UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of Science and Technology

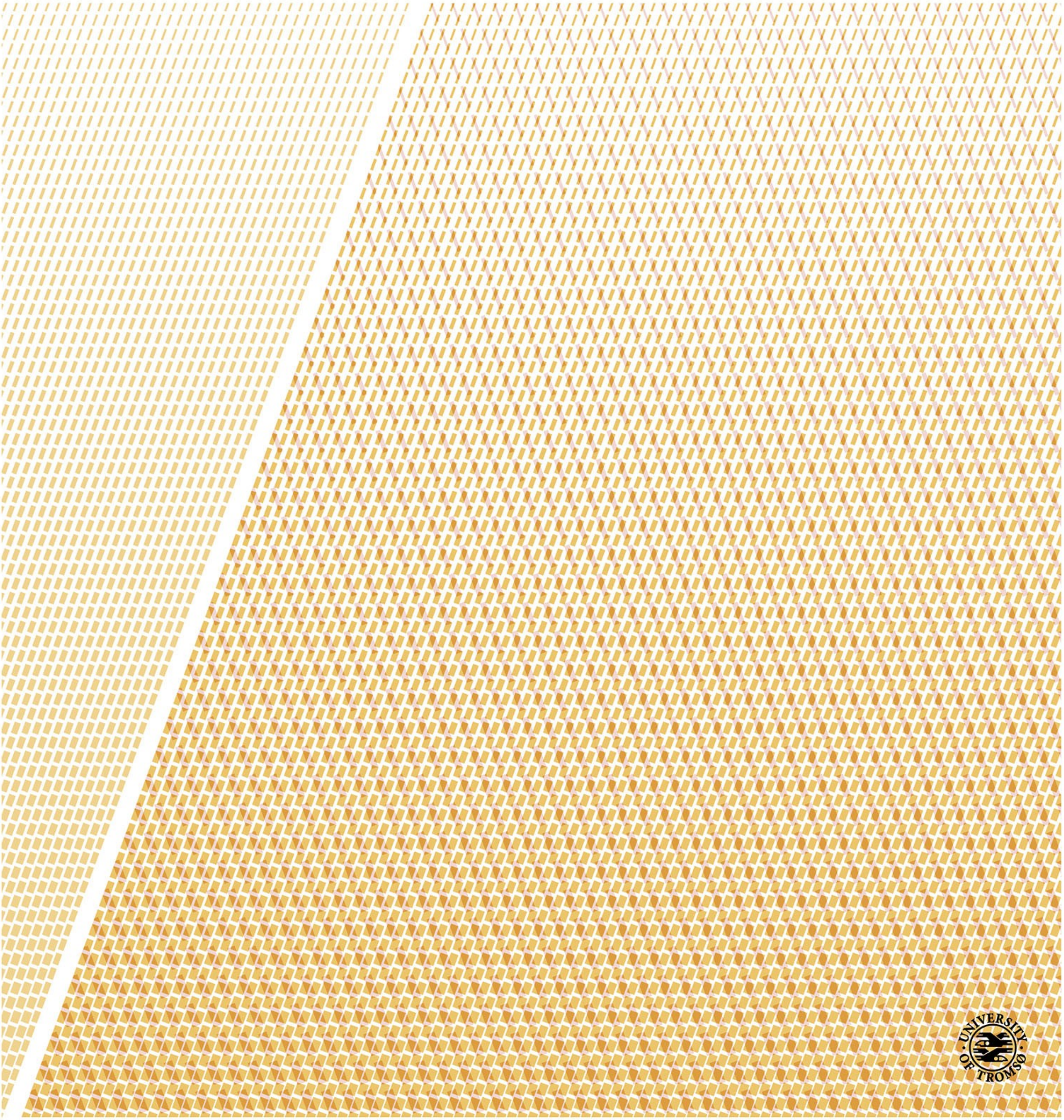
Department of Mathematics and Statistics

Advancing Unsupervised and Weakly Supervised Learning with Emphasis on Data-Driven Healthcare

—

Karl Øyvind Mikalsen

A dissertation for the degree of Philosophiae Doctor – November 2018



Abstract

In healthcare, vast amounts of data are stored digitally in the electronic health records (EHRs). EHRs contain patient-specific data in the form of unstructured free text notes as well as structured lab tests, diagnosis codes, etc., and represent a largely untapped source of clinically relevant information, which combined with advances in machine learning, have the potential to transform healthcare into a more data-driven direction.

Due to the complexity and poor quality of the EHRs, data-driven healthcare is facing many challenges. In this thesis, we address the challenge posed by lack of ground-truth labels and provide methodological solutions to challenges related with missing data, temporality, and high dimensionality. Towards that end, we present four lines of work where we develop novel unsupervised and weakly supervised learning methodology.

The first work presents a novel kernel for a type of data that frequently occur in the EHRs, namely multivariate time series with missing values. Key components in the method are clustering and ensemble learning, which ensure robustness to hyper-parameters and make the kernel well-suited as a component in unsupervised learning frameworks. Experiments on benchmark datasets demonstrate that the proposed kernel is robust to hyper-parameter choices and performs well in presence of missing data.

Next, we present a novel dimensionality reduction method, which is designed to account for many of the challenges data-driven healthcare is facing. One of them is high dimensionality, but in addition, the method is capable of exploiting noisy and partially labeled multi-label data, touching upon challenges related with lack of labels, domain complexity and noisy data. Extensive experiments on benchmark datasets, as well as a case study of patients suffering from chronic diseases, demonstrate the effectiveness of the proposed algorithm.

A main motivation for the third work is to take advantage of the fact that missing values and patterns often contain rich information about the clinical outcome of interest. We present a multivariate time series kernel, capable of exploiting this information to learn useful representations of incompletely observed time series data. Moreover, we also propose a novel semi-supervised kernel, capable of taking advantage of incomplete label information. The effectiveness of the proposed methods is demonstrated via experiments on benchmark data and a case study of patients suffering from infectious postoperative complications.

In the last work, we focus on another complication following major high-risk surgeries, namely postoperative delirium. It is a common complication among the elderly that often goes undetected, but might have serious consequences. We perform phenotyping using a weakly supervised learning framework, wherein clinical knowledge is used to generate a noisy labeled training set, which in turn is used to train classifiers. Experiments on a dataset collected from a Norwegian university hospital demonstrate the efficiency of the framework.

List of publications

The thesis is based on the following original journal papers.

I K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz and R. Jenssen, "**Time series cluster kernel for learning similarities between multivariate time series with missing data**", *Pattern Recognition*, Apr. 2018, Vol. 76, pp 569–581, doi: `https://doi.org/10.1016/j.patcog.2017.11.030`.

II K. Ø. Mikalsen, C. Soguero-Ruiz, F. M. Bianchi and R. Jenssen, "**Noisy multi-label semi-supervised dimensionality reduction**", submitted to *Pattern Recognition*, Sept. 2018.

III K. Ø. Mikalsen, C. Soguero-Ruiz, F. M. Bianchi, A. Revhaug and R. Jenssen, "**Time series cluster kernels to exploit informative missingness and incomplete label information**", submitted to *Pattern Recognition*, Nov. 2018.

IV K. Ø. Mikalsen, C. Soguero-Ruiz, K. Jensen, K. Hindberg, M. Gran, A. Revhaug, R.-O. Lindsetmo, S. O. Skrøvseth, F. Godtliebsen and R. Jenssen, "**Using anchors from free text in electronic health records to diagnose postoperative delirium**", *Computer Methods and Programs in Biomedicine*, 2017, Vol. 152, pp 105–114, doi: `https://doi.org/10.1016/j.cmpb.2017.09.014`.

Other papers

The following journal papers, manuscripts under review and other peer-reviewed publications also contribute to this thesis, but are not included.

5. J. N. Myhre, K. Ø. Mikalsen, S. Løkse and R. Jenssen, "**A robust clustering using a kNN mode seeking ensemble**", *Pattern Recognition*, Apr. 2018, Volume 76, pp 491–505.
6. K. Jensen, C. Soguero-Ruiz, K. Ø. Mikalsen, R. O. Lindsetmo, I. Kouskoumvekaki, M. Girolami, S. O. Skrovseth and K. M. Augestad, "**Analysis of free text in electronic health records for identification of cancer patient trajectories**", *Scientific Reports*, Apr. 2017, Volume 7.
7. K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, R. Jenssen, "**The time series cluster kernel**", published in *Proceedings of 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, Tokyo, Japan, Sep. 2017, pp. 1–6.
8. K. Ø. Mikalsen, C. Soguero-Ruiz, K. Jensen, K. Hindberg, M. Gran, A. Revhaug, R.-O. Lindsetmo, S. O. Skrøvseth, F. Godtliebsen and R. Jenssen, "**Predicting postoperative delirium using anchors**", poster presentation at *NIPS 2015 Workshop on Machine Learning in Healthcare*, Montreal, December 2015.

9. K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, S. O. Skrøvseth, R.-O. Lindsetmo, A. Revhaug, R. Jenssen, " **Learning similarities between irregularly sampled short multivariate time series from EHRs**", oral presentation at *3rd ICPR International Workshop on Pattern Recognition for Healthcare Analytics*, Cancun, Mexico, Dec. 2016, available at `https://sites.google.com/site/iwprha3/proceedings`.

10. K. Ø. Mikalsen, C. Soguero-Ruiz, K. Jensen, K. Hindberg, M. Gran, A. Revhaug, R.-O. Lindsetmo, S. O. Skrøvseth, F. Godtliebsen and R. Jenssen, "**Using anchors from free text to diagnose postoperative delirium from EHRs** ", poster presentation at Regional helseforskningskonferanse 2016, Tromsø, Norway, Nov. 2016.

11. M. A. Hansen, K. Ø. Mikalsen, M. Kampffmeyer, C. Soguero-Ruiz and R. Jenssen, "**Towards deep anchor learning**", published in *Proceedings of 2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, Las Vegas, USA, Mar. 2018, pp 315–318.

12. A. Storvik Strauman, F. M. Bianchi, K. Ø. Mikalsen, M. Kampffmeyer, C. Soguero-Ruiz, R. Jenssen, "**Classification of postoperative surgical site infections from blood measurements with missing data using recurrent neural networks**", published in *Proceedings of 2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, Las Vegas, USA, Mar. 2018, pp 307–310.

13. F. M. Bianchi, K. Ø. Mikalsen and R. Jenssen, "**Learning compressed representations of blood samples time series with missing data** ", published in *Proceedings of 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium, Apr. 2018,

14. J. N. Myhre, K. Ø. Mikalsen, S. Løkse and R. Jenssen, "**Consensus clustering using kNN mode seeking**", published in *Proceedings of 19th Scandinavian Conference on Image Analysis*, Copenhagen, Denmark, June 2015, pp 175–186.

15. J. N. Myhre, K. Ø. Mikalsen, S. Løkse and R. Jenssen, "**Robust non-parametric mode clustering**", published in *NIPS workshop on Adaptive and Scalable Nonparametric Methods in Machine Learning*, Barcelona, Spain, Dec 2016. `https://sites.google.com/site/nips2016adaptive/accepted-papers`

16. K. Ø. Mikalsen, C. Soguero-Ruiz, I. Mora-Jiménez, I. Caballero López Fando, and R. Jenssen, "**Using multi-anchors to identify patients suffering from multimorbidities**", accepted for IEEE International Conference on Bioinformatics and Biomedicine (BIBM), BHI workshop, Madrid, Spain, Dec. 2018.

17. F. M. Bianchi, L. Livi, K. Ø. Mikalsen, M. Kampffmeyer and R. Jenssen, "**Learning representations of multivariate time series with missing data using Temporal Kernelized Autoencoders**", arXiv preprint arXiv:1805.03473, submitted to *Pattern Recognition*, Aug. 2018, `https://arxiv.org/abs/1805.03473`.

18. P. Kocbek, A. Stozer, C. Soguero-Ruiz, G. Stiglic, K. Ø. Mikalsen, N. Fijacko, P. Povalej Brzan, R. Jenssen, S. O. Skrøvseth, and U. Maver", **Maximizing interpretability and cost-effectiveness of Surgical Site Infection (SSI) predictive models using feature-specific regularized logistic regression on preoperative temporal data**", submitted to *Computational and Mathematical Methods in Medicine*, Sept. 2018.

Acknowledgements

I have been privileged to enjoy the support and encouragement of many people during the creation of this thesis. I am thankful to everyone who made this possible.

First, I would like to express my sincere gratitude to my supervisor, Robert, for his guidance, support, constant encouragement, optimism and patience throughout this entire journey. I would also like to thank my co-supervisors Fred and Stein-Olav for their advice and helpful discussions.

I would like to express my gratitude to everyone at the University Hospital of North-Norway and the Norwegian Centre for E-health Research that I have collaborated with. In particular, I would like to mention Arthur, Rolv-Ole, Knut-Magne, Kristian, Mads, Kasper, Anne-Torill and Stein-Olav.

To Filippo, thank you for useful discussions, for your enthusiasm and guidance. Working with you has deepened my understanding and interests in machine learning.

To Cristina, thanks for your advice and helpful discussions, and for all the time and effort you have spent on our joint research projects. I would also like to thank other collaborators in Fuenlabrada.

To Jonas, Michael and Sigurd, thanks for insightful conversations about research and helpful advice. It has been a great pleasure working with you on our joint projects.

To my other co-authors, I am grateful that you welcomed me into the research community.

To everyone at the UiT Machine Learning Group, thanks for the good discussions and support. When I started my PhD, the group did not exist. It has therefore been fun and motivating to experience the growth in the number of team members in the last couple of years.

I also would like to thank my committee members for taking the time to read my thesis and attending the defense.

Last but definitely not least, I would like to thank my family and friends for their constant support!

Karl Øyvind Mikalsen,
Tromsø, November 2018.

Contents

List of abbreviations

ATC Anatomical Therapeutic Chemical.

CNN Convolutional Neural Network.

CPT Current Procedural Terminology.

DRG Diagnosis Related Groups.

EHR Electronic Health Records.

EM Expectation-Maximization.

GMM Gaussian Mixture Model.

GRU Gated Recurrent Units.

ICD International Classification of Diseases.

kNN k-Nearest Neighbors.

KPCA Kernel Principal Component Analysis.

LSTM Long Short-Term Memory.

MeSH Medical Subject Headings.

NCSP NOMESCO Classification of Surgical Procedures.

NLP Natural Language Processing.

PAC Probably Approximately Correct.

PCA Principal Component Analysis.

PU-learning Learning with Positive and Unlabeled data.

RBF Radial Basis Function.

RCN Random Classification Noise.

RKHS Reproducing Kernel Hilbert Space.

RNN Recurrent Neural Network.

SNOMED-CT Systematized Nomenclature of Medicine-Clinical Terms.

SVM Support Vector Machine.

TCK Time series Cluster Kernel.

UMLS Unified Medical Language System.

UNN University Hospital of North Norway.

WHO World Health Organization.

Chapter 1

Introduction

1.1 Data-driven healthcare

Major advances in healthcare such as the introduction of vaccines, anesthesia, antibiotics, randomized control trials and radiology imagery are examples of events from the 19th and 20th century that revolutionized healthcare and lead to improved quality of life for many people. Despite these enormous improvements, now in the 21st century, there are still tremendous unsolved challenges in healthcare and new challenges are continuously appearing.

For instance, one of the future challenges is related to the fact that the demographic is changing. For the first time in history there will be more people aged 65 and over than children under age 5 on the globe by 2020 (He et al., 2016). Along with changes in diet and lifestyle, aging is the main reason why chronic noncommunicable diseases such as cardiovascular diseases, diabetes, and cancer are increasing in prevalence and now represent the dominant healthcare burden globally (WHO, 2014; Marengoni et al., 2011). In general, the challenges that health is facing are profound, and major changes in current practice are needed (WHO, 2015).

In many disciplines such as e.g. marketing, financial services, and linguistics, to name a few, the combination of advancements in data science and a rapidly increasing amount of data being generated in digital format has led to new insights and solutions to existing challenges in these fields. Also in health, vast amounts of (biomedical) data are ubiquitously being recorded at the patient level. One source of such biomedical data is the Electronic Health

Records (EHR), which contains documentation of clinical and administrative encounters between the healthcare providers (physicians, nurses, etc.) and the patients (Jensen et al., 2012; Birkhead et al., 2015).

The EHRs were primarily developed for making healthcare more efficient from an operational standpoint and for billing purposes. However, these data undoubtedly represent a largely untapped source of clinical information that can be exploited via secondary use (Häyrinen et al., 2008; Botsis et al., 2010; Bellazzi and Zupan, 2008). Therefore, researchers already several years ago saw the potential to transform healthcare by developing autonomous monitoring systems as well as diagnosis and decision support tools based on data-driven approaches and *machine learning*, leaping forward quality of care for the individual patient, and thereby being one of the solutions to the challenges modern healthcare are facing (Savage, 2012; Groves et al., 2013; Murdoch and Detsky, 2013). This research direction, in which machine learning plays a key role, is the main focus of the thesis and will be referred to as *data-driven healthcare* hereafter. Figure 1.1 shows an illustration of what data-driven healthcare might look like in practice.

Data-driven healthcare is a rapidly evolving research field that is getting an increasing amount of attention. This is reflected by the vast amount of startups[1], initiatives[2], as well as research centers[3] all over the globe that are focusing on this topic. For instance, data-driven healthcare is a research area that is being pursued at IBM Research[4], and, in particular, via their Center for Computational Health. As a result of these efforts, many research articles showing great promise for data-driven healthcare, have been published in academic journals (Ng et al., 2015; Yu et al., 2016; Choi et al., 2016a; Dai et al., 2015; Caballero Barajas and Akella, 2015; Esteva et al., 2017; Choi

[1] For startups on data-driven healthcare, see e.g. BenevolentAI `https://benevolent.ai/`, Babylon `https://www.babylonhealth.com/`, and DataRobot `https://www.datarobot.com/healthcare/`.

[2] For initiatives on advancing ubiquitous data and services in health, see e.g. the Norwegian government's "One citizen – one journal" act `https://www.regjeringen.no/no/dokumenter/meld-st-9-20122013/id708609/`, and Big Data Technologies in Healthcare `http://www.bdva.eu/sites/default/files/Big%20Data%20Technologies%20in%20Healthcare.pdf`

[3] Examples of research centers include the Computational Health Informatics laboratory `http://www.robots.ox.ac.uk/~davidc/index.php`, SPHERE `https://www.irc-sphere.ac.uk/`, Google Research `https://ai.google/research/teams/brain/healthcare-biosciences`, Machine Learning in Medicine `https://www.mlim-cornell.club/`, BigMed `https://bigmed.no/`, Norwegian Centre for E-health Research `https://ehealthresearch.no/en/`

[4] `https://www.research.ibm.com/healthcare-and-life-sciences/`

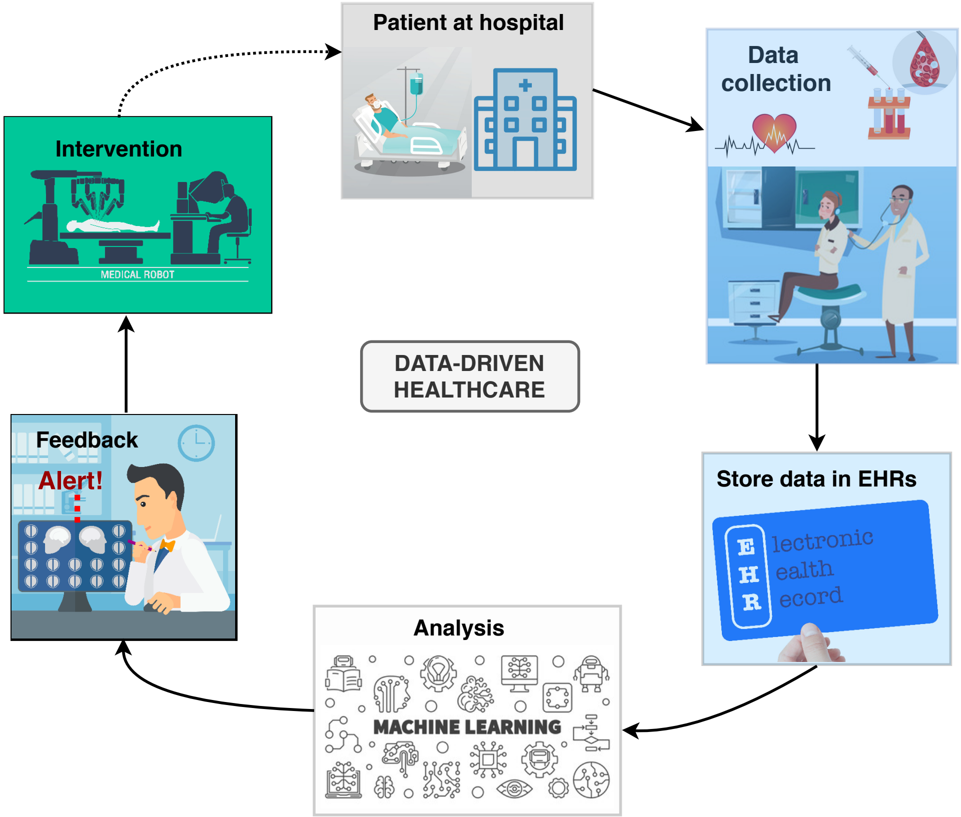


Figure 1.1: *Illustration of what data-driven healthcare might look like in practice. 1. The patient sees the doctor at the hospital. 2. Relevant tests are performed (labs, CT, MRI, etc.) and data are collected. 3. Data are stored in the EHRs, 4. Data-driven analysis using machine learning. 5. Provide clinical decision support to medical practitioner (e.g. warn the doctor that the patient is about to experience a complication). 6. Intervention (e.g. perform surgery).*

et al., 2016e; Rajkomar et al., 2018; Bai et al., 2018; Liu et al., 2018a).

A concrete example of an unsolved problem within healthcare is that there is a large number of postoperative complications. Approximately 25 percent of the patients undergoing high-risk surgeries suffer from at least one postoperative complication within 30 days of surgery. These complications are associated with severe consequences such as increased mortality, as exemplified by the fact that in hospitals in the United Kingdom alone, 20000 to 25000 deaths occur every year after surgical procedures (Findley, 2011). Another consequence of postoperative complications is that many of these patients are readmitted to the hospital, which leads to increased costs for the healthcare providers. One study demonstrated that even relatively modest reductions in complication rates (5% - 20%), can lead to cost savings in the range of 31 million to 124 million US dollars per year for Medicare[5] (Sweeney, 2013). Hence, the potential impact of prediction and prevention of postoperative complications is immense, not only for the well-being of the individual patients, but also in terms of cost optimization and resource allocation.

Recently, a couple of studies have shown great promise for data-driven healthcare as a means to predict postoperative complications such as e.g. anastomosis leakage (Soguero-Ruiz et al., 2016a; Watanabe et al., 2017; Soguero-Ruiz et al., 2016b), acute kidney injury (Kate et al., 2016), urinary tract infections (Taylor et al., 2018), and surgical site infections (Sanger et al., 2016; Soguero-Ruiz et al., 2015; Ke et al., 2017). Hence, a concrete example of a consequence of advances in data-driven healthcare is reduction in the number of postoperative complications.

Nevertheless, despite the many promising results reported in academic journals, the seemingly large availability of biomedical data, the vast amounts of startups and initiatives, and the many success stories and great promises reported in mainstream media (Scutti, 2017; Mukherjee, 2017; Comstock, 2017; Murgia, 2017; Bhardwaj, 2018), big data analytics and machine learning based-approaches have yet to see the same success in healthcare as in other fields (Lee and Yoon, 2017; Fröhlich et al., 2018). Data-driven healthcare is still only an emerging reality and has yet not transformed medicine. One of the main reasons for this is that there are still many unresolved challenges for data-driven healthcare. In the next section, we will briefly describe these challenges.

1.2 Challenges for data-driven healthcare

Despite that the challenges that data-driven healthcare are facing are well documented in the literature[6], due to the complexity of the human body,

[5] Medicare is a federal health insurance for Americans aged 65 years and older `https://www.medicare.gov/`.

[6] See e.g. (Jensen et al., 2012; Weiskopf and Weng, 2013; Hripcsak and Albers, 2012; Kuo et al., 2014; Hersh et al., 2013; Miotto et al., 2017; Yadav et al., 2018; Dinov, 2016;

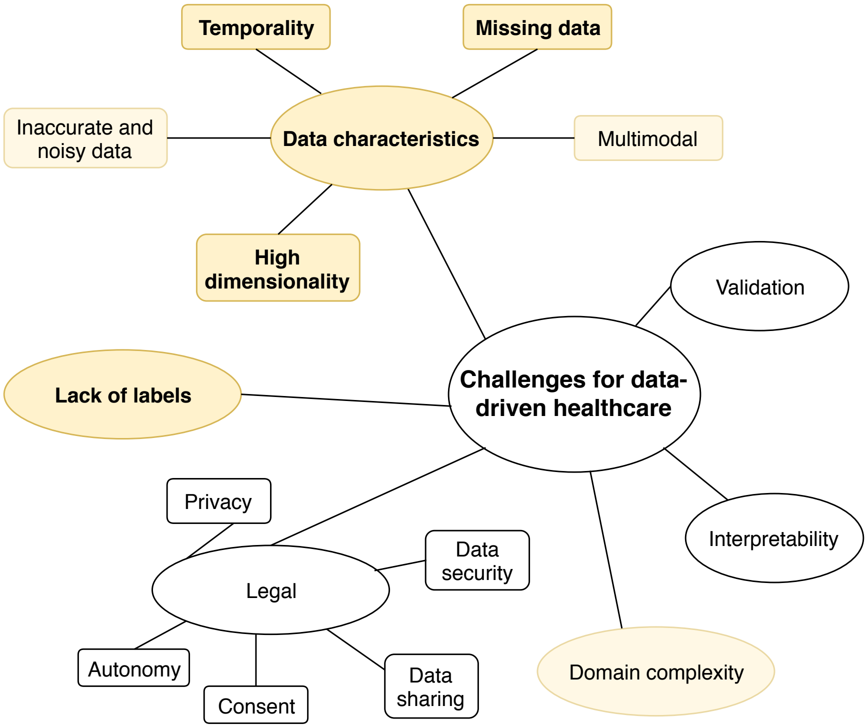


Figure 1.2: *Overview of the challenges that data-driven healthcare are facing. Challenges that we are providing solutions to are marked in yellow.*

the complexity of diseases, and the complexity of healthcare in general, it is difficult to give a complete overview of all challenges. Moreover, biomedical data have some uncommon characteristics that complicate analysis. In addition to the EHRs, these data also include e.g. clinical imagery, genomics data, and data collected from wearable devices. Nevertheless, in Fig 1.2, we have tried to provide an overview. We note that our perspective is slightly biased towards EHR-related challenges since these are the main focus of the thesis. A more detailed description of these challenges follows next.

Data characteristics. One of the main challenges for data-driven healthcare is that the very nature of the EHR data is uniquely complex. Such data have some special uncommon characteristics compared to other application domains. The data are characterized by:

Fröhlich et al., 2018; Xiao et al., 2018; Lee and Yoon, 2017; Ravı et al., 2017; Shickel et al., 2018; Häyrinen et al., 2008; Ching et al., 2018; Banda et al., 2018; Johnson et al., 2016).

Multiple modalities. EHRs are highly heterogeneous and consist of, for example, unstructured text in the form of nurses reports and surgical procedure notes. In addition, EHRs contain information about e.g. admissions, discharge, blood samples, histology, radiology (imagery), etc. Some of these data are stored in the form of structured codes for different medical conditions using e.g. the International Classification of Diseases (ICD), 10th version (WHO, 2004).

High dimensionality. Even if data extracted from EHRs often at first sight look "big", also in terms of number of patients, in practical clinical scenarios the number of patients available to train the models is often limited. On the other hand, because of the heterogeneity of biomedical data, the number of attributes describing each patient is often large compared to the number of patients (large p, small n) (Wang and Krishnan, 2014; Sinha et al., 2009; Lee and Yoon, 2017). A patient could for example be described by tens of thousands of genes and/or a vast amount of clinical parameters such as laboratory tests, drugs, codes, x-rays as well as unstructured free-text documents. Such high-dimensional data is a problem for most machine learning algorithms because of the *curse of dimensionality* (Friedman, 1997).

Inaccurate and noisy data. The patient records were primarily developed for billing purposes, and are also used by healthcare professionals to plan patient care, and to document and assess the care that is delivered (Häyrinen et al., 2008). This means that the EHRs do not constitute a traditional research database and therefore the data quality is in general worse than in other databases. Erroneous, inconsistent and instable data frequently occur.

Temporality. The EHR data are longitudinal in nature since the diseases and the patients' health statuses progress over time. However, many existing machine learning algorithms cannot deal with temporality, but assume static vector based inputs.

Missing data. The data are largely missing in many different ways, often as a result of not having been collected for research purposes (Wells et al., 2013; Hripcsak and Albers, 2012). The reason might be as simple a human error. For instance, it could happen that a clinician makes a mistake when he records the result of a lab test. Data are also missing because people usually visit healthcare providers only if they are sick or injured, i.e. there are no available data from the periods when the patients are healthy (which usually is most of the time). However, even for hospitalized patients missing data frequently occur, e.g. because the doctor thinks the patient is in good shape and therefore decides to not order a lab test. Either way, regardless of the reason why missing values occur, they pose a challenge for most machine learning methods and must be handled.

Lack of labels. In a data-driven healthcare setting, labels refer to gold standards, i.e. the true clinical outcomes or the true disease phenotypes for the patients of interest. These types of labels are typically not consistently captured in the EHRs and therefore not easily available. In addition, generating such ground-truth labels is often time consuming, expensive or even impossible. Lack of labels poses a challenge because an underlying assumption in the classical branch of machine learning, which is supervised learning, is that ground-truth labels are provided for the entire training set. In this scenario, machine learning is usually very powerful. However, when label information is either completely lacking or incomplete, learning data-driven algorithms usually becomes more difficult.

Legal issues. The EHRs contain private information about individual patients' lives that should be kept secret, and therefore, privacy and legal issues are important. However, this also poses a challenge for researchers since restriction of access to EHR data is an obstacle for the development of data-driven healthcare (Jensen et al., 2012).

Domain complexity. It is more complicated to understand a disease and the inner workings of the human body than an image or speech. The many data sources and the characteristics of the EHR data, which we described above, also contribute to increase the complexity.

Interpretability. Many machine learning algorithms can be good at predicting e.g. a disease onset, but typically it is difficult to interpret how the algorithm came to that conclusion (black box). However, understanding why the algorithms provide the recommendations they do is critical to convince the clinicians to trust the predictions. In addition, the General Data Protection Regulation was adopted by the European Union recently and gives for example patients "right to an explanation" (Goodman and Flaxman, 2016), which could be difficult if the predictions are made by machine learning algorithms.

Validation. To translate an algorithm into clinical practice requires rigorous validation, which is a complicated process that is both time consuming and expensive. Algorithms reported in academic journals are typically not sufficiently validated for clinical practice (Fröhlich et al., 2018).

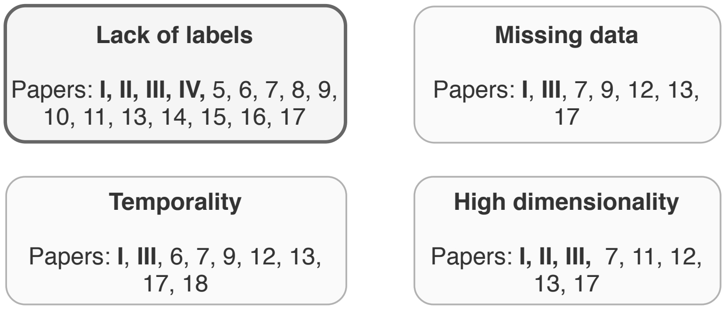

Lack of labels
Papers: **I, II, III, IV,** 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17
Missing data
Papers: **I**, **III**, 7, 9, 12, 13, 17
Temporality
Papers: **I**, **III**, 6, 7, 9, 12, 13, 17, 18
High dimensionality
Papers: **I, II, III,** 7, 11, 12, 13, 17

Figure 1.3: *Categorization of publications according to the objectives they deal with.*

1.3 Objectives

In this thesis, we focus on some of the above-mentioned challenges. Our main objective is to provide methodological solutions that address the challenge posed by *lack of labels*. All four included papers deal with this objective.

Secondary objectives are to provide methodological solutions to challenges related with

- missing data,
- temporality,
- high dimensionality.

In addition, we also touch upon challenges related with inaccurate and noisy data, multiple modalities, and domain complexity. Fig. 1.3 provides an overview of how the different publications relate to the objectives.

1.4 Proposed approaches

The work presented in this thesis is motivated by challenges that data-driven healthcare are facing, and particularly the challenges posed by lack of labels, missing data, temporality and high dimensionality. However, data-driven healthcare is not the only application domain in which the process of obtaining reliable ground-truth labels often is difficult. In e.g. computer vision (Xiao et al., 2015), audio and speech processing (Adavanne and Virtanen, 2017), to name a few, one experiences similar problems. Likewise, while temporal data frequently occur in healthcare, e.g. via lab measurements of hospitalized patients which naturally constitute multivariate time series subject to missing data, similar type of data also occur in other applications such as e.g. biology, finance and geosciences (Mudelsee, 2013; Lacasa et al., 2015). Further, high dimensionality is a challenge that almost all practical applications share. For these reasons, we take a general approach to solve these problems by developing novel machine learning methodology, which as such is not restricted to medical applications, but potentially can be applied to any domain facing similar challenges.

In this thesis, the key solution to our main objective (the challenge posed by lack of labels) is novel *unsupervised* and *weakly supervised* learning methodology. An illustrative explanation of these learning frameworks is provided in Fig. 1.4. Two of the works (Paper I and partly Paper III) present unsupervised learning frameworks in which no label information is provided. An alternative workaround to the lack of labels problem is to generate incomplete or inaccurate labels. The idea is that these labels can be created in a way that is less expensive and less time consuming than to create the labels manually. This is a situation we study in Paper II, IV and partly Paper III. For this purpose, we develop and employ semi-supervised learning frameworks that can deal with label noise.

The unsupervised and weakly supervised learning methods developed and employed in this thesis can be further divided into three sub-categories:

- Clustering.
- Semi-supervised learning with noisy labels.
- Representation learning (dimensionality reduction).

In Fig. 1.5, we have categorized the publications according to the three sub-categories of methods.

In addition to the challenge posed by lack of labels, we also present approaches to address the challenges posed by missing data, temporality, and high dimensionality. A key approach to cope with missing data and temporality is to analyze longitudinal EHR data subject to missing elements within the framework of *kernel methods*, and, in particular, to consider kernels for multivariate time series (Paper I and III). Regarding high dimensionality, in Paper II we present a novel dimensionality reduction method. Moreover, in the first paper we demonstrate that kernels could provide a useful tool to learn representations of high-dimensional multivariate time series – a tool we exploit in Paper III to learn representations of blood sample time series containing large amounts of missing data.

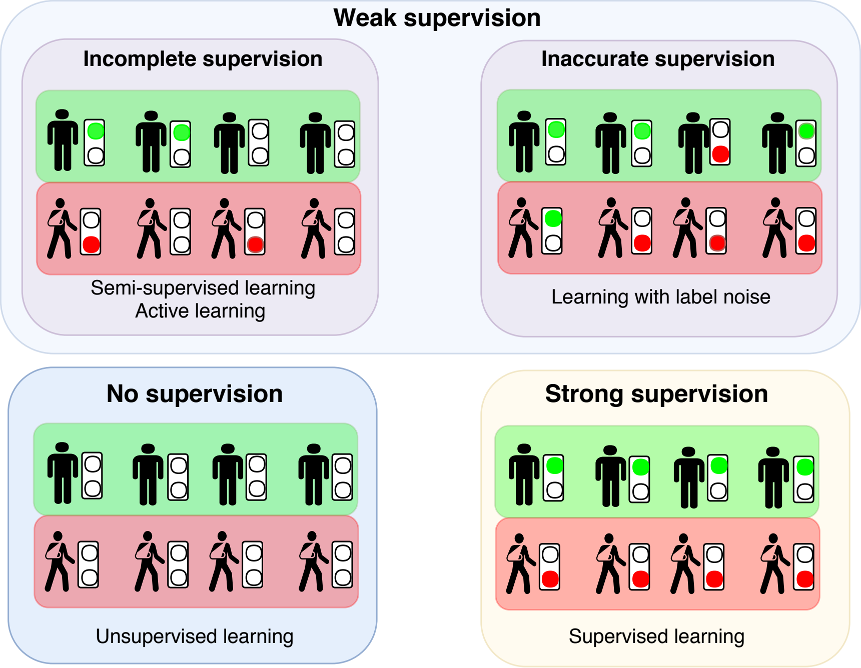


Figure 1.4: *Illustration of the concepts unsupervised and weakly supervised learning from a healthcare perspective. The patients in the red box have a particular clinical outcome of interest, whereas those in the green box do not. The fact that* ***no supervision*** *information is provided means that the clinical outcome (label) is unknown for all patients under study. In this situation, we employ unsupervised learning. In other cases,* ***weak supervision*** *information is provided. This could be in the form of incomplete supervision information, i.e. the clinical outcomes of interest are known for a subset of the patients. For this type of supervision information, we employ semi-supervised learning algorithms. Weak supervision information could also be provided in terms of inaccurate supervision, i.e. the clinical outcomes of interest are known for most patients, but some of the patients have been assigned wrong outcomes. Hence, the labels are noisy. These settings are different from the classical branch of machine learning, namely supervised learning, in which strong supervision information is provided in terms of labels for the entire training set.*

In this thesis, we evaluate how well the proposed approaches solve the objectives empirically on general domain benchmark datasets as well as real-world EHR data obtained via close collaborators at our local hospital and

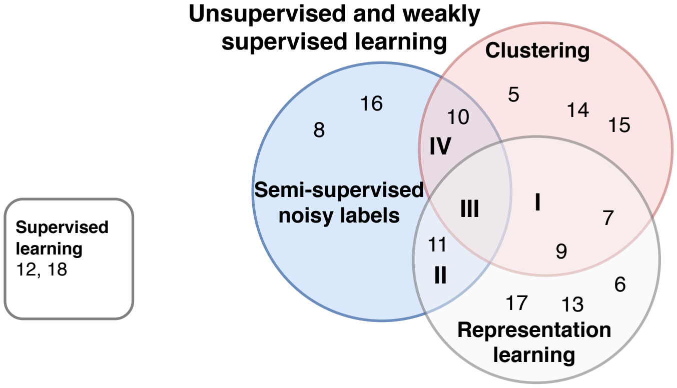

Unsupervised and weakly supervised learning
Clustering
Semi-supervised noisy labels
Representation learning
Supervised learning 12, 18
16
8
10
IV
5
14
15
III
I
7
11
II
9
6
17
13

Figure 1.5: *Methodological categorization of papers.*

a hospital in Spain. In particular, we study prediction and detection of postoperative complications related with gastrointestinal surgery (Paper III and IV). The majority of the patients under study undergo surgery for colorectal cancer, which is one of the most serious noncommunicable diseases. In Paper II, we provide a case study of patients that suffer from multiple noncommunicable diseases. Effectiveness of the proposed methods and solutions is evaluated in a relative manner (as opposed to absolute), i.e. we do not evaluate if the methods are effective, but if they are effective relative to existing methods.

1.5 Brief summary of papers

Paper I. In this paper, we present a novel methodology for computing the similarity between multivariate time series (temporal data) subject to missing data. Key components in the method are clustering and ensemble learning, which make the similarity measure robust to choice of hyper-parameters. For this reason, the proposed similarity measure, which also is a kernel, is well-suited when lack of labels is an issue and can be used as one component in a larger unsupervised learning framework.

Paper II. This paper presents a novel dimensionality reduction method, which is general and not necessarily restricted to healthcare applications. However, the method is designed in such a way that it accounts for many of the challenges data-driven healthcare is facing. One of them is obviously high dimensionality, but in addition, the method is capable of exploiting noisy and partially labeled multi-label data, touching upon challenges related with lack of labels, domain complexity and inaccurate data.

Paper III. The paper builds upon the work presented in Paper I and studies multivariate time series and missing data. A main motivation for this work is that, e.g. in healthcare, instead of having missing completely at random data, the missing values and missing patterns often contain rich information about the clinical outcome of interest. We present a kernel, which is capable of exploiting this information to learn a better representation of the incompletely observed time series data. Moreover, we also propose a novel semi-supervised kernel, capable of exploiting incomplete label information.

Paper IV. In this paper, we focus on detection of postoperative delirium, which is a quite common complication after major high-risk surgeries among the elderly. Delirium is a complication that often goes undetected, but might have serious consequences both for the patients and the caregivers. For these reasons, it is important to improve current detection models. However, getting access to large enough amounts of ground-truth labels to train the models is difficult. In this study, we build detection models using a weakly-supervised framework, in which supervision information is provided in terms of clinical knowledge. The clinical expertise is used to generate a noisy labeled training set, which in turn is used to train classifiers.

1.6 Organization of the thesis

The remainder of this thesis is organized into four parts, *machine learning for data-driven healthcare*, *methodology and context*, *summary of research*, and *included papers*. The first part contains two chapters. In Chapter 2, we provide a description of EHRs and, in particular, the data types these records contain. Chapter 3 presents examples of machine learning for EHRs. The methodology part is divided into three chapters, which in sum constitute the theoretical background for the research presented in this thesis. In Chapter 4, we provide an introduction to kernel methods. Chapter 5 presents unsupervised learning, whereas weakly supervised learning is described in Chapter 6. In the summary of research part, we provide a short overview of the scientific contribution of each paper in this thesis. We also add some concluding remarks and a discussion on future directions. The research papers are included in Part IV of this thesis. We also provide an appendix, which contains a statistical description of missing data mechanisms and a survey on common methods to deal with missing data.

Part I

Machine learning for data-driven healthcare

Chapter 2

Electronic health records

The EHR is an evolving concept and there exists several different definitions. The *Recommendation of 2 July 2008 on cross-border interoperability of electronic health record systems* (European Union) defined an EHR as "a comprehensive medical record or similar documentation of the past and present physical and mental state of health of an individual in electronic form, and providing for ready availability of these data for medical treatment and other closely related purposes"[1].

Other definitions make stronger assumptions and require that the records can be shared, contain information about the complete healthcare, are available instantly and securely to authorized personnel, or require that the EHR also contains information necessary to fulfill reporting obligations or disclosure obligations laid down in law or in compliance with the law [2] [3] (Gunter and Terry, 2005; Kierkegaard, 2011; Gerhard et al., 2013).

Nevertheless, despite the multitude of definitions, the intent of the EHR systems is usually that they can be shared across healthcare providers, specialists, clinicians and laboratories, etc., and, therefore, contain information about the *complete* healthcare of the patient. Therefore, the EHRs contain whole range of data – in different forms – including the patient's medical history, demographics, diagnoses, vital signs, medications, treatment plans,

[1] Commission Recommendation of 2 July 2008 on cross-border interoperability of electronic health record systems (notified under document number C ((2008) 3282).

[2] `https://www.healthit.gov/faq/what-electronic-health-record-ehr`

[3] `ehelse.no/standarder-kodeverk-og-referansekatalog/elektronisk-pasientjournal-epj`

immunization status, allergies, radiology images, free text notes, and laboratory test results. Moreover, irrespective of whether it is the government (via tax payers' money) or insurance companies that pay for the healthcare, the healthcare providers usually are reimbursed by documenting the care that they have provided in the EHRs via codes.

Next, we provide some examples of data types that are commonly contained in the EHRs.

2.1 EHR data types

Descriptive data. Normally, in all EHRs, one can find demographic details about patients such as age, sex, date of birth and death, religion, ethnicity, marital status, etc., as well as other descriptive data such as admission and discharge times.

Coded data. Coded data in the EHRs are recorded primarily for billing and administrative purposes. In particular, *diagnoses* and diagnostic and therapeutic *procedures* are often coded.

The patients' diagnoses are typically documented in the EHRs using codes, and for this purpose an international standard exists, namely World Health Organization (WHO)'s International Classification of Diseases. In Europe, most countries use the 10th version (ICD-10). However, e.g. in Spain and Portugal, the 9th version is still in use. Moreover, many country-specific modifications of ICD exists, and different countries have their own coding guidelines. The US is using the ICD-10 Clinical Modification.

As an illustrative example of an ICD (-10) code, we highlight the code *C18.1*. The letter 'C' indicate that this is a code that belong to the neoplasm-family (C and D). In particular, 'C' represents malignant neoplasms. Further, the number '18' indicates that the code represents a malignant neoplasm of a digestive organ (C15-C26), and more specifically, '18' represents colon. The digit after the period specifies where in the colon the neoplasm is, in this case in the appendix.

Some countries, like e.g. Spain and Portugal, also use ICD for classification of diagnostic and therapeutic procedures performed by physicians and other health care providers (*procedure coding*). More specifically, they use the ICD, 9th Revision - Clinical Modification. However, no international standard exists, and therefore the difference between the countries is larger for procedure codes than for diagnosis codes (Busse et al., 2011). For example, the US uses the Current Procedural Terminology (CPT) classification (AMA, 2007), whereas the Nordic countries use the NOMESCO Classification of Surgical Procedures (NCSP) (NOMESCO, 2011). The NCSP is divided into 15 main chapters describing surgical procedures related to the functional-anatomic body system, and 4 subsidiary chapters describing therapeutic and investigative surgical procedures.

The EHRs can also contain so-called Diagnosis Related Groups (DRG), which is a coding-system that classifies patient cases into categories with similar resource use. It is based on diagnoses and procedures, as well as age, sex, status at discharge and and the presence of complications or comorbidities. This coding system is typically used for reimbursement purposes, even though the DRG reimbursement practice could vary quite significantly from country to country (Mihailovic et al., 2016).

In addition to codes describing diagnoses and procedures, in some records, one can also find codes describing drugs. The Anatomical Therapeutic Chemical (ATC) Classification System(WHO, 2016) classifies drugs according to properties of the drug (therapeutic, pharmacological and chemical) and according to which organ or system the drug acts on. In more detail, the ATC codes are structured into five levels, referring to anatomical main groups, chemical substance, and therapeutic, pharmacological, chemical subgroups.

To illustrate the five levels in an ATC code, we highlight the code 'A10BA02'. The 1st level 'A' represents alimentary tract and metabolism, the 2nd level 'A10' drugs used for used for diabetes mellitus, 3rd level 'A10B' blood glucose lowering drugs, excluding insulins, 4th level 'A10BA' biguanides, whereas the 5th level 'A10BA02' represents metformin.

Vital signs and test results. Vital signs (body temperature, heart rate, blood pressure and breathing rate) are typically quite regularly documented by nurses for in-hospital patients.

Laboratory tests check samples of tissue or body fluids such as blood or urine to get more information about the health status of the patients. Information from these tests are usually recorded in the EHRs.

Clinical notes. A lot of the patient information in the EHRs is in free text. Clinical notes that contain free text include e.g. the admission journal, nursing notes, doctor notes, descriptive surgical reports, intensive care reports, hospital discharge summaries, reports of electrocardiogram and imaging studies (radiology reports), and administration records of intravenous medications and medication orders.

Unstructured free text from EHRs have some characteristics that make it different from other published text. For instance, clinical notes are characterized by that (i) a limited amount of time is spent on entering the text into the documents, simply because the document is a dictate of a conversation during a consultation, or because conversations are recorded and then later transcribed by a secretary; (ii) incomplete sentences and spelling errors are more common in medical text than in usual published text; and (iii) abbreviations and acronyms frequently occur. Even though the free text documents obviously contain a lot of information, from a computational and informatics point of view, the characteristics of the clinical notes pose many challenges.

Standard healthcare terminologies. In many EHR systems, the healthcare workers document some patient information using standardized terminologies. These are sometimes also referred to as medical ontologies, dictionaries or standard vocabularies. Among these, the most prominent example is the Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) (Stearns et al., 2001), which is the medical terminology of choice of both WHO and the International Health Terminology Standards Development Organization. SNOMED-CT is a systematic collection of clinical terms consisting of four core components: (i) numerical *codes* describing clinical terms, organized in hierarchies, (ii) textual *descriptions* of the codes, (iii) *relationships* between codes with similar meaning, and (iv) *reference sets* that groups clinical terms into sets. These sets can be used for cross-mapping to other standards. This means that SNOMED-CT also contains *coded data* and therefore the medical ontologies are closely connected to the coded data we mentioned above. Indeed, the ICD and ATC classification systems are examples other vocabularies (or ontologies).

We also include some examples of other commonly used terminologies:

- The US-specific medication terminology *RxNorm*[4] (Bennett, 2012).

[4] https://www.nlm.nih.gov/research/umls/rxnorm/

Terminology	Full name	Topic
ICD	International Classification of Diseases	Diagnoses
ATC	Anatomical Therapeutic Chemical	Medications
RxNorm	RxNorm	Medications
CPT	Current Procedural Terminology	Procedures
NCSP	NOMESCO Classification of Surgical Procedures	Procedures
LOINC	Logical Observation Identifiers, Names and Codes	Laboratory tests
DRG	Diagnosis Related Groups	Diagnos., procedu.
SNOMED-CT	Systematized Nomenclature of Medicine-Clinical Terms	General
UMLS	Unified Medical Language System	General
MeSH	Medical Subject Headings	PUBmed

Table 2.1: *Healthcare terminologies.*

- *Logical Observation Identifiers, Names and Codes* (LOINC)[5] (Forrey et al., 1996) is a commonly used standard vocabulary describing laboratory test results.

- Medical Subject Headings (MeSH)[6] is a controlled and hierarchically–organized vocabulary for indexing of medical journals in the MEDLINE/PubMed database[7]. Each article in the database is described by a set of MeSH terms. The MeSH terms can be mapped to other terminologies such as ICD-10 and ATC.

- MeSH Norwegian[8] [9] is the Norwegian version of MeSH. In addition, MeSH has been translated to many other languages.

- The Unified Medical Language System (UMLS)[10] (Lindberg et al., 1993) is a compendium of many vocabularies, which includes all abovementioned vocabularies.

In Tab. 2.1, we summarize the healthcare terminologies we have described in this chapter.

[5] https://loinc.org/

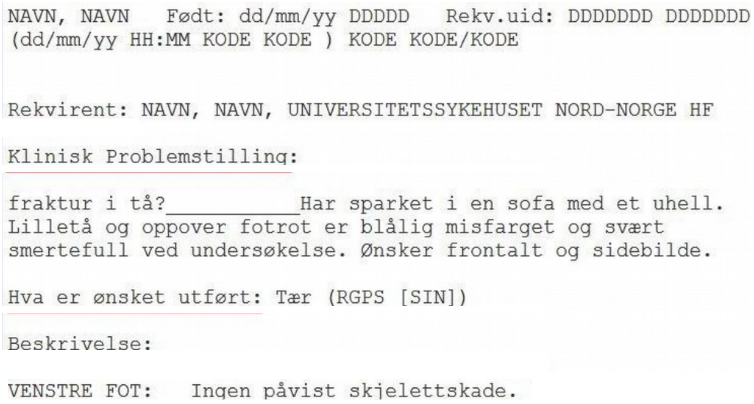
[6] https://www.nlm.nih.gov/mesh/

[7] https://www.ncbi.nlm.nih.gov/pubmed

[8] https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/MSHNOR/metadata.html

[9] http://mesh.uia.no

[10] https://www.nlm.nih.gov/research/umls/



```
NAVN, NAVN   Født: dd/mm/yy DDDDD   Rekv.uid: DDDDDDD DDDDDDD
(dd/mm/yy HH:MM KODE KODE ) KODE KODE/KODE


Rekvirent: NAVN, NAVN, UNIVERSITETSSYKEHUSET NORD-NORGE HF

Klinisk Problemstilling:

fraktur i tå?___________Har sparket i en sofa med et uhell.
Lilletå og oppover fotrot er blålig misfarget og svært
smertefull ved undersøkelse. Ønsker frontalt og sidebilde.

Hva er ønsket utført: Tær (RGPS [SIN])

Beskrivelse:

VENSTRE FOT:   Ingen påvist skjelettskade.
```

Figure 2.1: *Example of a fictive radiology report from University Hospital of North Norway (UNN).*

2.2 Tromsø EHR corpus

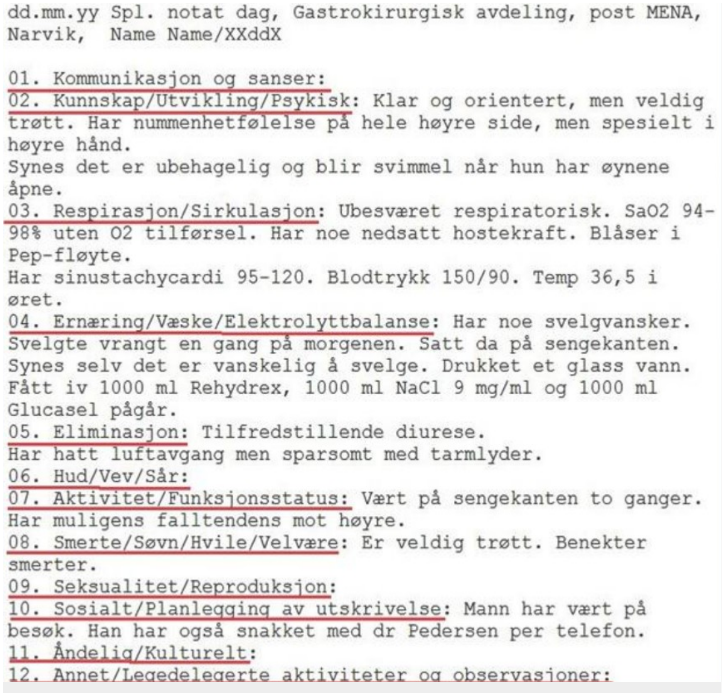
The data that are used in Paper III and Paper IV (and most of the Papers 5-18) are extracted from a real EHR system, obtained from the UNN. UNN has allocated resources for retrieving, pre-processing and making available EHR data from an entire department, namely the Department of Gastrointestinal Surgery for the years 2004-today. This longitudinal dataset contains more than 35000 unique patients and approximately 264 000 outpatient visits. Use of the data (de-anonymized) for research is granted by the Regional Committee for Medical Research Ethics (REK) and Norwegian Social Science Data Services (NSD).

The dataset contains the following sources of data.

- Procedure codes: More than 1 000 000 NCSP codes.
- Diagnosis codes: More than 1 000 000 ICD-10 codes.
- Laboratory tests: More than 1 600 000 lab tests.
- Free text notes: More than 1 800 000. There are hundreds of different document categories included in the database. The main patient journals are, however, the admission journals, nurse notes, doctor notes, descriptive surgical reports, intensive care reports, and discharge notes. The degree of structure in the documents varies, but many of them are completely unstructured (pure free text).
- Radiologic examinations: More than 60 000 radiology reports.
- Histology data: more than 500 000 pathology reports, including (re)-admittance and death dates.

Patient ID	1	2
Procedure code	JFB30	JFH20
Date of procedure	14.02.2010	24.04.2008
Age	69	46
Sex	Male	Female
Elective procedure	Yes	Yes
Main diagosis code	C182	K589
Open surgery	Yes	Yes
Stoma	No	Yes
ASA score	NA	3
Type of anesteshia	General and epidural	General anesthesia
Start of anesteshia	2010-02-14 10:01:00	2008-04-24 09:23:31
End of anesteshia	2010-02-14 12:39:52	2008-04-24 13:38:08
Start of surgery	2010-02-14 10:28:25	2008-04-24 10:05:00
End of surgery	2010-02-14 12:21:31	2008-04-24 13:25:25

Table 2.2: *Structured EHR data from UNN.*



```
dd.mm.yy Spl. notat dag, Gastrokirurgisk avdeling, post MENA,
Narvik,  Name Name/XXddX

01. Kommunikasjon og sanser:
02. Kunnskap/Utvikling/Psykisk: Klar og orientert, men veldig
trøtt. Har nummenhetfølelse på hele høyre side, men spesielt i
høyre hånd.
Synes det er ubehagelig og blir svimmel når hun har øynene
åpne.
03. Respirasjon/Sirkulasjon: Ubesværet respiratorisk. SaO2 94-
98% uten O2 tilførsel. Har noe nedsatt hostekraft. Blåser i
Pep-fløyte.
Har sinustachycardi 95-120. Blodtrykk 150/90. Temp 36,5 i
øret.
04. Ernæring/Væske/Elektrolyttbalanse: Har noe svelgvansker.
Svelgte vrangt en gang på morgenen. Satt da på sengekanten.
Synes selv det er vanskelig å svelge. Drukket et glass vann.
Fått iv 1000 ml Rehydrex, 1000 ml NaCl 9 mg/ml og 1000 ml
Glucasel pågår.
05. Eliminasjon: Tilfredstillende diurese.
Har hatt luftavgang men sparsomt med tarmlyder.
06. Hud/Vev/Sår:
07. Aktivitet/Funksjonsstatus: Vært på sengekanten to ganger.
Har muligens falltendens mot høyre.
08. Smerte/Søvn/Hvile/Velvære: Er veldig trøtt. Benekter
smerter.
09. Seksualitet/Reproduksjon:
10. Sosialt/Planlegging av utskrivelse: Mann har vært på
besøk. Han har også snakket med dr Pedersen per telefon.
11. Åndelig/Kulturelt:
12. Annet/Legedelegerte aktiviteter og observasjoner:
```

Figure 2.2: *Example of a fictive nurse note from UNN.*

Tab. 2.2 shows an example of structured data extracted for two fictive patients in connection with a surgical procedure performed at UNN. In Fig. 2.1 and Fig. 2.2, we show fictive examples of two document types from the EHRs from UNN, namely a radiology report and a nurse's note, respectively.

2.3 Fuenlabrada EHR corpus

The PhD-project was done in collaboration with our partners at the Universidad Rey Juan Carlos, Fuenlabrada, Spain, and for that reason we have also been using data collected from EHRs at the University Hospital of Fuenlabrada in this work.

In Paper II (and Paper 17), we used a dataset that was extracted from the EHRs at the University Hospital of Fuenlabrada, which is a public hospital in the southern area of Madrid, Spain, that covers a region with more than 200.000 inhabitants. The patient dataset consists of a structured subset of the patients' records from the year 2012. This subset contains information about time and place for the encounter with the health system, demographic data, pharmacy dispensation in the Madrid area, as well as information about diagnoses and procedures from patient encounters with primary and specialized care in the Fuenlabrada area. In total, there are more than 64000 patients in the dataset.

The information about diagnoses and procedures is provided in terms of codes according to the ICD-9 - Clinical Modification, whereas information about drugs is provided in terms of pharmacological dispensing codes according to the ATC classification systems (WHO, 2016).

In addition, the dataset contains information obtained from a specific Patient Classification System (PCS) (Davis and LaCour, 2016), namely the Clinical Risk Group (CRG) (Hughes et al., 2004). PCSs stratify patients according to different measures such as e.g. morbidity, health status, resource consumption, etc., based on information extracted during a certain a period of time. In particular, the CRGs provide useful information about the health status of patients potentially suffering from a multitude of chronic conditions. Each CRG is described by a five-digit code, where the first digit represents the core health status group, ranging from healthy to catastrophic (1 - 9). The three next digits represent the base risk group, whereas the last digit characterizes the severity-of-illness level.

Chapter 3

Examples of machine learning for EHRs

In this chapter, we review related work on machine learning for EHRs. We have sorted the work according to five non-mutually exlusive categories, i.e the papers do not necessarily exclusively belong to the category they are listed under. These categories are

- Phenotyping,
- Representation learning,
- Patient similarity,
- Predictive modeling,
- Other uses of machine learning for EHRs.

Figure 3.1 illustrate an example of a data-driven healthcare pipeline benefiting from the concepts discussed in Part I of this thesis, starting with the **raw EHR data**, followed by patient cohort identification via a machine learning driven **phenotyping** algorithm and **representation learning**, and finally, clinical decision support, for example via **predictive modeling** of diseases or **patient similarity** analytics for clinical knowledge extraction.

3.1 Phenotyping

Electronic phenotyping, EHR-based phenotyping, or simply just phenotyping, is the process of identifying patients with certain medical conditions

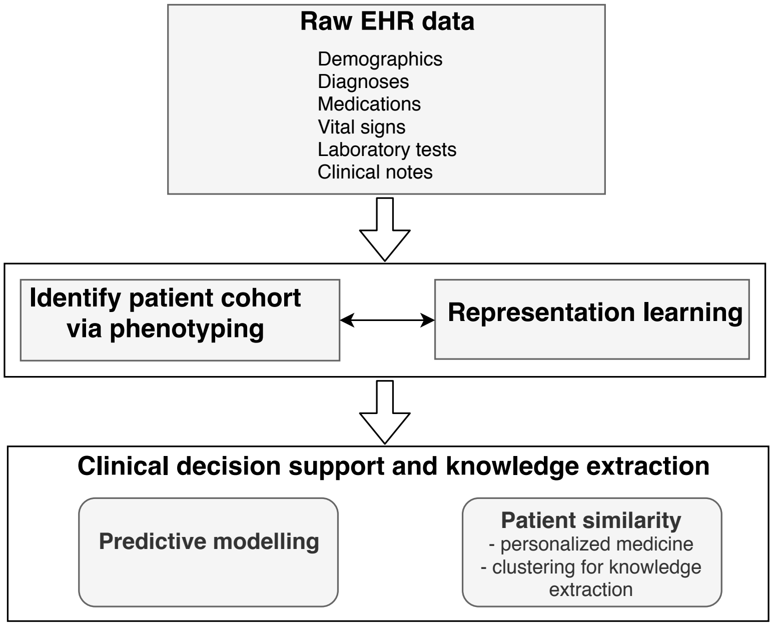


Figure 3.1: Illustration of a machine learning for EHR pipeline.

or characteristics of interest[1] (Yu et al., 2017a; Banda et al., 2017, 2018). Examples of phenotypes include specific diseases such as breast cancer, complex medical conditions such as stage III colorectal cancer and chronic obstructive pulmonary disease, and observable traits such as height and drug response (Wei and Denny, 2015). Phenotyping is one of the fundamental EHR research topics as it forms the basis of e.g. clinical decision support, translational research, population health analyses based on EHR data, and comparative effectiveness studies (Banda et al., 2018).

Patient cohort identification

A typical use of phenotyping algorithms is for patient cohort identification, i.e. finding cases (and controls) for certain phenotypes (Shivade et al., 2013; Yu et al., 2017a). There exist many works on uses of machine learning for this purpose.

[1] EHR-based phenotyping is, however, not well defined in literature and therefore its meaning is wide ranging (Shivade et al., 2013).

Huang et al. (2007) were among the first ones to use machine learning for the purpose of phenotyping in their study of type II diabetic patients. Carroll et al. (2011) proposed a framework for detecting rheumatoid arthritis using a support vector machine trained on Natural Language Processing (NLP)-derived concepts in addition to structured EHR data. Yu et al. (2015) developed a phenotyping algorithm to identify patients with rheumatoid arthritis who also suffered from coronary artery disease. In particular, they focused on investigating the use of automated extraction of NLP text features, which combined with structured codes, were used as input to a regularized logistic regression classifier. Teixeira et al. (2017) developed and evaluated several different phenotyping algorithms and categories of EHR information to identify hypertensive cases and controls.

The survey by Shivade et al. (2013) showed that cancer and diabetes are, by far, the two most common phenotypes to study. 49 of the included articles studied cancer or diabetes, whereas only 31 of the articles studied any other phenotype (heart failure, rheumatoid arthritis, cataract, pneumonia, etc.). However, there are also works focusing on more rare phenotypes, such as special types of voice disorders (Ghassemi et al., 2014b, 2016).

Reducing labeling efforts One line of research within phenotyping has focused on methods for exploiting noisy labeled training data in order to reduce labeling efforts. Examples include so-called anchor learning and silver standard learning (Halpern et al., 2016; Agarwal et al., 2016). These are two very similar frameworks. In (Agarwal et al., 2016), the silver standard labels were created using descriptive phrases from the clinical notes such as e.g. "type 2 diabetes mellitus". Halpern et al. (2016) created noisy labels using so-called anchors, which are highly informative, clinically relevant variables, typically defined by clinical experts. These methods of course provide some wrongly labeled instances, but according to the theory in noisy label learning (Simon, 1996; Aslam and Decatur, 1996), the error that these models make compared to identical models trained on clean labels is bounded and can be compensated for by using enough training examples. Bulk learning (Chiu and Hripcsak, 2017) is a hierarchical learning framework based on ensemble learning that uses a sparsely annotated training set to evaluate many phenotypes at once, which do not require much intervention of clinical experts. In particular, their focus was on phenotyping infectious diseases.

Unsupervised discovery of phenotypes

With their perspective article (Hripcsak and Albers, 2012), Hripscak and Albers introduced a shift from expert crafted phenotypes to electronic, simultaneous generation of many phenotypes via so-called *high-throughput phenotyping*. This idea was further elaborated on in a more recent perspective piece (Hripcsak and Albers, 2018), which also introduced a new term, namely *high-fidelity phenotyping*.

High-throughput phenotyping can be referred to as the process of mapping the raw data from the EHRs into medical concepts or representations that are meaningful to a medical expert, which in turn can be used for further research (Ho et al., 2014a; Albers et al., 2018). Hence, it can be thought of as a form of (unsupervised) representation learning, or dimensionality reduction, where the extracted features are informative (clinically meaningful) (Ho et al., 2014b). Then, the idea is that the extracted features actually are phenotypes themselves, either new ones or phenotypes known from before. Therefore, high-throughput phenotyping can also be referred to as unsupervised discovery of new phenotypes.

In particular, nonnegative tensor factorization (generalized nonnegative matrix factorization (Lee and Seung, 2001)) has been a very popular tool to use for designing models that performs high-throughput phenotyping, since it offers an effective approach to convert massive electronic health records into meaningful clinical concepts (phenotypes) (Ho et al., 2014a,b; Wang et al., 2015a; Chen et al., 2015; Gunasekar et al., 2016; Yang et al., 2017; Perros et al., 2017, 2018; Kim et al., 2017a,b; Henderson et al., 2018). However, many alternative approaches have also been proposed. For example, Pivovarov et al. (2015) presented the UPhenome model, a probabilistic graphical model for unsupervised phenotyping. Lasko et al. (2013) proposed a phenotype discovery method based on deep learning and Gaussian processes, whereas Che et al. (2015) introduced the deep computational phenotyping framework, in which deep neural networks were used to identify features associated with different diagnoses.

Other authors have investigated the use of topic models such as latent Dirichlet allocation (Blei et al., 2003). Ghassemi et al. (2014a) investigated the use of topic modeling to discover phenotypes from clinical narratives, whereas Chen et al. (2015) used latent Dirichlet allocation to translate EHR data into phenotype topics and investigated the portability of such topics across different institutions. Topic models also play a key role in PhenoLines (Glueck et al., 2018), a visualization tool for easier interpretation of the phenotype topics (disease subtype topics).

The work of Yu et al. (2015), which we have already briefly discussed, can be thought of as a combined approach for high-throughput phenotyping (unsupervised phenotype discovery) and phenotyping (patient cohort identification). This method was improved and refined with the *surrogate-assisted feature extraction* (SAFE) framework (Yu et al., 2017a). In the method, candidate features are selected by extracting medical concepts (UMLS concepts) using named entity recognition on articles from five sources[2]. The final task is to predict a target phenotype Y, and in order to do so, the corresponding ICD-9 and NLP (UMLS) counts are used to create noisy labels. The candidate features and noisy labels are fed into an elastic-net logistic regression, which is used to selected a subset of highly predictive features from the set of candidate features. The final phenotyping classifier (for the phenotype Y) is then trained using gold-standard labels and the selected subset of features. SAFE was used to identify patients suffering from coronary artery disease, rheumatoid arthritis, Crohn's disease, and ulcerative colitis.

Other work on phenotyping

Boland et al. (2015) proposed the Classification Approach for Extracting Severity Automatically from Electronic Health Records (CAESAR), a method for classifying severity at the phenotype-level based on random forests. By classifying severity at the phenotype-level, it is meant to distinguish between e.g. mild and severe variants of the same condition. A concrete example is acne and myocardial infarction. In contrast, patient-level severity determines if a given patient has a mild or severe form of the condition. With the goal to reduce labeling efforts, CAESAR was adapted using active learning (Settles, 2012) to the CAESAR-Active Learning Enhancement framework (Nissim et al., 2015, 2017).

One line of research aims particularly at accounting for the temporal and dynamic nature of the EHRs while performing phenotyping. Dagliati et al. (2017) used careflow mining (Quaglini et al., 2001) for electronic temporal

[2] Wikipedia `https://www.wikipedia.org/`, Merck Manuals `https://www.msdmanuals.com/`, Medscape `https://www.medscape.com/`, Mayo Clinic Diseases and Conditions `https://www.mayoclinic.org/diseases-conditions/index`, and MedlinePlus Medical Encyclopedia `https://medlineplus.gov/encyclopedia.html`.

phenotyping, whereas Liu et al. (2015) proposed a graph based framework for the same purpose. In this regard, we also highlight the phenotyping frameworks based on non-linear time series analysis (Albers et al., 2014; Hripcsak et al., 2015) and *the Care Pathway Explorer* (Perer et al., 2015), in which frequent sequence mining is used to extract sequences of medical events that can be visualized via the provided user interface.

3.2 Representation learning

Several works have focused on methods for learning domain appropriate representations of the EHR that account for its unique characteristics (multiple modalities, temporality, etc.) and can be useful for further analysis such as e.g. predictive modeling. We note that phenotyping (phenotype discovery, in particular) and learning EHR representations often are two closely connected tasks. Hence, several methods described in the previous section also belong to this section, and vice versa.

Representation of unstructured clinical notes

Bag-of-words (simple frequency counts of words) is often the standard choice for representing clinical text, and it has been shown that for a variety of clinical tasks, bag-of-words types of representations yield comparable performance to more advanced NLP methods (Jung et al., 2015). Latent Dirichlet allocation and other *topic models* have also been popular choices for representing clinical notes. For example, Rumshisky et al. (2016) examined the use latent Dirichlet allocation to decompose clinical notes into meaningful features, and used these features to predict early psychiatric readmission.

Miotto et al. (2016) proposed the *Deep patient* framework, an unsupervised deep feature learning method based on denoising autoencoders (Vincent et al., 2010). As input to the three-layer denoising autoencoders, they used latent Dirichlet allocation compressed vectors of bag-of-words counts of coded EHR data and concepts extracted from clinical notes. Beaulieu-Jones et al. (2016) also used denoising autoencoders to learn patient representations, but tested the framework only on synthetic data. In Paper 11, we used autoencoders to learn patient representations from free text nurses notes and explored the use of these representations in an anchor learning phenotyping framework.

Skip-gram Learning *word embeddings*, which are techniques for mapping words or phrases to real-valued vectors, has proven useful in many NLP tasks (Bengio et al., 2003). In particular, the skip-gram model (word2vec) of Mikolov et al. (2013) has been popular in use for EHR applications. Word2vec is based on two-layer neural networks and capable of capturing complicated relationships between words. For instance, Minarro-Giménez et al. (2014) used skip-gram to learn embeddings of medical concepts from unstructured text extracted from several different medical text corpora. Instead of training the skip-gram model directly on terms extracted from clinical notes, De Vine et al. (2014) first extracted UMLS concepts from the notes and thereafter trained the model over sequences of such concepts.

NLP for non-English clinical notes The extent of research on clinical NLP for non-English records is still quite limited, partly because of scarce availability of shared annotated corpora and medical dictionaries (Velupillai et al., 2015; Viani et al., 2017b). However, there are some works that have focused on creating representations of free text from non-English records.

Viani et al. (2017b) created a system for extracting and summarizing information from Italian EHRs based on a NLP pipeline in which one of the components was a Support Vector Machine (SVM). Viani et al. (2017a) explored the use of Recurrent Neural Network (RNN) (Elman, 1990) architectures for clinical event extraction from Italian EHRs. Dalianis and colleagues have several publications on NLP for Swedish clinical notes (Dalianis et al., 2012; Henriksson et al., 2014, 2015; Skeppstedt et al., 2014; Velupillai et al., 2014; Weegar et al., 2015; Jacobson and Dalianis, 2016; Perez et al., 2017).

We also mention that in Paper 6, we represented patients by converting Norwegian EHR free text into conceptual information. In more detail, the unstructured EHR text was matched with concepts corresponding to diseases, drugs and surgical procedures in MeSH Norwegian[3] using the Smith-Waterman algorithm (Smith and Waterman, 1981). Moreover, to learn to distinguish between real-time data describing the state of the patient, and retrospective data and noise (negations and internal communication etc.), three naive Bayes classifiers (Russell and Norvig, 2016) were trained to separate between real-time and history, real-time and noise, and history and noise, respectively. From the learned patient representation we created disease trajectories and used them for identification of cancer patients in need

[3] See Section 2.1 for an explanation of MeSH Norwegian.

of resource demanding treatment and/or readmission, and for identification of events enabling individual risk estimation of subsequent events. For a survey on clinical NLP in non-English languages, we refer to (Névéol et al., 2018).

Representation of longitudinal structured EHR data and/or unstructured EHR data

Skip-gram and GRAM Instead of free text notes, Choi et al. (2016f,d,b) focused on structured longitudinal visit records, and used skip-grams to learn representations of medical concepts such as e.g. ICD diagnoses codes, CPT procedure codes and SNOMED-CT codes. Farhan et al. (2016) modified the skip-gram models to support dynamic windows and thereby create contextual embedding representations of sequential medical events such as diagnoses, prescriptions, and laboratory tests. With the graph-based attention model (GRAM), Choi et al. (2017) focused particularly on creating representations of medical concepts that accounts for the hierarchical information inherent to medical ontologies. In an attempt to minimize the need for preprocessing and tuning of parameters, Bajor et al. (2018) proposed a representation learning framework that also was based on a methodology closely related to skip-gram.

Bayesian approaches and Gaussian processes. Many recent works have focused on modeling longitudinal clinical data using Bayesian approaches and Gaussian processes. Albers et al. (2018) developed a method for learning to represent, or automatically summarize, raw laboratory data by taking a Bayesian approach wherein parametric models (Weibull) and concepts from information theory (Kullback-Leibler divergence) played a central role. Ghassemi et al. (2015) used multi-task Gaussian process models for multivariate time series modeling of both physiological signals and clinical notes, and used the models to assess the severity of illnesses. Caballero Barajas and Akella (2015) used dynamic Bayesian networks to model clinical data as time series of topics and calculate probabilities of mortality. Krishnan et al. (2017) proposed the deep Markov model, which is a generative model where a multi-layer perceptron is used instead of linear emission and transition distributions, to model sequential data. A survey on dynamic (temporal) Bayesian networks applied to temporal clinical data can be found in (Orphanou et al., 2014).

Boltzmann machines and autoencoders Mehrabi et al. (2015) used diagnoses codes recorded over time as inputs to a Boltzmann machine (Aarts and Korst, 1989). In Paper 13, we proposed a framework for creating compressed representations of multivariate time series representations of blood tests using autoencoders (Hinton and Salakhutdinov, 2006) and kernel alignment with the Time series Cluster Kernel (TCK) (see Paper I). A key component in the framework was the deep kernelized autoencoder (Kampffmeyer et al., 2018).

RNNs A great deal of works apply RNNs to clinical time series data. (Lipton et al., 2015) used Long Short-Term Memory (LSTM) RNNs (Hochreiter and Schmidhuber, 1997) to establish a framework for multi label classification of diagnoses based on multivariate time series representations of structured clinical data such as vital signs and laboratory tests. This framework was extended and adapted in (Lipton et al., 2016) such that it directly modeled missing data using binary indicator variables instead of using heuristic imputation. Similarly, Pham et al. (2016) used LSTM RNNs to create representations of admission episodes, described by diagnoses and interventions. Paper 17 presents a representation learning framework for multivariate time series subject to missing data based on stacked bidirectional RNNs and kernel alignment with the TCK. Among other things, we used the method to learn vector representations of blood sample data from the Tromsø EHR corpus.

In the DOCTOR AI framework, Choi et al. (2016a) used RNNs with Gated Recurrent Units (GRU) (Cho et al., 2014; Chung et al., 2014) for representing longitudinal patient visit records and prediction of diagnosis, medication order and visit time. Che et al. (2018) proposed a novel GRU RNN architecture, namely GRU-D, to better account for informative missing patterns in the multivariate time series, whereas Vani et al. (2017) introduced a RNN architecture, called Grounded recurrent neural network, and used it to understand what medical concepts were mentioned and discussed in patients' discharge summaries.

Convolutional neural networks Suresh et al. (2017) focused on learning representations from all available ICU sources (vitals, labs, notes, demographics) to predict multiple invasive interventions. For this purpose they used both LSTM RNN and Convolutional Neural Network (CNN)s (Krizhevsky et al., 2012). Similarly, with the Health-ATM, Ma et al. (2018) proposed a hybrid RNN and CNN network to incorporate both attention and time-awareness. (Razavian et al., 2016b) also explored the use of both LSTM RNNs and CNNs. CNNs were also the key component in *Deepr*, a general framework for extracting features from EHRs and risk prediction (Nguyen et al., 2017). Rajkomar et al. (2018) proposed a to create patient representations by mapping raw EHR records to the Fast Healthcare Interoperability Resources (FHIR) format (Mandel et al., 2016) and demonstrated that deep learning methods trained on this representation can provide accurate predictive models.

Other methods Various other methods have also been employed to learn representations of longitudinal clinical data. Zhao et al. (2015b) explored three different 'bag-of-words' strategies, namely bag of events, bag of binned events, and bag of weighted events. Zhao et al. (2017) proposed to use different symbolic sequence representations of temporal clinical data. Key components were the use of symbolic aggregate approximation (SAX) (Lin et al., 2003) and time series subsequences (shapelets (Ye and Keogh, 2009)). Moskovitch et al. (2017) introduced a framework based on time interval mining analytics for longitudinal clinical data, and several different works have created representations using temporal association rules (Orphanou et al., 2016, 2018, 2014).

Dubois et al. (2017) explored the use of transfer learning (Pan et al., 2010) with RNNs to learn representations of clinical notes in cases when the training set is small (less than 1000 patients), but found that a conceptually simple NLP approach, called embed-and-aggregate, provided competitive results on various predictive modeling tasks. Embed-and-aggregate learns vector space embeddings for medical concepts in which each element is a real vector using the GloVE algorithm (Pennington et al., 2014) and large unannotated text corpora.

3.3 Patient similarity

Often, when the learned vectorial representations of (temporal) clinical data is used for further analysis (e.g. predictive modeling), one measures how similar, or dissimilar, the representations are using Euclidean distance. However, some times other metrics or measures of similarity can prove more useful, and therefore, as an alternative, one can also learn suitable measures for the *similarity* between pairs of patients.

Appropriate patient similarity measures are essential, for example, in order to allow meaningful stratification of patients into subgroups (Parimbelli et al., 2018; Brown, 2016; Hu et al., 2016). Once a suitable patient similarity measure is defined, it can be used to identify cohorts of patients that are similar relative to an index patient, enabling personalized predictions and personalized medicine (Sharafoddini et al., 2017). Patient similarity also play a key role in initiatives such as *Patients like me*[4].

Subtyping of diseases or patients One of the most common uses of patient similarity is for *disease subtyping* or *patient subtyping*. Using *clustering* one can, for example, identify subgroups of patients with similar disease evolution. Examples of diseases that have been subtyped using clustering include autism spectrum disorders (Doshi-Velez et al., 2014; Lingren et al., 2016), autoimmune diseases (Schulam et al., 2015), hypotension (Dai et al., 2017), Parkinson's disease (Lewis et al., 2005), cystic fibrosis and Crohn's disease (Chen et al., 2007), juvenile idiopathic arthritis (Cole et al., 2013) and hypertension (Chen et al., 2016a).

Clustering has also been used for other patient stratification tasks such as subgrouping of ICU patients (Vranas et al., 2017). Baytas et al. (2017) proposed a general patient subtyping framework based on so-called time-aware LSTM RNN Networks to irregular time intervals in the longitudinal EHRs. The network produced a vector representation of the patients, which in turn was clustered using k-means.

The majority of the papers mentioned above exploited existing clustering methods such as hierarchical clustering, k-means, consensus clustering, or model based clustering (see Section 5.1) in order to extract new knowledge for a specific disease. However, some authors have also been developing new clustering methods for patient stratification. These works include Bayesian biclustering (Khakabimamaghani and Ester, 2016) and mixture models for multivariate clinical time series with missing data (Marlin et al., 2012).

Li et al. (2015) presented a clustering framework based on topological data analysis (Carlsson, 2009) to identify type 2 diabetes subgroups using both molecular and clinical EHR data. Schulam et al. (2015) proposed the probabilistic subtyping model, which is a clustering method for time series of clinical markers obtained from routine visits to subgroup similar patients.

In Paper 5 (prior work presented in Paper 14 and 15), we developed a novel consensus clustering method based on mode seeking, which we in a case study applied to subtype patients undergoing major abdominal surgery based on free-text from nurses notes.

[4] `https://www.patientslikeme.com/`

Other patient similarity applications In addition to disease subtyping and patient stratification, patient similarity measures have also played a key role in various other EHR applications. These include:

- Personalized predictive modeling (Ng et al., 2015; Lee et al., 2015; Suo et al., 2017).
- Treatment recommendation (Zhang et al., 2014; Wang et al., 2015b).
- Exploring drug effects (Ghalwash et al., 2017).
- Clinical decision support (Gottlieb et al., 2013; Gallego et al., 2015).
- Visualization (Cahan and Cimino, 2016; Kwon et al., 2018).
- Predicting disease trajectories and disease progression modeling (Ebadollahi et al., 2010; Mould, 2012; Wang et al., 2014c).

Disease trajectories and disease progression modeling have also been studied under other frameworks such as dynamic topic modeling (Elibol et al., 2016) and Gaussian processes (Schulam and Saria, 2016; Futoma et al., 2016a,b). In this regard, we also note that Paper 6, which we discussed above, focuses on the identification of cancer patient trajectories.

Similarity measures Some works apply existing similarity measures to evaluate patient similarity. For example, Ebadollahi et al. (2010); Sun et al. (2010b,a, 2012); Ng et al. (2015) explored the use of *local supervised metric learning* (Wang and Zhang, 2007), which is a supervised similarity measure that has proven useful for patient similarity evaluation.

Other authors propose novel similarity measures, designed for different purposes. For example, Wang (2015); Huang et al. (2014) propose methods for measuring similarity between patients in terms of clinical or diagnostic patterns, Zhang et al. (2014); Ghalwash et al. (2017) designed drug similarity measures, Ramos et al. (2016) proposed a similarity measure for radiology reports, whereas Wang and Sun (2015a) proposed a general patient similarity framework.

The underlying methodology used for defining the similarity measures vary quite significantly between the different methods. For example, Suo et al. (2017); Zhu et al. (2016) built CNN-based similarity measures, whereas Zhan et al. (2016) used Mahalanobis distance and low-rank sparse feature selection as key components in the similarity measure, and (Sha et al., 2016) used the Smith-Waterman algorithm (Smith and Waterman, 1981) to define a similarity measure for temporal and irregularly sampled EHR data.

Kernels form a particular class of similarity measures as they satisfy the positive-semidefiniteness property (see Section 4). In further analysis, these measures can be used as inputs to kernel machines such as the SVM. Soguero-Ruiz et al. (2016a) used kernel methods to predict postoperative complications. Bogojeska et al. (2012) introduced the *history-alignment model*, which is a kernel approach that predicts whether the outcome of particular HIV therapy choices are successful. Parbhoo et al. (2017) modified the history-alignment model and combined it with model-based reinforcement learning (Sutton et al., 1998) via a mixture-of-experts approach (Jordan and Jacobs, 1994), and used the framework to recommend a HIV therapy.

Some works have focused on designing kernels for multivariate time series and applying them to EHR data. For example, Kale et al. (2014) applied a linear kernel (Euclidean distance), the global alignment kernel (Cuturi, 2011), vector autoregressive kernel (Cuturi and Doucet, 2011), kernelized locality sensitive hashing (Kulis and Grauman, 2012) as well as a non-valid (not positive-semidefinite) kernel, dynamic time warping (Berndt and Clifford, 1994), to three clinical datasets. We refer to (Kale et al., 2014) and Papers I and III for more related work on this topic, and to Section 4 for more background on kernel methods. As a remark, we also note that in Paper 13 and 17, a key component is *kernel alignment* (Cristianini et al., 2002), which is a measure of similarity either between two kernels or between a target function and a kernel.

3.4 Predictive modeling

A great deal of work on machine learning for EHRs focuses on predictive modeling, i.e. to learn models that can transform input data (e.g. EHRs) into a prediction of the outcome of a variable of interest (e.g. a disease). In particular, models for predicting the risk of developing certain diseases have been popular to study. Examples include type 2 diabetes (Razavian et al., 2016a; Albers et al., 2017; Dagliati et al., 2018), congestive heart failure (Choi et al., 2016e,c; Cheng et al., 2016), chronic obstructive pulmonary disease (Cheng et al., 2016), Parkinson's disease (Che et al., 2017), sepsis (Futoma et al., 2017a,b), among many others. Other authors have focused on complications and adverse events of various types, such as diabetes complications (Liu et al., 2018a), adverse drug events (Zhao et al., 2014a,b, 2015a), surgical site infection (Ke et al., 2017; Shankar et al., 2018; Soguero-Ruiz et al., 2015; Sanger et al., 2016), and anastomosis leakage (Soguero-

Ruiz et al., 2016a), to name a few.

Mortality prediction has been a popular task (Johnson and Mark, 2017; Lee et al., 2015; Aczon et al., 2017). In addition, researchers have focused on various other tasks, such as prediction of readmission (Rumshisky et al., 2016; Shameer et al., 2017; Xue et al., 2018), wound healing (Jung and Shah, 2015; Jung et al., 2016), future high-cost patients (Tamang et al., 2017), comorbidities (Yousefi et al., 2017), age (Wang et al., 2017b), drug interactions (Zhang et al., 2015), and onset of clinical interventions such as vasopressors and mechanical ventilation (Suresh et al., 2017; Ghassemi et al., 2017; Wu et al., 2017; Ren et al., 2018).

3.5 Other uses of machine learning for EHRs

It is impossible to provide a comprehensive and complete review of all research papers on uses of machine learning for EHRs. However, to conclude this chapter, we provide some examples of other uses that have not been discussed so far.

- Uncertainty-aware prediction using Bayesian modeling (Soleimani et al., 2018).
- Interpretable models (Wickstrøm et al., 2018; Bai et al., 2018; Wang et al., 2017a; Ross et al., 2017; Doshi-Velez and Kim, 2018; Zhang et al., 2018).
- Privacy and discrimination (Boag et al., 2018).
- Counterfactual models for reliable clinical decision support (Schulam and Saria, 2017).
- Behavioral modeling (Hao et al., 2017).
- Doctor recommendation (Guo et al., 2016).
- Risk profiling (Shah et al., 2015).
- Treatment recommendation and estimation of treatment response (Zhang et al., 2017; Raghu et al., 2017; Xu et al., 2016; Soleimani et al., 2017). Observational research via treatment pathways (Hripcsak et al., 2016).
- Identification of reference intervals for laboratory tests (Poole et al., 2016).
- Active surveillance of diagnostic accuracy (Schroeder et al., 2016).
- Learning knowledge bases linking diseases and symptoms (Rotmensch et al., 2017).

Part II

Methodology and context

Chapter 4

Kernel methods

This chapter presents background theory on kernel methods, which is relevant to Paper I and III. Kernel methods became popular in the late 90's and have since then been an important part of machine learning and pattern recognition (Shawe-Taylor and Cristianini, 2004; Jenssen, 2010, 2013; Gu and Sheng, 2017; Løkse et al., 2017; Kampffmeyer et al., 2018; Camps-Valls and Bruzzone, 2009; González et al., 2018).

While the theoretical and mathematical foundations of kernel methods might appear involved and difficult to understand for people not familiar with the field, the underlying idea is simple and intuitive: Most learning algorithms aim at learning a function f over a set $\mathcal{X}$. One of the easiest and most well understood such functions, notably for classification, regression and dimensionality reduction, is the linear function,

$$f(\mathbf{x}) = \langle \mathbf{a},\, \mathbf{x} \rangle = \mathbf{a}^T\mathbf{x}. \tag{4.1}$$

The linear function is easily interpretable, but in many practical scenarios the data are not linearly separable and therefore the linear methods do not provide the wanted performance. The idea in kernel methods is, however, to map the data to a new high-dimensional space where data are linearly separable. Hence, kernel methods are also linear in nature, but since they can be expressed solely in terms of inner-products and because of the so-called *kernel trick*, one can avoid explicit calculation of the high-dimensional representations of the data. Instead, the function f can be expressed in terms of a kernel k, which is computed in the input space,

$$f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i,\, \mathbf{x}). \tag{4.2}$$

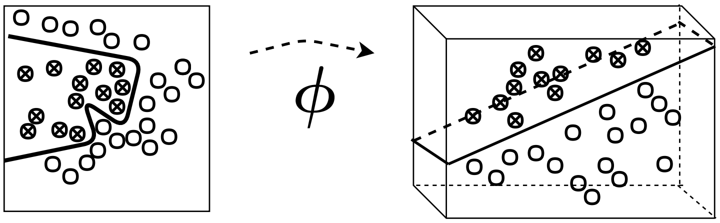


Figure 4.1: *Conceptual illustration of a kernel machine. Non-linear data are mapped to a higher-dimensional space where it is linearly separable.*

The function k measures how similar two elements $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ are via the value $k(\mathbf{x}, \mathbf{y})$, and since this function might be non-linear, kernel methods can also be applied to non-linear problems. We will return to the expression (4.2) later. However, before that we will take a look at the kernel k.

4.1 Kernels

In mathematics, the term *kernel* is highly ambiguous. Also within kernel methods, a kernel can be defined in several different, but equivalent ways.

Definition 1 *Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if there exists a $\mathbb{R}$-Hilbert space $\mathcal{H}$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that $\forall \mathbf{x},\ \mathbf{y} \in \mathcal{X}$,*

$$k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathcal{H}}. \tag{4.3}$$

The function Φ is often referred to as a *feature map* and $\mathcal{H}$ the *feature space* of the kernel k. Eq. (4.3) forms the basis of a key concept in kernel methods, namely the kernel trick. Any (linear) algorithm that can be expressed solely in terms of inner-products can be kernelized simply by replacing the inner-products with any kernel. Hence, by explicitly computing the kernel $k(\mathbf{x}, \mathbf{y})$ over the original input space, one can implicitly map the data to a possibly infinite-dimensional space, where hopefully the data are linearly separable. An equivalent definition of kernels, which is useful in practice, is as follows.

Definition 2 *Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a* *kernel if, for any $n \in \mathbb{N}$, $\{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^n$, and $\{c_i \in \mathbb{R}\}_{i=1}^n$*

$$k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i) \tag{4.4}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0. \tag{4.5}$$

Def. 2 states any *symmetric* (Eq. (4.4)) and *positive semidefinite* (Eq. (4.5)) function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be used to design a kernel. Another practical aspect, which is related to Def. 2, is that typically in kernel machines one has to solve optimization problems that contain terms of the form $\mathbf{c}^T K \mathbf{c}$, where K is a matrix of kernel evaluations ($K = \{k(\mathbf{x}_i, \mathbf{x}_j)\}$). Eq. (4.5) ensures that also the kernel matrix (Gram matrix) K is positive semidefinite, which implies that the optimization problems can be solved efficiently using *convex programming* and the algorithms converge to the relevant solution (Boyd and Vandenberghe, 2004). The interested reader can find a proof of the equivalence between Def. 1 and Def. 2 in (Schölkopf et al., 2002), i.e that it is possible to take one of them as definition and prove that the other follows from that, and vice versa.

Reproducing kernel Hilbert spaces In addition to the feature map view in Def. 1 and the positive semidefiniteness view in Def. 2, kernels can also be viewed from a functional analysis viewpoint (Hille and Phillips, 1996). In that respect, the concepts *reproducing kernels* and Reproducing Kernel Hilbert Space (RKHS) are of major importance.A reproducing kernel is defined as follows.

Definition 3 *Let $\mathcal{X}$ be a non-empty set, $\mathcal{H}$ a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$, and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a function. Then, k is a reproducing kernel if*

$$\mathcal{H} = \overline{span\{k(\mathbf{x}, \,\cdot\,) \mid \mathbf{x} \in \mathcal{X}\}}, \tag{4.6}$$

$$\langle f, k(\,\cdot\,, \mathbf{x})\rangle_{\mathcal{H}} = f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{X}, \forall f \in \mathcal{H}, \tag{4.7}$$

where $\bar{S}$ denotes completion of the set S. Eq. (4.6) states that k has to span $\mathcal{H}$, whereas Eq. (4.7) is commonly referred to as the reproducing property.

The definition of a RKHS is closely connected to the previous definition:

Definition 4 *A RKHS is a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathbb{R}$, with a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.*

These concepts are tightly connected to kernels. In fact, reproducing kernels are kernels, which means that every RKHS defines a kernel. Conversely, the Moore-Aronszajn theorem states that every kernel is associated with a unique RKHS (Aronszajn, 1950). Hence, the functional analysis viewpoint (Def. 3 and 4) is equivalent both to the positive-semidefiniteness viewpoint (Def. 2), and the feature map viewpoint (Def. 1).

Remark. So far we have not mentioned the well-known *Mercer's theorem*, which is commonly used to construct a feature space for a valid kernel. We note that the theorem is not required in itself, since the RKHS construction serves the same purpose (Shawe-Taylor and Cristianini, 2004). However, the advantage of using Mercer's theorem is that it defines the feature space explicitly in terms of feature vectors instead of using a function space.

4.2 Representer theorem

Now that we have explained some properties of the kernels, let us return to Eq. (4.2). The reason why the output of most kernel machines can be written in that form is that they can be formulated as regularized empirical risk optimization problems in a RKHS (Schölkopf et al., 2001). In fact, Eq. (4.2) is actually a direct consequence of the *Representer theorem* (Kimeldorf and Wahba, 1971), which can be stated as follows.

Theorem 1 *Let $\mathcal{X}$ be a non-empty set endowed with a kernel k and let $\mathcal{H}_k$ be its corresponding RKHS. Further, let $\{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^n$ and $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ be a function that is strictly increasing for the last argument. Then, any $f \in \mathcal{H}_k$ that minimizes the (regularized) empirical risk functional*

$$\Psi(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n), \|f\|_{\mathcal{H}_k}) \tag{4.8}$$

admits a representation of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i,\, \mathbf{x}), \quad \alpha_i \in \mathbb{R}. \tag{4.9}$$

This is a very strong result, because it states that a whole range of learning algorithms that are formulated as potentially infinite-dimensional empirical risk minimization problems have a solution that can be expressed as a finite linear combination of kernels centered at the training points. Hence, the

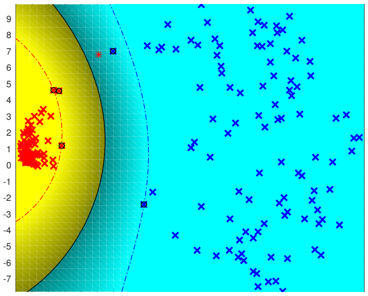


Figure 4.2: *Illustration of a SVM classifier trained on a simple 2D dataset.*

dimensionality of the solution is finite, and, at most, equal to the number of training points.

The SVM is the best known kernel machine and can be used for classification, regression, anomaly detection, and more (Cortes and Vapnik, 1995; Steinwart and Christmann, 2008). In the (canonical) support vector classifier, the empirical risk is given by

$$\Psi(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n),\|f\|_{\mathcal{H}_k}) = \frac{1}{\lambda}\sum_{i=1}^{n}\max(0,1-y_i f(\mathbf{x}_i)) + \|f\|_2^2 \qquad (4.10)$$

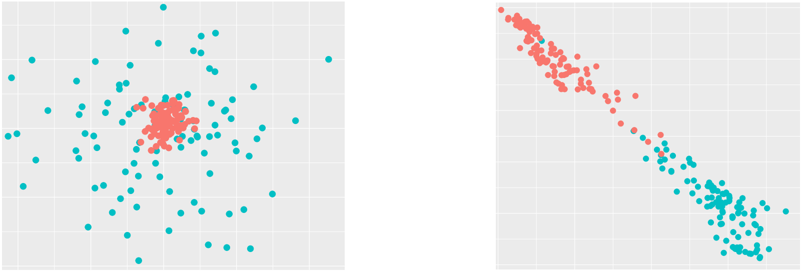
Even though it is not explicitly stated in Eq. (4.10), the empirical risk for the SVM is also a function of the labels y_i. We note that for the SVM, the representer theorem has to be slightly modified to account for the constant term b (Schölkopf et al., 2001), and the solution is given by $sign(\sum_{i=1}^{n}\alpha_i k(\mathbf{x}_i,\,\mathbf{x})+b)$. Fig. 4.2 shows a simple example of a SVM classifier trained to separate between two classes. The figure shows both the decision line between the classes, and the so-called support vectors.

4.3 Examples of kernel machines

The kernelized version of Principal Component Analysis (PCA)[1] for dimensionality reduction, namely Kernel Principal Component Analysis (KPCA), is an other well-known kernel machine (Schölkopf et al., 1997). It has been shown that for KPCA, the empirical risk can be written as

$$\Psi(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n),\|f\|_{\mathcal{H}_k}) = \psi(f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n)) + \phi(\|f\|_{\mathcal{H}_k}), \qquad (4.11)$$

[1] For a brief introduction to PCA, see Section 5.2.



(a) PCA. (b) KPCA.

Figure 4.3: *Dimensionality reduction of a synthetic two-class dataset consisting of two-variate time series using PCA and KPCA, respectively.*

where

$$\psi(f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)) = \begin{cases} 0, & \text{if } \sum\limits_{i=1}^{n} \left(f(\mathbf{x}_i) - \frac{1}{n} \sum_{j=1}^{n} f(\mathbf{x}_j) \right)^2 = n \\ \infty, & \text{otherwise,} \end{cases} \tag{4.12}$$

and ϕ is an arbitrary strictly increasing function (Schölkopf et al., 1999).

For convenience, we also summarize the KPCA-algorithm here. Given a dataset $\{\mathbf{x}_i\}_{i=1}^{n}$, a kernel K and a point $\mathbf{x} \in \mathbb{R}$, first compute the matrix $\mathcal{K} = (K(\mathbf{x}_i, \mathbf{x}_j))$, secondly find the k dominant $\{\lambda_i, \mathbf{a}_i\}$ s.t. $\mathcal{K}\mathbf{a}_i = n\lambda_i\mathbf{a}_i$ and $||\mathbf{a}_i||^2 = (n\lambda_i)^{-1}$ and then, eventually compute the projections $y(j) = \sum_{i=1}^{n} a_j(i) K(\mathbf{x}_i, \mathbf{x})$ for $j = 1, 2, .., k$.

Fig. 4.3 shows an example of dimensionality reduction of a synthetic two-class dataset consisting of 200 two-variate time series using PCA and KPCA with a non-linear time series kernel, respectively. KPCA is capable of creating a two-dimensional representation of the data wherein the classes are (almost) linearly separable, whereas this is not the case for PCA.

Examples of other kernel machines include kernel Fisher discriminant analysis (Mika et al., 1999), the kernel perceptron (Bordes et al., 2005), kernel k-means and spectral clustering (Dhillon et al., 2004), and kernel ridge regression (Shawe-Taylor and Cristianini, 2004), to name a few. We also note that while Gaussian processes (Rasmussen, 2004) were not originally formulated in terms of a regularized optimization problem in a RKHS, but rather in terms of marginal and conditional distributions, this family of methods is closely connected to the kernel methods (Kanagawa et al., 2018). One of the reasons is that a key component in Gaussian processes is to use positive-semidefinite kernel functions.

4.4 Examples of kernels

We have now seen that kernel methods are formulated in terms of notions like RKHS, optimization and regularization, and that at the end of the day, for most kernel methods, everything boils down to learning functions of the form $f(\cdot) = \sum_i \alpha_i k(\mathbf{x}_i, \, \cdot \,)$. More specifically, given a kernel k and training data $\{\mathbf{x}_i\}_{i=1}^n$, the algorithms learn the coefficients α_i.

The only problem is that the kernel is not "given". It actually has to be selected by the user, and, as we have seen, the selected kernel defines a unique Hilbert space of functions (Moore-Aronszajn theorem), which is exactly the function space from where the candidate functions that the algorithm aims at learning are selected. I.e., the fact that the kernel defines this function space, largely dictates what types of functions the kernel machine can learn. Hence, depending on what type of kernel that is selected, the properties of the kernel machine can change considerably.

An additional aspect is that the selected kernel should also be a good measure of how similar pairs of objects are. However, all kernels are not suitable similarity measures for any data type. Hence, the bottom line is that one must select an appropriate kernel for the data analysis task at hand. Next, we list examples of some of the most common kernels.

Polynomial kernel. $k(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \, \mathbf{y} \rangle + r)^d, \quad r \geq 0, \; d \in \mathbb{N}.$

Radial Basis Function (RBF). $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{y}\|^2\right), \quad \sigma > 0.$

Laplacian kernel. $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\alpha\|\mathbf{x} - \mathbf{y}\|\right), \quad \alpha > 0.$

Shared properties of these kernels are that

(i) they are dependent on hyper-parameters and it is often critical to tune these correctly to in order to achieve the desired performance;
(ii) they are designed for vector-based inputs.

For other examples of kernels, we refer to (Hofmann et al., 2008; Shawe-Taylor and Cristianini, 2004).

4.5 Why should we use kernel methods?

We have already discussed several advantages of kernel methods, but would like to highlight two additional aspects here.

Real-world data are not necessarily vectorial. One of the great advantages of kernel methods is that, unlike many traditional machine learning methods that are formulated only for vectors, positive semidefinite kernels can be defined on any non-empty set of any type of objects. Therefore, in addition to vectors, kernel methods can be applied to objects like e.g. graphs (Gärtner et al., 2003), histograms and probability measures (Hein and Bousquet, 2005), manifolds (Jayasumana et al., 2013), and multivariate time series that are of particular interest in this thesis (Paper I and III).

The reason why kernel methods can be straight-forwardly applied to non-vectorial data is that they can be thought of as a two-step procedure, where the first step consists in mapping n training points to a $n \times n$ similarity matrix using an appropriate kernel for the data at hand. Then, in the second step, one employs the learning algorithm with the similarity matrix as input. Hence, the learning part (second step) is independent of the complexity of the objects and the data type. In other words, the computational complexity of kernel methods is a function of the number of training points rather than the dimensionality of the input data. We note that this is not always an advantage. For instance, in large scale problems where the number of training points is very large, kernel methods are often very slow or not even feasible to train.

Real-world data are often heterogeneous. An additional aspect of kernel methods is *multi-modality*. An increasing amount of real-world datasets are collected from multiple sources. In fact, we have already seen that one of the challenges for data-driven healthcare is multi-modal data.

The classical way to deal with multi-modal data is to train individual models on separate modalities, and then, after training, aggregate the decision functions. On the other hand, in kernel methods, this problem is mitigated in a more elegant way, namely by exploiting the fact that the convex combination of multiple kernels is also a kernel. Hence, one can design powerful composite kernels (Lanckriet et al., 2004a; Noble et al., 2004; Lanckriet et al., 2004b; Ben-Hur and Noble, 2005; Soguero-Ruiz et al., 2016a). This is one of at least two uses of the more general multiple kernel learning framework (Gönen and Alpaydın, 2011). Multiple kernel learning can also be used as a mean to select an appropriate kernel for the problem at hand in a data-driven way. From that point of view, the kernel we propose in Paper I can be considered as a particular multiple kernel learning method.

Chapter 5

Unsupervised learning

Unsupervised learning algorithms aim at describing data and extracting knowlegde from data without access to a labeled training set. As a background for the research presented in this thesis, in this chapter we describe the most classical type of unsupervised learning, namely clustering. In addition, we discuss ensemble learning and provide a brief introduction to dimensionality reduction.

5.1 Clustering

This section introduces clustering and presents relevant background material for Paper I, III, and IV.

Clustering is typically based on the notion of similarity. Given a dataset and a measure of similarity, a clustering algorithm aims at identifying subsets (clusters), such that the similarities between pairs of data points from the same cluster are high and the similarities between pairs not from the same cluster are low. However, clustering algorithms are not necessarily based on this notion. In fact, there exists no universal definition of what a cluster is (Jain, 2010; Guyon et al., 2009; Filippone et al., 2008; Jain et al., 1999). However, in this thesis we stick to the above definition.

In addition to a wide range of biomedical applications (Doshi-Velez et al., 2014; Lingren et al., 2016; Elakkia and Narendran, 2016; Smistad et al., 2015; Schulam et al., 2015; Dai et al., 2017; Lewis et al., 2005; Chen et al., 2007;

Cole et al., 2013; Chen et al., 2016a; Vranas et al., 2017) (see also (Parimbelli et al., 2018) and references therein), clustering has also been applied in areas such as astrophysics (Anderson et al., 2014), computational chemistry (Downs and Barnard, 2002), traffic analysis (Gopalakrishnan et al., 2016), web mining (Runkler and Bezdek, 2003), remote sensing (Gómez-Chova et al., 2012), credit scoring analysis (Mancisidor et al., 2018), among many others.

The combination of the facts that (i) the number of applications of clustering is large, (ii) clustering is both task and data dependent, and (iii) clustering is subjective and no universal definition of a cluster exists, has resulted in a large number of different clustering algorithms. There are several possible ways to categorize clustering methods, for example, into *hard* and *soft* clustering algorithms. In hard clustering, each data point belongs completely to one cluster. On the other hand, in soft clustering, each data point potentially can belong to multiple clusters, i.e. the probability of being a member of a cluster can be non-zero for more than one cluster.

Alternatively, from a statistical perspective, clustering algorithms roughly fall into three distinct categories according to how they deal with probability density estimation: *combinatorial* algorithms, (parametric) *model-based* clustering (mixture models), and (nonparametric) *modal* clustering (Hastie et al., 2009; Menardi, 2015). All these three categories are relevant to this thesis. Next, we provide a brief description of these three categories of clustering algorithms. For a more complete survey on clustering, we point the interested reader to the survey papers by von Luxburg (2007); Vega-Pons and Ruiz-Shulcloper (2011); Filippone et al. (2008); Jain (2010); Menardi (2015).

5.1.1 Combinatorial algorithms

Combinatorial algorithms do not explicitly model an underlying probability density function. Instead, they work directly on the observed data. In these algorithms typically the dataset is divided into clusters in such a way that each data point belongs to exactly one cluster. Additionally, the user has to set the number of clusters in advance.

Some of the most popular clustering methods belong to this category, and examples include k-means (Florek et al., 1951), hierarchical clustering (Gower and Ross, 1969; Seifoddini, 1989), and spectral clustering (Ng et al., 2002;

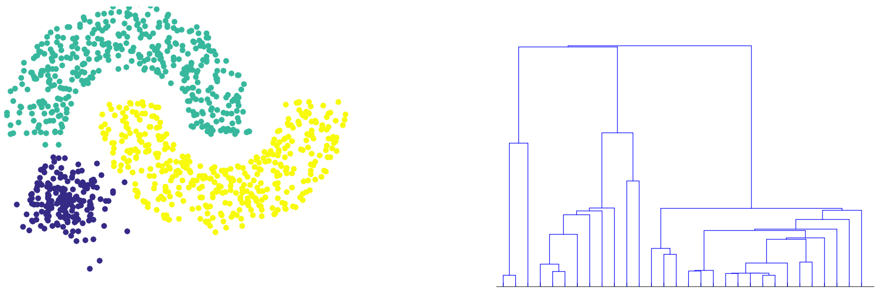
Løkse et al., 2017; Filippone et al., 2008).

k-means is an example of a *centroid*-based clustering algorithm. In these methods, one aims at learning cluster representatives (e.g. mean vectors), typically by optimizing a cost function. Similarities between data points are measured in terms of the Mahalanobis distance (elliptical metric), or the Euclidean distance (spherical metric) as a special case. For this reason, e.g. k-means can only identify convex clusters, i.e. linearly separable cluster structures.

In order to be able to identify non-convex clusters, k-means has been kernelized by mapping the input data to a high-dimensional space via a non-linear function (Schölkopf et al., 1998). Kernel k-means is closely connected to *spectral clustering* (Dhillon et al., 2004), which is a family of methods that exploits the spectrum of a similarity matrix to perform non-linear dimensionality reduction before clustering in the lower-dimensional space. Spectral clustering is often formulated as a graph-partition problem, wherein the objective is to minimize the normalized cut (Shi and Malik, 2000). These algorithms are also closely related to KPCA. In fact, KPCA combined with k-means is a spectral clustering algorithm.

Hierarchical clustering algorithms are different from those we have considered so far since they result in a set of nested clusters instead of a single clustering of the dataset. These algorithms can be divided into two subcategories, *agglomerative* and *divisive* methods. In the latter subcategory, one starts with only one cluster and then recursively divides the data points into more and more clusters, whereas in the agglomerative methods one initially consider each data point as a cluster and then recursively join similar clusters. In both cases, the result is a hierarchy of clusters, which often is organized as a tree (dendrogram). The agglomerative *linkage* methods (Gower and Ross, 1969; Seifoddini, 1989) are among the most prominent examples of hierarchical clustering algorithms.

Fig. 5.1a shows an example of a clustering of a synthetic two-dimensional dataset obtained via a hierarchical clustering algorithm. The corresponding dendrogram is shown in Fig. 5.1b. In hierarchical clustering, the number of clusters can be chosen according to the *longest lifetime* of the clusters. As we can see from the dendrogram, the longest lifetime in this example is achieved when the number of clusters is three.



(a) Example of a clustering of a synthetic two-dimensional dataset.

(b) Dendrogram.

Figure 5.1: Clustering example.

5.1.2 Model-based clustering

Model-based clustering, also referred to as distribution-based clustering or mixture modeling (McLachlan and Basford, 1988; Fraley and Raftery, 2002), assumes that the dataset is sampled Independent and Identically Distributed (IID) from a probability distribution. This distribution is assumed to be a mixture of several components. Each component of the mixture is associated with a cluster, and the data points are assigned to the cluster with the highest density according to the parametric model.

Mixture models can be thought of as generative models where data are generated according to a two-step random process as follows:

(i) select one out of K clusters by sampling from a (categorical) distribution $\pi = (\pi_1, \dots, \pi_K)$,

(ii) sample a data point according to the probability distribution, $p_k(X|\phi_k)$, of the selected cluster k.

The marginal distribution associated with this generative model is

$$p(X \mid \phi) = \sum_{k=1}^{K} \pi_k p_k(X \mid \phi_k), \tag{5.1}$$

where ϕ_k are the parameters that uniquely define the parametric model associated with cluster k and ϕ is the collection of parameters for all the K clusters. The parameters $\pi = (\pi_1, \dots, \pi_K)$ are commonly referred to as mixing coefficients, or mixing probabilities, and therefore satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

The mixture models are often expressed via latent variables, or more precisely, the cluster assignment of data point X is expressed in terms of a latent random variable, Z, with $Z = k$ if X belongs to cluster k. The data are then assumed to be generated according to

$$Z \sim Cat(\pi), \tag{5.2}$$
$$X \mid Z = k \sim p_k(X \mid \phi_k). \tag{5.3}$$

By marginalizing out Z, i.e. $p(X) = \sum_Z p(Z)p(X \mid Z)$, one obtain the expression in Eq. (5.1).

Hence, given a dataset $\{X_i\}_{i=1}^N$, each of the data points are assumed to be IID samples generated from the process described above. We note that, in practice, the user has to specify the distributions for each $p_k(X \mid \phi_k)$, and, typically, one uses the same parametric model for all clusters. For example, if the parametric models are chosen to be normal distributions, the clustering algorithm is referred to as a Gaussian Mixture Model (GMM).

During inference, the goal is to estimate the unknown cluster assignments Z_i via the posterior $P(Z_i \,|\, X_i)$, which can be calculated using Bayes' rule. In practice, this boils down to estimating the parameters $\theta = (\pi, \phi)$. The most common way of estimating θ is via maximum likelihood and the Expectation-Maximization (EM) algorithm (Bilmes, 1998). Alternatively, one can put priors over the parameters θ and use Bayesian approaches to estimate them. In that case, the model is a Bayesian mixture model. In Paper I and III, we consider Bayesian mixture models for multivariate time series where we put informative priors over the parameters to account for large amounts of missing data.

The mixture models discussed so far assume that the number of clusters is finite. However, also infinite (countable) mixture models exist (Rasmussen, 2000). These are commonly studied under the framework of Bayesian nonparametrics, and we refer to Gershman and Blei (2012); Hjort et al. (2010) for a more detailed description of these methods.

5.1.3 Modal clustering

Modal clustering is similar to model-based clustering in the sense that it is also based on probability density functions. The difference between the methods consists in that modal clustering algorithms take a nonparametric approach and aims at estimating distinct modes of the density directly. In these approaches, a cluster is defined as a connected high density region.

Modal clustering can be divided into two main categories, *mode seeking* and *level set* methods (Menardi, 2015). Level set clustering is based on the idea that the probability density function can be thresholded and thereafter the clusters can be identified as connected regions of high density (Chaudhuri et al., 2014; Cuevas et al., 2001; Stuetzle and Nugent, 2012). In mode seeking, the goal is to identify local maxima (modes) on the estimated density and use the modes to represent the clusters (Duin et al., 2012). The next step is that each data point is attracted to a mode via the gradient flow of the probability density function. All points in the basin of attraction for a local mode are assigned to the same cluster (Chacón, 2012).

Many mode seeking algorithms are modifications of the so-called *mean-shift* algorithm, originally proposed by Fukunaga and Hostetler (1975). The modification typically consists in proposing an alternative strategy for finding the local modes (Comaniciu and Meer, 2002; Georgescu et al., 2003; Cheng, 1995; Arias-Castro et al., 2016). The underlying idea in the mean shift algorithm is to estimate the probability density function, f, in a nonparametric way via a differentiable kernel[1] , K, as follows

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K\left(\|\mathbf{x}-\mathbf{x}_i\|\right), \tag{5.4}$$

and then perform gradient ascent on the estimated density $\hat{f}$. This results in an iterative scheme,

$$\mathbf{x} \leftarrow \mathbf{m}(\mathbf{x}), \tag{5.5}$$

where the *mean shift* vector $\mathbf{m}(\mathbf{x})$ is given by

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{N} \mathbf{x}_i \, K'\left(\mathbf{x}-\mathbf{x}_i\right)}{\sum_{i=1}^{N} K'\left(\mathbf{x}-\mathbf{x}_i\right)} - \mathbf{x}. \tag{5.6}$$

This iterative scheme is performed for each data point $\mathbf{x}_i$, $i = 1, \ldots, N$, and each point that converges to the same mode belongs to the same cluster.

The algorithm we proposed in Paper 5, which is also a part of the methodology in Paper IV presented in this thesis, is partly based on a variation of

[1] The kernels used in non-parametric density estimation should not be confused with kernel methods and the kernels discussed in Chapter 4. In density estimation, a kernel is a weighting function, which is real-valued (non-negative), symmetric and integrable. These kernels are often normalized such that they integrate to one over the real line.

mean shift, where the density is estimated by a k-Nearest Neighbors (kNN) density estimate (Duin et al., 2012). The kNN density is given by

$$\hat{f}(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{x}_k\|^2}, \tag{5.7}$$

where $\mathbf{x}_k$ is the $k-$th neighbor of $\mathbf{x}$. To create a robust clustering method, we used this algorithm in an *ensemble learning* framework, which we describe in the next section. For more details on kNN mode seeking and our algorithm, we refer to (Duin et al., 2012) and Paper 5.

An other variation of mode seeking, Modal EM (Li et al., 2007a), formulates mode seeking as an EM algorithm by considering density estimation as a GMM. We also mention *density based spatial clustering of applications with noise* (DBSCAN), and its extensions (Ester et al., 1996; Birant and Kut, 2007; Kisilevich et al., 2010), which is a popular clustering algorithm that is related to modal clustering since it captures modal regions and is more robust to noise. However, DBSCAN is different from the methods discussed above since it is not based on the notion of probability density functions (Menardi, 2015).

In the next subsection, we provide a brief introduction to *ensemble learning*, which is a key concept in Paper I and in many clustering algorithms.

5.1.4 Ensemble learning

Ensemble learning is a general learning framework that is not restricted to unsupervised learning. The underlying idea consists in combining a collection of many base models into a composite model. A good such ensemble model will have statistical, computational and representational advantages such as lower variance, lower sensitivity to local optima and a broader span of representable functions, respectively, compared to the individual models. Ensemble learning has been successfully adopted in both supervised and unsupervised learning, exemplified by the fact that many data competitions, such as the Imagenet large scale visual recognition challenge (Russakovsky et al., 2015) for computer vision, are often won by ensemble models[2].

In classification, a necessary and sufficient condition for an ensemble of classifiers to be better than any of its individual base models is *diversity* and *accuracy* (Hansen and Salamon, 1990), i.e. the base models cannot make

[2] `http://image-net.org/challenges/LSVRC/2016/results`

the same errors on new test data and have to perform better than random guessing. Examples of ensemble methods for classification include boosting trees and random forests, and a more thorough overview of these and other supervised ensemble methods can be found in (Hastie et al., 2009).

In unsupervised learning, ensemble methods have mainly been proposed for clustering (Fred and Jain, 2002; Monti et al., 2003; Strehl and Ghosh, 2003), but also for other purposes such as density estimation (Glodek et al., 2013) and for designing kernel functions. The probabilistic cluster kernel (Izquierdo-Verdiguier et al., 2015) for vectorial data, and the time series cluster kernels (Paper I and III) and learned pattern similarity (Baydogan and Runger, 2016) for time series, are examples of the latter. In this thesis, unsupervised ensemble learning, is a (smaller or larger) part of the methodology in Paper I, III and IV.

In ensemble clustering, also called consensus clustering, typically, one integrates the outcomes of the same or different (weak) clustering algorithms as they are trained under different, often randomly chosen, cluster settings (parameters, initialization or resampling) (Li et al., 2007a; Monti et al., 2003; Lourenço et al., 2015; Topchy et al., 2005; Strehl and Ghosh, 2003; Vega-Pons and Ruiz-Shulcloper, 2011; Jain, 2010). Moreover, consensus clustering can be considered as a two step clustering process, where the first step consists of running multiple clusterings of the data with different parameters, initializations and/or random subsets each time. In the second step, one measures the *consensus* over all the iterations to obtain the final clustering, which is supposed to be a more robust clustering than the single algorithm clusterings. The consensus over the ensemble can be measured in several different ways, which broadly can be divided into methods based on *median partition* and *co-association* (Li et al., 2007b; Fred, 2001; Vega-Pons and Ruiz-Shulcloper, 2011).

Median partition methods are based on a cost function formulation that aims at making the final clustering similar to the ensemble clusterings, whereas co-association methods create a *consensus matrix*, which contains (normalized) counts of how many times each pair of data points are clustered together. The consensus matrix forms a similarity matrix, which is used as input to a new clustering algorithm such as hierarchical clustering (done in e.g. (Fred, 2001; Fred and Jain, 2005) and Paper 5), or spectral clustering (done in Paper 14).

5.2 Dimensionality reduction

Dimensionality reduction is relevant for Paper II, and partly Paper I and III. Here, as a service for readers not familiar with machine learning, we provide a brief introduction to some basics on dimensionality reduction. For more details we point the interested reader to the surveys (Van Der Maaten et al., 2009; Burges et al., 2010; Khalid et al., 2014; Cunningham and Ghahramani, 2015; Wang and Sun, 2015b), and the related work section in Paper II.

The most well-known unsupervised dimensionality reduction method is PCA, which linearly transforms the data such that the covariance matrix becomes a diagonal matrix. By doing so, PCA transforms a dataset consisting of observations of potentially correlated variables into data points with linearly uncorrelated variables.

Mathematically, PCA can be described as follows. Assume that we have the random variable $\mathbf{x} \in \mathbb{R}^d$, where the covariance is given by the $d \times d$ matrix $\Sigma_{\mathbf{x}}$. PCA aims at finding a linear transformation

$$\mathbf{y} = T(\mathbf{x}) = A^T \mathbf{x}. \tag{5.8}$$

Covariance matrices are symmetric and positive semi-definite, and therefore we can find an orthonormal set $\{\mathbf{e}_i\}_{i=1}^d$ that satisfies

$$\Sigma_{\mathbf{x}} \mathbf{e}_i = \lambda_i \mathbf{e}_i, \text{ i = 1,2,...,d}, \tag{5.9}$$

where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d \geq 0$. Define $D = diag(\lambda_i)$ and $E = [\mathbf{e}_1 \, \mathbf{e}_2 \, ... \, \mathbf{e}_d]$. Thus we can write (5.9) as

$$\Sigma_{\mathbf{x}} E = DE. \tag{5.10}$$

Moreover, because of the symmetric and positive semi-definite property, the matrix E is orthogonal, i.e. $E^T = E^{-1}$, which in turn implies that

$$D = E^T \Sigma_{\mathbf{x}} E. \tag{5.11}$$

We also have that

$$\Sigma_{\mathbf{y}} \equiv E[(\mathbf{y} - \mu_{\mathbf{y}})(\mathbf{y} - \mu_{\mathbf{y}})^T] = E[A^T(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{x} - \mu_{\mathbf{x}})^T A] = A^T \Sigma_{\mathbf{x}} A \tag{5.12}$$

Let $A = E$ and we obtain that

$$\Sigma_{\mathbf{y}} = E^T \Sigma_{\mathbf{x}} E = D. \tag{5.13}$$

Thus the linear transformation $\mathbf{y} = E^T\mathbf{x}$, where E contains the eigenvectors of $\Sigma_{\mathbf{x}}$, makes the matrix $\Sigma_{\mathbf{y}}$ diagonal. The elements of $\Sigma_{\mathbf{y}}$ are the eigenvalues of $\Sigma_{\mathbf{x}}$. In order to reduce dimension one can decide to use only the k top eigenvalues and eigenvectors, i.e. let

$$\mathbf{y} = E_k^T\mathbf{x}, \tag{5.14}$$

where $k < d$ and $E_k = [\mathbf{e}_1\ \mathbf{e}_2\ ...\ \mathbf{e}_k]$. Since the eigenvectors are ordered by decreasing eigenvalue, they correspond to directions of decreasing variance in the data. Thus, the k-dimensional subspace span$\{\mathbf{e}_i\}_{i=1}^k$ capture more of the variance in the data than any other k-dimensional subspace.

PCA does not take label information into account and is therefore an example of an *unsupervised* dimensionality reduction method. The most prominent example of a *supervised* dimensionality reduction method is linear discriminant analysis(Fisher, 1936), which aims at finding the linear projection that maximizes the within-class similarity and at the same time minimizes the between-class similarity in the projected space. More specifically, in linear discriminant analysis the transformation matrix A (Eq. (5.8)) is given by

$$A = \text{argmax}_G\, Tr\left(\frac{G^T S_b G}{G^T S_w G}\right), \tag{5.15}$$

where the between-class scatter matrix S_b is given by

$$S_b = \sum_{c=1}^{C} n_c(\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^T, \tag{5.16}$$

and the within-class scatter matrix S_w is given by

$$S_w = \sum_{c=1}^{C}\sum_{i\in I_c}(\mathbf{x}_i - \mathbf{m}_c)(\mathbf{x}_i - \mathbf{m}_c)^T, \tag{5.17}$$

and Tr is the trace operator, C is the number of classes, $\mathbf{m}_c$ the mean of class c, $\mathbf{m}$ the global mean, n_c the number of data points in class c, $I_c = \{i \mid y_i = c\}$, and $y_i \in \{1, \ldots, C\}$ the label of data point $\mathbf{x}_i$, $i = 1, \ldots, N$. The optimization problem (5.15) can be solved via (generalized) eigen-decomposition.

Chapter 6

Weakly supervised learning

In this thesis, we refer to *weakly supervised learning* as the collection of learning algorithms that can learn from *weak supervision information*, as opposed to *strong supervision information* such as completely labeled training data. Weakly supervised learning aims to bypass the need of large amounts manually annotated data for training the models.

There are several types of weak supervision information (Zhou, 2017; Chapelle et al., 2006; Patrini et al., 2016; Frenay and Verleysen, 2014). These include

- *Incomplete supervision.* Only a subset of the training data is labeled.
- *Inaccurate supervision.* Labels are potentially noisy.
- *Inexact supervision.* Some supervision information is provided, but it is not as exact as desired.
- *Constraint supervision.* Some data points are subject to constraints.

Weak supervision information can also be a combination of two or more of these types.

Fig. 6.1 shows an overview of different types of supervision information and the most common learning frameworks to deal with the each type of supervision information.

Constraint supervision information can for example be additional information provided as pairwise must-link or cannot-link constraints, i.e. some data points *must* have, or *cannot* have, the same label. This setting is commonly

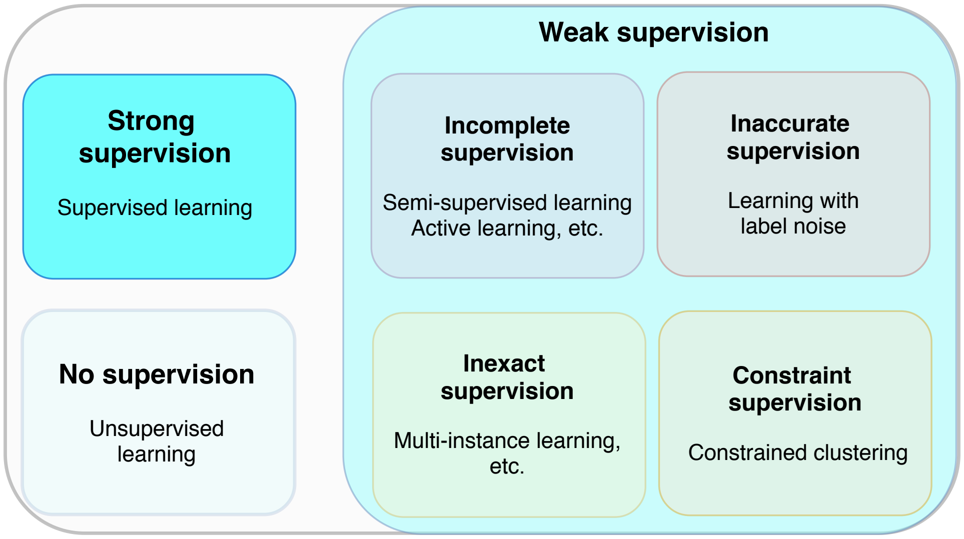


Figure 6.1: *Overview of common types of supervision information in machine learning, and learning paradigms that deal with the different types of supervision information.*

studied in so-called constrained clustering, also called semi-supervised clustering, which consists in performing clustering guided by constraints (Basu et al., 2004; Wang et al., 2008; Kulis et al., 2009; Wang et al., 2014b; Basu et al., 2008).

An example of a situation where inexact supervision information occurs is when labeled data are provided, but the labels are more coarse-grained than desired. As an example, in sentiment analysis of online reviews from e.g. Tripadvisor, coarse-grained document annotations could often be easy to obtain via star ratings. However, often these reviews also contain finer-grained annotations (e.g. pros and cons about a visit to a restaurant), which are not that easy to extract since they provided in the form of free text (Angelidis and Lapata, 2018). The typical learning framework to deal with this type of supervision is multi-instance learning (Foulds and Frank, 2010), which we do not discuss in more detail in this text. However, we refer to the excellent review paper by Zhou (2017) for a short survey on methods for inexact supervision. In the remainder of this chapter, we focus on incomplete and inaccurate supervision, which are the types of weak supervision that are relevant for the research presented in Paper II, III and IV.

6.1 Incomplete supervision

There are two main learning frameworks for dealing with incomplete supervision information, active learning Settles (2012) and semi-supervised learning (Chapelle et al., 2006).

In active learning, the underlying idea is that an algorithm trained on sparsely annotated data can perform better if it is allowed to select some unlabeled data points and get new labels for these. This is typically achieved via human intervention by letting the active learner pose queries to a human expert, or in general, an oracle. Hence, active learning assumes that human intervention is possible, whereas this is not the case in semi-supervised learning, which we describe next. For more details on active learning, we point the interested reader to the survey by Settles (2012).

Semi-supervised learning

While alternative and more general definitions exist (Chapelle et al., 2006), in this text we define semi-supervised learning as follows.

Definition 5 *Semi-supervised learning is the learning task aiming to learn from unlabeled data and weak supervision information provided in terms of the desired output for a subset of the data points in the training set.*

Hence, in semi-supervised learning, we are provided with a training set consisting of unlabeled data and some data points with known label for the particular learning task at hand. Therefore semi-supervised learning can naturally be thought of as a mix of the two classical branches in machine learning, unsupervised learning and supervised learning.

The final objective of semi-supervised learning can be either a classical *supervised* learning task, i.e. to learn a predictor, or an *unsupervised* learning task.

Regarding the latter case, we note that constrained clustering is not covered by the definition above since supervision information is provided in terms of constraints (on the labels) rather than explicit label information. On the other hand, semi-supervised clustering methods (Bair, 2013) which assume that some labels are known are covered by the definition.

Semi-supervised dimensionality reduction

An other classical unsupervised learning task that fall within this category is dimensionality reduction. In the context of semi-supervised learning, we refer to it as *semi-supervised dimensionality reduction.* There exist three different problem formulations (settings), which we describe below, that lead to semi-supervised dimensionality reduction. We note that only two of them are semi-supervised according to Def. 5.

Setting 1. *The ultimate goal of the learning task is dimensionality reduction, and therefore the desired output of the learning framework is the new representation in the low dimensional space. Weak supervision information is provided as the exact mapping of certain data points.*

According to our definition, algorithms developed for this purpose are semi-supervised. As an example, Yang et al. (2006) proposed several different methods for this purpose.

Setting 2. *The setting is classification, i.e. the desired outputs are class labels. As a goal in itself, as a preprocessing step, or as an integral part of the learning procedure, one performs dimensionality reduction. The training set consists of both unlabeled data and a subset of data points with known class labels.*

In this case, one can also perform dimensionality reduction in a semi-supervised manner. There are many examples of methods designed for this purpose, and these include (Cai et al., 2007; Lee et al., 2010) as well as the unified frameworks by Song et al. (2008); Nie et al. (2010b), in which semi-supervised versions of classical methods such as e.g. PCA, linear discriminant analysis, maximum margin criterion, locality preserving projections, and their kernelized versions, can be seen as special cases. This category is not restricted to standard multi-class classification. It may also be that the semi-supervised dimensionality reduction methods are designed for multi-labels[1] (Zhang and Zhou, 2007). In fact, the method proposed in Paper II is an example of the latter. We refer to that paper for more examples of such dimensionality reduction methods.

Setting 3. *The ultimate goal of the learning task is classification. However, in this case, weak supervision information is provided in terms of pairwise constraints.*

[1] In multi-label learning, each data point can potentially belong to multiple classes.

This setup is not covered by Def. 5. Examples of dimensionality reduction methods proposed for this purpose include the *semi-supervised dimensionality reduction* algorithm by Zhang et al. (2007) and some methods for non-negative matrix factorization (Chen et al., 2008; Wang et al., 2008).

Semi-supervised classification

In the remainder of this section, we focus on the former case described above, i.e. semi-supervised learning for classical supervised learning tasks. In more detail, we assume the following setting.

Given $\mathcal{F}$ a hypothesis space, $\mathcal{X}$ an input space, and $\mathcal{Y}$ an output (label) space, the goal of the learning task is to learn a predictor $f \in \mathcal{F}$, $f : \mathcal{X} \to \mathcal{Y}$. For this purpose, we assume that we are given a dataset of N data points, L of which are labeled $\{\mathbf{x}_i, y_i\}_{i=1}^{L} \overset{IID}{\sim} p(\mathbf{x}, y)$, and the remaining $U = N - L$ are unlabeled $\{\mathbf{x}_i\}_{i=L+1}^{N} \overset{IID}{\sim} p(\mathbf{x})$. Here, $p(\mathbf{x}, y)$ is an unknown joint distribution and $p(\mathbf{x})$ the corresponding marginal.

It is common to distinguish between *inductive* and *transductive* semi-supervised classifiers. In the former case, the goal is to learn a predictor f that performs better on unseen test data $\mathbf{x} \in \mathcal{X}$ than a predictor trained on only $\{\mathbf{x}_i, y_i\}_{i=1}^{L}$. On the other hand, in the transductive case, the goal is to classify the unlabeled training data $\{\mathbf{x}_i\}_{i=L+1}^{N}$.

In general, there are three assumptions about the underlying distribution of the data that are commonly made in semi-supervised learning (Chapelle et al., 2006):

1. Smoothness assumption: *If two points, $\mathbf{x}_i$ and $\mathbf{x}_j$, in a high-density region are close, then so should the corresponding targets y_i and y_j* be.

2. Cluster assumption: *Points that belong to the same cluster are likely to be of the same class.* or, equivalently *the decision boundaries between classes should lie in low-density regions.*

3. Manifold assumption: *The high-dimensional data (from $\mathcal{X}$) lie on a lower-dimensional manifold.*

If it should be possible to learn efficient models via semi-supervised learning, typically, at least one of these three assumption must hold. Next, we discuss some of the main semi-supervised learning approaches for classification.

Generative models There have been several attempts to learn generative mixture models for both labeled and unlabeled data (Miller and Uyar, 1997; Nigam et al., 2000). The difference between these models and the clustering mixture models is that the latent variables for the labeled data are not hidden, but known and equal to the class labels. The semi-supervised generative models incorporate this information in their formulation of the likelihood and the EM-algorithm. We refer to Section 5.1.2 for more details on mixture models. We also note that there have been attempts to combine generative and discriminative approaches for semi-supervised learning (Fujino et al., 2005).

Low-density separation methods As the name suggests, low-density separation methods aim at pushing the decision boundaries, $\{\mathbf{x} \in \mathcal{X} \,|\, f(\mathbf{x}) = 0\}$, away from the unlabeled points and into areas with low density. The semi-supervised support vector machines (S3VMs) are among the most prominent examples of methods in this category (Joachims, 1999; Chapelle and Zien, 2005; Li et al., 2013). The S3VMs are kernel machines in which the empirical risk (see Eq. (4.8)) typically takes the form

$$\begin{aligned}\Psi(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n), \|f\|_{\mathcal{H}_k}) = &\frac{1}{L}\sum_{i=1}^{L} \max(0, 1 - y_i f(\mathbf{x}_i)) + \lambda_1 \|f\|_2^2 \\ &+ \frac{\lambda_2}{U}\sum_{i=L+1}^{N} \max(0, 1 - |f(\mathbf{x}_i)|). \qquad (6.1)\end{aligned}$$

Fig. 6.2 shows an illustration of a SVM and a S3VM trained on a simple two-class and two-dimensional semi-supervised classification task, where there are only two labeled data points in each class. As we can see, S3VM accounts for the cluster assumption and places the decision boundary in the area with low density, whereas SVM only accounts for the labeled training points.

Heuristic approaches Several heuristic approaches to semi-supervised learning have been proposed. Common to these methods is that unlabeled data, in one way or another, are made use of within a supervised framework.

One approach is to first in an unsupervised way learn a representation, a metric, or a kernel, and then, apply a supervised learning algorithm only for the labeled subset using the learned representation, metric or kernel.

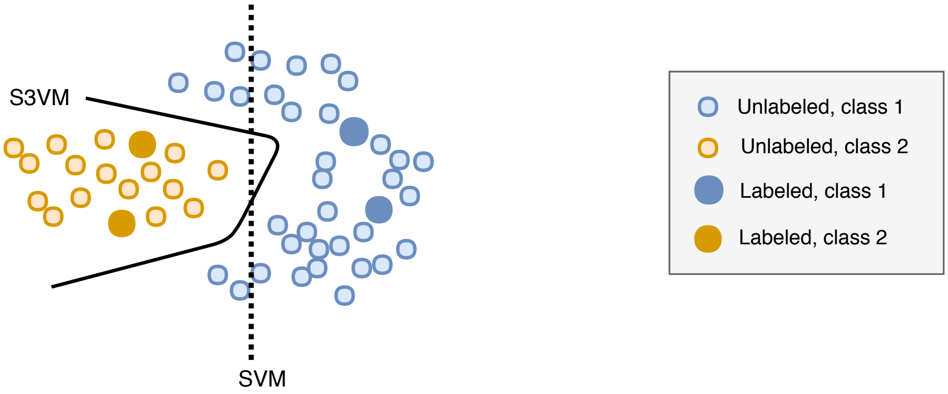


Figure 6.2: *Illustration of a SVM and a S3VM trained on a simple two-class and two-dimensional semi-supervised classification task. The decision boundary of the SVM is dashed.*

A second approach is self-training, also called bootstrapping[2], which refers to a repetitive procedure where a classifier is trained on the labeled data and then used to classify unlabeled data. By doing so, one obtains labels for previously unlabeled data. Typically, data points corresponding to the most confident predictions are added to the labeled subset, and then the procedure is repeated (Fazakis et al., 2016; Tanha et al., 2017).

Co-training (Blum and Mitchell, 1998), refers to an approach where two classifiers generate labels for one another by training on two different sets of features.

Graph-based methods Graph-based learning methods model the whole dataset as a graph. The nodes of the graph represents the data points, whereas the edges correspond to pairwise similarities between the patterns. In semi-supervised learning, the graph-based methods are typically transductive. For example, a common strategy is to use the graph to propagate label information to the unlabeled data points, i.e. perform *label propagation* (Zhu and Ghahramani, 2002; Yang et al., 2016; Belkin and Niyogi, 2003; Hensley et al., 2015; Zhu et al., 2003a; Nie et al., 2010a; Sandryhaila and Moura, 2013). This is a strategy we also employ in Paper II.

The structure of the graph is given by the adjacency matrix **A**, which can be

[2] Not related to bootstrapping in statistics.

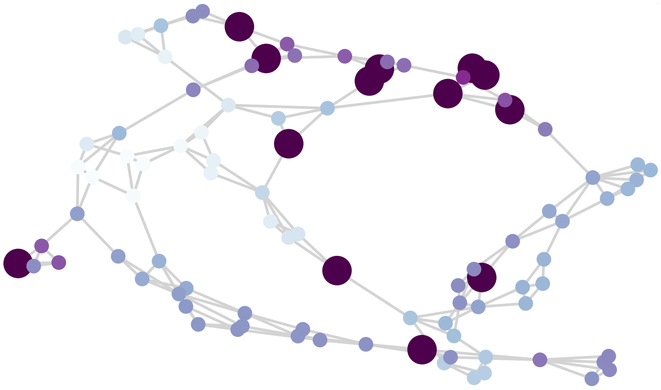


Figure 6.3: *Illustration of a kNN graph ($k = 3$). Large nodes represent data points that are labeled, whereas the small nodes represent data points that are unlabeled. An edge between two data points indicate that the similarity is 1. If there is no edge between data points then the similarity is 0.*

defined in many different ways and therefore affects the performance of the method. One can for example define a fully-connected graph, whose edges are weighted by the adjacency matrix given via the RBF-kernel,

$$A_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right). \tag{6.2}$$

Other formulations of $\mathbf{A}$ include a binary kNN graph, used for example by Yu et al. (2017b), where

$$A_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

with $\mathcal{N}(\cdot)$ denoting a neighborhood of size k. Fig. 6.3 shows an illustration of a kNN graph computed over a toy dataset consisting of both labeled and unlabeled data.

Yet another alternative is the linear neighborhood graph (Wang and Zhang, 2008), which is defined by

$$A_{ij} = \operatorname{argmin}\|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} A_{ij}\mathbf{x}_j\|_2^2, \ \text{s.t. } A_{ij} \geq 0 \text{ and } \sum_j A_{ij} = 1. \tag{6.4}$$

As a representative of the label propagation methods, we here briefly describe the *learning with local and global consistency* algorithm (Zhou et al., 2004).

Define the symmetrically normalized adjacency matrix

$$\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}, \text{ where } D_{ii} = \sum_j A_{ij}. \tag{6.5}$$

Let $\mathbf{Y}_L \in \mathbb{R}^{L \times C}$ be a one-hot representation of the labels (assuming C classes). Then, soft labels $\mathbf{F} = [\mathbf{F}_L; \mathbf{F}_U] \in \mathbb{R}^{N \times C}$ can be learned using the iterative procedure described by

$$\mathbf{F}^{(t+1)} = \alpha \mathbf{W}\mathbf{F}^{(t)} + (1-\alpha)\mathbf{Y}, \quad \alpha \in (0,1), \tag{6.6}$$

where the initial value $\mathbf{F}^{(0)} = \mathbf{Y} = [\mathbf{Y}_L; \mathbf{0}_{U \times C}]$. This procedure ensures smoothness (continuity) of the label assignments on the manifold, i.e. global consistency, whereas the second term minimizes discrepancy of the soft labels with their prior values (local consistency). With this method, in contrast to methods where the labeled data are clamped (Zhu and Ghahramani, 2002), the final value of the labels for the labeled points usually diverges from their initial value, i.e. $\mathbf{F}_L \neq \mathbf{Y}_L$. This can be beneficial in presence of *label noise* (Bengio et al., 2006) (see next section). It can be shown that the update iteration in (6.6) converges to

$$\mathbf{F} = (\mathbf{I} - \alpha \mathbf{W})^{-1}\mathbf{Y}, \tag{6.7}$$

which implies that the solution $\mathbf{F}$ can be found by solving a linear system of equations.

Learning with positive and unlabeled data Learning with Positive and Unlabeled data (PU-learning) can be considered as a special case of semi-supervised learning where it is assumed that the labeled set only consists of positive examples.

The traditional task in PU-learning is binary classification (Elkan and Noto, 2008; Mordelet and Vert, 2014; Du Plessis et al., 2014), but the framework has also been formulated for e.g. matrix completion (Hsieh et al., 2015), streaming networks (Chang et al., 2016), ranking and multi-label learning (Kanehira and Harada, 2016). Researchers have found PU-learning useful in several different application areas, such as text classification (Liu et al., 2003), bioinformatics (Cerulo et al., 2010), medicine (Halpern et al., 2014), to name a few.

In the case of classification, PU-learning takes a quite different approach compared to the semi-supervised methods described above that use unlabeled data for regularization purposes under the cluster or manifold assumption. Instead of being based on such restrictive distributional assumptions, PU-learning directly extracts label information from the unlabeled data.

The work of Elkan and Noto (2008) is particularly relevant to Paper IV. They consider binary classification with a problem setting as follows. Let $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{-1, 1\}$ be random input and target variables with an underlying joint density $p(\mathbf{x}, y)$. A set, U, of n_u unlabeled data points is sampled from the marginal density $p(\mathbf{x})$, which is given by

$$p(\mathbf{x}) = \pi p_+(\mathbf{x}) + (1 - \pi) p_-(\mathbf{x}), \tag{6.8}$$

where $p_+(\mathbf{x}) = p(\mathbf{x} \mid Y = 1)$, $p_-(\mathbf{x}) = p(\mathbf{x} \mid Y = -1)$, and $\pi = P(Y = 1)$ is the class-prior. A set P of n_+ data points from the positive class is then sampled from $p_+(\mathbf{x})$. Let $z \in \{-1, 1\}$ be random variable that takes the value 1 if a positive label is observed, i.e. if $\mathbf{x}$ belongs to P.

Then, Elkan and Noto (2008) assumes that the labeled positive data points form a random subset of all data points in the positive class, i.e.

$$P(z = 1 \mid y = 1,\ \mathbf{x}) = P(z = 1 \mid y = 1), . \tag{6.9}$$

Using the randomness assumption, it is shown that

$$\begin{aligned} P(z = 1 \mid \mathbf{x}) &= P(y = 1,\ z = 1 \mid \mathbf{x}) \\ &= P(y = 1 \mid \mathbf{x}) P(z = 1 \mid y = 1,\ \mathbf{x}) \\ &= P(y = 1 \mid \mathbf{x}) P(z = 1 \mid y = 1). \end{aligned} \tag{6.10}$$

It follows that

$$P(y = 1 \mid \mathbf{x}) = C \cdot P(z = 1 \mid \mathbf{x}), \tag{6.11}$$

where $C^{-1} = P(y = z \mid y = 1)$, i.e. the classifier trained on the positive and unlabeled data is off only by a constant factor, which implies that this classifier will provide the same ranking of new unseen data as a classifier trained on both negative and positive examples. One can choose any classifier and the constant C can be estimated empirically in several different ways.

This framework does, however, in most cases lead to biased risk estimators, and to account for that, Du Plessis et al. (2014, 2015) proposed unbiased risk estimators for PU-learning. Later, Kiryo et al. (2017) proposed PU-learning with non-negative risk estimators.

We also highlight the work of Niu et al. (2016), which theoretically studied under what circumstances PU-learning is expected to outperform regular supervised learning, and, in particular, how many unlabeled data points, n_u, are needed to outperform the corresponding supervised classifier trained on n_+ positive examples and n_- negative examples. Under mild assumptions on the distribution of the data and the function class, they showed that the estimation error bound of the risk minimizer in PU-learning is tighter than of supervised learning if

$$\frac{\pi}{\sqrt{n_+}} + \frac{1}{\sqrt{n_u}} < \frac{1-\pi}{\sqrt{n_-}}. \tag{6.12}$$

PU-learning has also been extended to incorporate negative data (Sakai et al., 2017). Hence, the setting is the same as in classical semi-supervised learning, but because of the properties of PU-learning the common distributional assumptions in semi-supervised learning do not have to be made.

6.2 Inaccurate supervision

Inaccurate supervision information occurs when some of the observed labels are corrupted and therefore do not coincide with the ground-truth labels. Such inaccurate labels are referred to as *noisy* labels (Frenay and Verleysen, 2014; Liu and Tao, 2016a; Natarajan et al., 2013). In this thesis, both Paper II and IV consider learning situations where label noise occurs.

There are many different sources of label noise. For example, when human experts (e.g. medical doctors) are involved in the annotation process, labeling errors naturally occur (Frenay and Verleysen, 2014). Some of the most common reasons are imperfect evidence (inadequate information), data-entry errors, and subjective labeling errors (e.g., experts do not agree on what is the correct label) (Smyth, 1996; Brodley and Friedl, 1999; Frenay and Verleysen, 2014).

Label noise may also occur when automated algorithms are involved in the labeling process. For example, Zhu et al. (2018) proposed a Bayesian fusion model designed for noisy labels provided by multiple imperfect automated algorithms. In other cases, a combination of human experts and automated algorithms is used for annotating the data, also resulting in label noise. This is the case in frameworks such as anchor learning (Halpern et al., 2016) and silver standard learning (Agarwal et al., 2016), in which both the so-called anchor variables and the silver standards are examples of noisy labels.

The fact that learning with label noise is a problem of great practical importance, has led to a great deal of work on the problem, both in terms of practical (McDonald et al., 2003; Pechenizkiy et al., 2006; Nettleton et al., 2010) and theoretical studies (Angluin and Laird, 1988; Natarajan et al., 2013; Bi and Jeske, 2010; Simon, 1996; Aslam and Decatur, 1996).

The theoretical studies of label noise often assume Random Classification Noise (RCN) (Natarajan et al., 2013), i.e. instead of ground-truth labels, the learning algorithm sees labels that have independently been corrupted with some small probability.

There are three main approaches to learning supervised classifiers with label noise: noise-robust models, data cleaning approaches, and noise-tolerant learning (Bouveyron and Girard, 2009; Frenay and Verleysen, 2014). Next, we discuss these three approaches.

Label noise-robust models A learning algorithm is robust to label noise if the model trained on noisy labeled data performs similarly on noise-free test data as the corresponding model trained on noise free data (Manwani and Sastry, 2013). Hence, label noise-robust models simply ignore the label-noise and are instead trained as if the labels were clean. Many learning algorithms, such as e.g. the SVM, tend to overfit to noisy training data, and therefore they do not generalize well to test data. However, it has been shown that certain algorithms are less influenced by the presence of noise than others. For example, Manwani and Sastry (2013); Ghosh et al. (2015) studied theoretically label noise-robust loss functions and found that 0-1 loss, sigmoid loss, ramp loss and probit loss are robust when the noise is uniform, whereas the exponential loss (Adaboost), log loss (logistic regression), and hinge loss (SVM) are not robust even in the case of uniform label noise. Further, Nettleton et al. (2010) showed empirically that the Naive Bayes classifier is more noise robust than many other standard classifiers.

Data cleaning approaches A second set of approaches adds a preprocessing step in which the noisy labels are cleaned before a classifier is trained in the standard way, i.e. by assuming noise-free labels. These methods rely on correctly identifying the corrupted labels. Corrupted labels are then either *relabeled* (altered) or *filtered* out.

There exist many different methods for identifying corrupted labels. Some approaches search for label mismatch among the *k-nearest neighbors* (Wilson and Martinez, 2000), whereas the *graph-based* methods represent training sets via neighborhood graphs and use the graph to detect corrupted labels (Lallich et al., 2002) (also see Paper II). It is also quite common to take an ensemble learning approach and train multiple classifiers on the noisy labeled data to look for disagreement between the methods (Brodley and Friedl, 1999; Zhu et al., 2003b).

Inherently label noise-tolerant learning algorithms A third set of approaches consists of algorithms that are designed to inherently account for label-noise e.g. by directly modeling the noise process.

Some works aim at making specific classifiers or algorithms tolerant to noise. For example, Lawrence and Schölkopf (2001) built a kernel Fisher discriminant classifier wherein they explicitly modeled the noise process as one component in a generative model. The perceptron algorithm has been modified in various ways to become noise-tolerant (Khardon and Wachman, 2007), whereas Sukhbaatar et al. (2014) studied how to train deep neural networks in presence of label noise.

Other works study noise tolerant learning from a more general perspective, typically, starting by making some assumptions on the noise process. Early work focused on Probably Approximately Correct (PAC) learning introduced by Valiant (1984). In this framework, a learner is given $\mathcal{F}$, a class of functions it can choose from, a training set $\{\mathbf{x}_i,\ y_i\}$, and ϵ and δ, accuracy and confidence parameters. The task is to find a close approximation of an unknown binary target function f. More precisely, a PAC-learner has to find a hypothesis function, $\hat{f} \in \mathcal{F}$, such that with high probability (at least $1-\delta$), the generalization error is lower than ϵ. An assumption in PAC-learning is that the labels are noise-free.

The works of Simon (1996); Aslam and Decatur (1996) studied PAC-learning in the presence of label noise, by assuming RCN, which is a simple noise model introduced by Angluin and Laird (1988) where each data point is assumed to be mislabeled independently and randomly with a fixed random classification error η. More precisely, Aslam and Decatur (1996) studied the increase in sample complexity in presence of label noise, i.e. how many noisy labeled training examples are needed for a PAC-learner, and obtained a lower bound on the number of training samples needed given by

$$N \geq C \frac{VC(\mathcal{F}) - \log \delta}{\epsilon(1 - 2\eta)^2}, \tag{6.13}$$

where C is some constant and VC (Vapnik and Chervonenkis, 2015), the Vapnik-Chervonenkis dimension of $\mathcal{F}$.

More recent works have considered the situation when the noise process is not uniform. One such situation is asymmetric RCN or class-conditional noise, i.e. the probability that a label is corrupted depends on which class the data point belongs to. Scott et al. (2013) proposed a general classification framework for class-conditional noise, whereas Natarajan et al. (2013) studied risk optimization of a particular surrogate loss function to obtain noise tolerant classifiers and proved that e.g. weighted logistic regression (King and Zeng, 2001) is noise tolerant. Liu and Tao (2016b) also considered class-conditional noise and focused on how to efficiently estimate the noise rate η, which in most practical scenarios is unknown.

A special type of class-conditional noise can occur in PU-learning, since by definition, the unlabeled class cannot contain label noise[3] . Hence, label noise can only occur among the positive examples. Learning with noisy positives and unlabeled data was studied by (Jain et al., 2016).

We conclude this chapter with an example, illustrating some practical consequences of the work of (Aslam and Decatur, 1996).

Example Note that Eq. (6.13) is also true in the case of clean labels, which corresponds to $\eta = 0$. Hence, given N RCN-noisy labeled examples with random classification error equal to η and by assuming a finite VC dimension, we can use the bound to estimate the number of clean labels, N_c, needed to obtain similar generalization error by looking at the fraction

$$\frac{N}{N_c} \approx \frac{C\frac{VC(\mathcal{F})-\log\delta}{\epsilon(1-2\eta)^2}}{C\frac{VC(\mathcal{F})-\log\delta}{\epsilon(1-2\cdot 0)^2}} = \frac{1}{(1-2\eta)^2}. \tag{6.14}$$

Hence, the number of clean labels needed is given by $N_c = (1-2\eta)^2 N$, which means that if, for example, the fraction of mislabeled examples is 0.05 in the noisy labeled dataset, one needs to annotate $N_c = 0.81N$ examples with clean labels to achieve similar performance given that the underlying assumptions hold. In other words, this means that if an automated algorithm is capable of creating (cheap) noisy labels for 5000 data points with a noise rate of 5%, human experts will have to manually label approximately 4000 data points to achieve the same performance.

[3] PU-learning can, however, be cast into a corrupted label setting (Natarajan et al., 2013; Menon et al., 2015).

Part III

Summary of research

Chapter 7

Summary of papers

Paper I - Time series cluster kernel for learning similarities between multivariate time series with missing data

The paper presents a new kernel function for multivariate time series that contain missing data, namely the TCK. The kernel is designed following an ensemble learning approach with GMMs as base models. A key to ensure robustness to missing data is to take a Bayesian approach and extend the GMMs with priors over the parameters. By doing so, the cluster means are forced to be smooth over time and, hence, less sensitive to missing elements. The parameters of the Bayesian GMMs are learned using maximum a posteriori EM. Figure 7.1 shows an illustration of how the TCK kernel is constructed.

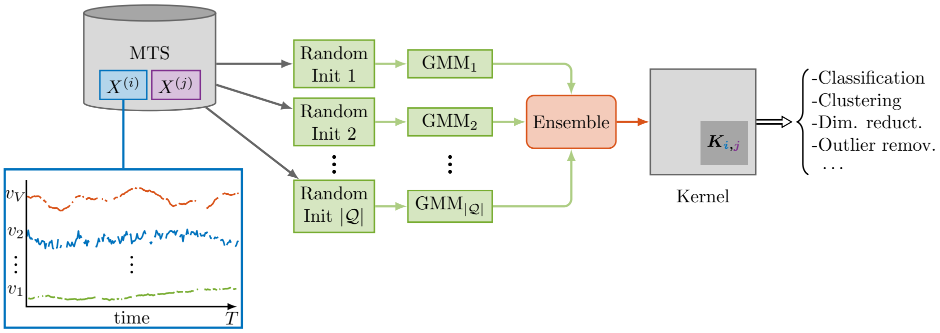


Figure 7.1: *Illustration of how the TCK kernel for multivariate time series (MTS) is constructed.*

The idea of using Bayesian mixture models for multivariate time series subject to missing data is not a novelty in itself. Marlin et al. (2012) used the same mixture models to cluster patients based on time series originating from EHRs. The novelty in our work is to use an ensemble approach to design a time series *kernel* from the GMMs. Ensemble learning ensures robustness to hyper-parameter changes, and therefore the TCK kernel is ideal to use as one component in an unsupervised learning scheme where cross-validation cannot be used to select hyper-parameters. Further, the ensemble approach makes the space of representable functions larger and enables the creation of a *kernel*, which is more general than a single clustering of the time series and can be used in a whole range of learning tasks, such as classification, clustering, dimensionality reduction, anomaly detection, etc.

The experimental results demonstrate that the TCK is robust to hyper-parameter choices, provides competitive results for multivariate time series without missing data and outperforms other kernels when missing data are present. There are few existing kernels designed for multivariate time series subject to missing data, and therefore we believe that the TCK can be a useful tool across a variety of applied domains in time series analysis.

An extended abstract (Paper 9) of preliminary work, leading to this paper, was presented at the 3rd International Workshop on Pattern Recognition for Healthcare Analytics, International Conference on Pattern Recognition (ICPR), Cancun, Mexico in 2016. In addition, a conference paper version (Paper 7) was presented at the IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Tokyo, Japan in 2017.

Contributions by the author The idea was conceived by myself and further developed in collaborations with the other co-authors. The implementation and experiments were carried out by myself with the help of Filippo Bianchi. I wrote the main draft of the manuscript.

Paper II - Noisy multi-label semi-supervised dimensionality reduction

The paper presents a novel dimensionality reduction method for partially and noisy labeled multi-label data. We call the method Noisy multi-label semi-supervised dimensionality reduction (NMLSDR). The proposed method can be considered as a two-stage procedure. In the first stage, we take a graph-based approach to simultaneously clean the noisy multi-labels and

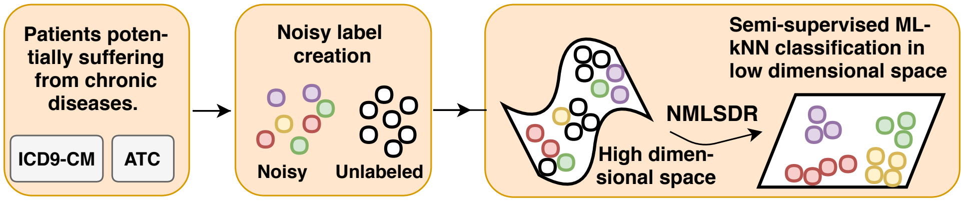

Patients potentially suffering from chronic diseases.
ICD9-CM
ATC
Noisy label creation
Noisy
Unlabeled
Semi-supervised ML-kNN classification in low dimensional space
NMLSDR
High dimensional space

Figure 7.2: *Illustration of the proposed framework applied in the case study in Paper II.*

find labels for the unlabeled data using a novel label propagation algorithm specially designed for this purpose. In the second stage, we learn a lower dimensional representation of the data by maximizing the dependence between the enlarged and cleaned multi-label space and the features in the projected space. As an additional contribution, we propose a novel framework for semi-supervised classification of noisy multi-label data.

Experiments on toy data and ten benchmark datasets demonstrate that NMLSDR is superior to baseline dimensionality reduction methods according to seven multi-label evaluation metrics. In a case study, we employ the proposed framework for semi-supervised classification of noisy multi-label data for EHR-based phenotyping. Our objective is to identify patients with certain chronic diseases. More specifically, the phenotypes we consider are hypertension, diabetes mellitus and a multi-morbidity, namely hypertension and diabetes mellitus. In our framework, we use clinical expertise to create a partially and noisy labeled dataset. An illustration of this framework is shown in Figure 7.2. We think that NMLSDR can be a useful method across a variety of applied domains, and particularly in healthcare applications, which we also illustrate in our real-world case study from healthcare.

Contributions by the author The idea was conceived by myself and further developed in collaborations with the other co-authors. The implementation and experiments were carried out by myself. I wrote the main draft of the manuscript.

Paper III - Time series cluster kernels to exploit informative missingness and incomplete label information

A main motivation for the third work is to take advantage of the fact that in data-driven healthcare missing values and patterns often contain rich information about the clinical outcomes of interest. To this end, we build upon the work presented in Paper I and propose a multivariate time series kernel capable of exploiting informative missingness to learn useful representations of incompletely observed time series data. In our approach, we create a representation of the missing patterns using masking, i.e. we represent the missing patterns using binary indicator time series. By doing so, we obtain multivariate time series consisting of both continuous and discrete attributes, which we model using mixed mode Bayesian mixture models.

Moreover, we also propose a novel semi-supervised kernel, capable of taking advantage of incomplete supervision information. To this end, we incorporate ideas from information theory to measure similarities between distributions. More specifically, we employ the Kullback-Leibler divergence to assign labels to unlabeled data.

In addition to experiments on benchmark data, we demonstrate the effectiveness of the proposed kernels trough a case study of patients suffering from infectious postoperative complications. More specifically, we consider the problem of identifying patients that get surgical site infection after having undergone colorectal cancer surgery. Surgical site infection is a common hospital-acquired infection, and is associated with increased mortality rate, prolonged hospitalization and increased risk of readmission. Similarly to several earlier studies, we base the analysis on only blood samples, which are naturally represented as multivariate time series subject to missing data. The methodology considered in the case study in illustrated in Fig. 7.3. Our results show that the proposed kernel is capable of exploiting the informative missing patterns in the blood sample time series to a much larger degree than the baselines we compare to.

We believe the proposed kernels will be particularly useful in the medical domain where lack of labels and large amounts of missing data are two characteristic challenges. However, the kernels are not limited to this domain. Other application domains facing similar challenges might also benefit from the use of these kernels.

Contributions by the author The idea was conceived by myself and the method was further developed in collaborations with the other co-authors. Implementation of the method and experiments were carried out by myself. I wrote the draft of the manuscript.

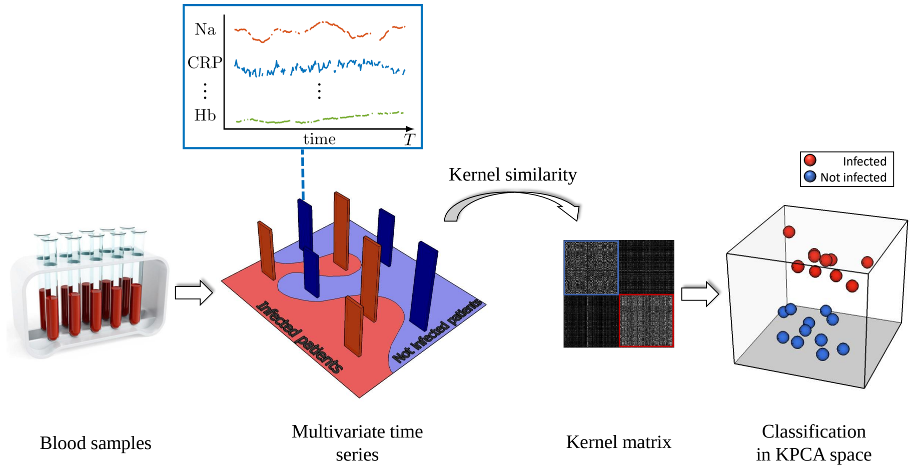
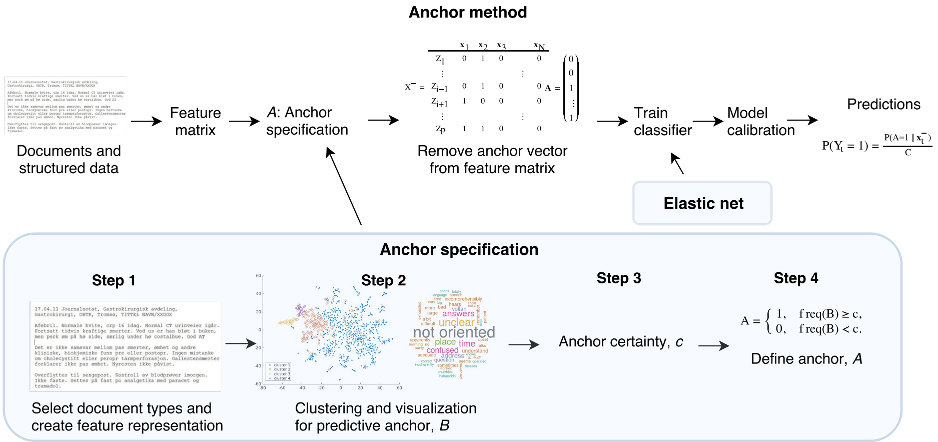


Figure 7.3: *Illustration of the methodology employed in the case study presented in Paper III.*

Paper IV - Using anchors from free text in electronic health records to diagnose postoperative delirium

This paper presents an approach for detecting postoperative delirium using free text documents from electronic health records without access to a labeled training set. The proposed methodology is based on a recent phenotyping algorithm in which noisy labeled training data are created semi-automatically by transforming key observations (anchors) into labels (Halpern et al., 2016). The anchors variables are highly informative for the phenotype of interest and are typically defined by clinical experts. Anchor learning is a PU-learning framework since the patients for which the anchor variable is present get a positive label, whereas nothing can be said for the patients that do not have the anchor.

The novelties in this paper are that we propose a novel approach for specifying anchors from free text documents, following an exploratory data analysis approach based on clustering and data visualization techniques. Additionally, we modify the existing anchor learning framework, by introducing a classifier that is better suited for low sample size problems. Experiments demonstrate that the proposed approach is well suited to detect postoperative delirium. An illustration of the methodology employed in this work is shown in Figure 7.4.


Anchor method
Documents and structured data
Feature matrix
A: Anchor specification
x_1 x_2 x_3 x_N
$X^- =$ z_1 z_{i-1} z_{i+1} z_p
$A = (0, 0, 1, \vdots, 1)$
Remove anchor vector from feature matrix
Train classifier
Model calibration
Predictions
$P(Y_t = 1) = \frac{P(A=1 \mid x_t^-)}{c}$
Elastic net
Anchor specification
Step 1
Step 2
Step 3
Step 4
cluster 1
cluster 2
cluster 3
cluster 4
not oriented
unclear
answers
place
time
confused
address
question
understand
vollan
incomprehensibly
Anchor certainty, c
$A = \begin{cases} 1, & freq(B) \geq c, \\ 0, & freq(B) < c. \end{cases}$
Define anchor, A
Select document types and create feature representation
Clustering and visualization for predictive anchor, B

17.04.11 Journalnotat, Gastrokirurgisk avdeling,
Gastrokirurgi, OBTR, Tromsø, TITTEL NAVN/XXDDX

Afebril. Normale hvite, crp 16 idag. Normal CT urinveier igår. Fortsatt tidvis kraftige smerter. Ved us er han bløt i buken, men perk øm på hø side, særlig under hø costalbue. God AT

Det er ikke samsvar mellom pas smerter, ømhet og andre kliniske, biokjemiske funn pre eller postopr. Ingen mistanke om cholecystitt eller peropr tarmperforasjon. Gallestensmerter forklarer ikke pas ømhet. Nyresten ikke påvist.

Overflyttes til sengepost. Kontroll av blodprøver imorgen. Ikke faste. Settes på fast po analgetika med paracet og tramadol.

Figure 7.4: *Graphical abstract for Paper IV.*

Another novelty of this work is to study the problem of detecting postoperative delirium using anchor learning. For this reason, in addition to being of interest to theoretically oriented readers, this work is also useful for clinical practitioners. Postoperative delirium is a complication that is often seen in geriatric patients undergoing major surgery. Despite that the consequences of this complication are potentially very serious, delirium is hard to detect and therefore often goes undiagnosed. It is also an example of a resource demanding complication that is under-coded, and thereby leads to too low reimbursement for the hospitals. Creating a method that accurately detects delirium might therefore help uncover the prevalence of this complication, providing clinically relevant knowledge, and a more correct income for the hospitals.

Preliminary work, which lead to this paper, was presented at the Workshop on Machine Learning in Healthcare, Conference on Neural Information Processing Systems Montreal (NIPS), December 2015 and Regional helseforskningskonferanse 2016, Tromsø, Norway, November 2016.

Contributions by the author I developed the method in close collaboration with the other co-authors. Implementation of the method and experiments were carried out by myself. I wrote the draft of the manuscript.

Chapter 8

Concluding remarks

In this thesis, we addressed the problem of getting access to large amounts of high quality labeled training data in data-driven healthcare. One of our solutions to this problem was to develop and employ machine learning methods that can deal with unlabeled training data, i.e. *unsupervised* methods. In the first and third work, we presented novel kernels for multivariate time series subject to missing data, which frequently occur in EHRs. These kernels are particularly useful when there is a lack of labels.

We also investigated the use of clinical expertize to extract partially and noisy labeled training data in a semi-automatic manner. In the second paper, we developed a semi-supervised dimensionality reduction method, specially designed for this type of data. The method can deal with multi-label data, and we therefore foresee that it can be a very useful tool e.g. in studies of patients suffering from multi-morbidities, which we demonstrated in a case study. The third work also presented a time series kernel capable of exploiting *weak supervision* information. In the last work, we created partially and noisy labeled training data. However, in this case, the partially labeled subset is supposed to contain only positive examples, leading to a PU-learning problem. Using this framework, we were able to accurately identify patients suffering from postoperative delirium, a common complication among the elderly after high-risk surgeries.

In addition to the challenge posed by lack of labels, we also advanced data-driven healthcare by addressing challenges related to other characteristics of EHR data, such as *missing data*, *temporality*, and *high dimensionality*. In fact, one of the main focuses of the first and third work was to deal with temporal data in presence of missing elements, whereas in the second work we presented a dimensionality reduction method. In Paper I, we also demonstrated that the TCK could be useful for learning lower dimensional representations of multivariate time series by using it as a kernel in KPCA. In Paper III, we learned representations of blood sample time series containing large amounts of missing values by exploiting the fact that the missing patterns were informative.

We conclude that, with the four lines of research presented in this thesis, we contributed to advance the field of data-driven healthcare, mainly by addressing the challenges posed by lack of labels, missing data, temporality and high dimensional data, and we made theoretical contributions to the fields of unsupervised and weakly supervised learning.

8.1 Limitations and further work

We acknowledge that every research paper has both strengths and weaknesses. Therefore, we end this part by providing a discussion of limitations, usefulness and future work for the research presented in this thesis.

Paper I. We mentioned in Paper I that in future work it would be interesting to investigate if the use of more general covariance structures in the mixture models and/or hidden Markov models (Panuccio et al., 2002) as base models could improve TCK. However, we have also identified several weaknesses that should be considered as future work.

An underlying assumption in the TCK is that the missingness mechanism (see Appendix A) is *missing at random* (MAR). However, our experiments demonstrated that even though in practice the missingness mechanism is *missing not at random* (MNAR), the TCK could still provide the desired performance. Nevertheless, we acknowledge that the assumption of MAR is a limitation. For instance, temporal data extracted from EHRs in some cases contain missing values that are MNAR and informative. Therefore, it would be interesting to create a kernel with weaker assumptions on the missingness mechanism.

Much real-world time series data are irregularly sampled, i.e. the time intervals are of different length. It would therefore be useful to modify the TCK to account for irregular sampling. We also believe that the TCK can be modified to better account for the cluster assumption (points in high density regions belong to the same cluster), for instance using concepts from information theory such as divergence. Further, it would be interesting to weigh the contribution from the individual base models in a data-driven way. For instance, using ideas from multiple kernel learning (Gönen and Alpaydın, 2011). Finally, we also mention that, even though the TCK can be implemented via an embarrassingly parallel procedure, it would be useful to work on reducing the computational complexity.

Paper II. In the experimental section of this work, in addition to evaluating the proposed method visually for some datasets, we combined the NMLSDR with a popular multi-label classifier, namely the multi-label k-nearest neighbor classifier (ML-kNN) (Zhang and Zhou, 2007). By doing so, we could quantitatively evaluate the quality of the embeddings learned by the NMLSDR and compare to alternative dimensionality reduction methods. However, it should be noticed that many other multi-label classifiers exist (Tahir et al., 2012; Madjarov et al., 2012; Xu, 2013; Chen et al., 2016b; Liu et al., 2018b; Wang et al., 2014a; Trajdos and Kurzynski, 2015, 2018; Zhuang et al., 2018). It would be interesting to investigate if the proposed method outperforms alternative dimensionality reduction methods in conjunction with other classifiers as well.

When performing label propagation using a graph there are a couple choices that can be made, which possibly could influence the result. More precisely, there are two main components that affect the outcome of label propagation; the particular method chosen and how the graph is constructed. Both of these two components are important (Zhu, 2005, 2006). In our work, we employed a kNN neighborhood graph with binary weights. However, it would be interesting to investigate how sensitive NMLSDR is to the choices made for constructing the graph.

In this work, we considered a case study of patients suffering from multimorbidities and showed that the proposed method performed well for this purpose. However, we restricted the study to only consider patients suffering from at most two different chronic conditions. In future work, we would like to extend this and consider simultaneous phenotyping of many multimorbidities.

Paper III. With this work, we addressed one of the main limitations of Paper I, namely that the missingness mechanism was assumed to be ignorable. To this end, we introduced mixed mode mixture models into the ensemble learning framework to model both the data and the missing patterns. The discrete modality (the missing patterns) was modeled using Bernoulli distributions. One reason for using a relatively simple model such as the Bernoulli distribution is related to the fact the we were using an ensemble learning approach. More precisely, two necessary conditions in ensemble learning are *diversity* and *accuracy* (see Sec. 5.1.4). However, often there is a trade-off between these two conditions. Making the base models more flexible could lead to improved individual accuracy, but could also come at the cost of decreased diversity since the probability that the base models make the same errors on test data might increase with increased flexibility (different base models might overtrain for the same reason). Hence, it is not necessarily the case that the time series cluster kernels will improve if we use more flexible base models (e.g. general covariance structure instead of diagonal covariance, incorporating time dependence and attribute dependence for the discrete modality, etc.). Nevertheless, we acknowledge that there might exist other approaches to design the base models such that one in a better way captures the missing patterns. Thanks to the "modularity" of our framework, it is possible to seamlessly extend our model to include a more sophisticated formulation to model the missing data.

To make the kernels better suited to situations when some labels are provided, we also proposed methods to incorporate both strong and weak supervision information into the procedure for learning the kernel. This was done in an intermediate and independent processing step. However, it would be interesting to incorporate label information into the training of parameters of the base models as well. In this regard, one could consider to use similar frameworks to the ones presented by Miller and Uyar (1997); Xing et al. (2013) as base models.

Paper IV. A long discussion of limitations and further work is provided in the paper. However, we would like to highlight a few more aspects here.

One of the main limitations of anchor learning is that in practice it could be difficult to find reliable anchors. In particular, the more the condition $P(Y = 1 \mid A = 1) = 1$, is broken, the more corrupted the positive labels will get. We introduced a problem-specific method to define anchors for postoperative delirium such that thes condition holds to a larger degree.

However, an alternative approach, which probably is more general than our solution, would be to explicitly account for label noise among the positive examples. In this regard, recent theoretical and practical advances in PU-learning are highly relevant. For instance, Jain et al. (2016) developed an effective algorithm for high-dimensional data, which is robust to label noise in the positive examples. Menon et al. (2015) developed an alternative algorithm for the same purpose. These two algorithms were key components in the MutPred2 framework, a software package genetic and molecular data, developed by Pejaver et al. (2017). We believe that these methods could also make anchor learning more robust to label noise.

The current formulation of anchor learning is based on the PU-learning framework introduced by Elkan and Noto (2008). This framework does, however, only lead to unbiased risk estimators if the class-conditional densities for the positive and negative class have non-overlapping support. This is an unrealistic assumption, and if it was true, then any sufficiently flexible classifier would do a perfect job to separate the classes. Hence, we believe that the performance can be improved using frameworks such as the ones proposed by (Du Plessis et al., 2014, 2015) in which the risk estimators are unbiased. Moreover, to formulate anchor learning via flexible models such as deep neural networks would also be interesting. In that case, we think the framework proposed by Kiryo et al. (2017) is useful.

We also think that the work of Niu et al. (2016) could be useful to incorporate into anchor learning. By doing so, one could theoretically estimate how many patients one needs to manually annotate with ground-truth labels in order to achieve similar performance as anchor learning.

Finally, we mention that an important next step for researchers working with data-driven healthcare is to also start to translate promising algorithms to clinical practice so that the final outcome of research is not just a performance gain reported in an academic journal, but also leads to improved healthcare. There are probably many reasons why the impact on current clinical practice has been low until now; a mismatch between the performance of the methods and unrealistically high expectations, difficulties related to privacy, legal aspects, ethics and interpretation of predictions, and lack of prospective clinical trials to validate and demonstrate benefits compared to current practice (Fröhlich et al., 2018). To improve this in the future, probably large interdisciplinary efforts are needed. Healthcare workers, data scientists, politicians, regulatory agencies, etc., have to go together to establish a common ground for what is reasonable to expect (avoid hype) as the outcome and benefits of data-driven healthcare and to establish practical guidelines and plans for how to effectively implement data-driven healthcare solutions into clinical practice.

Part IV

Included papers

Chapter 9

Paper I

Pattern Recognition 76 (2018) 569–581

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Time series cluster kernel for learning similarities between multivariate time series with missing data

Karl Øyvind Mikalsen[a,b,*], Filippo Maria Bianchi[b,c], Cristina Soguero-Ruiz[b,d], Robert Jenssen[b,c]

[a] Department of Mathematics and Statistics, UiT The Arctic University of Norway, Tromsø, Norway
[b] UiT Machine Learning Group, Norway
[c] Department of Physics and Technology, UiT, Tromsø, Norway
[d] Department of Signal Theory and Communications, Telematics and Computing, Universidad Rey Juan Carlos, Fuenlabrada, Spain

ARTICLE INFO

Article history:
Received 23 March 2017
Revised 11 October 2017
Accepted 30 November 2017
Available online 6 December 2017

Keywords:
Multivariate time series
Similarity measures
Kernel methods
Missing data
Gaussian mixture models
Ensemble learning

ABSTRACT

Similarity-based approaches represent a promising direction for time series analysis. However, many such methods rely on parameter tuning, and some have shortcomings if the time series are multivariate (MTS), due to dependencies between attributes, or the time series contain missing data. In this paper, we address these challenges within the powerful context of kernel methods by proposing the robust *time series cluster kernel* (TCK). The approach taken leverages the missing data handling properties of Gaussian mixture models (GMM) augmented with informative prior distributions. An ensemble learning approach is exploited to ensure robustness to parameters by combining the clustering results of many GMM to form the final kernel.

We evaluate the TCK on synthetic and real data and compare to other state-of-the-art techniques. The experimental results demonstrate that the TCK is robust to parameter choices, provides competitive results for MTS without missing data and outstanding results for missing data.

1. Introduction

Time series analysis is an important and mature research topic, especially in the context of univariate time series (UTS) prediction [1–4]. The field tackles real world problems in many different areas such as energy consumption [5], meteorology [6], climate studies [7], biology [8], medicine [9–12] and finance [13]. However, the need for analysis of multivariate time series (MTS) [14] is growing in modern society as data is increasingly collected simultaneously from multiple sources over time, often plagued by severe missing data problems [15,16]. These challenges complicate analysis considerably, and represent open directions in time series analysis research. The purpose of this paper is to answer such challenges, which will be achieved within the context of the powerful *kernel methods* [17,18] for reasons that will be discussed below.

Time series analysis approaches can be broadly categorized into two families: (i) *representation methods*, which provide high-level features for representing properties of the time series at hand, and (ii) *similarity measures*, which yield a meaningful similarity between different time series for further analysis [19,20].

Classic representation methods are for instance Fourier transforms, wavelets, singular value decomposition, symbolic aggregate approximation, and piecewise aggregate approximation [21–25]. Time series may also be represented through the parameters of model-based methods such as Gaussian mixture models (GMM) [26–28], Markov models and hidden Markov models (HMMs) [29–31], time series bitmaps [32] and variants of ARIMA [33–35]. An advantage with parametric models is that they can be naturally extended to the multivariate case. For detailed overviews on representation methods, we refer the interested reader to [19,20,36].

Of particular interest to this paper are similarity-based approaches. Once defined, such similarities between pairs of time series may be utilized in a wide range of applications, such as classification, clustering, and anomaly detection [37]. Time series similarity measures include for example dynamic time warping (DTW) [38], the longest common subsequence (LCSS) [39], the extended Frobenius norm (Eros) [40], and the Edit Distance with Real sequences (EDR) [41], and represent state-of-the-art performance in UTS prediction [19]. However, many of these measures cannot straightforwardly be extended to MTS such that they take relations between different attributes into account [42]. The learned

[*] Corresponding author at: Department of Mathematics and Statistics, Faculty of Science and Technology, UiT – The Arctic University of Norway, TromsØ N-9037, Norway.

E-mail address: karl.o.mikalsen@uit.no (K.Ø. Mikalsen).

https://doi.org/10.1016/j.patcog.2017.11.030
0031-3203/© 2017 Elsevier Ltd. All rights reserved.

pattern similarity (LPS) is an exception, based on the identification of segments-occurrence within the time series, which generalizes naturally to MTS [43] by means of regression trees where a bag-of-words type compressed representation is created, which in turn is used to compute the similarity.

A similarity measure that also is positive semi-definite (psd) is a *kernel* [18]. Kernel methods [18,44,45] have dominated machine learning and pattern recognition over two decades and have been very successful in many fields [46–49]. A main reason for this success is the well understood theory behind such methods, wherein nonlinear data structures can be handled via an implicit or explicit mapping to a reproducing kernel Hilbert space (RKHS) [50,51] defined by the choice of kernel. Prominent examples of kernel methods include the support vector machine (SVM) [52] and kernel principal component analysis (kPCA) [53].

However, many similarities (or equivalently dissimilarities) are non-metric as they do not satisfy the triangle-inequality, and in addition most of them are not psd and therefore not suited for kernel methods [54,55]. Attempts have been made to design kernels from non-metric distances such as DTW, of which the global alignment kernel (GAK) is an example [56]. There are also promising works on deriving kernels from parametric models, such as the probability product kernel [57], Fisher kernel [58], and reservoir based kernels [59]. Common to all these methods is however a strong dependence on a correct hyperparameter tuning, which is difficult to obtain in an unsupervised setting. Moreover, many of these methods cannot naturally be extended to deal with MTS, as they only capture the similarities between individual attributes and do not model the dependencies between multiple attributes [42]. Equally important, these methods are not designed to handle missing data, an important limitation in many existing scenarios, such as clinical data where MTS originating from electronic health records (EHRs) often contain missing data [9–11,60].

In this work, we propose a new kernel for computing similarities between MTS that is able to handle missing data without having to resort to imputation methods [61]. We denote this new measure as the *time series cluster kernel* (TCK). Importantly, the novel kernel is robust and designed in an unsupervised manner, in the sense that no critical hyperparameter choices have to be made by the user. The approach taken is to leverage the missing data handling properties of GMM modeling following the idea of [26], where robustness to sparsely sampled data is ensured by extending the GMM using informative prior distributions. However, we are not fitting a single parametric model, but rather exploiting an ensemble learning approach [62] wherein robustness to hyperparameters is ensured by joining the clustering results of many GMM to form the final kernel. This is to some degree analogous to the approaches taken in [63] and [64]. More specifically, each GMM is initialized with different numbers of mixture components and random initial conditions and is fit to a randomly chosen subsample of the data, attributes and time segment, through an embarrassingly parallel procedure. This also increases the robustness against noise. The posterior assignments provided by each model are combined to form a kernel matrix, i.e. a psd similarity matrix. This opens the door to clustering, classification, etc., of MTS within the framework of kernel methods, benefiting from the vast body of work in that field. The procedure is summarized in Fig. 1.

In the experimental section we illustrate some of the potentials of the TCK by applying it to classification, clustering, dimensionality reduction and visualization tasks. In addition to the widely used DTW, we compare to GAK and LPS. The latter inherits the decision tree approach to handle missing data, is similar in spirit to the TCK in the sense of being based on an ensemble strategy [43], and is considered the state-of-the-art for MTS. As an additional contribution, we show in Appendix A that the LPS is in fact a kernel itself, a result that to the authors best knowledge has not been proven before. The experimental results demonstrate that TCK is very robust to hyperparameter choices, provides competitive results for MTS without missing data and outstanding results for MTS with missing data. This we believe provides a useful tool across a variety of applied domains in MTS analysis, where missing data may be problematic.

The remainder of the paper is organized as follows. In Section 2 we present related works, whereas in Section 3, we give the background needed for building the proposed method. In Section 4 we provide the details of the TCK, whereas in Section 5 we evaluate it on synthetic and real data and compare to LPS, GAK and DTW. Section 6 contains conclusions and future work.

2. Related work

While several (dis)similarity measures have been defined over the years to compare time series, many of those measures are not psd and hence not suitable for kernel approaches. In this section we review some of the main kernels functions that have been proposed for time series data.

The simplest possible approach is to treat the time series as vectors and apply well-known kernels such as a linear or radial basis kernel [17]. While this approach works well in some circumstances, time dependencies and the relationships among multiple attributes in the MTS are not explicitly modeled.

DTW [38] is one of the most commonly used similarity measures for UTS and has become the state-of-the-art in many practical applications [65–68]. Several formulations have been proposed to extend DTW to the multidimensional setting [42,69]. Since DTW does not satisfy the triangle inequality, it is not negative definite and, therefore, one cannot obtain a psd kernel by applying an exponential function to it [70]. Such an indefinite kernel may lead to a non-convex optimization problem (e.g., in an SVM), which hinders the applicability of the model [54]. Several approaches have been proposed to limit this drawback at the cost of more complex and costly computations. In [71,72] ad hoc spectral transformations were employed to obtain a psd matrix. Cuturi et al. [56] designed a DTW-based kernel using global alignments (GAK). Marteau and Gibet proposed an approach that combines DTW and edit distances with a recursive regularizing term [55].

Conversely, there exists a class of (probabilistic) kernels operating on the configurations of a given parametric model, where the idea is to leverage the way distributions capture similarity. For instance, the Fisher kernel assumes an underlying generative model to explain all observed data [58]. The Fisher kernel maps each time series x into a feature vector U_x, which is the gradient of the log-likelihood of the generative model fit on the dataset. The kernel is defined as $K(x_i, x_j) = U_{x_i}^T \mathcal{I}^{-1} U_{x_j}$, where $\mathcal{I}$ is the fisher information matrix. Another example is the probability product kernel [57], which is evaluated by means of the Bhattacharyya distance in the probability space. A further representative is the marginalized kernel [73], designed to deal with objects generated from latent variable models. Given two visible variables, x and x' and two hidden variables, h and h', at first, a joint kernel $K_z(z,\ z')$ is defined over the two combined variables $z = (x, h)$ and $z' = (x', h')$. Then, a marginalized kernel for visible data is derived from the expectation with respect to hidden variables: $K(x, x') = \sum_h \sum_{h'} p(h|x) p(h'|x') K_z(z, z')$. The posterior distributions are in general unknown and are estimated by fitting a parametric model on the data.

In several cases, the assumption of a single parametric model underlying all the data may be too strong. Additionally, finding the most suitable parametric model is a crucial and often difficult task, which must be repeated every time a new dataset is processed. This issue is addressed by the autoregressive kernel [63], which

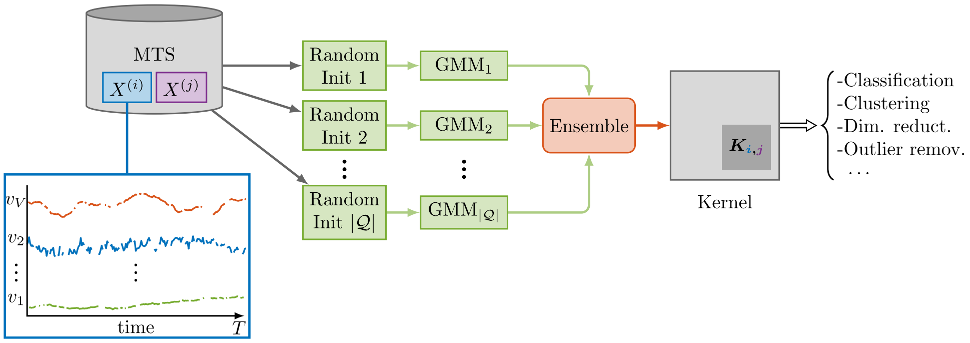


Fig. 1. Schematic depiction of the procedure used to compute the TCK.

evaluates the similarity of two time series on the corresponding likelihood profiles of a vector autoregressive model of a given order, across all possible parameter settings, controlled by a prior. The kernel is then evaluated as the dot product in the parameter space of such profiles, used as sequence representations. The reservoir based kernels [59], map the time series into a high dimensional, dynamical feature space, where a linear readout is trained to discriminate each signal. These kernels fit reservoir models sharing the same fixed reservoir topology to all time series. Since the reservoir provides a rich pool of dynamical features, it is considered to be “generic” and, contrarily to kernels based on a single parametric model, it is able to represent a wide variety of dynamics for different datasets.

The methodology we propose is related to this last class of kernels. In order to create the TCK, we fuse the framework of representing time series via parametric models with similarity and kernel based methods. More specifically, the TCK leverages an ensemble of multiple models that, while they share the same parametric form, are trained on different subset of data, each time with different, randomly chosen initial conditions.

3. Background

In this section we provide a brief background on kernels, introduce the notation adopted in the remainder of the paper and provide the frameworks that our method builds on. More specifically, we introduce the diagonal covariance GMM for MTS with missing data, the extended GMM framework with empirical priors and the related procedure to estimate the parameters of this model.

3.1. Background on kernels

Thorough overviews on kernels can be found in [17,18,52,70]. Here we briefly review some basic definitions and properties, following [52].

Definition 1. Let $\mathcal{X}$ be a non-empty set. A function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *kernel* if there exists a $\mathbb{R}$ -Hilbert space $\mathcal{H}$ and a map $\Phi: \mathcal{X} \to \mathcal{H}$ such that $\forall x,\ y \in \mathcal{X},\ \ k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$.

From this definition it can be shown that a kernel is symmetric and psd, meaning that $\forall n \geq 1$, $\forall (a_1, \ldots, a_n) \in \mathbb{R}^n$, $\forall (x_1, \ldots, x_n) \in \mathcal{X}^n$, $\Sigma_{i,\,j} a_i a_j K(x_i,\ x_j) \geq 0$. Of major importance in kernel methods are also the concepts of reproducing kernels and reproducing kernel Hilbert spaces (RKHS), described by the following definition.

Definition 2. Let $\mathcal{X}$ be a non-empty set, $\mathcal{H}$ a Hilbert space and $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a function. k is a reproducing kernel, and $\mathcal{H}$ a RKHS, if $\forall x \in \mathcal{X},\ \ \forall f \in \mathcal{H},\ \ k(\cdot, x) \in \mathcal{H}$ and $\langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$ (reproducing property).

These concepts are highly connected to kernels. In fact reproducing kernels are kernels, and every kernel is associated with a unique RKHS (Moore–Aronszajn theorem), and vice versa. Moreover, the *representer theorem* states that every function in an RKHS that optimizes an empirical risk function can be expressed as a linear combination of kernels centered at the training points. These properties have very useful implications, e.g. in an SVM, since an infinite dimensional empirical risk minimization problem can be simplified to a finite dimensional problem and the solution is included in the linear span of the kernel function evaluated at the training points.

3.2. MTS with missing data

We define a UTS, x, as a sequence of real numbers ordered in time, $x = \{x(t) \in \mathbb{R} \mid t = 1, 2, \ldots, T\}$. The independent time variable, t, is without loss of generality assumed to be discrete and the number of observations in the sequence, T, is the *length* of the UTS.

A MTS X is defined as a (finite) sequence of UTS, $X = \{x_v \in \mathbb{R}^T \mid v = 1, 2, \ldots, V\}$, where each attribute, x_v, is a UTS of length T. The number of UTS, V, is the *dimension* of X. The length T of the UTS x_v is also the length of the MTS X. Hence, a V-dimensional MTS, X, of length T can be represented as a matrix in $\mathbb{R}^{V \times T}$.

Given a dataset of N MTS, we denote $X^{(n)}$ the nth MTS. An incompletely observed MTS is described by the pair $(X^{(n)}, R^{(n)})$, where $R^{(n)}$ is a binary MTS with entry $r_v^{(n)}(t) = 0$ if the realization $x_v^{(n)}(t)$ is missing and $r_v^{(n)}(t) = 1$ if it is observed.

3.3. Diagonal covariance GMM for MTS with missing data

A GMM is a mixture of G components, with each component belonging to a normal distribution. Hence, the components are described by the mixing coefficients θ_g, means μ_g and covariances Σ_g. The mixing coefficients θ_g satisfy $0 \leq \theta_g \leq 1$ and $\sum_{g=1}^{G} \theta_g = 1$.

We formulate the GMM in terms of a latent random variable Z, represented as a G-dimensional one-hot vector, whose marginal distribution is given by $p(Z \mid \Theta) = \prod_{g=1}^{G} \theta_g^{Z_g}$. The conditional distribution for the MTS X, given Z, is a multivariate normal distribution, $p(X \mid Z_g = 1,\ \Theta) = \mathcal{N}(X \mid \mu_g, \Sigma_g)$. Hence, the GMM can be described by its probability density function (pdf), given by

$$p(X) = \sum_{Z} p(Z) p(X \mid Z,\ \Theta) = \sum_{g=1}^{G} \theta_g \mathcal{N}(X \mid \mu_g, \Sigma_g). \tag{1}$$

The GMM described by Eq. (1) holds for completely observed data and a general covariance. However, in the diagonal covariance GMM considered in this work, the following assumptions are made. The MTS are characterized by time-dependent means, expressed by $\mu_g = \{\mu_{gv} \in \mathbb{R}^T \mid v = 1, \ldots, V\}$, where μ_{gv} is a UTS, whereas the covariances are constrained to be constant over time. Accordingly, the covariance matrix is $\Sigma_g = diag\{\sigma_{g1}^2, \ldots, \sigma_{gV}^2\}$, being σ_{gv}^2 the variance of attribute v. Moreover, the data is assumed to be *missing at random* (MAR), i.e. the missing elements are only dependent on the observed values.

Under these assumptions, missing data can be analytically integrated away, such that imputation is not needed [74], and the pdf for the incompletely observed MTS (X, R) is given by

$$p(X \mid R,\ \Theta) = \sum_{g=1}^{G} \theta_g \prod_{v=1}^{V} \prod_{t=1}^{T} \mathcal{N}(x_v(t) \mid \mu_{gv}(t), \sigma_{gv})^{r_v(t)} \tag{2}$$

The conditional probability of Z given X, can be found using Bayes' theorem,

$$\begin{aligned} \pi_g &\equiv P(Z_g = 1 \mid X,\ R,\ \Theta) \\ &= \frac{\theta_g \prod_{v=1}^{V} \prod_{t=1}^{T} \mathcal{N}(x_v(t) \mid \mu_{gv}(t), \sigma_{gv})^{r_v(t)}}{\sum_{g=1}^{G} \theta_g \prod_{v=1}^{V} \prod_{t=1}^{T} \mathcal{N}(x_v(t) \mid \mu_{gv}(t), \sigma_{gv})^{r_v(t)}}. \end{aligned} \tag{3}$$

θ_g can be thought of as the prior probability of X belonging to component g, and therefore Eq. (3) describes the corresponding posterior probability.

To fit a GMM to a dataset, one needs to learn the parameters $\Theta = \{\theta_g,\ \mu_g, \sigma_g\}_{g=1}^{G}$. The standard way to do this is to perform maximum likelihood expectation maximization (EM) [75]. However, to be able to deal with large amounts of missing data, one can introduce informative priors for the parameters and estimate them using maximum a posteriori expectation maximization (MAP-EM) [26]. This ensures each cluster mean to be smooth over time and clusters containing few time series, to have parameters similar to the mean and covariance computed over the whole dataset. We summarize this procedure in the next subsection (see Ref. [26] for details).

3.4. MAP-EM diagonal covariance GMM augmented with empirical prior

To enforce smoothness, a kernel-based Gaussian prior is defined for the mean, $P(\mu_{gv}) = \mathcal{N}(\mu_{gv} \mid m_v, S_v)$. m_v are the empirical means and the prior covariance matrices, S_v, are defined as $S_v = s_v \mathcal{K}$, where s_v are empirical standard deviations and $\mathcal{K}$ is a kernel matrix, whose elements are $\mathcal{K}_{tt'} = b_0 \exp(-a_0(t-t')^2)$, $t, t' = 1, \ldots, T$. a_0, b_0 are user-defined hyperparameters. An inverse Gamma distribution prior is put on the standard deviation σ_{gv}, $P(\sigma_{gv}) \propto \sigma_{gv}^{-N_0} \exp(-\frac{N_0 s_v}{2\sigma_{gv}^2})$, where N_0 is a user-defined hyperparameter. We denote $\Omega = \{a_0, b_0, N_0\}$ the set of hyperparameters. Estimates of parameters Θ are found using MAP-EM [76,77], according to Algorithm 1.

Algorithm 1 MAP-EM diagonal covariance GMM.

```
Input Dataset $\{(X^{(n)}, R^{(n)})\}_{n=1}^{N}$, hyperparameters $\Omega$ and number of mixtures $G$.
1: Initialize the parameters $\Theta$.
2: E-step. For each MTS $X^{(n)}$, evaluate the posterior probabilities using current parameter estimates,$\pi_g^{(n)} = P(Z_g = 1 \mid X^{(n)},\ R^{(n)},\ \Theta)$.
3: M-step. Update parameters using the current posteriors
     $\theta_g = N^{-1} \sum_{n=1}^{N} \pi_g^{(n)}$
     $\sigma_{gv}^2 = \left(N_0 + \sum_{n=1}^{N} \sum_{t=1}^{T} r_v^{(n)}(t)\ \pi_g^{(n)}\right)^{-1} \times \left(N_0 s_v^2 + \sum_{n=1}^{N} \sum_{t=1}^{T} r_v^{(n)}(t)\ \pi_g^{(n)} \left(x_v^{(n)}(t) - \mu_{gv}(t)\right)^2\right)$
     $\mu_{gv} = \left(S_v^{-1} + \sigma_{gv}^{-2} \sum_{n=1}^{N} \pi_g^{(n)} \mathrm{diag}(r_v^{(n)})\right)^{-1} \times \left(S_v^{-1} m_v + \sigma_{gv}^{-2} \sum_{n=1}^{N} \pi_g^{(n)} \mathrm{diag}(r_v^{(n)})\ x_v^{(n)}\right)$
4: Repeat steps 2 and 3 until convergence.
Output Posteriors $\Pi^{(n)} \equiv \left(\pi_1^{(n)}, \ldots, \pi_G^{(n)}\right)^T$ and mixture parameters $\Theta$.
```

4. Time series cluster kernel (TCK)

Methods based on GMM, in conjunction with EM, have been successfully applied in different contexts, such as density estimation and clustering [78]. As a major drawback, these methods often require to solve a non-convex optimization problem, whose outcome depends on the initial conditions [77,79]. The model described in the previous section depends on initialization of parameters Θ and the chosen number of clusters G [26]. Moreover, three different hyper-parameters, a_0, b_0, N_0, have to be set. In particular, modeling the covariance in time is difficult; choosing a too small hyperparameter a_0 leads to a degenerate covariance matrix that cannot be inverted. On the other hand, a too large value would basically remove the covariance such that the prior knowledge is not incorporated. Furthermore, a single GMM provides a limited descriptive flexibility, due to its parametric nature.

Ensemble learning has been adopted both in classification, where classifiers are combined through e.g. bagging or boosting [80–82], and clustering [83–85]. Typically, in ensemble clustering one integrates the outcomes of the same algorithm as it processes different data subsets, being configured with different parameters or initial conditions, in order to capture local and global structures in the underlying data [84,86] and to provide a more stable and robust final clustering result. Hence, the idea is to combine the results of many weaker models to deliver an estimator with statistical, computational and representational advantages [62], which are lower variance, lower sensitivity to local optima and a broader span of representable functions, respectively.

We propose an ensemble approach that combines multiple GMM, whose diversity is ensured by training the models on subsamples of data, attributes and time segments, using different numbers of mixture components and random initialization of Θ and hyperparameters. Thus, we generate a model robust to parameters and noise, also capable of capturing different levels of granularity in the data. To ensure robustness to missing data, we use the diagonal covariance GMM augmented with the informative priors described in the previous section as base models in the ensemble.

Potentially, we could have followed the idea of [87] to create a density function from an ensemble of GMM. Even though several methods rely on density estimation [78], we aim on deriving a *similarity measure*, which provides a general-purpose data representation, fundamental in many applications in time-series analysis, such as classification, clustering, outlier detection and dimensionality reduction [37].

Moreover, we ensure the similarity measure to be psd, i.e. a *kernel*. Specifically, the linear span of posterior distributions π_g, formed as G-vectors, with ordinary inner product, constitutes a Hilbert space. We explicitly let the *feature map* Φ be these posteriors. Hence, the TCK is an inner product between two distributions and therefore forms a linear kernel in the space of posterior distributions. Given an ensemble of GMM, we create the TCK using the fact that the sum of kernels is also a kernel.

4.1. Method details

To build the TCK kernel matrix, we first fit different diagonal covariance GMM to the MTS dataset. To ensure diversity, each GMM model uses a number of components from the inter-

Algorithm 2 TCK kernel. Training phase.

```
Input Training data $\{(X^{(n)}, R^{(n)})\}_{n=1}^{N}$ , $Q$ initializations, $C$ maximal
      number of mixture components.
Initialize kernel matrix $K = 0_{N \times N}$.
for $q \in \mathcal{Q}$ do
    Compute posteriors $\Pi^{(n)}(q) \equiv \left(\pi_1^{(n)}, \dots, \pi_{q_2}^{(n)}\right)^T$,
    $n = 1, \dots, N$, by applying Algorithm 1 with $q_2$ clusters and
    by randomly selecting,
      i. hyperparameters $\Omega(q)$,
     ii. a time segment $\mathcal{T}(q)$ of length $T_{min} \le |\mathcal{T}(q)| \le T_{max}$,
    iii. a subset of attributes, $\mathcal{V}(q) \subset (1, \dots, V)$, with cardinality
         $V_{min} \le |\mathcal{V}(q)| \le V_{max}$,
     iv. a subset of MTS, $\eta(q) \subset (1, \dots, N)$, with cardinality $N_{min} \le$
         $|\eta(q)| \le N$,
      v. initialization of the mixture parameters $\Theta(q)$.
    Update kernel matrix, $K_{nm} = K_{nm} + \Pi^{(n)}(q)^T \Pi^{(m)}(q)$,
    $n, m = 1, \dots, N$.
end for
Output $K$ TCK kernel matrix, time segments $\mathcal{T}(q)$, subsets of at-
       tributes $\mathcal{V}(q)$, subsets of MTS $\eta(q)$, GMM parameters $\Theta(q)$ and
       posteriors $\Pi^{(n)}(q)$.
```

val [2, C]. For each number of components, we apply Q different random initial conditions and hyperparameters. We let $\mathcal{Q} = \{q = (q_1, q_2) \mid q_1 = 1, \dots Q,\ q_2 = 2, \dots, C\}$ be the index set keeping track of initial conditions and hyperparameters (q_1), and the number of components (q_2). Moreover, each model is trained on a random subset of MTS, accounting only a random subset of variables $\mathcal{V}$, with cardinality $|\mathcal{V}| \le V$, over a randomly chosen time segment $\mathcal{T}, |\mathcal{T}| \le T$. The inner products of the posterior distributions from each mixture component are then added up to build the TCK kernel matrix, according to the ensemble strategy [88]. Algorithm 2 describes the details of the method.

In order to be able to compute similarities with MTS not available at the training phase, one needs to store the time segments $\mathcal{T}(q)$, subsets of attributes $\mathcal{V}(q)$, GMM parameters $\Theta(q)$ and posteriors $\Pi^{(n)}(q)$. Then, the TCK for such out-of-sample MTS is evaluated according to Algorithm 3.

Algorithm 3 TCK kernel. Test phase.

```
Input Test set $\left\{(X^{*(m)}, R^{*(m)})\right\}_{m=1}^{M}$, time segments $\mathcal{T}(q)$, subsets of
      attributes $\mathcal{V}(q)$, subsets of MTS $\eta(q)$, GMM parameters $\Theta(q)$
      and posteriors $\Pi^{(n)}(q)$.
Initialize kernel matrix $K^* = 0_{N \times M}$.
for $q \in \mathcal{Q}$ do
    Compute posteriors $\Pi^{*(m)}(q)$, $m = 1, \dots, M$ by applying Eq.
    (3) with mixture parameters $\Theta(q)$.
    Update kernel matrix, $K^*_{nm} = K^*_{nm} + \Pi^{(n)}(q)^T \Pi^{*(m)}(q)$, $n =$
    $1, \dots, N$, $m = 1, \dots, M$.
end for
Output $K^*$ TCK test kernel matrix
```

4.2. Parameters and robustness

The maximal number of mixture components in the GMM, C, should be set high enough to capture the local structure in the data. On the other hand, it should be set reasonably lower than the number of MTS in the dataset in order to be able to estimate the parameters of the GMM. Intuitively, a high number of realizations Q improves the robustness of the ensemble of clusterings. However, more realizations comes at the expense of an increased computational cost. In the end of next section we show experimentally that it is not critical to correctly tune these two hyperparameters as they just have to be set high enough.

Through empirical evaluations we have seen that none the other hyperparameters are critical. We set default hyperparameters as follows. The hyperparameters are sampled according to a uniform distribution from pre-defined intervals. Specifically, we let $a_0 \in (0.001, 1)$, $b_0 \in (0.005, 0.2)$ and $N_0 \in (0.001, 0.2)$. The subsets of attributes are selected randomly by sampling according to a uniform distribution from $\{2, \dots, V_{\max}\}$. The lower bound is set to two, since we want to allow the algorithm to learn possible interdependencies between at least two attributes. The time segments are sampled from $\{1, \dots, T\}$ and the length of the segments are allowed to vary between $T_{\min}$ and $T_{\max}$. In order to be able to capture some trends in the data we set $T_{\min} = 6$. We let the minimal size of the subset of MTS be 80% of the dataset.

We do acknowledge that for long MTS the proposed method becomes computationally demanding, as the complexity scales as $\mathcal{O}(T^3)$. Moreover, there is a potential issue in Eq. (3) since multiplying together very small numbers both in the nominator and denominator could yield to numerically unstable expressions close to 0/0. While there is no theoretical problem, since the normal distribution is never exactly zero, the posterior for some outliers could have a value close to the numerical precision. In fact, since the posterior assignments are numbers lower than 1, the value of their product can be small if V and T are large. We address this issue by putting upper thresholds on the length of the time segments, $T_{\max}$, and number of attributes, $V_{\max}$, which is justified by the fact that the TCK is learned using an ensemble strategy. Moreover, to avoid problems for outliers we put a lower bound on the value for the conditional distribution for $x_v(t)$ at $\mathcal{N}(3 \mid 0, 1)$. In fact, it is very unlikely that a data point generated from a normal distribution is more than three standard deviations away from the mean.

4.3. Algorithmic complexity

4.3.1. Training complexity

The computational complexity of the EM procedure is dominated by the update of the mean, whose cost is $\mathcal{O}(2T^3 + NVT^2)$. Hence, for G components and I iterations, the total cost is $\mathcal{O}(IG(2T^3 + NVT^2))$. The computation of the TCK kernel involves both the MAP-EM estimation and the kernel matrix generation for each $q \in \mathcal{Q}$, whose cost is upper-bounded by $\mathcal{O}(N^2C)$. The cost of a single evaluation q is therefore bounded by $\mathcal{O}(N^2C + IC(2T_{\max}^3 + NV_{\max}T_{\max}^2))$. We underline that the effective computational time can be reduced substantially through parallelization, since each instance $q \in \mathcal{Q}$ can be evaluated independently. As we can see, the cost has a quadratic dependence on N, which becomes the dominating term in large datasets. We note that in spectral methods the eigen-decomposition costs $\mathcal{O}(N^3)$ with a consequent complexity higher than TCK for large N.

4.3.2. Testing complexity

For a test MTS one has to evaluate $|\mathcal{Q}|$ posteriors, with a complexity bounded by $\mathcal{O}(CV_{\max}T_{\max})$. The complexity of computing the similarity with the N training MTS is bounded by $\mathcal{O}(NC)$. Hence, for each $q \in \mathcal{Q}$, the testing complexity is $\mathcal{O}(NC + CV_{\max}T_{\max})$. Note that also the test phase is embarrassingly parallelizable.

4.4. Properties

In this section we demonstrate that TCK is a proper kernel and we discuss some of its properties. We let $\mathcal{X} = \mathbb{R}^{V \times T}$ be the space of V-variate MTS of length T and $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be the TCK.

Theorem 1. *K is a kernel.*

Proof. According to the definition of TCK, we have $K(X^{(n)}, X^{(m)}) = \sum_{q\in\mathcal{Q}} k_q(X^{(n)}, X^{(m)})$, where $k_q(X^{(n)}, X^{(m)}) = \Pi^{(n)}(q)^T\Pi^{(m)}(q)$. Since the sum of kernels is a kernel, it is sufficient to demonstrate that k_q is a kernel. We define $\mathcal{H}_q = \{f = \sum_{n=1}^N \alpha_n\Pi^{(n)}(q) \mid N \in \mathbb{N},\ X^{(1)}, \ldots, X^{(N)} \in \mathcal{X},\ \alpha_1, \ldots, \alpha_N \in \mathbb{R}\}$. Since $\mathcal{H}_q$ is the linear span of posterior probability distributions, it is closed under addition and scalar multiplication and therefore a vector space. Furthermore, we define an inner product in $\mathcal{H}_q$ as the ordinary dot-product in $\mathbb{R}^{q_2}$, $\langle f, f'\rangle_{\mathcal{H}_q} = f^T f'$. □

Lemma 1. *$\mathcal{H}_q$ with $\langle\cdot,\cdot\rangle_{\mathcal{H}_q}$ is a Hilbert space.*

Proof. $\mathcal{H}_q$ is equipped with the ordinary dot product, has finite dimension q_2 and therefore is isometric to $\mathbb{R}^{q_2}$. □

Lemma 2. *k_q is a kernel.*

Proof. Let $\Phi_q : \mathcal{X} \to \mathcal{H}_q$ be the mapping given by $X \to \Pi(q)$. It follows that $\langle\Phi_q(X^{(n)}), \Phi_q(X^{(m)})\rangle_{\mathcal{H}_q} = \langle\Pi(q)^{(n)}, \Pi(q)^{(m)}\rangle_{\mathcal{H}_q} = (\Pi(q)^{(n)})^T\Pi(q)^{(m)} = k_q(X^{(n)}, X^{(m)})$. □

Now, let $\mathcal{H}$ be the Hilbert space defined via direct sum, $\mathcal{H} = \bigoplus_{q\in\mathcal{Q}} \mathcal{H}_q$. $\mathcal{H}$ consists of the set of all ordered tuples $\mathbf{\Pi}^{(n)} = (\Pi^{(n)}(1), \Pi^{(n)}(2), \ldots, \Pi^{(n)}(|\mathcal{Q}|))$. An induced inner product on $\mathcal{H}$ is $\langle\mathbf{\Pi}^{(n)}, \mathbf{\Pi}^{(m)}\rangle_{\mathcal{H}} = \sum_{q\in\mathcal{Q}}\langle\Pi^{(n)}(q), \Pi^{(m)}(q)\rangle_{\mathcal{H}_q}$. If we let $\Phi : \mathcal{X} \to \mathcal{H}$ be the mapping given by $X^{(n)} \to \mathbf{\Pi}^{(n)}$, it follows that $\langle\Phi(X^{(n)}), \Phi(X^{(m)})\rangle_{\mathcal{H}} = \langle\mathbf{\Pi}^{(n)}, \mathbf{\Pi}^{(m)}\rangle_{\mathcal{H}} = \sum_{q\in\mathcal{Q}} k_q(X^{(n)}, X^{(m)}) = K(X^{(n)}, X^{(m)})$.

This result and its proof unveil important properties of TCK. (i) K is symmetric and psd; (ii) the feature map Φ is provided explicitly; (iii) K is a linear kernel in the Hilbert space of posterior probability distributions $\mathcal{H}$; (iv) the induced distance d, given by

$$\begin{aligned} d^2(X^{(n)}, X^{(m)}) &= \langle\Phi(X^{(n)}) - \Phi(X^{(m)}), \Phi(X^{(m)}) - \Phi(X^{(m)})\rangle_{\mathcal{H}} \\ &= K(X^{(n)}, X^{(n)}) - 2\,K(X^{(n)}, X^{(m)}) + K(X^{(m)}, X^{(m)}) \end{aligned}$$

is a pseudo-metric as it satisfies the triangle inequality, takes non-negative values, but, in theory, it can vanish for $X^{(n)} \neq X^{(m)}$.

5. Experiments and results

The proposed kernel is very general and can be used as input in many learning algorithms. It is beyond the scope of this paper to illustrate all properties and possible applications for TCK. Therefore we restricted ourselves to classification, with and without missing data, dimensionality reduction and visualization. We applied the proposed method to one synthetic and several benchmark datasets. The TCK was compared to three other similarity measures, DTW, LPS and the fast global alignment kernel (GAK) [56]. DTW was extended to the multivariate case using both the *independent* (DTW i) and *dependent* (DTW d) version [69]. To evaluate the robustness of the similarity measures, they were trained unsupervisedly also in classification experiments, without tuning hyperparameters by cross-validation. In any case, cross-validation is not trivial in multivariate DTW, as the best window size based on individual attributes is not well defined [43].

For the classification task, to not introduce any additional, unnecessary parameters, we chose to use a nearest-neighbor (1NN) classifier. This is a standard choice in time series classification literature [89]. Even though the proposed method provides a kernel, by doing so, it is easier to compare the different properties of the similarity measures directly to each other. Performance was measured in terms of *classification accuracy* on a test set.

To perform dimensionality reduction we applied kPCA using the two largest eigenvalues of the kernel matrices. The different kernels were visually assessed by plotting the resulting mappings with the class information color-coded.

Table 1
Clustering performance, measured in terms of CA and ARI, on simulated VAR(1) datasets for TCK and GMM.

	TCK	GMM	TCK$_{UTS}$	TCK$_{\rho=0}$
CA	0.990	0.910	0.775	0.800
ARI	0.961	0.671	0.299	0.357

The TCK was implemented in R and Matlab, and the code is made publicly available at [90]. In the experiments we used the same parameters on all datasets. We let $C = 40$ and $Q = 30$. For the rest of the parameters we used the default values discussed in Section 4.2. The only exception is for datasets with less than 100 MTS, in that case we let the maximal number of mixtures be $C = 10$. The hyperparameter dependency is discussed more thoroughly in the end of this section.

For the LPS we used the Matlab implementation provided by Baydogan [91]. We set the number of trees to 200 and number of segments to 5. Since many of the time series we considered were short, we set the minimal segment length to 15% of the length of MTS in the dataset. The remaining hyperparameters were set to default. For the DTW we used the *R* package *dtw* [92]. The GAK was run using the Matlab Mex implementation provided by Cuturi [93]. In accordance with [93] we set the bandwidth σ to two times the median distance of the MTS in the training set, scaled by the square root of the median length of the MTS. The triangular parameter was set to 0.2 times the median length.

In contrast to the TCK and LPS, the DTW and GAK do not naturally deal with missing data and therefore we imputed the overall mean for each attribute and time interval.

5.1. Synthetic example: vector autoregressive model

We first applied TCK in a controlled experiment, where we generated a synthetic MTS dataset with two classes from a first-order vector autoregressive model, VAR(1) [4], given by

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} + \begin{pmatrix} \rho_x & 0 \\ 0 & \rho_y \end{pmatrix}\begin{pmatrix} x_1(t-1) \\ x_2(t-1) \end{pmatrix} + \begin{pmatrix} \xi_1(t) \\ \xi_2(t) \end{pmatrix} \tag{4}$$

To make $x_1(t)$ and $x_2(t)$ correlated with $\mathrm{corr}(x_1(t), x_2(t)) = \rho$, we chose the noise term s.t., $\mathrm{corr}(\xi_1(t), \xi_2(t)) = \rho\,(1 - \rho_x\rho_y)\,[(1 - \rho_x^2)(1 - \rho_y^2)]^{-1}$. For the first class, we generated 100 two-variate MTS of length 50 for the training and 100 for the test, from the VAR(1)-model with parameters $\rho = \rho_x = \rho_y = 0.8$ and $\mathbb{E}[(x_1(t), x_2(t))^T] = (0.5, -0.5)^T$. Analogously, the MTS of the second class were generated using parameters $\rho = -0.8$, $\rho_x = \rho_y = 0.6$ and $\mathbb{E}[(x_1(t), x_2(t))^T] = (0, 0)^T$. On these synthetic data, in addition to dimensionality reduction and classification with and without missing data, we also performed spectral clustering on the TCK matrix in order to be able to compare TCK directly to a single diagonal covariance GMM optimized using MAP-EM.

5.1.1. Clustering

Clustering performance was measured in terms of *adjusted rand index* (ARI) [94] and *clustering accuracy* (CA). CA is the maximum bipartite matching (*map*) between cluster labels (l_i) and ground-truth labels (y_i), defined as $\mathrm{CA} = N^{-1}\sum_{i=1}^N \delta(y_i, \mathrm{map}(l_i))$, where $\delta(\cdot,\cdot)$ is the Kronecker delta and $\mathrm{map}(\cdot)$ is computed with the Hungarian algorithm [95].

The single GMM was run with $a_0 = 0.1$, $b_0 = 0.1$ and $N_0 = 0.01$. Table 1 show that spectral clustering on the TCK achieves a considerable improvement compared to GMM clustering and verify the efficacy of the ensemble and the kernel approach with respect

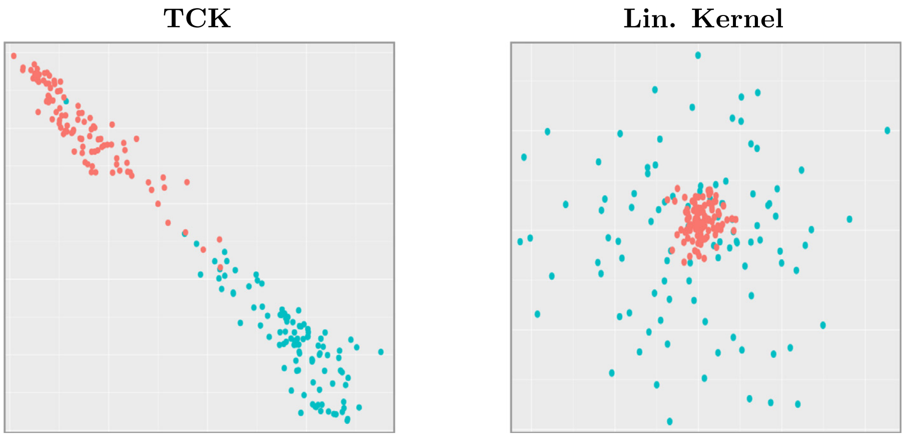


Fig. 2. Projection of the VAR(1) dataset to two dimensions using kPCA with the TCK and a linear kernel. The different colors indicate the true labels of the MTS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

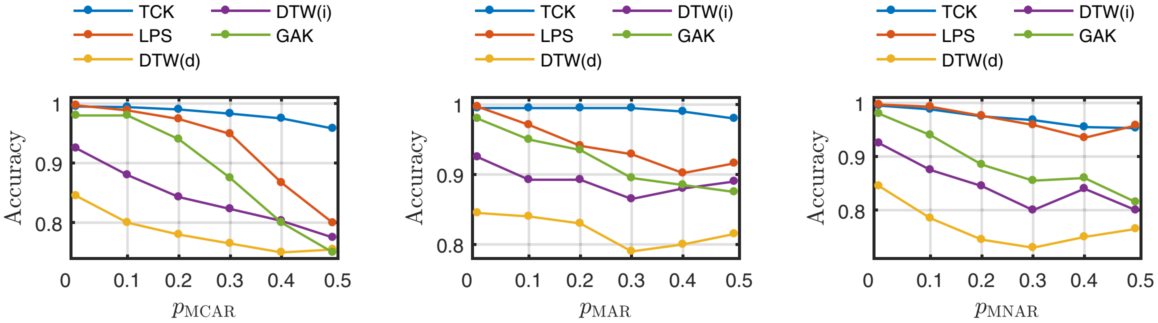


Fig. 3. Classification accuracy on simulated VAR(1) dataset of the 1NN-classifier configured with a (dis)similarity matrix obtained using LPS, DTW (d), DTW (i), GAK and TCK. We report results for three different types of missingness, with an increasing percentage of missing values.

Table 2
Description of benchmark time series datasets. Columns 2–5 show the number of attributes, samples in training and test set, and classes, respectively. $T_{\min}$ is the length of the shortest MTS in the dataset and $T_{\max}$ the longest MTS. T is the length of the MTS after the transformation.

Datasets	Attributes	Train	Test	Classes	$T_{\min}$	$T_{\max}$	T	Source
ItalyPower	1	67	1029	2	24	24	24	UCR
Gun Point	1	50	150	2	150	150	150	UCR
Synthetic control	1	300	300	6	60	60	60	UCR
PenDigits	2	300	10,692	10	8	8	8	UCI
Libras	2	180	180	15	45	45	23	UCI
ECG	2	100	100	2	39	152	22	Olszewski
uWave	3	200	4278	8	315	315	25	UCR
Char. Traj.	3	300	2558	20	109	205	23	UCI
Robot failure LP1	6	38	50	4	15	15	15	UCI
Robot failure LP2	6	17	30	5	15	15	15	UCI
Robot failure LP3	6	17	30	4	15	15	15	UCI
Robot failure LP4	6	42	75	3	15	15	15	UCI
Robot failure LP5	6	64	100	5	15	15	15	UCI
Wafer	6	298	896	2	104	198	25	Olszewski
Japanese vowels	12	270	370	9	7	29	15	UCI
ArabicDigits	13	6600	2200	10	4	93	24	UCI
CMU	62	29	29	2	127	580	25	CMU
PEMS	963	267	173	7	144	144	25	UCI

to a single GMM. Additionally, we evaluated TCK by concatenating the MTS as a long vector and thereby treating the MTS as an UTS (TCK_{UTS}) and on a different VAR(1) dataset with the attributes uncorrelated ($\mathrm{TCK}_{\rho=0}$). The superior performance of TCK with respect to these two approaches illustrates that, in addition to accounting for similarities within the same attribute, TCK also leverages interaction effects between different attributes in the MTS to improve clustering results.

5.1.2. Dimensionality reduction and visualization

To evaluate the effectiveness of TCK as a kernel, we compared kPCA with TCK and kPCA with a linear kernel (ordinary PCA). Fig. 2 shows that TCK maps the MTS on a line, where the two classes are well separated. On the other hand, PCA projects one class into a compact blob in the middle, whereas the other class is spread out. Learned representations like these can be exploited by learning algorithms such as an SVM. In this case, a linear classifier will perform well on the TCK representation, whereas for the other representation a non-linear method is required.

5.1.3. Classification with missing data

To investigate the TCK capability of dealing with missing data in a classification task, we removed values from the synthetic dataset according to three missingness patterns: *missing completely at ran-*

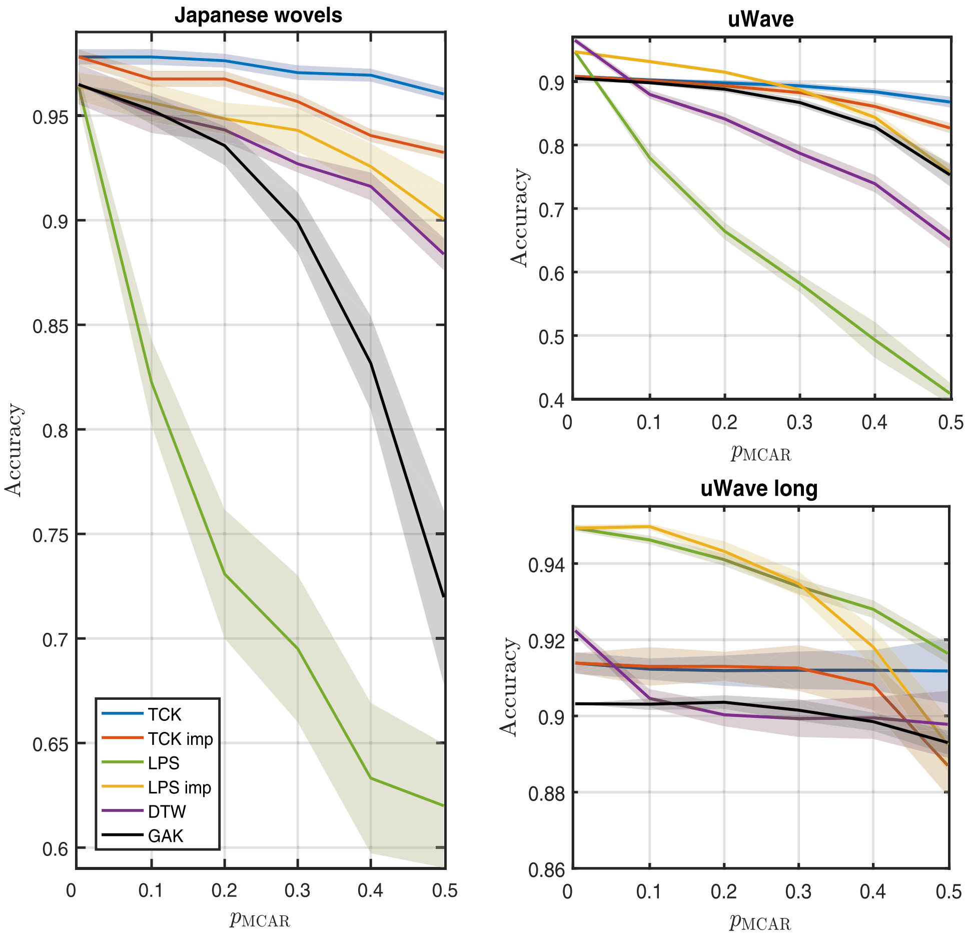


Fig. 4. Classification accuracies with different proportions of MCAR data for *Japanese vowels* and *uWave*. *uWave long* represents the uWave dataset where the MTS have their original length ($T = 315$). Shaded areas represent standard deviations calculated over 10 independent runs.

Table 3
Classification accuracy on different UTS and MTS benchmark datasets obtained using TCK, LPS, DTW (i), DTW (d) and GAK in combination with a 1NN-classifier. The best results are highlighted in bold.

Datasets	TCK	LPS	DTW (i)	DTW (d)	GAK
ItalyPower	0.922	0.933	0.918	0.918	**0.950**
Gun Point	0.923	0.790	**1.000**	**1.000**	0.900
Synthetic control	**0.987**	0.975	0.937	0.937	0.870
Pen digits	0.904	0.928	0.883	0.900	**0.945**
Libras	0.799	**0.894**	0.878	0.856	0.811
ECG	**0.852**	0.815	0.810	0.790	0.840
uWave	0.908	**0.945**	0.909	0.844	0.905
Char. Traj.	0.953	**0.961**	0.903	0.905	0.935
Robot failure LP1	**0.890**	0.836	0.720	0.640	0.720
Robot failure LP2	0.533	**0.707**	0.633	0.533	0.667
Robot failure LP3	**0.703**	0.687	0.667	0.633	0.633
Robot failure LP4	0.848	**0.914**	0.880	0.840	0.813
Robot failure LP5	0.596	**0.688**	0.480	0.430	0.600
Wafer	**0.982**	0.981	0.963	0.961	0.967
Japanese vowels	**0.978**	0.964	0.965	0.865	0.965
ArabicDigits	0.945	**0.977**	0.962	0.965	0.966
CMU	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
PEMS	**0.878**	0.798	0.775	0.763	0.763

dom (MCAR), *missing at random* (MAR) and *missing not at random* (MNAR) [74]. To simulate MCAR, we uniformly sampled the elements to be removed. Specifically, we discarded a ratio p_{MCAR} of the values in the dataset, varying from 0 to 0.5. To simulate MAR, we let $x_i(t)$ have a probability p_{MAR} of being missing, given that $x_j(t) > 0.5$, $i \neq j$. Similarly, for MNAR we let $x_i(t)$ have a probability p_{MNAR} of being missing, given that $x_i(t) > 0.5$. We varied the probabilities from 0 to 0.5 to obtain different fractions of missing data.

For each missingness pattern, we evaluated the performance of a 1NN classifier configured with TCK, LPS, DTW (d), DTW (i) and GAK. Classification accuracies are reported in Fig. 3. First of all, we see that in absence of missing data, the performance of TCK and LPS are approximately equal, whereas the two versions of DTW and GAK yield a lower accuracy. Then, we notice that the accuracy for the TCK is quite stable as the amount of missing data increases, for all types of missingness patterns. For example, in the case of MCAR, when the amount of missing data increases from 0 to 50%, accuracy decreases to from 0.995 to 0.958. Likewise, when p_{MNAR} increases from 0 to 0.5, accuracy decreases from 0.995 to 0.953. This indicates that our method, in some cases, also works well for data that are MNAR. On the other hand, we notice that for MCAR and MAR data, the accuracy obtained with LPS decreases much faster than for TCK. GAK seems to be sensitive to all three types of missing data. Performance also diminishes quite fast in the DTW variants, but we also observe a peculiar behavior as the accuracy starts to increase again when the missing ratio increases. This can be interpreted as a side effect of the imputation procedure implemented in DTW. In fact, the latter replaces some noisy data with a mean value, hence providing a regularization bias that benefits the classification procedure.

5.2. Benchmark time series datasets

We applied the proposed method to multivariate benchmark datasets from the UCR and UCI databases [96,97] and other published work [98,99], described in Table 2. In order to also illustrate TCK's capability of dealing with UTS, we randomly picked

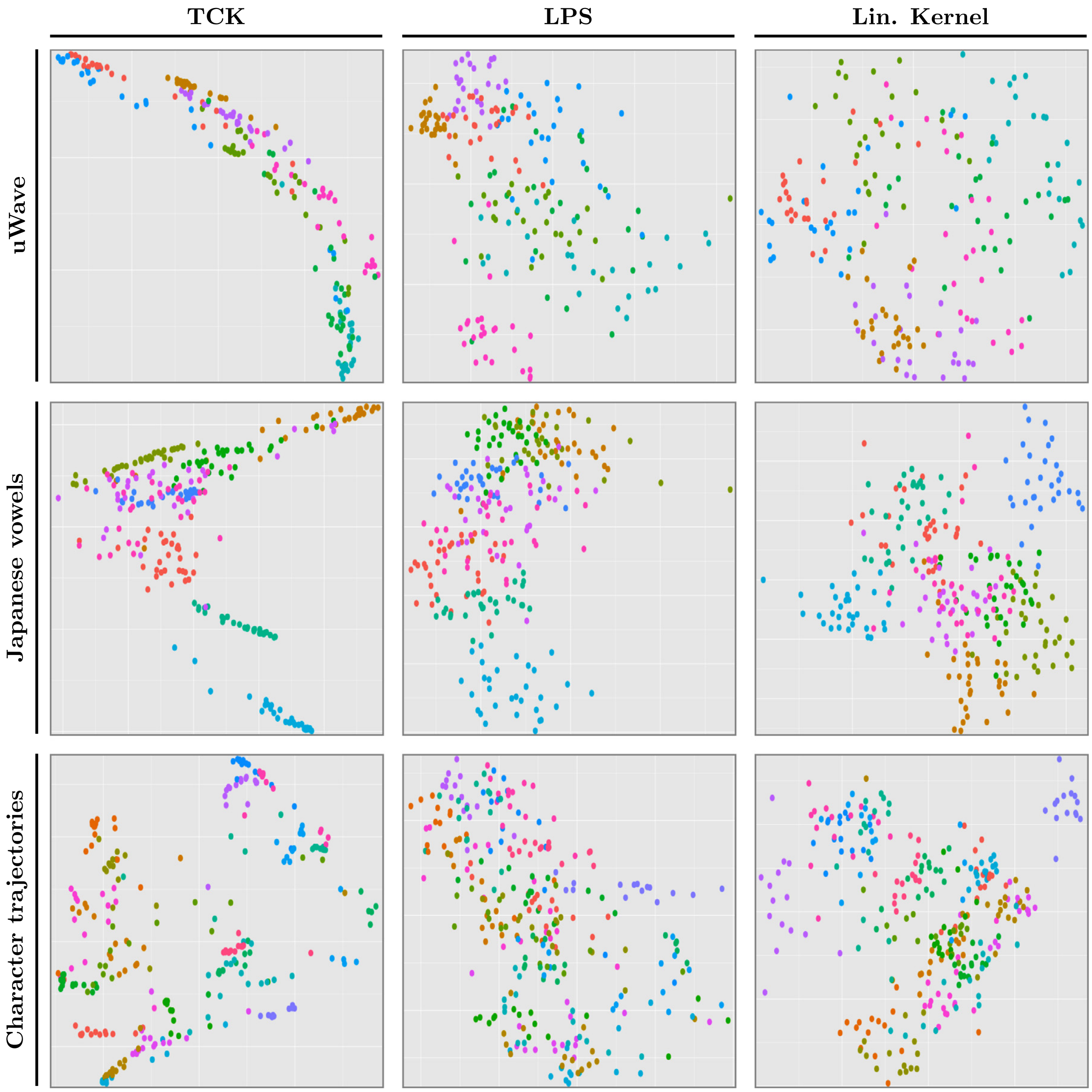


Fig. 5. Projection of three MTS datasets onto the two top principal components when different kernels are applied. The different colors indicate true class labels. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

three univariate datasets from the UCR database; *ItalyPower, Gun Point* and *Synthetic control*. Some of the multivariate datasets contain time series of different length. However, the proposed method is designed for MTS of the same length. Therefore we followed the approach of Wang et al. [100] and transformed all the MTS in the same dataset to the same length, T, determined by $T = \lceil \frac{T_{\max}}{\lceil \frac{T_{\max}}{25} \rceil} \rceil$, where $T_{\max}$ is the length of the longest MTS in the dataset and $\lceil\,\rceil$ is the ceiling operator. We also standardized to zero mean and unit standard deviation. Since decision trees are scale invariant, we did not apply this transformation for LPS (in accordance with [43]).

5.2.1. Classification without missing data

Initially we considered the case of no missing data and applied a 1NN-classifier in combination with the five different (dis)similarity measures. Table 3 shows the mean classification accuracies, evaluated over 10 runs, obtained on the benchmark time series datasets. First, we notice that the dependent version of DTW, in general, gives worse results than the independent version. Second, TCK gives the best accuracy for 8 out of 18 datasets. LPS and GAK are better than the competitors for 8 and 3 datasets, respectively. The two versions of DTW achieve the highest accuracy for Gun Point. On CMU all methods reach a perfect score. We also see that TCK works well for univariate data and gives comparable accuracies to the other methods.

5.2.2. Classification with missing data

We used the *Japanese vowels* and *uWave* datasets to illustrate the TCKs ability to classify real-world MTS with missing data. We removed different fractions of the values completely at random (MCAR) and ran a 1NN-classifier equipped with TCK, LPS, DTW (i) and GAK. We also compared to TCK and LPS with imputation of the mean. Mean classification accuracies and standard deviations, evaluated over 10 runs, are reported in Fig. 4.

On the Japanese vowels dataset the accuracy obtained with LPS decreases very fast as the fraction of missing data increases and is greatly outperformed by LPS imp. The performance of GAK also diminishes quickly. The accuracy obtained with DTW (i) decreases from 0.965 to 0.884, whereas TCK imp decreases from 0.978 to 0.932. The most stable results are obtained using TCK: as the ratio of missing data increases from 0 to 0.5, the accuracy decreases

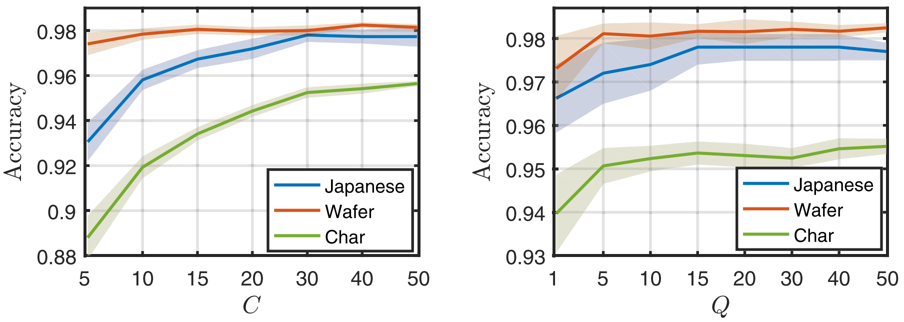


Fig. 6. Accuracies for (left) $Q = 30$ and varying C, and (right) $C = 40$ and varying Q, over three datasets. Shaded areas represent standard deviations calculated over 10 replications.

from 0.978 to 0.960. We notice that, even if TCK imp yields the second best results, it is clearly outperformed by TCK.

Also for the uWave dataset the accuracy decreases rapidly for LPS, DTW and GAK. The accuracy for TCK is 0.908 for no missing data, is almost stable up to 30% missing data and decreases to 0.868 for 50% missing data. TCK imp is outperformed by TCK, especially beyond 20% missingness. We notice that LPS imp gives better results than LPS also for this dataset. For ratios of missing data above 0.2 TCK gives better results than LPS imp, even though in absence of missingness the accuracy for LPS is 0.946, whereas TCK yields 0.908 only.

To investigate how TCK works for longer MTS, we classified the uWave dataset with MTS of original length, 315. In this case the LPS performs better than for the shorter MTS, as the accuracy decreases from 0.949 to 0.916. We also see that the accuracy decreases faster for LPS imp. For the TCK the accuracy increased from 0.908, obtained on uWave with MTS of length 25, to 0.914 on this dataset. TCK still gives a lower accuracy than LPS when there is no missing data. However, we see that TCK is very robust to missing data, since the accuracy only decreases to 0.912 when the missing ratio increases to 0.5. TCK imp performs equally well up to 30% missing data, but performs poorly for higher missing ratios.

These results indicate that, in contrast to LPS, TCK is not sensitive to the length of the MTS. It can deal equally well with short MTS and long MTS.

5.2.3. Dimensionality reduction and visualization

In Fig. 5 we have plotted the two principal components of *uWave, Japanese vowels* and *Character trajectory*, obtained with kPCA configured with TCK, LPS and a linear kernel. We notice a tendency in LPS and linear kernel to produce blob-structures, whereas the TCK creates more compact and separated embeddings. For example, for Japanese vowels TCK is able to isolate two classes from the rest.

5.3. Sensitivity analysis

The hyperparameters in the TCK are: maximum number of mixtures C, number of randomizations Q, segment length, subsample size η, number of attributes, hyperparameters Ω and initialization of GMM parameters Θ. However, all of them except C and Q, are chosen randomly for each $q \in \mathcal{Q}$. Hence, the only hyperparameters that have to be set by the user are C and Q.

We have already argued that the method is robust and not sensitive to the choice of these hyperparameters. Here, we evaluate empirically TCK's dependency on the chosen maximum number of mixture components C and of randomizations Q, on the three datasets *Japanese vowels, Wafer* and *Character trajectories*. Fig. 6 (left) shows the classification accuracies obtained using TCK in combination with a 1NN-classifier on the three datasets by fixing $Q = 30$ and varying C from 5 to 50. We see that the accuracies are very stable for C larger than 15–20. Even for $C = 10$, the accuracies are not much lower. Next, we fixed $C = 40$ and varied Q from 5 to 50. Fig. 6 (right) shows that the accuracies increase rapidly from $Q = 1$, but also that the it stabilizes quite quickly. It appears sufficient to choose $Q > 10$, even if the standard errors are a bit higher for lower Q. These results indicate that it is not critical to tune the hyperparameters C and Q correctly, which is important if the TCK should be learned in an unsupervised way.

Table 4
Running times (s) for computing the similarity between the test and training set for two datasets. The time in brackets represents time used to train the models for the methods that need training. For the PEMS dataset we used the original 963 attributes, but also ran the models on subsets consisting of 100, 10 and 2 attributes, respectively. For the uWave dataset we varied the length from $T = 315$ to $T = 25$.

PEMS	$V = 963$	$V = 100$	$V = 10$	$V = 2$
TCK	3.6 (116)	3.5 (115)	2.5 (84)	1.2 (31)
LPS	22 (269)	3.3 (33)	1.3 (4.5)	0.9 (2.9)
GAK	514	52	5.8	1.6
DTW (i)	1031	119	13	3.5
uWave	$T = 315$	$T = 200$	$T = 100$	$T = 25$
TCK	42 (46)	39 (45)	41 (46)	27 (35)
LPS	26 (17)	17 (11)	11 (7)	6.6 (2.5)
GAK	28	25	21	20
DTW (i)	506	244	110	59

5.4. Computational time

All experiments were run using an Ubuntu 14.04 64-bit system with 64 GB RAM and an Intel Xeon E5-2630 v3 processor. We used the low-dimensional *uWave* and the high-dimensional *PEMS* dataset to empirically test the running time of the TCK. To investigate how the running time is affected by the length and number of variables of the MTS, for the PEMS dataset we selected $V = \{963, 100, 10, 2\}$ attributes, while for the uWave dataset we let $T = \{315, 200, 100, 25\}$. Table 4 shows the running times (s) for TCK, LPS, GAK and DTW (i) on these datasets. We observe that the TCK is competitive to the other methods and, in particular, that its running time is not that sensitive to increased length or number of attributes.

6. Conclusions

We have proposed a novel similarity measure and kernel for multivariate time series with missing data. The robust time series cluster kernel was designed by applying an ensemble strategy to probabilistic models. TCK can be used as input in many different learning algorithms, in particular in kernel methods.

The experimental results demonstrated that the TCK (1) is robust to hyperparameter settings, (2) is competitive to established methods on prediction tasks without missing data and (3) is better than established methods on prediction tasks with missing data.

In future works we plan to investigate whether the use of more general covariance structures in the GMM, or the use of HMMs as base probabilistic models, could improve TCK.

Conflict of interest

The authors have no conflict of interest related to this work.

Acknowledgments

This work (Robert Jenssen and Filippo Bianchi) is partially supported by the Research Council of Norway over FRIPRO grant no. 234498 on developing the *Next Generation Learning Machines*. Cristina Soguero-Ruiz is supported by Research Project TEC2016-75361-R from the Spanish Government.

The authors would like to thank Sigurd Løkse and Michael Kampffmeyer for useful discussions, and Jonas Nordhaug Myhre for proof reading the article.

Appendix A

Theorem 2. *LPS is a kernel.*

Proof. The LPS similarity between two time series $X^{(n)}$ and $X^{(m)}$ is computed from the LPS representation, given by the frequency vectors $H(X^{(n)})$ and $H(X^{(m)})$, where $H(X^{(n)}) = [h_{1,1}^{(n)}, \dots, h_{R,J}^{(n)}] \in \mathbb{N}_0^{RJ}$ being $h_{r,j}^{(n)} \in \mathbb{N}_0$ the number of segments of $X^{(n)}$ contained in the leaf r of tree j and J the number of trees [43]. Let $N_s = T - L - 1$ be the total number of segments of length L in the MTS X of length T. Without loss of generality we assume that N_s and R, the total number of leaves, are constant in all trees. The LPS similarity reads

$$S\big(X^{(n)}, X^{(m)}\big) = \frac{1}{RJ} \sum_{r=1}^{R} \sum_{j=1}^{J} \min\left(h_{r,j}^{(n)}, h_{r,j}^{(m)}\right) \in [0,1]. \qquad \text{(A.1)}$$

We notice that, if we ignore the normalizing factor, Eq. (A.1) is the computation of the intersection between $H(X^{(n)})$ and $H(X^{(m)})$. In order to complete the proof, we now introduce an equivalent binary representation of the frequency vectors in the leaves. We represent the leaf r of the tree j as a binary sequence, with $h_{r,\,j}$ 1s in front and 0s $N_s - h_{r,j}$ in the remaining positions

$$\bar{H}(X) = \left[\underbrace{\overbrace{1,\dots,1}^{h_{1,1}},\overbrace{0,\dots,0}^{N_s-h_{1,1}}}_{\text{leaf}\,(1,1)},\dots,\underbrace{\overbrace{1,\dots,1}^{h_{r,j}},\overbrace{0,\dots,0}^{N_s-h_{r,j}}}_{\text{leaf}\,(r,j)},\dots,\underbrace{\overbrace{1,\dots,1}^{h_{R,J}},\overbrace{0,\dots,0}^{N_s-h_{R,J}}}_{\text{leaf}\,(R,J)} \right] \in \{0,1\}^{N_sRJ}.$$

The intersection between $H(X^{(n)})$ and $H(X^{(m)})$, yielded by Eq. (A.1), can be expressed as a bitwise operation through dot product

$$\big(H(X^{(n)}) \wedge H(X^{(m)})\big) = \bar{H}(X^{(n)})^T \bar{H}(X^{(m)}), \qquad \text{(A.2)}$$

which is a linear kernel in the linear span of the LPS representations, which is isometric to $\mathbb{R}^{N_sRJ}$. □

References

[1] W. Vandaele, Applied Time Series and Box-Jenkins Models, 1983.
[2] C. Chatfield, The Analysis of Time Series: An Introduction, CRC Press, 2016.
[3] J.D. Cryer, N. Kellet, Time Series Analysis, vol. 101, Springer, 1986.
[4] R.H. Shumway, D.S. Stoffer, Time Series Analysis and Its Applications: With R Examples, Springer Science & Business Media, 2010.
[5] F. Iglesias, W. Kastner, Analysis of similarity measures in times series clustering for the discovery of building energy patterns, Energies 6 (2) (2013) 579–597.
[6] M. Das, S.K. Ghosh, Data-driven approaches for meteorological time series prediction: a comparative study of the state-of-the-art computational intelligence techniques, Pattern Recognit. Lett. (2017), doi:10.1016/j.patrec.2017.08.009.
[7] M. Ji, F. Xie, Y. Ping, A dynamic fuzzy cluster algorithm for time series, Abstr. Appl. Anal. 2013 (2013) 7 pages.
[8] M. Pyatnitskiy, I. Mazo, M. Shkrob, E. Schwartz, E. Kotelnikova, Clustering gene expression regulators: new approach to disease subtyping, PLoS One 9 (1) (2014) 1–10.
[9] K. Häyrinen, K. Saranto, P. Nykänen, Definition, structure, content, use and impacts of electronic health records: a review of the research literature, Int. J. Med. Inf. 77 (5) (2008) 291–304.
[10] C. Soguero-Ruiz, W.M. Fei, R. Jenssen, K.M. Augestad, J.-L. Rojo-Álvarez, I. Mora-Jiménez, R.-O. Lindsetmo, S.O. Skrøvseth, Data-driven temporal prediction of surgical site infection, in: AMIA Annual Symposium Proceedings, vol. 2015, American Medical Informatics Association, 2015, pp. 1164–1173.
[11] C. Soguero-Ruiz, K. Hindberg, I. Mora-Jiménez, J.L. Rojo-Álvarez, S.O. Skrøvseth, F. Godtliebsen, K. Mortensen, A. Revhaug, R.-O. Lindsetmo, K.M. Augestad, R. Jenssen, Predicting colorectal surgical complications using heterogeneous clinical data and kernel methods, J. Biomed. Inf. 61 (2016) 87–96.
[12] A. Gharehbaghi, P. Ask, A. Babic, A pattern recognition framework for detecting dynamic changes on cyclic time series, Pattern Recognit. 48 (3) (2015) 696–708.
[13] Y.-C. Hsu, A.-P. Chen, A clustering time series model for the optimal hedge ratio decision making, Neurocomputing 138 (2014) 358–370.
[14] R.S. Tsay, Multivariate Time Series Analysis: With R and Financial Applications, John Wiley & Sons, 2013.
[15] O. Anava, E. Hazan, A. Zeevi, Online time series prediction with missing data., in: ICML, 2015, pp. 2191–2199.
[16] F. Bashir, H.L. Wei, Handling missing data in multivariate time series using a vector autoregressive model based imputation (var-im) algorithm: part i: var-im algorithm versus traditional methods, in: 24th Mediterranean Conference on Control and Automation, 2016, pp. 611–616.
[17] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2001.
[18] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[19] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E. Keogh, Experimental comparison of representation methods and distance measures for time series data, Data Min. Knowl. Discovery 26 (2) (2013) 275–309.
[20] S. Aghabozorgi, A.S. Shirkhorshidi, T.Y. Wah, Time-series clustering–a decade review, Inf. Syst. 53 (C) (2015) 16–38.
[21] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast subsequence matching in time-series databases, in: Proceedings of the 1994 ACM SIGMOD International Conference on Management of data, ACM, 1994, pp. 419–429.
[22] K.-P. Chan, A.W.-C. Fu, Efficient time series matching by wavelets, in: Proceedings 15th International Conference on Data Engineering, IEEE, 1999, pp. 126–133.
[23] F. Korn, H.V. Jagadish, C. Faloutsos, Efficiently supporting ad hoc queries in large datasets of time sequences, in: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, ACM, 1997, pp. 289–300.
[24] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Min. Knowl. Discovery 15 (2) (2007) 107–144.
[25] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowl. Inf. Syst. 3 (3) (2001) 263–286.
[26] B.M. Marlin, D.C. Kale, R.G. Khemani, R.C. Wetzel, Unsupervised pattern discovery in electronic health care data using probabilistic clustering models, in: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, ACM, 2012, pp. 389–398.
[27] F. Bashir, A. Khokhar, D. Schonfeld, Automatic object trajectory-based motion recognition using Gaussian mixture models, in: IEEE International Conference on Multimedia and Expo, IEEE, 2005, pp. 1532–1535.
[28] F.I. Bashir, A.A. Khokhar, D. Schonfeld, Object trajectory-based activity classification and recognition using hidden Markov models, IEEE Trans. Image Process. 16 (7) (2007) 1912–1919.
[29] M. Ramoni, P. Sebastiani, P. Cohen, Bayesian clustering by dynamics, Mach. Learn. 47 (1) (2002) 91–121.
[30] A. Panuccio, M. Bicego, V. Murino, A Hidden Markov Model-based approach to sequential data clustering, in: Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Springer, 2002, pp. 734–743.
[31] B. Knab, A. Schliep, B. Steckemetz, B. Wichern, Model-based clustering with hidden Markov models and its application to financial time-series data, in: Between Data Science and Applied Data Analysis, Springer, 2003, pp. 561–569.
[32] N. Kumar, V.N. Lolla, E. Keogh, S. Lonardi, C.A. Ratanamahatana, L. Wei, Time-series bitmaps: a practical visualization tool for working with large time series databases, in: Proceedings of the Fifth SIAM International Conference on Data Mining, SIAM, 2005, pp. 531–535.
[33] M. Corduas, D. Piccolo, Time series clustering and classification by the autoregressive metric, Comput. Stat. Data Anal. 52 (4) (2008) 1860–1872.
[34] Y. Xiong, D.-Y. Yeung, Mixtures of arma models for model-based time series clustering, in: IEEE International Conference on Data Mining, IEEE, 2002, pp. 717–720.
[35] K.S. Tuncel, M.G. Baydogan, Autoregressive forests for multivariate time series modeling, Pattern Recognit. 73 (2018) 202–215.
[36] T.-C. Fu, A review on time series data mining, Eng. Appl. Artif. Intell. 24 (1) (2011) 164–181.

[37] J. Han, J. Pei, M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2011.

[38] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1994, pp. 359–370.

[39] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, E. Keogh, Indexing multi-dimensional time-series with support for multiple distance measures, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 216–225.

[40] K. Yang, C. Shahabi, An efficient k nearest neighbor search for multivariate time series, Inf. Comput. 205 (1) (2007) 65–98.

[41] L. Chen, M.T. Özsu, V. Oria, Robust and fast similarity search for moving object trajectories, in: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, ACM, 2005, pp. 491–502.

[42] Z. Bankó, J. Abonyi, Correlation based dynamic time warping of multivariate time series, Expert Syst. Appl. 39 (17) (2012) 12814–12823.

[43] M.G. Baydogan, G. Runger, Time series representation and similarity based on local autopatterns, Data Min. Knowl. Discovery 30 (2) (2016) 476–509.

[44] R. Jenssen, Kernel entropy component analysis, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 847–860.

[45] R. Jenssen, Entropy-relevant dimensions in the kernel feature space: cluster-capturing dimensionality reduction, IEEE Signal Process. Mag. 30 (4) (2013) 30–39.

[46] B. Schölkopf, K. Tsuda, J.-P. Vert, Kernel Methods in Computational Biology, MIT Press, 2004.

[47] G. Camps-Valls, L. Bruzzone, Kernel Methods for Remote Sensing Data Analysis, John Wiley & Sons, 2009.

[48] C. Soguero-Ruiz, K. Hindberg, J.L. Rojo-Álvarez, S.O. Skrøvseth, F. Godtliebsen, K. Mortensen, A. Revhaug, R.O. Lindsetmo, K.M. Augestad, R. Jenssen, Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records, IEEE J. Biomed. Health Inf. 20 (5) (2016) 1404–1415.

[49] B. Boecking, S.K. Chalup, D. Seese, A.S. Wong, Support vector clustering of time series data with alignment kernels, Pattern Recognit. Lett. 45 (Suppl C) (2014) 129–135, doi:10.1016/j.patrec.2014.03.015.

[50] B. Schölkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, in: International Conference on Computational Learning Theory, Springer, 2001, pp. 416–426.

[51] A. Berlinet, C. Thomas-Agnan, Reproducing Kernel Hilbert Spaces in Probability and Statistics, Springer Science & Business Media, 2011.

[52] I. Steinwart, A. Christmann, Support Vector Machines, Springer Science & Business Media, 2008.

[53] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, in: International Conference on Artificial Neural Networks, Springer, 1997, pp. 583–588.

[54] B. Haasdonk, C. Bahlmann, Learning with distance substitution kernels, in: Joint Pattern Recognition Symposium, Springer, 2004, pp. 220–227.

[55] P.-F. Marteau, S. Gibet, On recursive edit distance kernels with application to time series classification, IEEE Trans. Neural Netw. Learn. Syst. 26 (6) (2015) 1121–1133.

[56] M. Cuturi, Fast global alignment kernels, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 929–936.

[57] T. Jebara, R. Kondor, A. Howard, Probability product kernels, J. Mach. Learn. Res. 5 (2004) 819–844.

[58] T.S. Jaakkola, M. Diekhans, D. Haussler, Using the Fisher kernel method to detect remote protein homologies, in: Proceedings of the International Conference on Intelligent Systems for Molecular Biology, vol. 99, 1999, pp. 149–158.

[59] H. Chen, F. Tang, P. Tino, X. Yao, Model-based kernel for efficient time series analysis, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 392–400.

[60] Z. Liu, M. Hauskrecht, Learning adaptive forecasting models from irregularly sampled multivariate clinical data, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1273–1279.

[61] A.R.T. Donders, G.J. van der Heijden, T. Stijnen, K.G. Moons, Review: a gentle introduction to imputation of missing values, J. Clin. Epidemiol. 59 (10) (2006) 1087–1091.

[62] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, Berlin, Heidelberg, 2000, pp. 1–15.

[63] M. Cuturi, A. Doucet, Autoregressive kernels for time series, 2011 1101.0673.

[64] E. Izquierdo-Verdiguier, R. Jenssen, L. Gómez-Chova, G. Camps-Valls, Spectral clustering with the probabilistic cluster kernel, Neurocomputing 149 (C) (2015) 1299–1304.

[65] Q. Cai, L. Chen, J. Sun, Piecewise statistic approximation based similarity measure for time series, Knowl. Based Syst. 85 (2015) 181–195.

[66] C.A. Ratanamahatana, E. Keogh, Three myths about dynamic time warping data mining, in: Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, 2005, pp. 506–510.

[67] J. Lines, A. Bagnall, Time series classification with ensembles of elastic distance measures, Data Min. Knowl. Discovery 29 (3) (2015) 565–592.

[68] J. Zhao, L. Itti, ShapeDTW: shape dynamic time warping, Pattern Recognit. 74 (Suppl C) (2018) 171–184, doi:10.1016/j.patcog.2017.09.020.

[69] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, E. Keogh, Generalizing DTW to the multi-dimensional case requires an adaptive approach, Data Min. Knowl. Discovery 31 (1) (2017) 1–31.

[70] C. Berg, J.P. Christensen, P. Ressel, Harmonic analysis on semigroups: theory of positive definite and related functions, Graduate Texts in Mathematics, vol. 100, first ed., Springer, 1984.

[71] G. Wu, E.Y. Chang, Z. Zhang, Learning with non-metric proximity matrices, in: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM, 2005, pp. 411–414.

[72] Y. Chen, E.K. Garcia, M.R. Gupta, A. Rahimi, L. Cazzanti, Similarity-based classification: concepts and algorithms, J. Mach. Learn. Res. 10 (2009) 747–776.

[73] K. Tsuda, T. Kin, K. Asai, Marginalized kernels for biological sequences, Bioinformatics 18 (Suppl 1) (2002) S268–S275.

[74] D.B. Rubin, Inference and missing data, Biometrika 63 (3) (1976) 581–592.

[75] J.A. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models, Int. Comput. Sci. Inst. 4 (510) (1998) 126.

[76] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B (1977) 1–38.

[77] G. McLachlan, T. Krishnan, The EM Algorithm and Extensions, vol. 382, John Wiley & Sons, 2007.

[78] T.J. Hastie, R.J. Tibshirani, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics, Springer, 2009.

[79] C.J. Wu, On the convergence properties of the EM algorithm, Ann. Stat. 11 (1983) 95–103.

[80] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.

[81] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996), 1996, pp. 148–156.

[82] B.K. Iwana, V. Frinken, K. Riesen, S. Uchida, Efficient temporal pattern recognition by means of dissimilarity space embedding with discriminative prototypes, Pattern Recognit. 64 (Suppl C) (2017) 268–276, doi:10.1016/j.patcog.2016.11.013.

[83] A.L.N. Fred, A.K. Jain, Evidence accumulation clustering based on the k-means algorithm, in: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition, Springer, 2002, pp. 442–451.

[84] S. Monti, P. Tamayo, J. Mesirov, T. Golub, Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data, Mach. Learn. 52 (1–2) (2003) 91–118.

[85] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.

[86] S. Vega-Pons, J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, Int. J. Pattern Recognit. Artif. Intell. 25 (03) (2011) 337–372.

[87] M. Glodek, M. Schels, F. Schwenker, Ensemble Gaussian mixture models for probability density estimation, Comput. Stat. 28 (1) (2013) 127–138.

[88] A.L. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 835–850.

[89] R.J. Kate, Using dynamic time warping distances as features for improved time series classification, Data Min. Knowl. Discovery 30 (2) (2016) 283–312.

[90] K.Ø. Mikalsen, Time series cluster kernel (TCK) Matlab implementation, 2017, http://site.uit.no/ml.

[91] LPS Matlab implementation, http://www.mustafabaydogan.com/files/viewdownload/18-learned-pattern-similarity-lps/60-multivariate-lps-matlab-implementation.html. Accessed: 2017-03-07.

[92] T. Giorgino, Computing and visualizing dynamic time warping alignments in R: the dtw package, J. Stat. Softw. 031 (i07) (2009) 1–24.

[93] Fast global alignment kernel Matlab implementation, http://www.marcocuturi.net/GA.html. Accessed: 2017-06-20.

[94] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.

[95] H.W. Kuhn, B. Yaw, The Hungarian method for the assignment problem, Naval Res. Logist. Q. 2 (1955) 83–97.

[96] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The UCR time series classification archive, 2015, http://www.cs.ucr.edu/eamonn/time_series_data/. Accessed: 2016-12-17.

[97] M. Lichman, UCI machine learning repository, 2013, http://archive.ics.uci.edu/ml. Accessed: 2016-10-29.

[98] Carnegie Mellon University motion capture database, 2014, http://mocap.cs.cmu.edu. Accessed: 2017-1-13.

[99] R.T. Olszewski, Generalized feature extraction for structural pattern recognition in time-series data, 2001 Ph.D. thesis. Pittsburgh, PA, USA

[100] L. Wang, Z. Wang, S. Liu, An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm, Expert Syst. Appl. 43 (2016) 237–249.

Karl Øyvind Mikalsen received the M.Sc. degree in applied mathematics in 2014 at UiT – The Arctic University of Norway, Tromsø, Norway, where he currently is working towards a Ph.D. degree in machine learning. His research interests include machine learning for healthcare, time series analysis, kernel methods, clustering and semi-supervised methods.

Filippo Maria Bianchi received the B.Sc. in computer engineering (2009), the M.Sc. in artificial intelligence and robotics (2012) and Ph.D. in machine learning (2016) from Sapienza University. He worked 2 years as research assistant at Ryerson University. He is currently a postdoc at UiT. Research interests include graph-matching, reservoir computing and deep-learning.

Cristina Soguero-Ruiz got the Ph.D. in 2015, winning the Orange Best Ph.D. Award at Rey Juan Carlos University, where she is an assistant professor, and is in addition an associate member in the Machine Learning Group at University of Tromsø. Her research interests include machine learning and healthcare analytics.

Robert Jenssen directs the UiT Machine Learning Group: http://site.uit.no/ml. The group is advancing research on deep learning and kernel machines, as well as healthcare analytics, remote sensing, and industrial applications. Jenssen is an associate editor of Pattern Recognition, an IEEE TC MLSP member, and on the IAPR Governing Board.

Chapter 10

Paper II

Noisy multi-label semi-supervised dimensionality reduction

Karl Øyvind Mikalsen[a,b,*], Cristina Soguero-Ruiz[b,c], Filippo Maria Bianchi[d,b], Robert Jenssen[d,b]

[a]*Dept. of Mathematics and Statistics, UiT The Arctic University of Norway, Tromsø, Norway*
[b]*UiT Machine Learning Group*
[c]*Dept. of Signal Theory and Comm., Telematics and Computing, Universidad Rey Juan Carlos, Fuenlabrada, Spain*
[d]*Dept. of Physics and Technology, UiT, Tromsø, Norway*

Abstract

Noisy labeled data represent a rich source of information that often are easily accessible and cheap to obtain, but label noise might also have many negative consequences if not accounted for. How to fully utilize noisy labels has been studied extensively within the framework of standard supervised machine learning over a period of several decades. However, very little research has been conducted on solving the challenge posed by noisy labels in non-standard settings. This includes situations where only a fraction of the samples are labeled (semi-supervised) and each high-dimensional sample is associated with multiple labels. In this work, we present a novel semi-supervised and multi-label dimensionality reduction method that effectively utilizes information from both noisy multi-labels and unlabeled data. With the proposed *Noisy multi-label semi-supervised dimensionality reduction (NMLSDR)* method, the noisy multi-labels are denoised and unlabeled data are labeled simultaneously via a specially designed label propagation algorithm. NMLSDR then learns a projection matrix for reducing the dimensionality by maximizing the dependence between the enlarged and denoised multi-label space and the features in the projected space. Extensive experiments on synthetic data, as well as benchmark datasets, demonstrate the effectiveness of the proposed algorithm and show that it outperforms state-of-the-art multi-label feature extraction algorithms. Finally, we illustrate the benefits of the proposed method in a realistic healthcare case study, achieving statistically significant gains compared to the previous state-of-the-art on the problem of identifying patients suffering from multiple chronic diseases.

Keywords: Noisy labels, Multi-label learning, Semi-supervised learning, Dimensionality reduction, Healthcare case study

1. Introduction

Supervised machine learning crucially relies on the accuracy of the *observed labels* associated with the training samples [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Observed labels may be corrupted and, therefore, they do not necessarily coincide with the true class of the samples. Such inaccurate labels are also referred to as *noisy* [2, 11, 4]. Label noise can occur because of imperfect evidence or fatigue on the part of the labeler, e.g. in healthcare where a medical doctor may be annotating or labeling thousands of patients manually, potentially making mistakes in the process [12, 5]. In other cases, noisy labels may result from the use of frameworks such as anchor learning [13, 14] or silver standard learning [15], which have received interest for instance in healthcare analytics [16, 17]. A review of various sources of label noise can be found in [2].

In standard supervised machine learning settings, the challenge posed by noisy labels has been studied extensively. For example, many noise-tolerant versions of well-known classifiers have been proposed, including discriminant analysis [8, 18], logistic regression [9], the k-nearest neighbor classifier [19], boosting algorithms [20, 21], perceptrons [22, 23], support vector machines [24], deep neural networks [7, 25, 26]. Others have proposed more general classification frameworks that are not restricted to particular classifiers [4, 11].

However, very little research has been conducted on solving the challenge posed by noisy labels in non-standard settings, where the magnitude of the noisy label problem is increased considerably. One good example (among many) of such a non-standard setting occurs for instance within the healthcare domain, used here as an illustrative case-study. Non-standard settings include (i) *Semi-supervised learning* [27], referring to a situation where only a few (noisy) labeled data points are available, making the impact of noise in those few labels more prevalent, and where information must also jointly be inferred from unlabeled data points. In healthcare, it may be realistic to obtain some labels through a (imperfect) manual labeling process, but the vast amount of data remains unlabeled; (ii) *Multi-label learning*, wherein objects may not belong exclusively to one category. This situation occurs frequently in a number of domains, including healthcare, where for instance a patient could suffer from multiple chronic diseases; (iii) High-dimensional data, where the abundance of features and the limited (noisy) labeled data, lead to a curse of dimensionality problem. In such situations, *dimensionality reduction* (DR) [28] is useful, either as a pre-processing step, or as an integral part of the learning procedure. This is a well-known challenge in health, where the

[*] Corresponding author at: Department of Mathematics and Statistics, Faculty of Science and Technology, UiT – The Arctic University of Norway, N-9037 Tromsø, Norway

number of patients in the populations under study frequently is small, but heterogeneous potential sources of data features from electronic health records for each patient may be enormous [29, 30, 31, 32].

In this paper, and to the best of our knowledge, we propose the first noisy label, semi-supervised and multi-label DR machine learning method, which we call the *Noisy multi-label semi-supervised dimensionality reduction (NMLSDR)* method. Towards that end, we propose a label propagation method that can deal with noisy multi-label data. Label propagation [33, 34, 35, 36, 37, 38, 39], wherein one propagates the labels to the unlabeled data in order to obtain a fully labeled dataset, is one of the most successful and fundamental frameworks within semi-supervised learning. However, in contrast to many of these methods that clamp the labeled data, in our multi-label propagation method we allow the labeled part of the data to change labels during the propagation to account for noisy labels. In the second part of our algorithm we aim at learning a lower dimensional representation of the data by maximizing the feature-label dependence. Towards that end, similarly to other DR methods [40, 41], we employ the Hilbert-Schmidt independence criterion (HSIC) [42], which is a non-parametric measure of dependence.

The NMLSDR method is a DR method, which is general and can be used in many different settings, e.g. for visualization or as a pre-processing step before doing classification. However, in order to test the quality of the NMLSDR embeddings, we (preferably) have to use some quantitative measures. For this purpose, a common baseline classifier such as the multi-label k-nearest neighbor (ML-kNN) classifier [43] has been applied to the low-dimensional representations of the data [44, 45]. Even though this is a valid way to measure the quality of the embeddings, to apply a supervised classifier in a semi-supervised learning setting is not a realistic setup since one suddenly assumes that all labels are known (and correct). Therefore, as an additional contribution, we introduce a novel framework for semi-supervised classification of noisy multi-label data.

In our experiments, we compare NMLSDR to baseline methods on synthetic data, benchmark datasets, as well as a real-world case study, where we use it to identify the health status of patients suffering from potentially multiple chronic diseases. The experiments demonstrate that for partially and noisy labeled multi-label data, NMLSDR is superior to existing DR methods according to seven different multi-label evaluation metrics and the Wilcoxon statistical test.

In summary, the contributions of the paper are as follows.

- A new semi-supervised multi-label dimensionality reduction method based on dependence maximization that is robust to noisy labels.
- A novel framework for semi-supervised classification of noisy multi-label data.
- A comprehensive experimental section that illustrate the effectiveness of the NMLSDR, and in particular, a real-world case study where the proposed framework is used to identify the health status of patients with multiple chronic diseases.

The remainder of the paper is organized as follows. Related work is reviewed in Sec. 2. In Sec. 3, we describe our proposed NMLSDR method and the novel framework for semi-supervised classification of noisy multi-label data. Sec. 4 describes experiments on synthetic and benchmark datasets, whereas Sec. 5 is devoted to the case study where we study chronically ill patients. We conclude the paper in Sec. 6.

2. Related work

In this section we review related unsupervised, semi-supervised and supervised DR methods.[1]

Unsupervised DR methods do not exploit label information and can therefore straightforwardly be applied to multi-label data by simply ignoring the labels. For example, principal component analysis (PCA) aims to find the projection such that the variance of the input space is maximally preserved [47]. Other methods aim to find a lower dimensional embedding that preserves the manifold structure of the data, and examples of these include Locally linear embedding [48], Laplacian eigenmaps [49] and ISOMAP [50].

One of the most well-known supervised DR methods is linear discriminative analysis (LDA) [51], which aims at finding the linear projection that maximizes the within-class similarity and at the same time minimizes the between-class similarity. LDA has been extended to multi-label LDA (MLDA) in several different ways [52, 53, 54, 55, 56]. The difference between these methods basically consists in the way the labels are weighted in the algorithm. Following the notation in [56], wMLDAb [52] uses binary weights, wMLDAe [53] uses entropy-based weights, wMLDAc [54] uses correlation-based weights, wMLDAf [55] uses fuzzy-based weights, whereas wMLDAd [56] uses dependence-based weights.

Canonical correlation analysis (CCA) [57] is a method that maximizes the linear correlation between two sets of variables, which in the case of DR are the set of labels and the set of features derived from the projected space. CCA can be directly applied also for multi-labels without any modifications. Multi-label informed latent semantic indexing (MLSI) [58] is a DR method that aims at both preserving the information of inputs and capturing the correlations between the labels. In the Multi-label least square (ML-LS) method one extracts a common subspace that is assumed to be shared among multiple labels by solving a generalized eigenvalue decomposition problem [59].

In [40], a supervised method for doing DR based on dependence maximization [42] called Multi-label dimensionality reduction via dependence maximization (MDDM) was introduced. MDDM attempts to maximize the feature-label dependence using the Hilbert-Schmidt independence criterion and was originally formulated in two different ways. MDDMp is

[1] DR may be obtained both by feature extraction, i.e. by a data transformation, and by feature selection [46]. Here, we refer to DR in the sense of feature extraction.

based on orthonormal projection directions, whereas MDDMf makes the projected features orthonormal. Yu et al. showed that MDDMp can be formulated using least squares and added a PCA term to the cost function in a new method called Multi-label feature extraction via maximizing feature variance and feature-label dependence simultaneously (MVMD) [41].

The most closely related existing DR methods to NMLSDR are the semi-supervised multi-label methods. The Semi-supervised dimension reduction for multi-label classification method (SSDR-MC) [60], Coupled dimensionality reduction and classification for supervised and semi-supervised multilabel learning [61], and Semisupervised multilabel learning with joint dimensionality reduction [62] are semi-supervised multi-label methods that simultaneously learn a classifier and a low dimensional embedding.

Other semi-supervised multi-label DR methods are semi-supervised formulations of the corresponding supervised multi-label DR method. Blascho et al. introduced semi-supervised CCA based on Laplacian regularization [63]. Several different semi-supervised formulations of MLDA have also been proposed. Multi-label dimensionality reduction based on semi-supervised discriminant analysis (MSDA) adds two regularization terms computed from an adjacency matrix and a similarity correlation matrix, respectively, to the MLDA objective function [64]. In the Semi-supervised multi-label dimensionality reduction (SSMLDR) [44] method one does label propagation to obtain soft labels for the unlabeled data. Thereafter the soft labels of all data are used to compute the MLDA scatter matrices. An other extension of MLDA is Semi-supervised multi-label linear discriminant analysis (SMLDA) [65], which later was modified and renamed Semi-supervised multi-label dimensionality reduction based on dependence maximization (SMDRdm) [45]. In SMDRdm the scatter matrices are computed based on only labeled data. However, a HSIC term is also added to the familiar Rayleigh quotient containing the two scatter matrices, which is computed based on soft labels for both labeled and unlabeled data obtained in a similar way as in SSMLDR.

Common to all these methods is that none of them explictly assume that the labels can be noisy. In SSMLDR and SMDRdm, the labeled data are clamped during the label propagation and hence cannot change. Moreover, these two methods are both based on LDA, which is known heavily affected by outliers, and consequently also wrongly labeled data [66, 67, 68].

3. The NMLSDR method

We start this section by introducing notation and the setting for noisy multi-label semi-supervised linear feature extraction, and thereafter elaborate on our proposed NMLSDR method.

3.1. Problem statement

Let $\{x_i\}_{i=1}^n$ be a set of n D-dimensional data points, $x_i \in \mathbb{R}^D$. Assume that the data are ordered such that the l first of the data points are labeled and u are unlabeled, $l + u = n$. Let X be a $n \times d$ matrix with the data points as row vectors.

Assume that the number of classes is C and let $Y_i^L \in \{0, 1\}^C$ be the label-vector of data point x_i, $i = 1, \ldots, l$. The elements are given by $Y_{ic}^L = 1$, $c = 1, \ldots, C$ if data point x_i belongs to the $c-$th class and $Y_{ic}^L = 0$ otherwise. Define the label matrix $Y^L \in \{0, 1\}^{l \times C}$ as the matrix with the known label-vectors Y_i^L, $i = 1, \ldots, l$ as row vectors and let $Y^U \in \{0, 1\}^{u \times C}$ be the corresponding label matrix of the unknown labels.

The objective of linear feature extraction is to learn a projection matrix $P \in \mathbb{R}^{D \times d}$ that maps a data point in the original feature space $x \in \mathbb{R}^D$ to a lower dimensional representation $z \in \mathbb{R}^d$,

$$z = P^T x, \tag{1}$$

where $d < D$ and P^T denotes the transpose of the matrix P.

In our setting, we assume that the label matrix Y^L is potentially noisy and that Y^U is unknown. The first part of our proposed NMLSDR method consists of doing label propagation in order to learn the labels Y^U and update the estimate of Y^L. We do this by introducing soft labels $F \in \mathbb{R}^{n \times C}$ for the label matrix $Y = \begin{pmatrix} Y^L \\ Y^U \end{pmatrix}$, where F_{ic} represents the probability that data point x_i belong to the $c-th$ class. We obtain F with label propagation and thereafter use F to learn the projection matrix P. However, we start by explaining our label propagation method.

3.2. Label propagation using a neighborhood graph

The underlying idea of label propagation is that similar data points should have similar labels. Typically, the labels are propagated using a neighborhood graph [33]. Here, inspired by [69], we formulate a label propagation method for multi-labels that is robust to noise. The method is as follows.

Step 1. First, a neighbourhood graph is constructed. The graph is described by its adjacency matrix W, which can be designed e.g. by setting the entries to

$$W_{ij} = \exp(-\sigma^{-2} \|x_i - x_j\|^2), \tag{2}$$

where $\|x_i - x_j\|$ is the Euclidean distance between the datapoints x_i and x_j, and σ is a hyperparameter. Alternatively, one can use the Euclidian distance to compute a k-nearest neighbors (kNN) graph where the entries of W are given by

$$W_{ij} = \begin{cases} 1, & \text{if } x_i \text{ among } x_j\text{'s } k\text{NN or } x_j \text{ among } x_i\text{'s } k\text{NN} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Step 2. Symmetrically normalize the adjacency matrix W by letting

$$\tilde{W} = D^{-1/2} W D^{-1/2}, \tag{4}$$

where D is a diagonal matrix with entries given by $d_{ii} = \sum_{k=1}^n W_{ik}$.

Step 3. Calculate the stochastic matrix

$$T = \tilde{D}^{-1} \tilde{W}, \tag{5}$$

where $\tilde{d}_{ii} = \sum_{k=1}^n \tilde{W}_{ik}$. The entry T_{ij} can now be considered as the probability of a transition from node i to node j along the edge between them.

Step 4. Compute soft labels $F \in \mathbb{R}^{n \times C}$ by iteratively using the following update rule

$$F(t + 1) = I_\alpha T F(t) + (I - I_\alpha) Y, \tag{6}$$

where I_α is a $n\times n$ diagonal matrix with the hyperparameters α_i, $0 \le \alpha_i < 1$, on the diagonal. To initialize F, we let $F(0) = Y$, where the unlabeled data are set to $Y^U_{ic} = 0, c = 1, \dots, C$.

3.2.1. Discussion

Setting $\alpha_i = 0$ for the labeled part of the data corresponds to clamping of the labels. However, this is not what we aim for in the presence of noisy labels. Therefore, a crucial property of the proposed framework is to set $\alpha_i > 0$ such that the labeled data can change labels during the propagation.

Moreover, we note that our extension of label propagation to multi-labels is very similar to the single-label variant introduced in [69], with the exception that we do not add the outlier class, which is not needed in our case. In other extensions to the multi-label label propagation [44, 45], the label matrix Y is normalized such that the rows sum to 1, which ensures that the output of the algorithm F also has rows that sum to 1. In the single-label case this makes sense in order to maintain the interpretability of probabilities. However, in the multi-label case the data points do not necessarily exclusively belong to a single class. Hence, the requirement $\sum_c F_{ic} = 1$ does not make sense since then x_i can maximally belong to one class if one think of F as a probability and require the probability to be 0.5 or higher in order to belong to a class.

On the other hand, in our case, a simple calculation shows that $0 \le F_{ic}(t+1) \le 1$:

$$\begin{aligned} F_{ic}(t+1) &= \alpha_i \sum_{m=1}^{n} T_{im} F_{mc}(t) + (1-\alpha_i) Y_{ic} \\ &\le \alpha_i \sum_{m=1}^{n} T_{im} + (1-\alpha_i) = \alpha_i + (1-\alpha_i) = 1, \end{aligned} \qquad (7)$$

since $F_{ic}(t) \le 1$ and $Y_{ic} \le 1$. However, we do not necessarily have that $\sum_c F_{ic} = 1$.

From matrix theory it is known that, given that $I - I_\alpha T$ is nonsingular, the solution of the linear iterative process (6) converges to the solution of

$$(I - I_\alpha T)F = (I - I_\alpha)Y, \qquad (8)$$

for any initialization $F(0)$ if and only if $I_\alpha T$ is a *convergent matrix* [70] (spectral radius $\rho(I_\alpha T) < 1$). $I_\alpha T$ is obviously convergent if $0 \le \alpha_i < 1\ \forall i$. Hence, we can find the soft labels F by solving the linear system given by Eq. (8).

Moreover, F_{ic} can be interpreted as the probability that datapoint x_i belongs to class c, and therefore, if one is interested in hard label assignments, $\tilde{Y}$, these can be found by letting $\tilde{Y}_{ic} = 1$ if $F_{ic} > 0.5$ and $\tilde{Y}_{ic} = 0$ otherwise.

3.3. Dimensionality reduction via dependence maximization

In this section we explain how we use the labels obtained using label propagation to learn the projection matrix P.

The motivation behind dependence maximization is that there should be a relation between the features and the label of an object. This should be the case also in the projected space. Hence, one should try to maximize the dependence between the feature similarity in the projected space and the label similarity. A common measure of such dependence is the Hilbert-Schmidt independence criterion (HSIC) [42], defined by

$$HSIC(X,Y) = \frac{1}{(n-1)^2} tr(KHLH), \qquad (9)$$

where tr denotes the trace of a matrix. $H \in \mathbb{R}^{n\times n}$ is given by $H_{ij} = \delta_{ij} - n^{-1}$, where $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. K is a kernel matrix over the feature space, whereas L is a kernel computed over the label space.

Let the projection of x be given by the projection matrix $P \in \mathbb{R}^{D\times d}$ and function $\Phi : \mathbb{R}^D \to \mathbb{R}^d$, $\Phi(x) = P^T x$. We select a linear kernel over the feature space, and therefore the kernel function is given by

$$\mathcal{K}(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = \langle P^T x_i, P^T x_j \rangle = P^T x_i x_j^T P \qquad (10)$$

Hence, given data $\{x_i\}_{i=1}^n$, the kernel matrix can be approximated by $K = XP^T PX^T$.

The kernel over the label space, $\mathcal{L}$, is given via the labels $y_i \in \{0,1\}^C$. One possible such kernel is the linear kernel

$$\mathcal{L}(y_i, y_j) = \langle y_i, y_j \rangle. \qquad (11)$$

However, in our semi-supervised setting, some of the labels are unknown and some are noisy. Hence, the kernel $\mathcal{L}$ cannot be computed. In order to enable DR in our non-standard problem, we propose to estimate the kernel using the labels obtained via our label propagation method. For the part of the data that was labeled from the beginning we use the hard labels, $\tilde{Y}^L$, obtained from the label propagation, whereas for the unlabeled part we use the soft labels, F^U. Hence, the kernel is approximated via $L = \tilde{F}\tilde{F}^T$, where $\tilde{F} = \begin{pmatrix} \tilde{Y}^L \\ F^U \end{pmatrix}$. The reason for using the hard labels obtained from label propagation for the labeled part is that we want some degree of certainty for those labels that change during the propagation (if the soft label F^L_{ic} changes with less than 0.5 from its initial value 0 or 1 during the propagation, the hard label Y^L_{ic} does not change).

The constant term, $(n-1)^{-2}$, in Eq. (9) is irrelevant in an optimization setting. Hence, by inserting the estimates of the kernels into Eq. (9), the following objective function is obtained,

$$\Psi(P) = tr(HXP^T PX^T H\tilde{F}\tilde{F}^T) = tr(P^T X^T H\tilde{F}\tilde{F}^T HXP). \qquad (12)$$

Note that the matrix $X^T H\tilde{F}\tilde{F}^T HX$ is symmetric. Hence, by requiring that the projection directions are orthogonal and that the new dimensionality is d, the following optimization problem is obtained

$$\begin{gathered} \arg\max_P \Psi(P) = \arg\max_P tr(P^T (X^T H\tilde{F}\tilde{F}^T HX)P), \qquad (13) \\ s.t.\ P \in \mathbb{R}^{D\times d},\ PP^T = I. \end{gathered}$$

As a consequence of the Courant-Fisher characterization [71], it follows that the maximum is achieved when P is an orthonormal basis corresponding to the d largest eigenvalues. Hence, P can be found by solving the eigenvalue problem

$$X^T H\tilde{F}\tilde{F}^T HXP = \Lambda P. \qquad (14)$$

The dimensionality of the projected space, d, is upper bounded by the rank of $\tilde{F}\tilde{F}^T$, which in turn is upper bounded by the number of classes C. Hence, d cannot be set larger than C.

3.4. Semi-supervised classification for noisy multi-label data

The multi-label k-nearest neighbor (ML-kNN) classifier [43] is a widely adopted classifier for multi-label classification. However, similarly to many other classifiers, its performance can be hampered if the dimensionality of the data is too high. Moreover, the ML-kNN classifier only works in a completely supervised setting. To resolve these problems, as an additional contribution of this work, we introduce a novel framework for semi-supervised classification of noisy multi-label data, consisting of two steps. In the first step, we compute a low dimensional embedding using NMLSDR. The second step consists of applying a semi-supervised ML-kNN classifier. For this classifier we use our label propagation method on the learned embedding to obtain a fully labeled dataset, and thereafter apply the ML-kNN classifier.

4. Experiments

In this paper, we have proposed a method for computing a low-dimensional embedding of noisy, partially labeled multi-label data. However, it is not a straightforward task to measure how well the method works. Even though the method is definitely relevant to real-world problems (illustrated in the case study in Sec. 5), the framework cannot be directly applied to most multi-label benchmark datasets since most of them are completely labeled, and the labels are assumed to be clean. Moreover, the NMLSDR provides a low dimensional embedding of the data, and we need a way to measure how good the embedding is. If the dimensionality is 2 or 3, this can to some degree be done visually by plotting the embedding. However, in order to quantitatively measure the quality and simultaneously maintain a realistic setup, we will apply our proposed end-to-end framework for semi-supervised classification and dimensionality reduction. In our experiments, this realistic semi-supervised setup will be applied in an illustrative example on synthetic data and in the case study.

A potential disadvantage of using a semi-supervised classifier, is that it does not necessarily isolate effect of the DR method that is used to compute the embedding. For this reason, we will also test our method on some benchmark datasets, but in order to keep everything coherent, except for the method used to compute the embedding, we compute the embedding using NMLSDR and baseline DR methods based on only the noisy and partially labeled multi-label training data. Thereafter, we assume that the true multi-labels are available when we train the ML-kNN classifier on the embeddings.

The remainder of this section is organized as follows. First we describe the performance measures we employed, baseline DR methods, and how we select hyper-parameters. Thereafter we provide an illustrative example on synthetic data, and secondly experiments on the benchmark data. The case study is described in the next section.

4.1. Evaluation metrics

Evaluation of performance is more complicated in a multi-label setting than for traditional single-labels. In this work, we decide use the seven different evaluation criteria that were employed in [40], namely Hamming loss (HL), Macro F1-score (MaF1), Micro F1 (MiF1), Ranking loss (RL), Average precision (AP), One-error (OE) and Coverage (Cov).

HL simply evaluates the number of times there is a mismatch between the predicted label and the true label, i.e.

$$HL = \sum_{i=1}^{n} \frac{\|\hat{y}_i \oplus y_i\|_1}{nC}, \tag{15}$$

where $\hat{y}_i$ denotes the predicted label vector of data point x_i and $\oplus$ is the XOR-operator. MaF1 is obtained by first computing the F1-score for each label, and then averaging over all labels.

$$MaF1 = \frac{1}{C} \sum_{c=1}^{C} \frac{2 \sum_{i=1}^{n} \hat{y}_{ic} y_{ic}}{\sum_{i=1}^{n} \hat{y}_{ic} + \sum_{i=1}^{n} y_{ic}}, \tag{16}$$

MiF1 calculates the F1 score on the predictions of different labels as a whole,

$$MiF1 = \frac{2 \sum_{i=1}^{n} \sum_{c=1}^{C} \hat{y}_{ic} y_{ic}}{\sum_{i=1}^{n} \sum_{c=1}^{C} \hat{y}_{ic} + \sum_{i=1}^{n} \sum_{c=1}^{C} y_{ic}}, \tag{17}$$

We note that HL, MiF1 and MaF1 are computed based on hard labels assignments, whereas the four other measures are computed based on soft labels. In all of our experiments, we obtain the hard labels by putting a threshold at 0.5.

RL computes the average ratio of reversely ordered label pairs of each data point. AP evaluates the average fraction of relevant labels ranked higher than a particular relevant label. OE gives the ratio of data points where the most confident predicted label is wrong. Cov gives an average of how far one needs to go down on the list of ranked labels to cover all the relevant labels of the data point. For a more detailed description of these measures, we point the interested reader to [72].

In this work, we modify four of the evaluation metrics such that all of them take values in the interval $[0, 1]$ and “higher always is better”. Hence, we define

$$HL' = 1 - HL, \tag{18}$$

$$RL' = 1 - RL, \tag{19}$$

$$OE' = 1 - OE, \tag{20}$$

and normalized coverage (Cov’) by

$$Cov' = 1 - Cov/(C - 1). \tag{21}$$

4.2. Baseline dimensionality reduction methods

In this work, we consider the following other DR methods: CCA, MVMD, MDDMp, MDDMf and four variants of MLDA, namely wMLDAb, wMLDAe, wMLDAc and wMLDAd. These methods are supervised and require labeled data, and are therefore trained only on the labeled part of the training data. In addition, we compare to a semi-supervised method, SSMLDR, which we adapt to noisy multi-labels by using the label propagation algorithm we propose in this paper instead of the label propagation method that was originally proposed in SSMLDR.

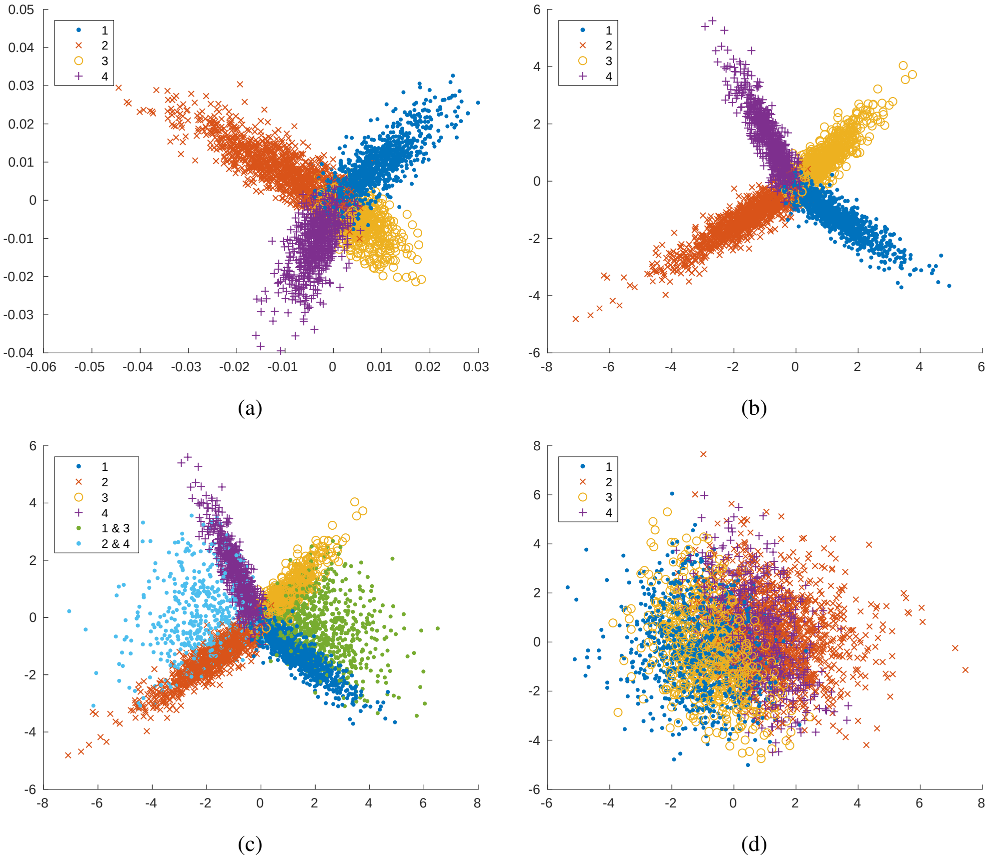


Figure 1: 3 dimensional embedding of the synthetic dataset obtained using (a) SSMLDR; (b) NMLSDR; (c) NMLSDR with multi-classes included; and (d) PCA.

4.3. Hyper-parameter selection

For the ML-kNN classifier we set $k = 10$. The effect of varying the number of neighbors will be left for further work. In order to learn the NMLSDR embedding we use a kNN-graph with $k = 10$ and binary weights. Moreover, we set $\alpha_i = 0.6$ for labeled data and $\alpha_i = 0.999$ for unlabeled data. By doing so, one ensures that an unlabeled datapoint is not affected by its initial value, but gets all contribution from the neighbors during the propagation.

4.4. Illustrative example on synthetic toy data

Dataset description. To test the framework in a controlled experiment, a synthetic dataset is created as follows.

A dataset of size 8000 samples is created, where each of the data points has dimensionality 320. The number of classes is set to 4, and we generate 2000 samples from each class. 30% from class 1 also belong to class 2, and vice versa. 20% from class 2 also belong to class 3 and vice versa, whereas 25% from class 3 also belong to class 4 and vice versa.

A sample from class i is generated by randomly letting 10% of the features in the interval $\{20(i-1)+1, \ldots, 20i\}$ take a random integer value between 1 and 10. Since there are 4 classes, this means that the first 80 features are directly dependent on the class-membership.

For the remaining 240 features we consider 20 of them at the time. We randomly select 50% of the 8000 samples and randomly let 20% of the 20 features take a random integer value between 1 and 10. We repeat this procedure for the 12 different sets of 20 features $\{20(i-1)+1, \ldots, 20i\}$, $i = 5, 6, \ldots, 16$.

All features that are not given a value using the procedure described above are set to 0. Noise is injected into the labels by randomly flipping a fraction $p = 0.1$ of the labels and we make the data partially labeled by removing 50 % of the labels. 2000 of the samples are kept aside as an independent test set. We note that noisy labels are often easier and cheaper to obtain than true labels and it is therefore not unreasonable that the fraction of labeled examples is larger than what it commonly is in traditional semi-supervised learning settings.

Results. We apply the NMLSDR method in combination with the semi-supervised ML-kNN classifier as explained above and compare to SSMLDR. We create two baselines by, for both of these methods, using a different value for the hyperparameter α_i for the labeled part of the data, namely 0, which corresponds to clamping. We denote these two baselines by SSMLDR* and NMLSDR*. In addition, we compare to baselines that only utilize the labeled part of the data, namely the supervised DR methods explained above in combination with a ML-kNN classifier. The data is standardized to 0 mean and 1 in standard

Method	HL'	RL'	AP	OE'	Cov'	MaF1	MiF1
CCA	0.863	0.884	0.898	0.852	0.816	0.787	0.785
MVMD	0.906	0.912	0.924	0.897	0.836	0.850	0.849
MDDMp	0.906	0.911	0.924	0.897	0.836	0.851	0.850
MDDMf	0.859	0.888	0.900	0.855	0.819	0.785	0.783
wMLDAb	0.844	0.871	0.885	0.831	0.807	0.754	0.750
wMLDAe	0.864	0.885	0.899	0.855	0.818	0.790	0.788
wMLDAc	0.865	0.887	0.900	0.857	0.818	0.787	0.785
wMLDAd	0.869	0.891	0.907	0.869	0.822	0.788	0.786
SSMLDR*	0.863	0.883	0.899	0.859	0.814	0.796	0.793
SSMLDR	0.879	0.898	0.910	0.871	0.827	0.817	0.814
NMLSDR*	0.907	0.919	0.929	0.903	0.842	0.861	0.859
NMLSDR	**0.913**	**0.925**	**0.935**	**0.912**	**0.846**	**0.868**	**0.866**

Table 1: Performance of different embeddings on the synthetic dataset.

deviation and we let the dimensionality of the embedding be 3.

Fig. 1a and 1b show the embeddings obtained obtained using SSMLDR and NMLSDR, respectively. For ivisualization purposes, we have only plotted those datapoints that exclusively belong to one class. In Fig. 1c, we have added two of the multi-classes for the NMLSDR embedding. For comparison, we also added the embedding obtained using PCA in Fig. 1d. As we can see, in the PCA embedding the classes are not separated from each other, whereas in the NMLSDR and SSMLDR embeddings the classes are aligned along different axes. It can be seen that the classes are better separated and more compact in the NMLSDR embedding than the SSMLDR embedding. Fig. 1c shows that the data points that belong to multiple classes are placed where they naturally belong, namely between the axes corresponding to both of the classes they are member of.

Tab. 1 shows the results obtained using the different methods on the synthetic dataset. As we can see, our proposed method gives the best performance for all metrics. Moreover, NMLSDR with $\alpha_i^L = 0$, which corresponds to clamping of the labeled data during label propagation gives the second best results but cannot compete with our proposed method, in which the labels are allowed to change during the propagation to account for noisy labels. We also note that, even though the SSMLDR improves the MLDA approaches that are based on only the labeled part of the data, it gives results that are considerably worse than NMLSDR.

4.5. Benchmark datasets

Experimental setup. We consider the following benchmark datasets [2]: Birds, Corel, Emotions, Enron, Genbase, Medical, Scene, Tmc2007 and Yeast. We also add our synthetic toy dataset as a one of our benchmark datasets (described in Sec. 4.4). These datasets are shown in Tab. 2, along with some useful characteristics. In order to be able to apply our framework to the benchmark datasets, we randomly flip 10 % of the labels to generate noisy labels and let 30 % of the data points training sets be labeled. All datasets are standardized to zero mean and standard deviation one.

We apply the DR methods to the partially and noisy labeled multi-label training sets in order to learn the projection matrix P, which in turn is used to map the D-dimensional training and test sets to a $d-$dimensional representation. d is set as large as possible, i.e. to $C - 1$ for the MLDA-based methods and C for the other methods. Then we train a ML-kNN classifier using the low-dimensional training sets, assuming that the true multi-labels are known and validate the performance on the low-dimensional test sets.

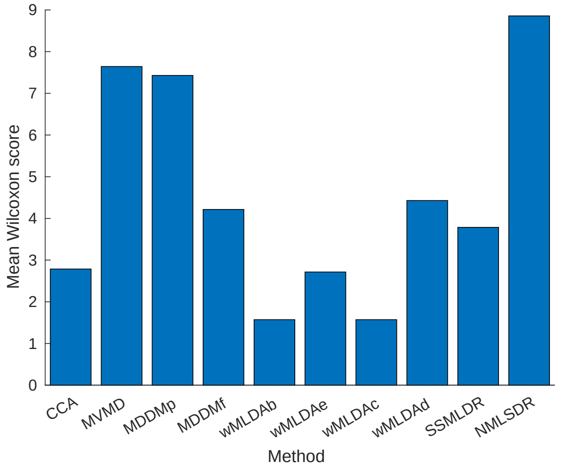


Figure 2: Mean of the Wilcoxon score obtained over the 7 different metrics.

In total we are evaluating the performance over 10 different datasets and across 7 different performance measures for all the feature extraction methods we use. Hence, to investigate which method performs better according to the different metrics, we also report the number of times each method gets the highest value of each metric. In addition, we compare all pairs of methods by using a Wilcoxon signed rank test with 5% significance level [73]. Similarly to [56], if method A performs better than B according to the test, A is assigned the score 1 and B the score 0. If the null hypothesis (method A and B perform equally) is not rejected, both A and B are assigned an equal score of 0.5.

Results. Tab. 3 shows results in terms of HL'. NMLSDR gets best HL'-score for eight of the datasets and achieves a maximal Wilcoxon score, i.e performs statistically better than all nine other methods according to the test at a 5 % significance level. The second best method MDDMp gets the highest HL' score for three datasets and Wilcoxon score of 7.5. From Tab. 4 we see that NMLSDR achieves the highest RL'-score seven times and a Wilcoxon score of 8.5. The second best method is MVMD, which obtains three of the highest RL' values and a Wilcoxon score of 8.0.

Tab. 5 shows performance in terms of AP. The highest AP score is achieved for NMLSDR for eight datasets and it gets a maximal Wilcoxon score of 9.0. According to the Wilcoxon score second place is tied between MVMD and MDDMp. However, MVMD gets the highest AP score for two datasets, whereas MDDMp does not get the highest score for any of them. OE' is presented in Tab. 6. We can see that NMLSDR gets a maximal Wilcoxon score and the highest OE' score for seven datasets. MVMD is number two with a Wilcoxon score

[2] Downloaded from mulan.sourceforge.net/datasets-mlc.html

Dataset	Domain	Train instances	Test instances	Attributes	Labels	Cardinality
Birds	audio	322	323	260	19	1.06
Corel	scene	5188	1744	500	153	2.87
Emotions	music	391	202	72	6	1.81
Enron	text	1123	579	1001	52	3.38
Genbase	biology	463	199	99	25	1.26
Medical	text	645	333	1161	39	1.24
Scene	scene	1211	1196	294	6	1.06
Tmc2007	text	3000	7077	493	22	2.25
Toy	synthetic	6000	2000	320	4	1.38
Yeast	biology	1500	917	103	14	4.23

Table 2: Description of benchmark datasets considered in our experiments.

	CCA	MVMD	MDDMp	MDDMf	wMLDAb	wMLDAe	wMLDAc	wMLDAd	SSMLDR	NMLSDR
Birds	0.947	0.950	0.950	0.947	0.948	0.949	0.949	0.949	0.949	0.951
Corel	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980	0.980
Emotions	0.715	0.771	0.778	0.711	0.696	0.714	0.709	0.717	0.786	0.787
Enron	0.941	0.950	0.950	0.942	0.941	0.941	0.941	0.940	0.938	0.950
Genbase	0.989	0.996	0.996	0.988	0.990	0.991	0.988	0.989	0.994	0.997
Medical	0.976	0.974	0.974	0.976	0.974	0.975	0.975	0.976	0.966	0.975
Scene	0.810	0.899	0.900	0.809	0.810	0.814	0.817	0.810	0.873	0.897
Tmc2007	0.914	0.928	0.928	0.912	0.911	0.911	0.911	0.916	0.922	0.929
Toy	0.836	0.894	0.894	0.839	0.821	0.831	0.831	0.854	0.861	0.903
Yeast	0.780	0.791	0.790	0.782	0.785	0.783	0.781	0.781	0.793	0.793
Best values	2	2	3	2	1	1	1	2	2	**8**
Wilcoxon	2.0	7.0	7.5	2.5	2.0	3.0	2.5	3.5	6.0	**9.0**

Table 3: Performance in terms of 1 - Hamming loss (HL') across 10 different benchmark datasets.

	CCA	MVMD	MDDMp	MDDMf	wMLDAb	wMLDAe	wMLDAc	wMLDAd	SSMLDR	NMLSDR
Birds	0.715	0.766	0.767	0.734	0.709	0.718	0.719	0.725	0.681	0.771
Corel	0.800	0.808	0.808	0.800	0.799	0.799	0.800	0.800	0.801	0.814
Emotions	0.695	0.824	0.824	0.709	0.693	0.700	0.676	0.714	0.829	0.845
Enron	0.894	0.911	0.911	0.893	0.893	0.892	0.891	0.893	0.883	0.914
Genbase	0.993	0.995	0.995	0.993	0.994	0.992	0.992	0.991	0.995	1.000
Medical	0.925	0.952	0.949	0.925	0.916	0.921	0.919	0.945	0.856	0.946
Scene	0.585	0.900	0.898	0.629	0.574	0.583	0.572	0.616	0.853	0.898
Tmc2007	0.831	0.906	0.906	0.830	0.830	0.830	0.831	0.847	0.872	0.910
Toy	0.871	0.909	0.909	0.870	0.849	0.865	0.861	0.888	0.887	0.926
Yeast	0.806	0.820	0.819	0.811	0.810	0.809	0.806	0.803	0.818	0.816
Best values	0	3	0	0	0	0	0	0	0	**7**
Wilcoxon	3.0	8.0	7.5	4.5	1.5	2.0	2.0	5.0	3.0	**8.5**

Table 4: Performance in terms of 1 - Ranking loss (RL') across 10 different benchmark datasets.

	CCA	MVMD	MDDMp	MDDMf	wMLDAb	wMLDAe	wMLDAc	wMLDAd	SSMLDR	NMLSDR
Birds	0.389	0.499	0.500	0.426	0.374	0.392	0.379	0.424	0.357	0.502
Corel	0.260	0.277	0.277	0.261	0.265	0.263	0.263	0.268	0.266	0.288
Emotions	0.669	0.781	0.773	0.686	0.672	0.687	0.666	0.704	0.799	0.808
Enron	0.592	0.669	0.670	0.583	0.584	0.582	0.580	0.578	0.526	0.675
Genbase	0.963	0.990	0.993	0.964	0.960	0.968	0.963	0.969	0.984	0.997
Medical	0.673	0.722	0.716	0.666	0.644	0.674	0.669	0.723	0.446	0.725
Scene	0.491	0.836	0.835	0.534	0.481	0.488	0.475	0.521	0.781	0.834
Tmc2007	0.584	0.714	0.713	0.587	0.579	0.576	0.577	0.623	0.662	0.721
Toy	0.882	0.921	0.921	0.880	0.862	0.880	0.875	0.900	0.897	0.933
Yeast	0.732	0.748	0.747	0.731	0.733	0.733	0.729	0.725	0.745	0.741
Best values	0	2	0	0	0	0	0	0	0	**8**
Wilcoxon	3.5	7.5	7.5	4.0	1.0	3.5	1.0	5.0	3.0	**9.0**

Table 5: Performance in terms of Average precision (AP) across 10 different benchmark datasets.

of 8.0 and two best values.

Tab. 7 shows Cov'. NMLSDR gets a maximal Wilcoxon score and the highest Cov' value for seven datasets. Despite that MVMD gets the highest Cov' for three datasets and MDDMp for none of the datasets, the second best Wilcoxon score is 7.5 and tied between MVMD and MDDMp. MaF1 is shown in Tab. 8. The best method, which is our proposed method gets a maximal Wilcoxon score and the highest MaF1 value

	CCA	MVMD	MDDMp	MDDMf	wMLDAb	wMLDAe	wMLDAc	wMLDAd	SSMLDR	NMLSDR
Birds	0.273	0.419	0.407	0.314	0.250	0.273	0.250	0.297	0.203	0.419
Corel	0.250	0.261	0.262	0.252	0.255	0.254	0.253	0.267	0.260	0.283
Emotions	0.535	0.673	0.644	0.564	0.535	0.589	0.550	0.589	0.718	0.728
Enron	0.620	0.762	0.762	0.610	0.587	0.604	0.606	0.579	0.544	0.765
Genbase	0.950	0.990	0.995	0.955	0.935	0.960	0.950	0.965	0.980	0.995
Medical	0.583	0.607	0.592	0.589	0.538	0.583	0.577	0.628	0.323	0.619
Scene	0.265	0.732	0.729	0.319	0.258	0.264	0.247	0.303	0.656	0.727
Tmc2007	0.527	0.650	0.648	0.531	0.523	0.519	0.516	0.578	0.604	0.656
Toy	0.821	0.888	0.887	0.819	0.785	0.821	0.811	0.850	0.849	0.903
Yeast	0.760	0.755	0.749	0.740	0.747	0.751	0.748	0.744	0.751	0.739
Best values	1	2	1	0	0	0	0	1	0	**7**
Wilcoxon	3.5	8.0	7.0	4.0	1.0	3.5	1.0	5.0	3.0	**9.0**

Table 6: Performance in terms of 1 - One error (OE') across 10 different benchmark datasets.

	CCA	MVMD	MDDMp	MDDMf	wMLDAb	wMLDAe	wMLDAc	wMLDAd	SSMLDR	NMLSDR
Birds	0.821	0.851	0.852	0.830	0.818	0.824	0.824	0.831	0.808	0.860
Corel	0.601	0.617	0.617	0.603	0.600	0.599	0.601	0.603	0.603	0.628
Emotions	0.563	0.684	0.679	0.579	0.567	0.565	0.554	0.587	0.679	0.696
Enron	0.738	0.762	0.763	0.736	0.737	0.736	0.734	0.736	0.724	0.768
Genbase	0.983	0.984	0.984	0.983	0.985	0.981	0.981	0.980	0.985	0.991
Medical	0.918	0.941	0.939	0.917	0.909	0.913	0.911	0.936	0.859	0.939
Scene	0.637	0.899	0.898	0.672	0.625	0.633	0.624	0.663	0.860	0.898
Tmc2007	0.740	0.835	0.835	0.741	0.740	0.739	0.741	0.762	0.790	0.840
Toy	0.809	0.837	0.837	0.807	0.794	0.805	0.802	0.822	0.820	0.849
Yeast	0.513	0.533	0.532	0.526	0.526	0.523	0.519	0.518	0.530	0.528
Best values	0	3	0	0	0	0	0	0	0	**7**
Wilcoxon	2.5	7.5	7.5	4.5	2.0	2.5	1.5	5.0	3.0	**9.0**

Table 7: Performance in terms of 1 - Normalized coverage (Cov') across 10 different benchmark datasets.

	CCA	MVMD	MDDMp	MDDMf	wMLDAb	wMLDAe	wMLDAc	wMLDAd	SSMLDR	NMLSDR
Birds	0.011	0.079	0.076	0.027	0.002	0.000	0.000	0.039	0.006	0.104
Corel	0.012	0.023	0.022	0.014	0.010	0.010	0.010	0.019	0.010	0.021
Emotions	0.381	0.599	0.604	0.419	0.366	0.385	0.371	0.415	0.623	0.649
Enron	0.044	0.102	0.105	0.048	0.043	0.049	0.044	0.065	0.063	0.101
Genbase	0.520	0.561	0.603	0.514	0.497	0.515	0.497	0.442	0.558	0.630
Medical	0.153	0.168	0.164	0.159	0.135	0.126	0.133	0.197	0.038	0.175
Scene	0.059	0.705	0.707	0.132	0.084	0.055	0.041	0.098	0.569	0.700
Tmc2007	0.183	0.419	0.418	0.189	0.171	0.177	0.175	0.212	0.349	0.434
Toy	0.732	0.830	0.828	0.741	0.709	0.722	0.724	0.758	0.776	0.845
Yeast	0.266	0.318	0.323	0.276	0.281	0.279	0.248	0.233	0.321	0.342
Best values	0	1	2	0	0	0	0	1	0	**6**
Wilcoxon	2.5	7.5	7.5	5.0	2.0	2.0	1.0	3.5	5.0	**9.0**

Table 8: Performance in terms of Macro F1-score (MaF1) across 10 different benchmark datasets.

	CCA	MVMD	MDDMp	MDDMf	wMLDAb	wMLDAe	wMLDAc	wMLDAd	SSMLDR	NMLSDR
Birds	0.036	0.178	0.172	0.063	0.006	0.000	0.000	0.065	0.019	0.197
Corel	0.017	0.033	0.031	0.019	0.013	0.013	0.013	0.031	0.015	0.033
Emotions	0.459	0.630	0.639	0.450	0.404	0.448	0.430	0.460	0.652	0.666
Enron	0.351	0.523	0.530	0.413	0.340	0.378	0.369	0.310	0.346	0.518
Genbase	0.882	0.953	0.959	0.872	0.885	0.902	0.873	0.881	0.932	0.968
Medical	0.459	0.501	0.495	0.505	0.400	0.440	0.455	0.498	0.212	0.496
Scene	0.066	0.700	0.702	0.142	0.086	0.058	0.041	0.102	0.584	0.698
Tmc2007	0.421	0.589	0.586	0.443	0.440	0.438	0.438	0.485	0.540	0.590
Toy	0.729	0.828	0.826	0.739	0.706	0.719	0.721	0.756	0.774	0.843
Yeast	0.573	0.605	0.607	0.577	0.582	0.584	0.555	0.548	0.609	0.626
Best values	0	1	2	1	0	0	0	0	0	**7**
Wilcoxon	2.5	8.0	7.5	5.0	1.5	2.5	2.0	4.0	3.5	**8.5**

Table 9: Performance in terms of Micro F1-score (MiF1) across 10 different benchmark datasets.

for six datasets. Tab. 9 shows MiF1. NMLSDR achieves 8.5 in Wilcoxon score and has the highest MiF1 score for seven datasets.

In total, NMLSDR consistently gives the best performance for all seven evaluation metrics. Moreover, in order to summarize our findings, we compute the mean Wilcoxon score across all seven performance metrics and plot the result in Fig. 2. If we sort these results, we get NMLSDR (8.86), MVMD (7.64), MDDMp (7.43), wMLDAd (4.43), MDDMf (4.21), SSMLDR (3.79), CCA (2.79), wMLDAe (2.71) and wMLDAb/wMLDAc (1.57). The best method, which is our proposed method, gets a mean value that is 1.22 higher than number two. The second best method is MVMD, slightly better than MDDMp. The best MLDA-based method is wMLDAd, which is ranked 4th, however, with a much lower mean value than the three best methods. The semi-supervised extension of MLDA (SSMLDR) is ranked 6th and is actually performing worse that wMLDAd, which is a bit surprising. However, SSMLDR also uses a binary weighting scheme, and should therefore be considered as a semi-supervised variant of wMLDAb, which it performs considerably better than. wMLDAb and wMLDAc give the worst performance of all the 10 methods.

The main reason why the MLDA-based approaches in general perform worse than the other DR methods is probably related to what we discussed in Sec. 2, namely that LDA-based approaches are heavily affected by outliers and wrongly labeled data. More concretely, the fact that the number of labeled data points are relatively few and that the labels are noisy, leads to errors in the scatter matrices that even might amplify since one has to invert a matrix to solve the generalized eigenvalue problem. The semi-supervised extension of MLDA, SSMLDR, improves quite much compared to wMLDAb, but the starting point is so bad that even though it improves, it cannot compete with the best methods. On the other hand, the MDDM-based methods (MVMD and MDDMp) are not so sensitive to label noise and the fact that there are few labels, and therefore these methods can perform quite well even though they are trained only on the labeled subset. Hence, the reasons to the good performance of NMLSDR are probably that MDDMp is the basis of NMLSDR, and that NMLSDR in addition uses our label propagation method to improve.

5. Case study

In this section, we describe a case study where we study patients potentially suffering from multiple chronic diseases. This healthcare case study reflects the need for label noise-tolerant methods in a non-standard situation (semi-supervised learning, multiple labels, high dimensionality). The objective is to identify patients with certain chronic diseases, more specifically hypertension and/or diabetes mellitus. In order to do so, we take an approach where we use clinical expertise to create a partially and noisy labeled dataset, and thereafter apply our proposed end-to-end framework, namely NMLSDR for dimensionality reduction in combination with semi-supervised ML-kNN to classify these patients. An overview of the framework employed in the case study is shown in Fig. 3.

Chronic diseases. According to The World Health Organisation, a disease is defined as chronic if one or several of the following criteria are satisfied: the disease is permanent, requires special training of the patient for rehabilitation, is caused by non-reversible pathological alterations, or requires a long period of supervision, observation, or care. The two most prevalent chronic diseases for people over 64 years are those that we study in this paper, namely hypertension and diabetes mellitus [74]. These types of diseases represent an increasing problem in modern societies all over the world, which to a large degree is due to a general increase in life expectancy, along with an increased prevalence of chronic diseases in an aging population [75]. Moreover, the economical burden associated with these chronic conditions is high. For example, in 2017, treatment of diabetic patients accounted for 1 out of 4 healthcare dollars in the United States [76]. Hence, in the future, a significant amount of resources must be devoted to the care of chronic patients and it will be important not only to improve the patient care, but also more efficiently allocate the resources spent on treatment of these diseases.

5.1. Data

In this case study, we study a dataset consisting of patients that potentially have one or more chronic diseases. All of these patients got some type of treatment at University Hospital of Fuenlabrada, Madrid (Spain) in the year 2012. The patients are described by diagnosis codes following the International Classification of Diseases 9th revision, Clinical Modification (ICD9-CM) [77], and pharmacological dispensing codes according to Anatomical Therapeutic Chemical (ATC) classification systems [78]. Some preprocessing steps are considered. Similarly to [79, 80], the ICD9-CM and ATC codes are represented using frequencies, i.e, for each patient, we consider all encounters with the health system in 2012 and we count how many times each ICD9-CM and ATC code appear in the electronic health record. In total there are 1517 ICD9-CM codes and 746 ATC codes. However, all codes that appear for less than 10 patients across the training set are removed. After this feature selection, the dimensionality of the data is 455, of which 267 represent ICD9-CM codes and 188 represent ATC codes.

We do have access to ground truth labels that indicate what type of chronic disease(s) the patients have. These are provided by a patient classification system developed by the company 3M [81]. This classification system stratify patients into so-called Clinical Risk Groups (CRG) that indicate what type(s) of chronic disease the patient has and the severity based on the patient encounters with the health system during a period of time, typically one year. A five-digit classification code is used to assign each patient to a severity risk group. The first digit of the CRG is the core health status group, ranging from healthy (1) to catastrophic (9); the second to fourth digits represents the base 3M CRG; and the fifth digit is used for characterizing the severity-of-illness levels.

For the purpose of this work, the ground truth labels are only used for cohort selection and final evaluation of our models. For the remaining parts they are considered unknown. To select a cohort, we consider the first four digits of the CRGs to analyze the the following chronic conditions: CRG-1000 (healthy), which contains 46835 individuals; CRG-5192 (hypertension) with 12447 patients; CRG-5424 (diabetes), which has 2166 patients; and CRG-6144 (hypertension and diabetes), with a total of 3179 patients. We employ an undersampling strategy and randomly select 2166 patients from each of the four categories, and thereby obtain balanced classes. An independent test set is

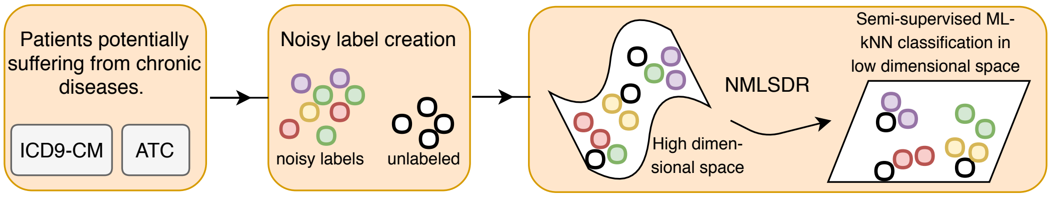

Patients potentially suffering from chronic diseases.
ICD9-CM
ATC
Noisy label creation
noisy labels
unlabeled
NMLSDR
High dimensional space
Semi-supervised ML-kNN classification in low dimensional space

Figure 3: Illustration of proposed framework applied to identify patients with chronic diseases.

created by randomly selecting 20 % of these patients. Hence, the training set contains 6932 patients and the test set 1732 patients.

5.2. *Rule-based creation of noisy labeled training data using clinical knowledge*

There are some important ICD9-CM codes and ATC-drugs that are strongly correlated with hypertension and diabetes, respectively. These are verified by our clinical experts and described in Tab. 10. In particular, the ICD9-CM code 250 is important for diabetes because it is the code for *diabetes mellitus*. Similarly, the ICD9-CM codes 401-405 are important for hypertension because they describe different types of hypertension.

In this case study we are interested in four groups, namely those that have hypertension, those that have diabetes, those that have both, and those that do not have any these two chronic diseases. Thanks to the clinical expertise and the information that they provided us with, which is summarized in Tab. 10, we can create a partially and noisy labeled dataset using the following set of rules.

1. Those that have the ICD codes 250 and any of the codes 401-405 are assigned to both the hypertension and diabetes class.
2. Those that have the ICD code 250, but none of the 7 ICD9-CM codes and 64 ATC drugs listed by the clinicians as indicators for hypertension, are labeled with diabetes.
3. Those that have any of the ICD9-CM codes 401-405, but none of the 4 ICD9-CM codes for diabetes or 12 ATC drugs for diabetes, are labeled with hypertension.
4. Those that do not have any of the ICD9-CM codes or ATC drugs listed up in Tab. 10 are labeled as healthy.
5. The remaining patients do not get a label.

In total, this leads to 1734 in the healthy class, 2547 in the hypertension class, 1971 in the diabetes class. 1302 of the patients in the hypertension class also belongs to the diabetes class. 1982 of the patients do not get a label using the the routine described above. To be able to examine for statistical significance, we randomly select 1000 of the noisy labeled patients and 1000 of the unlabeled patients. By doing so, we can repeat the experiments several times and test for significance using a pairwise t-test. We do the repetition 10 times and let the significance level be 95%.

5.2.1. *Performing feature extraction and classification*

After having obtained the partially and noisy labeled multi-label dataset, we do feature extraction using NMLSDR, followed by semi-supervised multi-label classification, exactly in the same manner as we did it for the synthetic toy data in Section 4.4. In this case study, we use the same evaluation metrics, hyper-parameters and baseline feature extraction methods as explained in Sec. 4.1. The dimensionality of the embedding is set to 2 for all embedding methods.

5.3. *Results*

Tab. 11 shows the performance of the different DR methods on the task of classifying patients with chronic diseases in terms of seven different evaluation metrics. According to the pairwise t-test, our method achieves the best performance for all metrics. Second place is tied between MDDMp and MVMD. The semi-supervised variant of MLDA, namely SSMLDR, performs better than the supervised counterparts (wMLDAb, wMLDAc, wMLDAd, wMLDAe) and is consistently ranked 4th according to all metrics. Interestingly, the more advanced weighting schemes in wMLDAc and wMLDAd actually lead to worse results than what the simple weights in wMLDAb and wMLdAe give. CCA gives the worst performance according to 4 of the evaluation measures, for the 3 other measures the difference between CCA and wMLDAd is not significant.

Fig. 4 shows plots of the two-dimensional embeddings of the chronic patients obtained using four different DR methods, namely MDDMp, wMLDAb, NMLSDR and SSMLDR. The different colors and markers represent the true CRG-labels of the patients. As we can see, visually the MDDMp and NMLSDR embeddings look quite similar. The healthy patients are squeezed together in a small area (purple dots), and the yellow dots that represent patients that have both diabetes and hypertension are placed between the blue dots, which are those that have only hypertension, and the red dots, which represent the patient that only have diabetes. Intuitively, this placement makes sense. On the other hand, the embedding obtained using SSMLDR does not look similar to its counterpart obtained using wMLDAb, and it is easy to see why the performance of wMLDAb is worse.

6. Conclusions

In this paper we have introduced the NMLSDR method, a dimensionality reduction method for partially and noisy labeled multi-label data. To our knowledge, NMLSDR is the only

Chronicity	ATC codes	ICD9-CM codes
Hypertension	C01AA, C01BA, C01BA, C01BC, C01BD, C01CA, C01CB, C01CX, C01DA, C01DX, C01EB, C02AB, C02AC, C02CA, C02DB, C02DC, C02DD, C02K, C02LC, C03AA, C03AX, C03BA, C03CA, C03DA C03EA, C03EB, C04AD, C04AE, C04AX, C05AA, C05AD, C05AE, C05AX, C05BA, C05BB, C05BX, C05CA, C05CX, C07AA, C07AB, C07AG, C07B, C07G, C07D, C07E, C07X, C08CA, C08DA, C08DB, C08GA, C09AA, C09BA, C09BB, C09CA, C09DA, C09DB, C09XA, C10AA, C10AB, C10AC, C10AD, C10AX, C10BA, C10BX	362, 401, 402, 403, 404, 405, 760
Diabetes	A10AB, A10AC, A10AD, A10AE, A10AF, A10BA, A10BB, A10BD, A10BFM, A10BGM, A10BH, A10BX,	250, 588, 648, 775

Table 10: ICD9-CM codes and ATC codes associated with hypertension and diabetes.

Method	HL'	RL'	AP	OE'	Cov'	MaF1	MiF1
CCA	0.782 ± 0.009	0.823 ± 0.008	0.866 ± 0.006	0.755 ± 0.011	0.798 ± 0.004	0.712 ± 0.012	0.741 ± 0.011
MVMD	0.875 ± 0.006	0.930 ± 0.006	0.942 ± 0.004	0.894 ± 0.006	0.861 ± 0.005	0.853 ± 0.008	0.858 ± 0.006
MDDMp	0.875 ± 0.006	0.930 ± 0.005	0.942 ± 0.003	0.895 ± 0.006	0.861 ± 0.005	0.853 ± 0.008	0.858 ± 0.006
MDDMf	0.811 ± 0.010	0.853 ± 0.012	0.888 ± 0.009	0.798 ± 0.017	0.815 ± 0.006	0.750 ± 0.015	0.774 ± 0.013
wMLDAb	0.794 ± 0.007	0.844 ± 0.012	0.883 ± 0.008	0.788 ± 0.017	0.810 ± 0.008	0.731 ± 0.012	0.744 ± 0.011
wMLDAe	0.805 ± 0.008	0.856 ± 0.009	0.891 ± 0.006	0.801 ± 0.014	0.818 ± 0.005	0.749 ± 0.013	0.763 ± 0.012
wMLDAc	0.790 ± 0.007	0.842 ± 0.008	0.882 ± 0.004	0.783 ± 0.009	0.810 ± 0.005	0.729 ± 0.012	0.745 ± 0.011
wMLDAd	0.779 ± 0.013	0.838 ± 0.012	0.874 ± 0.008	0.770 ± 0.016	0.805 ± 0.008	0.720 ± 0.017	0.729 ± 0.018
SSMLDR	0.839 ± 0.005	0.889 ± 0.009	0.911 ± 0.006	0.839 ± 0.012	0.835 ± 0.008	0.799 ± 0.007	0.811 ± 0.005
NMLSDR	**0.882 ± 0.005**	**0.939 ± 0.004**	**0.950 ± 0.003**	**0.909 ± 0.006**	**0.867 ± 0.005**	**0.864 ± 0.007**	**0.865 ± 0.005**

Table 11: Results in terms of 7 evaluation measures (average±std) obtained by doing feature extraction using different methods, followed by semi-supervised ML-kNN classification, on partially and noisy labeled chronicity data. The best performing methods according to each of the 7 metrics are marked in bold, where the statistical significance is examined using a pairwise t-test at 95% significance level.

method the can explicitly deal with this type of data. Key components in the method are a label propagation algorithm that can deal with noisy data and maximization of feature-label dependence using the Hilbert-Schmidt independence criterion. Our extensive experimental sections show that NMLSDR is a good dimensionality reduction method in settings where one has access to partially and noisy labeled multi-label data.

In the future, we will investigate more thoroughly the effect of using different weighting schemes in NMLSDR, similarly to how it is done in MLDA with wMLDAb, wMLDAc, wMLDAd and wMDLAd.

Acknowledgments

This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the *Next Generation Learning Machines*. Cristina Soguero-Ruiz is partially supported by project TEC2016-75361-R from Spanish Government and by Project DTS17/00158 from Institute of Health Carlos III (Spain). The authors would like to thank clinicians at University Hospital of Fuenlabrada for their helpful advice and comments on various issues examined in the case study.

References

[1] D. F. Nettleton, A. Orriols-Puig, A. Fornells, A study of the effect of different types of noise on the precision of supervised learning techniques, Artificial Intelligence Review 33 (4) (2010) 275–306. doi:10.1007/s10462-010-9156-z.

[2] B. Frenay, M. Verleysen, Classification in the presence of label noise: A survey, IEEE Transactions on Neural Networks and Learning Systems 25 (5) (2014) 845–869. doi:10.1109/TNNLS.2013.2292894.

[3] X. Zhu, X. Wu, Class noise vs. attribute noise: A quantitative study, Artificial Intelligence Review 22 (3) (2004) 177–210. doi:10.1007/s10462-004-0751-8.

[4] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, A. Tewari, Learning with noisy labels, in: Advances in neural information processing systems, 2013, pp. 1196–1204.

[5] M. Pechenizkiy, S. Puuronen, A. Tsymbal, O. Pechenizkiy, Class noise and supervised learning in medical domains: The effect of feature extraction, in: 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)(CBMS), Vol. 00, 2006, pp. 708–713. doi:10.1109/CBMS.2006.65.

[6] J. A. Aslam, S. E. Decatur, On the sample complexity of noise-tolerant learning, Information Processing Letters 57 (4) (1996) 189–195. doi:10.1016/0020-0190(96)00006-3.

[7] T. Xiao, T. Xia, Y. Yang, C. Huang, X. Wang, Learning from massive noisy labeled data for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2691–2699.

[8] P. A. Lachenbruch, Discriminant analysis when the initial samples are misclassified, Technometrics 8 (4) (1966) 657–662.

[9] Y. Bi, D. R. Jeske, The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise, Journal of Multivariate Analysis 101 (7) (2010) 1622–1637. doi:10.1016/j.jmva.2010.03.001.

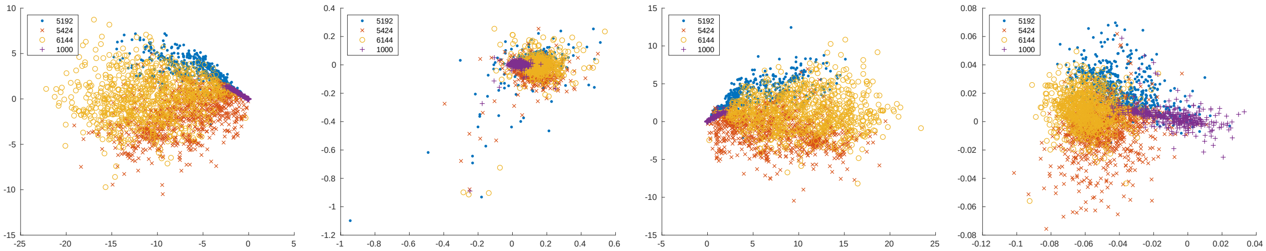
[10] D. Angluin, P. Laird, Learning from noisy examples, Machine Learning 2 (4) (1988) 343–370. doi:10.1023/A:1022873112823.

[11] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, IEEE Transactions on pattern analysis and machine intelligence 38 (3) (2016) 447–461.

[12] A. E. W. Johnson, M. M. Ghassemi, S. Nemati, K. E. Niehaus, D. A. Clifton, G. D. Clifford, Machine learning and decision support in critical care, Proceedings of the IEEE 104 (2) (2016) 444–466. doi:10.1109/JPROC.2015.2501978.

[13] Y. Halpern, S. Horng, Y. Choi, D. Sontag, Electronic medical record phenotyping using the anchor and learn framework, Journal of the American Medical Informatics Association 23 (4) (2016) 731–740. doi:10.1093/jamia/ocw011.

[14] K. Ø. Mikalsen, C. Soguero-Ruiz, K. Jensen, K. Hindberg, M. Gran, A. Revhaug, R.-O. Lindsetmo, S. O. Skrøvseth, F. Godtliebsen,



(a) (b) (c) (d)

Figure 4: Plot two-dimensional embeddings of the chronic patients obtained using four different DR methods: (a) MDDMp. (b) wMLDAb (c) NMLSDR (d) SSMLDR. The different colors and markers represent the true CRG-labels of the patients.

R. Jenssen, Using anchors from free text in electronic health records to diagnose postoperative delirium, Computer Methods and Programs in Biomedicine 152 (2017) 105–114. doi:10.1016/j.cmpb.2017.09.014.

[15] V. Agarwal, T. Podchiyska, J. M. Banda, V. Goel, T. I. Leung, E. P. Minty, T. E. Sweeney, E. Gyang, N. H. Shah, Learning statistical models of phenotypes using noisy labeled training data, Journal of the American Medical Informatics Association 23 (6) (2016) 1166–1173. doi:10.1093/jamia/ocw028.

[16] A. Callahan, N. H. Shah, Chapter 19 - Machine learning in healthcare, in: A. Sheikh, K. M. Cresswell, A. Wright, D. W. Bates (Eds.), Key Advances in Clinical Informatics, Academic Press, 2017, pp. 279 – 291. doi:10.1016/B978-0-12-809523-2.00019-4.

[17] J. M. Banda, M. Seneviratne, T. Hernandez-Boussard, N. H. Shah, Advances in electronic phenotyping: From rule-based definitions to machine learning models, Annual Review of Biomedical Data Science 1 (1) (2018) 53–68. doi:10.1146/annurev-biodatasci-080917-013315.

[18] N. D. Lawrence, B. Schölkopf, Estimating a kernel fisher discriminant in the presence of label noise, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 306–313.

[19] D. R. Wilson, T. R. Martinez, Reduction techniques for instance-based learning algorithms, Machine learning 38 (3) (2000) 257–286.

[20] P. M. Long, R. A. Servedio, Random classification noise defeats all convex potential boosters, Machine learning 78 (3) (2010) 287–304.

[21] R. A. McDonald, D. J. Hand, I. A. Eckley, An empirical comparison of three boosting algorithms on real data sets with artificial class noise, in: International Workshop on Multiple Classifier Systems, Springer, 2003, pp. 35–44.

[22] T. Bylander, Learning linear threshold functions in the presence of classification noise, in: Proceedings of the Seventh Annual Conference on Computational Learning Theory, COLT '94, ACM, New York, NY, USA, 1994, pp. 340–347. doi:10.1145/180139.181176.

[23] K. Crammer, A. Kulesza, M. Dredze, Adaptive regularization of weight vectors, in: Advances in neural information processing systems, 2009, pp. 414–422.

[24] B. Biggio, B. Nelson, P. Laskov, Support vector machines under adversarial label noise, in: Asian Conference on Machine Learning, 2011, pp. 97–112.

[25] A. Vahdat, Toward robustness against label noise in training deep discriminative neural networks, in: Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5596–5605.

[26] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, L. Qu, Making deep neural networks robust to label noise: A loss correction approach, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[27] O. Chapelle, B. Scholkopf, E. A. Zien, Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews], IEEE Transactions on Neural Networks 20 (3) (2009) 542–542. doi:10.1109/TNN.2009.2015974.

[28] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th Edition, Academic Press, Inc., Orlando, FL, USA, 2008.

[29] C. H. Lee, H.-J. Yoon, Medical big data: promise and challenges, Kidney research and clinical practice 36 (1) (2017) 3.

[30] P. B. Jensen, L. J. Jensen, S. Brunak, Mining electronic health records: towards better research applications and clinical care, Nature Reviews Genetics 13 (6) (2012) 395–405.

[31] J. Andreu-Perez, C. C. Y. Poon, R. D. Merrifield, S. T. C. Wong, G. Yang, Big data for health, IEEE Journal of Biomedical and Health Informatics 19 (4) (2015) 1193–1208. doi:10.1109/JBHI.2015.2450362.

[32] R. Miotto, F. Wang, S. Wang, X. Jiang, J. T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Briefings in Bioinformaticsdoi:10.1093/bib/bbx044.

[33] X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, Technical Report CMU-CALD-02-107), Carnegie Mellon University.

[34] X. Zhu, Z. Ghahramani, J. D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the 20th International conference on Machine learning (ICML-03), 2003, pp. 912–919.

[35] Z. Yang, W. W. Cohen, R. Salakhutdinov, Revisiting semi-supervised learning with graph embeddings, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48, JMLR. org, 2016, pp. 40–48.

[36] M. Belkin, P. Niyogi, Using manifold stucture for partially labeled classification, in: Advances in neural information processing systems, 2003, pp. 953–960.

[37] A. Sandryhaila, J. M. F. Moura, Discrete signal processing on graphs, IEEE Transactions on Signal Processing 61 (7) (2013) 1644–1656. doi:10.1109/TSP.2013.2238935.

[38] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Advances in neural information processing systems, 2004, pp. 321–328.

[39] M. Fan, X. Zhang, L. Du, L. Chen, D. Tao, Semi-supervised learning through label propagation on geodesics, IEEE transactions on cybernetics.

[40] Y. Zhang, Z.-H. Zhou, Multilabel dimensionality reduction via dependence maximization, ACM Transactions on Knowledge Discovery from Data 4 (3) (2010) 14:1–14:21.

[41] J. Xu, J. Liu, J. Yin, C. Sun, A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously, Knowledge-Based Systems 98 (2016) 172–184. doi:10.1016/j.knosys.2016.01.032.

[42] A. Gretton, O. Bousquet, A. Smola, B. Schölkopf, Measuring statistical dependence with Hilbert-Schmidt norms, in: International conference on algorithmic learning theory, Springer, 2005, pp. 63–77.

[43] M.-L. Zhang, Z.-H. Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038 – 2048. doi:10.1016/j.patcog.2006.12.019.

[44] B. Guo, C. Hou, F. Nie, D. Yi, Semi-supervised multi-label dimensionality reduction, in: Data Mining (ICDM), 2016 IEEE 16th International Conference on, IEEE, 2016, pp. 919–924.

[45] Y. Yu, J. Wang, Q. Tan, L. Jia, G. Yu, Semi-supervised multi-label dimensionality reduction based on dependence maximization, IEEE Access 5 (2017) 21927–21940. doi:10.1109/ACCESS.2017.2760141.

[46] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of machine learning research 3 (Mar) (2003) 1157–1182.

[47] I. Jolliffe, Principal component analysis, in: International encyclopedia of statistical science, Springer, 2011, pp. 1094–1096.

[48] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally

linear embedding, science 290 (5500) (2000) 2323–2326.

[49] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Advances in neural information processing systems, 2002, pp. 585–591.

[50] J. B. Tenenbaum, V. De Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, science 290 (5500) (2000) 2319–2323.

[51] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of eugenics 7 (2) (1936) 179–188.

[52] C. H. Park, M. Lee, On applying linear discriminant analysis for multi-labeled problems, Pattern recognition letters 29 (7) (2008) 878–887.

[53] W. Chen, J. Yan, B. Zhang, Z. Chen, Q. Yang, Document transformation for multi-label feature selection in text categorization, in: 7th IEEE International Conference on Data Mining, 2007, pp. 451–456.

[54] H. Wang, C. Ding, H. Huang, Multi-label linear discriminant analysis, in: European Conference on Computer Vision, Springer, 2010, pp. 126–139.

[55] X. Lin, X.-W. Chen, Mr. kNN: soft relevance for multi-label classification, in: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM, 2010, pp. 349–358.

[56] J. Xu, A weighted linear discriminant analysis framework for multi-label feature extraction, Neurocomputing 275 (2018) 107–120. doi:10.1016/j.neucom.2017.05.008.

[57] D. R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural computation 16 (12) (2004) 2639–2664.

[58] K. Yu, S. Yu, V. Tresp, Multi-label informed latent semantic indexing, in: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2005, pp. 258–265.

[59] S. Ji, L. Tang, S. Yu, J. Ye, A shared-subspace learning framework for multi-label classification, ACM Transactions on Knowledge Discovery from Data (TKDD) 4 (2) (2010) 8.

[60] B. Qian, I. Davidson, Semi-supervised dimension reduction for multi-label classification, in: Proc. AAAI Conf. Artif. Intell., Vol. 10, 2010, pp. 569–574.

[61] M. Gönen, Coupled dimensionality reduction and classification for supervised and semi-supervised multilabel learning, Pattern Recognition Letters 38 (2014) 132–141. doi:https://doi.org/10.1016/j.patrec.2013.11.021.

[62] T. Yu, W. Zhang, Semisupervised multilabel learning with joint dimensionality reduction, IEEE Signal Process. Lett. 23 (6) (2016) 795–799.

[63] M. B. Blaschko, J. A. Shelton, A. Bartels, C. H. Lampert, A. Gretton, Semi-supervised kernel canonical correlation analysis with application to human fMRI, Pattern Recognition Letters 32 (11) (2011) 1572–1583.

[64] H. Li, P. Li, Y.-j. Guo, M. Wu, Multi-label dimensionality reduction based on semi-supervised discriminant analysis, Journal of Central South University of Technology 17 (6) (2010) 1310–1319.

[65] Y. Yu, G. Yu, X. Chen, Y. Ren, Semi-supervised multi-label linear discriminant analysis, in: International Conference on Neural Information Processing, Springer, 2017, pp. 688–698.

[66] M. Hubert, K. V. Driessen, Fast and robust discriminant analysis, Computational Statistics & Data Analysis 45 (2) (2004) 301–320. doi:10.1016/S0167-9473(02)00299-2.

[67] C. Croux, C. Dehon, Robust linear discriminant analysis using S-estimators, Canadian Journal of Statistics 29 (3) (2001) 473–493.

[68] M. Hubert, P. J. Rousseeuw, S. Van Aelst, High-breakdown robust multivariate methods, Statistical science (2008) 92–119.

[69] F. Nie, S. Xiang, Y. Liu, C. Zhang, A general graph-based semi-supervised learning with novel class discovery, Neural Computing and Applications 19 (4) (2010) 549–555.

[70] C. D. Meyer, Jr, R. J. Plemmons, Convergent powers of a matrix with applications to iterative methods for singular linear systems, SIAM Journal on Numerical Analysis 14 (4) (1977) 699–705.

[71] Y. Saad, Chapter 1 - Background in matrix theory and linear algebra, in: Numerical Methods for Large Eigenvalue Problems, Manchester University Press, 1992, pp. 1–27. doi:10.1137/1.9781611970739.ch1.

[72] X.-Z. Wu, Z.-H. Zhou, A unified view of multi-label performance measures, arXiv preprint arXiv:1609.00288.

[73] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (Jan) (2006) 1–30.

[74] A. Soni, E. Mitchell, Expenditures for commonly treated conditions among adults age 18 and older in the U.S. civilian noninstitutionalized population, 2013, Statistical Brief.

[75] A. Calderón-Larrañaga, D. L. Vetrano, G. Onder, L. A. Gimeno-Feliu, C. Coscollar-Santaliestra, A. Carfí, M. S. Pisciotta, S. Angleman, R. J. Melis, G. Santoni, et al., Assessing and measuring chronic multimorbidity in the older population: a proposal for its operationalization, Journals of Gerontology Series A: Biomedical Sciences and Medical Sciences 72 (10) (2016) 1417–1423.

[76] American Diabetes Association, Economic costs of diabetes in the US in 2017, Diabetes Care 41 (5) (2018) 917–928.

[77] Centers for Disease Control and Prevention, International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) (2011).

[78] WHO, Collaborating Centre for Drug Statistics Methodology, Guidelines for ATC classification and DDD assignment, 2016.

[79] C. Soguero-Ruiz, A. A. Díaz-Plaza, P. de Miguel Bohoyo, J. Ramos-López, M. Rubio-Sánchez, A. Sánchez, I. Mora-Jiménez, On the use of decision trees based on diagnosis and drug codes for analyzing chronic patients, in: International Conference on Bioinformatics and Biomedical Engineering, Springer, 2018, pp. 135–148.

[80] A. Sanchez, C. Soguero-Ruiz, I. Mora-Jiménez, F. Rivas-Flores, D. Lehmann, M. Rubio-Sánchez, Scaled radial axes for interactive visual feature selection: A case study for analyzing chronic conditions, Expert Systems with Applications 100 (2018) 182–196.

[81] R. F. Averill, N. Goldfield, J. Eisenhandler, J. Hughes, B. Shafir, D. Gannon, L. Gregg, F. Bagadia, B. Steinbeck, N. Ranade, et al., Development and evaluation of Clinical Risk Groups (CRGs), Wallingford, CT: 3M Health Information Systems, 1999.

Chapter 11

Paper III

Time series cluster kernels to exploit informative missingness and incomplete label information

Karl Øyvind Mikalsen[a,b,*], Cristina Soguero-Ruiz[b,c], Filippo Maria Bianchi[d,b], Arthur Revhaug[e,f,g], Robert Jenssen[d,b]

[a] *Dept. of Mathematics and Statistics, UiT The Arctic University of Norway, Tromsø, Norway*
[b] *UiT Machine Learning Group*
[c] *Dept. of Signal Theory and Comm., Telematics and Computing, Universidad Rey Juan Carlos, Fuenlabrada, Spain*
[d] *Dept. of Physics and Technology, UiT, Tromsø, Norway*
[e] *Dept. of Gastrointestinal Surgery, University Hospital of North Norway (UNN), Tromsø, Norway*
[f] *Clinic for Surgery, Cancer and Women's Health, UNN, Tromsø, Norway*
[g] *Institute of Clinical Medicine, UiT, Tromsø, Norway*

Abstract

The time series cluster kernel (TCK) provides a powerful tool for analysing multivariate time series subject to missing data. TCK is designed using an ensemble learning approach in which Bayesian mixture models form the base models. Because of the Bayesian approach, TCK can naturally deal with missing values without resorting to imputation and the ensemble strategy ensures robustness to hyperparameters, making it particularly well suited for unsupervised learning.

However, TCK assumes missing at random and that the underlying missingness mechanism is ignorable, i.e. uninformative, an assumption that does not hold in many real-world applications, such as e.g. medicine. To overcome this limitation, we present a kernel capable of exploiting the potentially rich information in the missing values and patterns, as well as the information from the observed data. In our approach, we create a representation of the missing pattern, which is incorporated into mixed mode mixture models in such a way that the information provided by the missing patterns is effectively exploited. Moreover, we also propose a semi-supervised kernel, capable of taking advantage of incomplete label information to learn more accurate similarities.

Experiments on benchmark data, as well as a real-world case study of patients described by longitudinal electronic health record data who potentially suffer from hospital-acquired infections, demonstrate the effectiveness of the proposed methods.

Keywords: Multivariate time series, Kernel methods, Missing data, Informative missingness, Semi-supervised learning

1. Introduction

Multivariate time series (MTS) frequently occur in a whole range of practical applications such as medicine, biology, and climate studies, to name a few. A challenge that complicates the analysis is that real-world MTS are often subject to large amounts of missing data. Traditionally, missingness mechanisms have been categorized into missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR) [1]. The main difference between these mechanisms consists in whether the missingness is ignorable (MCAR and MAR) or non-ignorable (MNAR) [1, 2, 3]. In e.g. medicine, non-ignorable missingness can occur when the missing patterns R are related to the disease under study Y. In this case, the distribution of the missing patterns for diseased patients is not equal to the corresponding distribution for the control group, i.e. $p(R \mid Y = 1) \neq p(R \mid Y = 0)$. Hence, the missingness is *informative* [4, 5, 6]. By contrast, uninformative missingness will be referred to as *ignorable* in the remainder of this paper.

Both ignorable and informative missingness occur in real-world data. An example from medicine of ignorable missingness occurs e.g. if a clinician orders lab tests for a patient and the tests are performed, but because of an error the results are not recorded. On the other hand, informative missingness could occur if it is decided to not perform lab tests because the doctor thinks the patient is in good shape. In the latter case, the missing values and patterns potentially contain rich information about the diseases and clinical outcomes for the patient. Efficient data-driven approaches aiming to extract knowledge, perform predictive modeling, etc., must be capable of capturing this information.

Various methods have been proposed to handle missing data in MTS [7, 8, 9]. One simple approach is to create a *complete* dataset by discarding the time series with missing data. However, this gives unbiased predictions only if the missingness mechanism is MCAR. As an alternative, a preprocessing step involving *imputation* of missing values with some estimated value, such as the mean, is common. Other so-called *single imputation* methods exploit machine learning based methods such as multilayer perceptrons, self-organizing maps, k-nearest neighbors, recurrent neural networks and regression-based imputation [10, 11]. Alternatively, one can impute missing values using various smoothing and interpolation techniques [12, 10]. Among these, a prominent example is the last observation carried forward (LOCF) scheme that imputes the last non-missing

[*] Corresponding author at: Department of Mathematics and Statistics, Faculty of Science and Technology, UiT – The Arctic University of Norway, N-9037 Tromsø, Norway

value for the following missing values. Limitations of imputation methods are that they introduce additional bias and they ignore uncertainty associated with the missing values.

Multiple imputation [13] resolves this problem, to some extent, by estimating the missing values multiple times and thereby creating multiple complete datasets. Thereafter, e.g. a classifier is trained on all datasets and the results are combined to obtain the final predictions. However, despite that multiple imputation and other imputation methods can give satisfying results in some scenarios, these are ad-hoc solutions that lead to a multi-step procedure in which the missing data are handled separately and independently from the rest of the analysis. Moreover, the information about which values are actually missing (the missing patterns) is lost, i.e. imputation methods cannot exploit informative missingness.

Due to the aforementioned limitations, several research efforts have been devoted over the last years to process incomplete time series without relying on imputation [6, 14, 15, 16, 17, 18, 19]. In this regard, powerful kernel methods have been proposed, of which the recently proposed *time series cluster kernel* (TCK) [20] is a prominent example. The TCK is designed using an ensemble learning approach in which Bayesian mixture models form the base models. An advantage of TCK, compared to imputation methods, is that the missing data are handled automatically and no additional tasks are left to the user. Multiple imputation instead requires a careful selection of the imputation model and other variables are needed to do the imputation [7], which particularly in an unsupervised setting can turn out to be problematic.

A shortcoming of the TCK is that unbiased predictions are only guaranteed for ignorable missingness, i.e. the kernel cannot take advantage of informative missing patterns frequently occurring in medical applications. To overcome this limitation, in this work, we present a novel time series cluster kernel, TCK$_{IM}$. In our approach, we create a representation of the missing patterns using masking, i.e. we represent the missing patterns using binary indicator time series. By doing so, we obtain MTS consisting of both continuous and discrete attributes. To model these time series, we introduce mixed mode Bayesian mixture models, which can effectively exploit information provided by the missing patterns.

The time series cluster kernels are particularly useful in unsupervised settings. In many practical applications such as e.g. medicine it is not feasible to obtain completely labeled training sets [21], but in some cases it is possible to annotate a few samples with labels, i.e. incomplete label information is available. In order to exploit the incomplete label information, we propose a semi-supervised MTS kernel, ssTCK. In our approach, we incorporate ideas from information theory to measure similarities between distributions. More specifically, we employ the Kullback-Leibler divergence to assign labels to unlabeled data.

Experiments on benchmark MTS datasets and a real-world case study of patients suffering from hospital-acquired infections, described by longitudinal electronic health record data, demonstrate the effectiveness of the proposed TCK$_{IM}$ and ssTCK kernels.

The remainder of this paper is organized as follows. Section 2 presents background on MTS kernels. The two proposed kernels are described in Section 3 and 4, respectively. Experiments on synthetic and benchmark datasets are presented in Section 5, whereas the case study is described in Section 6. Section 7 concludes the paper.

2. Multivariate time series kernels to handle missing data

Kernel methods have been of great importance in machine learning for several decades and have applications in many different fields [22, 23, 24]. Within the context of time series, a *kernel* is a similarity measure that also is positive semi-definite [25]. Once defined, such similarities between pairs of time series may be utilized in a wide range of applications, such as classification or clustering, benefiting from the vast body of work in the field of kernel methods. Here we provide an overview of MTS kernels, and describe how they deal with missing data.

The simplest of all kernel functions is the linear kernel, which for two data points represented as vectors, x and y, is given by the inner product $\langle x, y \rangle$, possibly plus a constant c. One can also apply a linear kernel to pairs of MTS once they are unfolded into vectors. However, by doing so the information that they are MTS and there might be inherent dependencies in time and between attributes, is then lost. Nevertheless, in some cases such a kernel can be efficient, especially if the MTS are short [26]. If the MTS contain missing data, the linear kernel requires a preprocessing step involving e.g. imputation.

The most widely used time series similarity measure is *dynamic time warping* (DTW) [27], where the similarity is quantified as the alignment cost between the MTS. More specifically, in DTW the time dimension of one or both of the time series is warped to achieve a better alignment. Despite the success of DTW in many applications, similarly to many other similarity measures, it is non-metric and therefore cannot non-trivially be used to design a positive semi-definite kernel [28]. Hence, it is not suited for kernel methods in its original formulation. However, because of its popularity there have been attempts to design kernels exploiting the DTW. For example, Cuturi et al. designed a DTW-based kernel using global alignments [29]. An efficient version of the global alignment kernel (GAK) is provided in [30]. The latter has two hyperparameters, namely the kernel bandwidth and the triangular parameter. GAK does not naturally deal with missing data and incomplete datasets, and therefore also requires a preprocessing step involving imputation.

Two MTS kernels that can naturally deal with missing data without having to resort to imputation are the *learned pattern similarity* (LPS) [31] and TCK. LPS generalizes the well-known autoregressive modelsto local autopatterns using multiple lag values for autocorrelation. These autopatterns are supposed to capture the local dependency structure in the time series and are learned using a tree-based (random forest) learning strategy. More specifically, a time series is represented as a matrix of segments. Randomness is injected to the learning process by randomly choosing time segment (column in the matrix) and lag p for each tree in the random forest. A bag-of-words type compressed representation is created from the output of the leaf-nodes for each tree. The final time series representation is created by concatenating the representation obtained from the individual trees, which in turn are used to compute the similarity using a histogram intersection kernel [32].

The TCK is based on an ensemble learning approach wherein robustness to hyperparameters is ensured by joining the clustering results of many Gaussian mixture models (GMM) to form the final kernel. Hence, no critical hyperparameters have to be tuned by the user, and the TCK can be learned in an unsupervised manner. To ensure robustness to sparsely sampled data, the GMMs that are the base models in the ensemble, are extended using informative prior distributions such that the missing data is explicitly dealt with. More specifically, the TCK matrix is built by fitting GMMs to the set of MTS for a range of number of mixture components. The idea is that by generating partitions at different resolutions, one can capture both the local and global structure of the data. Moreover, to capture diversity in the data, randomness is injected by for each resolution (number of components) estimating the mixture parameters for a range of random initializations and randomly chosen hyperparameters. In addition, each GMM sees a random subset of attributes and segments in the MTS. The posterior distributions for each mixture component are then used to build the TCK matrix by taking the inner product between all pairs of posterior distributions. Eventually, given an ensemble of GMMs, the TCK is created in an additive way by using the fact that the sum of kernels is also a kernel.

Despite that LPS and TCK kernels share many properties, the way missing data are dealt with is very different. In LPS, the missing data handling abilities of decision trees are exploited. Along with ensemble methods, fuzzy approaches and support vector solutions, decision trees can be categorized as *machine learning approaches for handling missing data* [10], i.e. the missing data are handled naturally by the machine learning algorithm. One can also argue that the way missing data are dealt with in the TCK belongs to this category, since an ensemble approach is exploited. However, it can also be categorized as a *likelihood-based approach* since the underlying models in the ensemble are Gaussian mixture models. In the likelihood-based approaches, the full, incomplete dataset is analysed using maximum likelihood (or maximum a posteriori, equivalently), typically in combination with the expectation-maximization (EM) algorithm [7, 9]. These approaches assume that the missingness is ignorable.

3. Time series cluster kernel to exploit informative missingness

In this section, we present the novel time series cluster kernel, TCK$_{IM}$, which is capable of exploiting informative missingness.

A key component in the time series cluster kernel framework is ensemble learning, in which the basic idea consists in combining a collection of many base models into a composite model. A good such composite model will have statistical, computational and representational advantages such as lower variance, lower sensitivity to local optima and is capable of representing a broader span functions (increased expressiveness), respectively, compared to the individual base models [33]. Key to achieve this is *diversity* and *accuracy* [34], i.e. the base models cannot make the same errors on new test data and have to perform better than random guessing. This can be done by integrating multiple outcomes of the same (weak) base model as it is trained under different, often randomly chosen, settings (parameters, initialization, subsampling, etc.) to ensure diversity [35].

In the TCK$_{IM}$ kernel, the base model is a mixed mode Bayesian mixture model. Next, we provide the details of this model.

Notation

The following notation is used. A multivariate time series (MTS) X is defined as a (finite) combination of univariate time series (UTS), $X = \{x_v \in \mathbb{R}^T \mid v = 1, 2, \ldots, V\}$, where each attribute, x_v, is a UTS of length T. The number of UTS, V, is the *dimension* of X. The length T of the UTS x_v is also the length of the MTS X. Hence, a V–dimensional MTS, X, of length T can be represented as a matrix in $\mathbb{R}^{V\times T}$. Given a dataset of N MTS, we denote $X^{(n)}$ the n-th MTS. An incompletely observed MTS is described by the pair $U^{(n)} = (X^{(n)}, R^{(n)})$, where $R^{(n)}$ is a binary MTS with entry $r_v^{(n)}(t) = 0$ if the realization $x_v^{(n)}(t)$ is missing and $r_v^{(n)}(t) = 1$ if it is observed.

Mixed mode mixture model

Assume that a MTS $U = (X, R)$ is generated from two modes. X is a V-variate real-valued MTS ($X \in \mathbb{R}^{V\times T}$), whereas R is a V-variate binary MTS ($R \in \{0, 1\}^{V\times T}$). Further, we assume that U is generated from a finite mixture density,

$$p(U \mid \Phi, \Theta) = \sum_{g=1}^{G} \theta_g f(U \mid \phi_g), \tag{1}$$

where G is the number of components, f is the density of the components parametrized by $\Phi = (\phi_1, \ldots, \phi_G)$, and $\Theta = (\theta_1, \ldots, \theta_g)$ are the mixing coefficients, $0 \leq \theta_G \leq 1$ and $\sum_{g=1}^{G} \theta_g = 1$.

Now, introduce a latent random variable Z, represented as a G-dimensional one-hot vector $Z = (Z_1, \ldots, Z_G)$, whose marginal distribution is given by $p(Z \mid \Theta) = \prod_{g=1}^{G} \theta_g^{Z_g}$. The unobserved variable Z records the membership of U and therefore $Z_g = 1$ if U belongs to component g and $Z_g = 0$ otherwise. Hence, $p(U \mid Z,\ \Phi) = \prod_{g=1}^{G} f(U \mid \phi_g)^{Z_g}$, and therefore it follows that

$$p(U, Z \mid \Phi, \Theta) = p(U \mid Z,\ \Phi)p(Z \mid \Theta) = \prod_{g=1}^{G} \left[f(U \mid \phi_g)\theta_g \right]^{Z_g} \tag{2}$$

$U = (X, R)$ consists of two modalities X and R. We now naively assume that

$$f(U \mid \phi_g) = f(X \mid R, \mu_g, \Sigma_g) f(R \mid \beta_g), \tag{3}$$

where $f(X \mid R, \mu_g, \Sigma_g)$ is a density function given by

$$f(X \mid R, \mu_g, \Sigma_g) = \prod_{v=1}^{V} \prod_{t=1}^{T} \mathcal{N}(x_v(t) \mid \mu_{gv}(t), \sigma_{gv})^{r_v(t)}, \quad (4)$$

and $f(R \mid \beta_g)$ is a probability mass given by

$$f(R \mid \beta_g) = \prod_{v=1}^{V} \prod_{t=1}^{T} \beta_{gvt}^{r_v(t)} (1 - \beta_{gvt})^{1-r_v(t)}. \quad (5)$$

The parameters of each component are $\phi_g = (\mu_g, \Sigma_g, \beta_g)$, where $\mu_g = \{\mu_{gv} \in \mathbb{R}^T \mid v = 1, ..., V\}$ is a time-dependent mean (μ_{gv} is a UTS of length T), $\Sigma_g = diag\{\sigma_{g1}^2, ..., \sigma_{gV}^2\}$ is a time-constant diagonal covariance matrix in which σ_{gv}^2 is the variance of attribute v, and $\beta_{gvt} \in [0, 1]$ are the parameters of the Bernoulli mixture model (5). The idea is that even though the missingness mechanism is ignored in $f(X \mid R, \mu_g, \Sigma_g)$, which is only computed over the observed data, the Bernoulli term $f(R \mid \beta_g)$ will capture information from the missing patterns.

The conditional probability of Z given U, can be found using Bayes' theorem,

$$\pi_g \equiv P(Z_g = 1 \mid U, \Phi, \Theta)$$
$$= \frac{\theta_g \prod\limits_{v=1}^{V} \prod\limits_{t=1}^{T} \Big[\mathcal{N}(x_v(t) \mid \mu_{gv}(t), \sigma_{gv}) \beta_{gvt}\Big]^{r_v(t)} (1 - \beta_{gvt})^{1-r_v(t)}}{\sum\limits_{g=1}^{G} \theta_g \prod\limits_{v=1}^{V} \prod\limits_{t=1}^{T} \Big[\mathcal{N}(x_v(t) \mid \mu_{gv}(t), \sigma_{gv}) \beta_{gvt}\Big]^{r_v(t)} (1 - \beta_{gvt})^{1-r_v(t)}}. \quad (6)$$

Similarly to [20], we introduce a Bayesian extension and put informative priors over the parameters of the normal distribution, which enforces smoothness over time and that clusters containing few time series, to have parameters similar to the mean and covariance computed over the whole dataset. A kernel-based Gaussian prior is defined for the mean, $P(\mu_{gv}) = \mathcal{N}\left(\mu_{gv} \mid m_v, \, S_v\right)$. m_v are the empirical means and the prior covariance matrices, S_v, are defined as $S_v = s_v \mathcal{K}$, where s_v are empirical standard deviations and $\mathcal{K}$ is a kernel matrix, whose elements are $\mathcal{K}_{tt'} = b_0 \exp(-a_0(t - t')^2)$, $\quad t, t' = 1, \ldots, T$. a_0, b_0 are user-defined hyperparameters. An inverse Gamma distribution prior is put on the standard deviation σ_{gv}, $P(\sigma_{gv}) \propto \sigma_{gv}^{-N_0} \exp\left(-\frac{N_0 s_v}{2\sigma_{gv}^2}\right)$, where N_0 is a user-defined hyperparameter. We denote $\Omega = \{a_0, b_0, N_0\}$ the set of hyperparameters.

Then, given a dataset $\{U^{(n)}\}_{n=1}^N$, the parameters $\{\Phi, \Theta\}$ can be estimated using maximum a posteriori expectation maximization (MAP-EM) [36, 37]. This leads to Algorithm 1.

3.1. Forming the kernel

We now explain how the mixed mode mixture model is used to form the TCK$_{IM}$ kernel.

We use the mixed mode Bayesian mixture model as the base model in an ensemble approach. To ensure diversity, we vary the number of components for the base models by sampling from a set of integers $\mathcal{I}_C = \{I, \ldots, I + C\}$. For each number of components, we apply Q different random initial conditions and hyperparameters. We let $\mathcal{Q} = \{q = (q_1, q_2) \mid q_1 = 1, \ldots Q, \, q_2 \in \mathcal{I}_C\}$ be the index set keeping track of initial conditions and hyperparameters (q_1), and the number of components (q_2). Each base model q is trained on a random subset of MTS $\{(X^{(n)}, R^{(n)})\}_{n \in \eta(q)}$. Moreover, for each q, we select random subsets of variables $\mathcal{V}(q)$ as well as random time segments $\mathcal{T}(q)$.

Algorithm 1 MAP-EM for mixed mode mixture model

```
Require: Dataset $\{U^{(n)} = (X^{(n)}, R^{(n)})\}_{n=1}^N$, hyperparameters $\Omega$ and
    number of mixtures $G$.
1: Initialize the parameters $\Theta = (\theta_1, \ldots, \theta_G)$ and $\Phi = \{\mu_g, \sigma_g, \beta_g\}_{g=1}^G$.
2: E-step. For each MTS $U^{(n)}$, evaluate the posterior probabilities
    using Eq. (6) with the current parameter estimates.
3: M-step. Update parameters using the current posteriors
    $\theta_g = N^{-1} \sum_{n=1}^N \pi_g^{(n)}$
    $\sigma_{gv}^2 = \frac{N_0 s_v^2 + \sum_{n=1}^N \sum_{t=1}^T r_v^{(n)}(t) \, \pi_g^{(n)} (x_v^{(n)}(t) - \mu_{gv}(t))^2}{N_0 + \sum_{n=1}^N \sum_{t=1}^T r_v^{(n)}(t) \, \pi_g^{(n)}}$
    $\mu_{gv} = \frac{S_v^{-1} m_v + \sigma_{gv}^{-2} \sum_{n=1}^N \pi_g^{(n)} \mathrm{diag}(r_v^{(n)}) \, x_v^{(n)}}{S_v^{-1} + \sigma_{gv}^{-2} \sum_{n=1}^N \pi_g^{(n)} \mathrm{diag}(r_v^{(n)})}$
    $\beta_{gvt} = (\sum_{n=1}^N \pi_g^{(n)})^{-1} \sum_{n=1}^N \pi_g^{(n)} r_v^{(n)}(t)$
4: Repeat step 2-3 until convergence.
Ensure: Posteriors $\Pi^{(n)} \equiv \left(\pi_1^{(n)}, \ldots, \pi_G^{(n)}\right)^T$ and parameter estimates $\Theta$
    and $\Phi$.
```

The inner products of the normalized posterior distributions from each mixture component are then added up to build the TCK$_{IM}$ kernel matrix. Note that, in addition to introducing novel base models to account for informative missingness, we also modify the kernel by normalizing the vectors of posteriors to have unit length in the l_2-norm. This provides an additional regularization that may increase the generalization capability of the learned model. The details of the method are presented in Algorithm 2. The kernel for MTS not available during training can be evaluated according to Algorithm 3.

4. Semi-supervised time series cluster kernel

This section presents a semi-supervised MTS kernel, ssTCK, capable of exploiting incomplete label information. In ssTCK, the base mixture models are learned exactly in the same way as in TCK or TCK$_{IM}$. I.e. if there is no missing data, or the missingness is ignorable, the base models will be the Bayesian GMMs. Conversely, if the missingness is informative, the base models are the mixed mode Bayesian mixture models presented in the previous section. Both approaches will associate each MTS $X^{(n)}$ with a q_2-dimensional posterior $\Pi^{(n)} \equiv \left(\pi_1^{(n)}, \ldots, \pi_{q_2}^{(n)}\right)^T$, where $\pi_g^{(n)}$ represents the probability that the MTS belongs to component g and q_2 is the total number of components in the base mixture model.

In ssTCK, label information is incorporated in an intermediate processing step in which the posteriors $\Pi^{(n)}$ are transformed, before the transformed posteriors are sent into Algorithm 2 or 3. More precisely, the transformation consists in mapping the posterior for the mixture components to a class "posterior" (probability), i.e. we seek to find a function $\mathcal{M} : \; [0, 1]^{q_2} \to [0, 1]^{N_c}$,

Algorithm 2 Time series cluster kernel. Training phase.

```
Require: Training set of MTS $\{(X^{(n)}, R^{(n)})\}_{n=1}^{N}$ , $Q$ initializations, set of integers $\mathcal{I}_C$ controlling number of components for each base model.
Initialize kernel matrix $K = 0_{N \times N}$.
for $q \in Q$ do
    Compute posteriors $\Pi^{(n)}(q) \equiv (\pi_1^{(n)}, \dots, \pi_{q_2}^{(n)})^T$, by fitting a mixed mode mixture model with $q_2$ clusters to the dataset and by randomly selecting:
        i. hyperparameters $\Omega(q)$,
        ii. a time segment $\mathcal{T}(q)$ of length $T_{min} \leq |\mathcal{T}(q)| \leq T_{max}$ to extract from each $X^{(n)}$ and $R^{(n)}$,
        iv. a subset of attributes $\mathcal{V}(q)$, with cardinality $V_{min} \leq |\mathcal{V}(q)| \leq V_{max}$, to extract from each $X^{(n)}$ and $R^{(n)}$,
        vi. a subset of MTS, $\eta(q)$, with $N_{min} \leq |\eta(q)| \leq N$,
        vii. initialization of the mixture parameters $\Theta(q)$ and $\Phi(q)$.
    Update kernel matrix, $K_{nm} = K_{nm} + \frac{\Pi^{(n)}(q)^T \Pi^{(m)}(q)}{\|\Pi^{(n)}(q)\| \cdot \|\Pi^{(m)}(q)\|}$.
end for
Ensure: $K$ kernel matrix, time segments $\mathcal{T}(q)$, subsets of attributes $\mathcal{V}(q)$, subsets of MTS $\eta(q)$, parameters $\Theta(q)$, $\Phi(q)$ and posteriors $\Pi^{(n)}(q)$.
```

Algorithm 3 Time series cluster kernel. Test phase.

```
Require: Test set $\{X^{*(m)}\}_{m=1}^{M}$, time segments $\mathcal{T}(q)$ subsets of attributes $\mathcal{V}(q)$, $\mathcal{V}_R(q)$, subsets of MTS $\eta(q)$, parameters $\Theta(q)$, $\Phi(q)$ and posteriors $\Pi^{(n)}(q)$.
Initialize kernel matrix $K^* = 0_{N \times M}$.
for $q \in Q$ do
    Compute posteriors $\Pi^{*(m)}(q)$, $m = 1, \dots, M$ using the mixture parameters $\Theta(q)$, $\Phi(q)$.
    Update kernel matrix, $K^*_{nm} = K^*_{nm} + \frac{\Pi^{(n)}(q)^T \Pi^{*(m)}(q)}{\|\Pi^{(n)}(q)\| \cdot \|\Pi^{*(m)}(q)\|}$.
end for
Ensure: $K^*$ test kernel matrix.
```

$\Pi^{(n)} \xrightarrow{\mathcal{M}} \tilde{\Pi}^{(n)}$. Hence, we want to exploit the incomplete label information to find a transformation that merges the q_2 components of the mixture model into N_c clusters, where N_c is the number of classes.

The mapping $\mathcal{M}$ can be thought of as a (soft) N_c-class classifier, and hence there could be many possible ways of learning $\mathcal{M}$. However, choosing a too flexible classifier for this purpose leads to an increased risk of overfitting and could also unnecessarily increase the algorithmic complexity. For these reasons, we restrict ourselves to searching for a linear transformation

$$\mathcal{M}(\Pi^{(n)}) = W^T \Pi^{(n)}, \quad W \in [0, 1]^{q_2 \times N_c}. \tag{7}$$

Since the N_c-dimensional output $\tilde{\Pi}^{(n)} = \mathcal{M}(\Pi^{(n)})$ should represent a probability distribution, we add the constraint $\sum_{i=1}^{N_c} W_{ji} = 1$, $j = 1, \dots, q_2$.

A natural first step is to first assume that the label information is complete and look at the corresponding supervised kernel. In the following two subsections, we describe our proposed methods for learning the transformation $\mathcal{M}$ in supervised and semi-supervised settings, respectively.

Algorithm 4 Supervised posterior transformation

```
Require: Posteriors $\{\Pi^{(n)}\}_{n=1}^{N}$ from mixture models consisting of $q_2$ components and labels $\{y^{(n)}\}_{n=1}^{N}$,
for $i = 1, \dots, q_2$, $j = 1, \dots, N_c$ do
    Compute $W_{ij} = \frac{\sum_{n=1}^{N} y_j^{(n)} \pi_i^{(n)}}{\sum_{n=1}^{N} y_j^{(n)}}$.
    $W_{ij} = \frac{W_{ij}}{\sum_{j=1}^{N_c} W_{ij}}$.
end for
Transform training and test posteriors via $\tilde{\Pi} = W^T \Pi$
Ensure: Transformed posteriors $\tilde{\Pi}^{(n)}$
```

4.1. Supervised time series cluster kernel (sTCK)

Supervised setting. Each base mixture model consists of q_2 components, and we assume that the number of components is greater or equal to the number of classes N_c. Further, assume that each MTS $X^{(n)}$ in the training set is associated with a N_c–dimensional one-hot vector $y^{(n)}$, which represents its label. Hence, the labels of the training set can be represented via a matrix $Y \in \{0, 1\}^{N \times N_c}$, where N is the number of MTS in the training set.

We approach this problem by considering one component at the time. For a given component g, the task is to associate it with a class. One natural way to do this is to identify all members of component g and then simply count how many times each label occur. To account for class imbalance, one can then divide each count by the number of MTS in the corresponding class. One possible option would then be to assign the component to the class with the largest normalized count. However, by doing so, one is not accounting for uncertainty/disagreement within the component. Hence, a more elegant alternative is to simply use the normalized counts as the weights in the matrix W. Additionally, one has to account for that each MTS can simultaneously belong to several components, i.e. each MTS $X^{(n)}$ has a only soft membership to the component g, determined by the value $\pi_g^{(n)}$. This can be done using $\Pi^{(n)}$ as weights in the first step. This procedure is summarized in Algorithm 4.

4.2. Semi-supervised time series cluster kernel (ssTCK)

Setting. Assume that the labels $\{y^{(n)}\}_{n=1}^{L}$, $L < N$, are known and $\{y^{(n)}\}_{n=L+1}^{N}$ are unknown.

In this setting, if one naively tries to apply Algorithm 4 based on only the labeled part of the dataset, one ends up dividing by 0s. The reason is that some of the components in the mixture model will contain only unlabeled MTS (the soft label analogy is that the probability that any of the labeled MTS belong to that particular component is zero or very close to zero). Hence, we need a way to assign labels to the components that do not contain any labeled MTS.

Note that each component is described by a probability distribution. A natural measure of dissimilarity between probability distributions is the Kullback-Leibler (KL) divergence [38]. Moreover, since the components are described by parametric distributions, the KL divergence has a simple closed-form expression. The KL divergence between two components, i and j,

Algorithm 5 Semi-supervised posterior transformation

```
Require: Posteriors $\{\Pi^{(n)}\}_{n=1}^{N}$ from mixture models consisting of $q_2$
    components, labels $\{y^{(n)}\}_{n=1}^{L}$, and hyperparameter $h$.
for $i = 1, \dots, q_2$, $j = 1, \dots, N_c$ do
    Compute $W_{ij} = \frac{\sum_{n=1}^{N} y_j^{(n)} \pi_i^{(n)}}{\sum_{n=1}^{N} y_j^{(n)}}$.
end for
for all $k$ s.t. $\sum_{j=1}^{N_c} W_{kj} < h$ do
    Let $\mathcal{L} = \{l \ s.t. \ \sum_{j=1}^{N_c} W_{lj} \geq h\}$
    $W_{kj} = W_{lj}$ where $l = \arg\min_{l \in \mathcal{L}} D_{KL}^{S}(f^{(k)} \parallel f^{(l)})$.
end for
for $i = 1, \dots, q_2$, $j = 1, \dots, N_c$ do
    $W_{ij} = \frac{W_{ij}}{\sum_{j=1}^{N_c} W_{ij}}$.
end for
Transform training or test posterior via $\tilde{\Pi} = W^T \Pi$
Ensure: Transformed posteriors $\tilde{\Pi}^{(n)}$
```

in our Bayesian GMM is given by

$$D_{KL}(f^{(i)} \parallel f^{(j)}) = \frac{1}{2}\Big(\sum_{v=1}^{V} \sum_{t=1}^{T} \sigma_{iv}^{2} \sigma_{jv}^{-2} + \sigma_{jv}^{-2}(\mu_{jv}(t) - \mu_{iv}(t))^2 - 1 + \log(\sigma_{jv}^{2}) - \log(\sigma_{iv}^{2}) \Big), \tag{8}$$

where $f^{(i)} = f(X \mid R, \mu_i, \Sigma_i)$ is the density given in Eq. (4). The KL-divergence can be made symmetric via the transformation

$$D_{KL}^{S}(f^{(i)} \parallel f^{(j)}) = \frac{1}{2}\left(D_{KL}(f^{(i)} \parallel f^{(j)}) + D_{KL}(f^{(j)} \parallel f^{(i)}) \right). \tag{9}$$

The underlying idea in our semi-supervised framework is to learn the transformation W for the clusters with only unlabeled points by finding the nearest cluster (in the D_{KL}^{S}-sense) that contain labeled points. This leads to Algorithm 5.

5. Experiments on synthetic and benchmark datasets

The experiments in this paper consists of two parts. The purpose of the first part was to demonstrate within a controlled environment situations where the proposed TCK$_{IM}$ and ssTCK kernels might prove more useful than the TCK. In the second part (Sec. 6), we present a case study from a real-world medical application in which we compared to several baseline methods.

In the first part, we considered synthetic and benchmark datasets. The following experimental setup was considered. We performed kernel principal component analysis (KPCA) using time series cluster kernels and let the dimensionality of the embedding be 10. Thereafter, we trained a kNN-classifier with $k = 1$ on the embedding and evaluated performance in terms of classification accuracy on an independent test set. We let $Q = 30$ and $\mathcal{I}_C = \{N_c, \dots, N_c + 20\}$. An additional hyperparameter h was introduced for ssTCK. We set h to 10^{-1} in our experiments. We also standardized each attribute to zero mean and unit standard deviation.

Table 1: Accuracy on the synthetic VAR(1) dataset.

	Unsupervised	Semi-supervised	Supervised
TCK	0.826	0.854	0.867
TCK$_{IM}$	0.933	0.967	0.970

5.1. Synthetic example

To illustrate the effectiveness of the proposed methods, we first considered a controlled experiment in which a synthetic MTS dataset with two classes was sampled from a first-order vector autoregressive model,

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} + \begin{pmatrix} \rho_1 & 0 \\ 0 & \rho_2 \end{pmatrix} \begin{pmatrix} x_1(t-1) \\ x_2(t-1) \end{pmatrix} + \begin{pmatrix} \xi_1(t) \\ \xi_2(t) \end{pmatrix} \tag{10}$$

To make $x_1(t)$ and $x_2(t)$ correlated with $\text{corr}(x_1(t), x_2(t)) = \rho$, we chose the noise term s.t., $\text{corr}\,(\xi_1(t), \xi_2(t)) = \rho\,(1 - \rho_1\rho_2)\,[(1 - \rho_1^2)(1 - \rho_2^2)]^{-1}$. For the first class ($y = 1$), we generated 100 two-variate MTS of length 50 for the training and 100 for the test, from the VAR(1)-model with parameters $\rho = \rho_1 = \rho_2 = 0.8$ and $\mathbb{E}[(x_1(t), x_2(t))^T \mid y = 1] = (0.5, -0.5)^T$. Analogously, the MTS of the second class ($y = 2$) were generated using parameters $\rho = -0.8$, $\rho_1 = \rho_2 = 0.6$ and $\mathbb{E}[(x_1(t), x_2(t))^T \mid y = 2] = (0, 0)^T$.

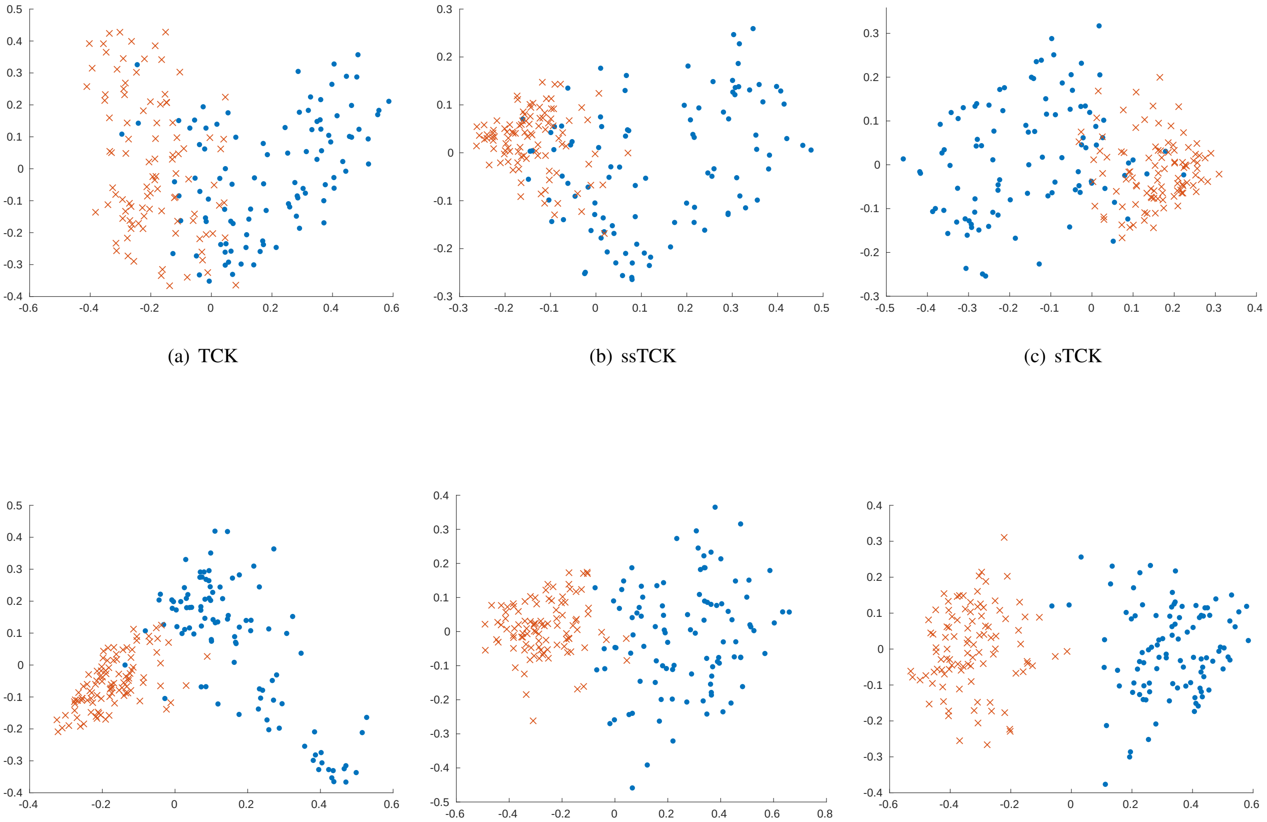
To simulate MNAR and inject informative missing patterns, we let $x_i^{(n)}(t)$ have a probability $p^{(n)}$ of being missing, given that $x_i^{(n)}(t) > -1$, $i = 1, 2$. We let $p^{(n)} = 0.9$ if $y^{(n)} = 1$ and $p_{(n)} = 0.8$ otherwise. By doing so, the missing ratio was roughly 63% in both classes.

Tab. 1 shows the accuracy on the test data for the different kernels. As expected, the TCK gives the lowest accuracy, 0.826. The ssTCK improves the accuracy considerably (0.854), and the supervised version (sTCK) gives further improvement (0.867). However, as we can see, the effect of explicitly modeling the missingness mechanism in the TCK$_{IM}$ is larger. In this case the accuracy increases from 0.826 to 0.933. The two corresponding embeddings are plotted in Fig. 1(a) and 1(d), respectively. In the TCK embedding, there are many points from different classes that overlap with each other, whereas for the TCK$_{IM}$ the number of overlapping points is much lower.

The ssTCK$_{IM}$ improves the accuracy to 0.967 (from 0.933 for TCK$_{IM}$ and 0.854 for ssTCK). The two embeddings obtained using the semi-supervised methods are shown in Fig. 1(b) and 1(e). The supervised version sTCK$_{IM}$ yields a slight improvement in terms of accuracy compared to ssTCK$_{IM}$ (0.970 vs 0.967). Plots of the supervised embeddings are shown in Fig. 1(c) and 1(f). We can see that for the sTCK$_{IM}$ the classes are clearly separated.

5.2. Performance of ssTCK on benchmark datasets

The purpose of the experiments reported in the following paragraph was to evaluate the impact of incorporating incomplete label information in the ssTCK. Towards that end, we considered benchmark datasets and artificially modified the number of labeled MTS in the training sets. We applied the proposed ssTCK to four MTS benchmark datasets from the UCR and UCI databases [39, 40] and other published work [41], described in Tab. 2. Since some of the datasets contain MTS of



(a) TCK

(b) ssTCK

(c) sTCK

(d) TCK_{IM}

(e) ssTCK_{IM}

(f) sTCK_{IM}

Figure 1: Plot of the two-dimensional KPCA representation of the synthetic data obtained using 6 different time series cluster kernels. The datapoints are color-coded according to their labels.

Table 2: Description of benchmark time series datasets. Column 2 to 5 show the number of attributes, samples in training and test set, and number of classes, respectively. T_{min} is the length of the shortest MTS in the dataset and T_{max} the longest MTS. T is the length of the MTS after the transformation.

Datasets	Attributes	Train	Test	N_c	T_{min}	T_{max}	T	Source
uWave	3	200	4278	8	315	315	25	UCR
Char.Traj.	3	300	2558	20	109	205	23	UCI
Wafer	6	298	896	2	104	198	25	Olsz.
Japan.vow.	12	270	370	9	7	29	15	UCI

Table 3: Classification accuracy for benchmark datasets obtained using TCK, ssTCK and sTCK.

Datasets	TCK	ssTCK	sTCK
Char. Traj.	0.908	0.928	0.934
uWave	0.867	0.881	0.894
Wafer	0.956	0.970	0.970
Japanese vowels	0.946	0.962	0.968

varying length, we followed the approach of Wang et al. [42] and transformed all the MTS in the same dataset to the same length, T, determined by $T = \left\lceil \frac{T_{max}}{\lceil \frac{T_{max}}{25} \rceil} \right\rceil$, where T_{max} is the length of the longest MTS in the dataset and $\lceil\ \rceil$ is the ceiling operator. The number of labeled MTS was set to $\max\{20, 3 \cdot N_c\}$. ssTCK was compared to ordinary TCK and sTCK (assuming complete label information in the latter case).

Tab. 3 shows the performance of ssTCK for the 4 benchmark datasets. As we can see, compared to TCK, the accuracy in general increases using ssTCK. For the Wafer dataset, ssTCK yields the same performance as the supervised kernel. For the three other datasets, the performance of ssTCK is slightly worse than sTCK. These experiments demonstrate that ssTCK is capable of exploiting incomplete label information.

Further, we created 8 synthetic datasets by randomly removing 50% and 80%, respectively, of the values in each of the 4 benchmark datasets. As we can see from the results presented in Tab. 4, also in presence of missing data the accuracy in general increases using ssTCK, compared to TCK.

For comparison, in Tab. 4 we also added the results obtained

Table 4: Classification accuracy for benchmark datasets obtained using TCK, ssTCK and sTCK.

Missing rate	Datasets	TCK	ssTCK	sTCK	GAK	Linear	LPS
50%	Char. Traj.	0.751	0.780	0.797	0.588	0.589	0.127
	uWave	0.812	0.834	0.850	0.828	0.813	0.411
	Wafer	0.956	0.970	0.972	0.792	0.791	0.823
	Japanese vowels	0.929	0.948	0.947	0.827	0.824	0.746
80%	Char. Traj.	0.282	0.310	0.331	0.194	0.192	0.062
	uWave	0.589	0.592	0.603	0.441	0.464	0.234
	Wafer	0.926	0.934	0.934	0.796	0.805	0.819
	Japanese vowels	0.809	0.836	0.847	0.473	0.489	0.389

Table 5: Classification accuracy on synthetic benchmark datasets that contain missing data.

Correlation	TCK	TCK$_B$	TCK$_0$	TCK$_{IM}$	TCK	TCK$_B$	TCK$_0$	TCK$_{IM}$
		Wafer				**Japanese vowels**		
0.2	0.951	0.951	0.951	**0.955**	0.938	**0.954**	0.951	0.940
0.4	**0.961**	0.953	0.955	**0.961**	0.932	0.938	0.938	**0.941**
0.6	0.961	0.900	0.965	**0.996**	0.922	0.946	0.924	**0.962**
0.8	0.958	0.893	0.963	**1.000**	0.922	0.924	0.935	**0.968**
		uWave				**Character trajectories**		
0.2	0.763	0.457	0.755	**0.841**	**0.854**	0.742	0.847	0.851
0.4	0.807	0.587	0.813	**0.857**	0.851	0.788	0.842	**0.867**
0.6	0.831	0.674	0.837	**0.865**	0.825	0.790	0.824	**0.871**
0.8	0.834	0.699	0.844	**0.884**	0.839	0.707	0.853	**0.901**

using three other kernels; GAK, the linear kernel, and LPS. GAK and the linear kernel cannot process incomplete MTS and therefore we created complete datasets using mean imputation for these two kernels. LPS[1] was run using default hyperparameters, with the exception that we adjusted the segment length to be sampled from the interval $[6, 0.8T]$ to account for the relatively short MTS in our datasets. In accordance with [43], for GAK[2] we set the bandwidth σ to 0.1 times the median distance of all MTS in the training set scaled by the square root of the median length of all MTS, and the triangular parameter to 0.2 times the median length of all MTS. Distances were measured using the canonical metric induced by the Frobenius norm. In the linear kernel we set the constant c to 0. As we can see, the performance of these kernels is considerably worse than the time series cluster kernels for 7 out of 8 datasets. For uWave with 50% missingness, the performance of GAK and the linear kernel is similar to the TCK kernels.

5.3. *Exploiting informative missingness in synthetic benchmark datasets*

To evaluate the effect of modeling the missing patterns in TCK$_{IM}$, we generated 8 synthetic datasets by manually injecting missing elements into the Wafer and Japanese vowels datasets using the following procedure. For each attribute $v \in \{1, \dots, V\}$, a number $c_v \in \{-1, 1\}$ was randomly sampled with equal probabilities. If $c_v = 1$, the attribute v is positively correlated with the labels, otherwise negatively correlated. For each MTS $X^{(n)}$ and attribute, a missing rate γ_{nv} was sampled from the uniform distribution $\mathcal{U}[0.3 + E \cdot c_v \cdot (y^{(n)} - 1), 0.7 + E \cdot c_v \cdot (y^{(n)} - 1)]$. This ensures that the overall missing rate of each dataset is approximately 50%. $y^{(n)} \in \{1, \dots N_c\}$ is the label of the MTS $X^{(n)}$ and E is a parameter, which we tune for each dataset in such a way that the absolute value of the Pearson correlation between the missing rates for the attributes γ_v and the labels $y^{(n)}$ takes the values {0.2, 0.4, 0.6, 0.8}, respectively. The higher the value of the Pearson correlation, the higher is the informative missingness.

Tab. 5 shows the performance of the proposed TCK$_{IM}$ and three baseline models (TCK, TCK$_B$, and TCK$_0$). The first baseline is ordinary TCK, which ignores the missingness mechanism. For the Wafer dataset, the performance of this baseline was quite similar across all four settings. For the Japanese vowels dataset, the performance actually decreases as the information in the missing patterns increases. In the second baseline, TCK$_B$, we tried to model the missing patterns by concatenating the binary missing indicator MTS R to the MTS X and creating a new MTS with $2V$ attributes. Then, we trained ordinary TCK on this representation. For the Wafer dataset, the performance decreases considerably as the informative missingness increases. For the Japanese vowels, this baseline yields the best performance when the correlation is 20%. However, the performance actually decreases as the informative missingness increases. Hence, informative missingness is not captured with this baseline. In the last baseline, TCK$_0$, we investigated if it is possible to capture informative missingness by imputing zeros for the missing values and then training the TCK on the imputed data. This baseline yields similar performance across all 4 settings for the Wafer dataset, and for Japanese vowels, TCK$_0$ has a similar behaviour as TCK$_B$, i.e. it does not capture informative missing patterns. The proposed TCK$_{IM}$ achieves the best

[1] Matlab implementation: http://www.mustafabaydogan.com/

[2] Matlab implementation: http://www.marcocuturi.net/GA.html

accuracy for 7 out of 8 settings and has the expected behaviour, namely that the accuracy increases as the correlation between missing values and class labels increases. The performance is similar to TCK when the amount of information in the missing patterns is low, whereas TCK is clearly outperformed when the informative missingness is high. This demonstrates that TCK$_{IM}$ effectively utilizes informative missing patterns.

To also test if TCK$_{IM}$ is capable of exploiting other types of informative missingness, we generated 8 synthetic datasets from uWave and Character trajectories using the following approach. Both of these datasets consists of 3 attributes. For each attribute $v \in \{1, \ldots, V\}$, a number $c_v \in \{-1, 1\}$ was randomly sampled with equal probabilities. For the attribute(s) with $c_v = -1$, we let it be negatively correlated with the labels by sampling the missing rate γ_{nv} from $\mathcal{U}[0.7 - E \cdot (y^{(n)} - 1), 1 - E \cdot (y^{(n)} - 1)]$. For the attribute with $c_v = 1$, we let it be positively correlated with the labels by sampling the missing rate γ_{nv} from $\mathcal{U}[0.3 + E \cdot (y^{(n)} - 1), 0.6 + E \cdot (y^{(n)} - 1)]$. We let each element with $x_v^{(n)}(t) > \mu_v$ have a probability γ_{nv} of being missing, where μ_v is the mean of attribute v computed over the complete dataset. The fact that the probability of being missing depends on the missing values means that, within each class, the missingness mechanism is MNAR. We tuned the parameter E such that the mean absolute value of the Pearson correlation between γ_v and the labels took the values {0.2, 0.4, 0.6, 0.8}. By doing so, the overall missing rate was approximately 32% for uWave and 45% for the Characters. However, we note that in this case the overall missing rate varies slightly as a function of the Pearson correlation.

Tab. 5 shows the performance on the 8 synthetic datasets created from uWave and Char. traj. One thing to notice here is the poor performance of TCK$_B$. This demonstrates the importance of using the mixed mode mixtures to model the two modalities in $U = (X, R)$. To naively apply TCK based on the GMMs to the concatenated MTS do not provide the desired performance. Further, we see that TCK$_{IM}$ achieves the best accuracy for 7 out of 8 settings and the accuracy increases as the correlation increases. For the Characters, the performance of TCK$_{IM}$ is similar to TCK for low correlation but increases as the missingness information increases, whereas the performance of TCK actually decreases. One possible explanation is that for this dataset, two of the variables were positively correlated with the labels and therefore the missing rate increases with increasing correlation. Regarding the results for uWave, it is a bit surprising that the largest difference in performance between TCK and TCK$_{IM}$ occurs when the correlation is lowest. There might be several reasons to this: a peculiarity of the dataset and/or that the MNAR missingness created missing patterns that negatively affect TCK.

6. Case study: Detecting infections among patients undergoing colon rectal cancer surgery

In this case study, the focus was to detect Surgical Site Infection (SSI), which is one of the most common types of nosocomial infections [44] and represents up to 30% of hospital-acquired infections [45, 46]. The importance of the topic of SSI prediction is reflected in several recent initiatives. For instance, the current study is part of a larger research effort by the current team, on SSI prediction and detection of postoperative adverse events related to gastrointestinal surgery within the context of improving the *quality of surgery* [21, 24, 47, 48, 49, 50]. Clearly, the reason for this massive interest is that a reduction in the number of postoperative complications such as SSI will be of great benefit both for the patients and for the society.

Table 6: List of extracted blood tests and their corresponding missing rates.

Attribute nr.	Blood test	Missing rate
1	Hemoglobin	0.646
2	Leukocytes	0.727
3	C-Reactive Protein	0.691
4	Potassium	0.709
5	Sodium	0.712
6	Creatinine	0.867
7	Thrombocytes	0.921
8	Albumin	0.790
9	Carbamide	0.940
10	Glucose	0.921
11	Amylase	0.952

Many studies have shown that laboratory tests, and blood tests in particular, are especially important predictors for SSI, both pre- and post-operatively [51, 49, 52, 53, 48, 54, 55, 56, 57, 58, 59]. Therefore, blood tests provided the basis also for this case study.

6.1. Data collection

Ethics approval for the parent study was obtained from the Data Inspectorate and the Ethics Committee at the University Hospital of North Norway (UNN) [50]. In [50], a cohort consisting of 7741 patients was identified by extracting the electronic health records for all patients that underwent a gastrointestinal surgical procedure at UNN in the years 2004–2012. In this case study, we were particularly interested in detecting SSI, which is an infection particularly associated with colorectal cancer surgery [60]. Therefore, patients who did not undergo this type of surgery were excluded, reducing the size of the cohort to 1137 patients.

In collaboration with a clinician (author A. R.), we extracted data for 11 of the most common blood tests from the patient's EHRs. The value of a patient's blood test, e.g. his or hers hemoglobin level, can be considered as a continuous variable over time. However, blood tests are usually measured on a daily basis, and therefore, for the purpose of the current analysis, we discretized time and let each time interval be one day. Hence, the blood samples could naturally be represented as MTS and needed no further feature preprocessing in our framework.

All blood tests were not available every day for each patient, which means that the dataset contained missing data, and we expected the missing patterns to be informative since whether a test is performed depends on whether the doctor thinks it is needed. We focused on detection of SSI within 10 days after surgery and therefore the length of the time series is 10. Patients with no recorded lab tests during the period from postoperative

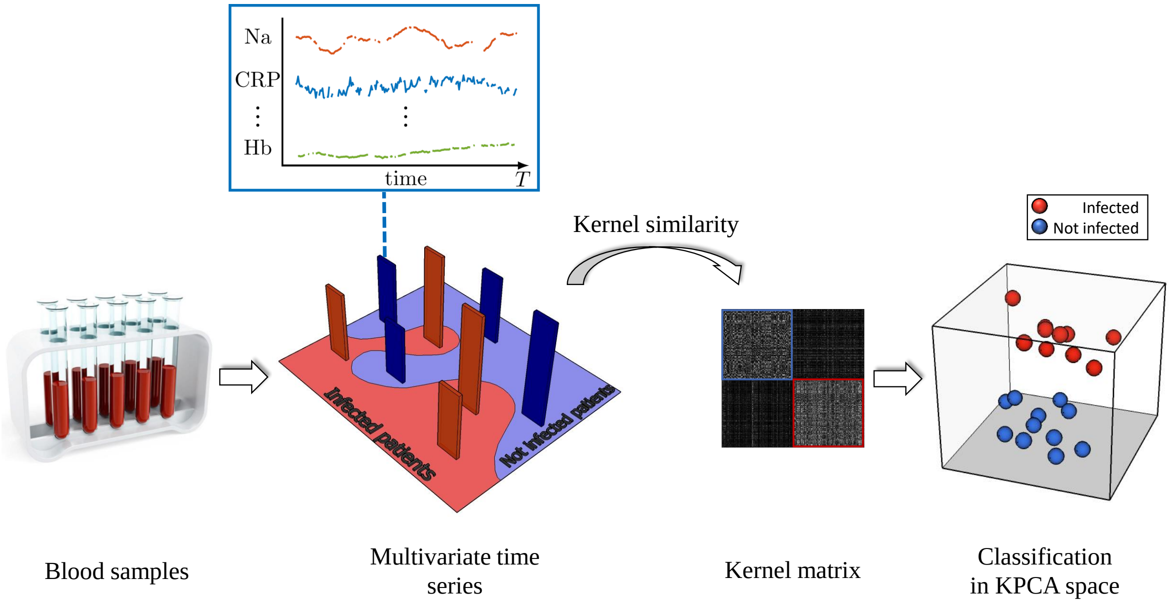


Figure 2: *Overview of the approach taken to detect postoperative SSI from MTS blood samples.*

day 1 until day 10 were removed from the cohort, which lead to a final cohort consisting of 858 patients. The average proportion of missing data in the cohort was 80.7%. Tab. 6 shows a list of the blood tests we considered in this study and their corresponding missing rate.

Guided by input from clinicians, the International Classification of Diseases (ICD10) or NOMESCO Classification of Surgical Procedures (NCSP) codes related to severe postoperative complications were considered to identify the patients in the cohort that developed postoperative SSI. Patients that did not have these codes and did not have the word "infection" in any of their postoperative text documents were considered as controls. This lead to a dataset with 227 infected patients (cases) and 631 non-infected patients (control).

6.2. Experimental setup

The objective of this case study was to evaluate how the proposed MTS kernels perform in a real-world application from medicine. We would like to emphasize that the proposed kernels are mainly designed for situations when there are no, or only a few, ground-truth labels available. However, in order to evaluate the quality of these kernels, we adopted a supervised scheme. Hence, we followed the scheme presented in Fig. 2, i.e. we computed the kernel from the MTS representations of the blood tests and performed KPCA, followed by kNN classification in the KPCA space. We set the dimensionality of the KPCA-representation to 10 in all experiments. The number of neighbors k was set using 5-fold cross validation.

Four baseline kernels were considered, namely TCK, LPS, GAK and the linear kernel. GAK and the linear kernel cannot work on incomplete datasets, and therefore, we created 2 complete datasets using mean and LOCF imputation. In order to investigate if it is possible to better exploit the information from the missing patterns for the LPS, GAK and linear kernels, we also created baselines by concatenating the binary indicator MTS $R^{(n)}$ to the MTS $X^{(n)}$.

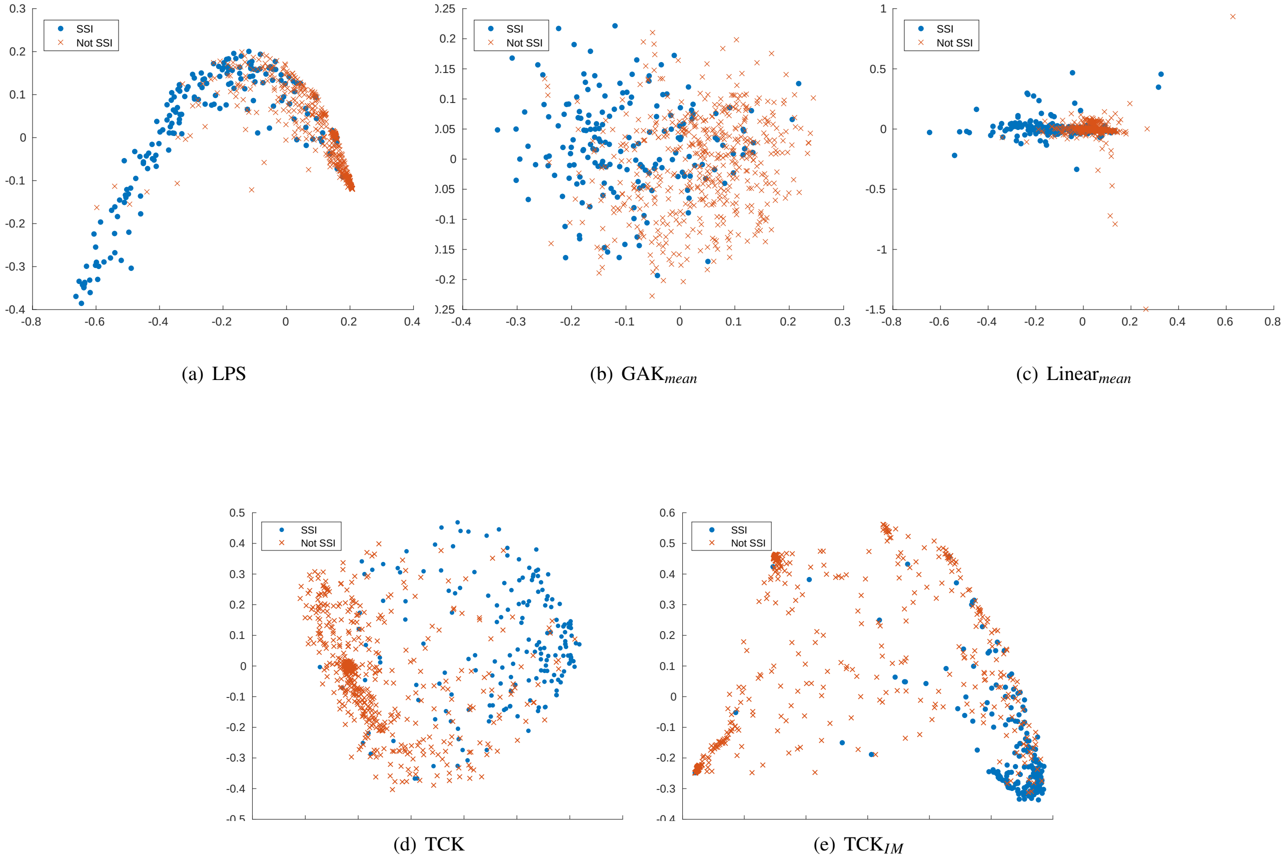
We performed 5-fold cross validation and reported results in terms of F1-score, sensitivity, specificity and accuracy. Sensitivity is the fraction of actual positives (has SSI) correctly classified as positive, whereas specificity is the fraction of actual negatives that are correctly classified as negative. F1-score is the harmonic mean of precision and sensitivity, where precision is the fraction of actual positives among all those that are classified as positive cases.

6.3. Results

Tab. 7 shows the performance in terms of 4 evaluation metrics for 11 baseline kernels as well as the proposed TCK$_{IM}$ kernel on the task of detecting patients suffering from SSI. We see that the kernels that rely on imputation performs much worse than other kernels in terms of F1-score, sensitivity and accuracy. These methods do, however, achieve a high specificity. However, any classifier can achieve a specificity of 1 simply by classifying all cases as negative, but this of course leads to lower F1-score and sensitivity. The main reasons why these methods do not perform better are probably that the imputation methods introduce strong biases into the data and that the missingness mechanism is ignored. The TCK and LPS kernels perform quite similarly across all 4 evaluation metrics (LPS slightly better). The F1-score, sensitivity and accuracy achieved for these methods are considerably higher than the corresponding scores for the GAK and linear kernel. One of the reasons why these methods perform better than the imputation methods is that ignoring the missingness leads to lower bias than replacing missing values with biased estimates. The performance of the linear kernel

Table 7: Performance (mean ± se) on the SSI dataset.

	Kernel	F1-score	Sensitivity	Specificity	Accuracy
Ignore missingness	TCK	0.726 ± 0.045	0.678 ± 0.035	0.930 ± 0.024	0.863 ± 0.023
	LPS	0.746 ± 0.035	0.696 ± 0.056	0.939 ± 0.019	0.875 ± 0.016
Impute	GAK$_{LOCF}$	0.570 ± 0.045	0.484 ± 0.059	0.924 ± 0.022	0.808 ± 0.017
	GAK$_{mean}$	0.629 ± 0.046	0.502 ± 0.059	0.966 ± 0.023	0.843 ±0.016
	Linear$_{LOCF}$	0.557 ± 0.058	0.480 ± 0.073	0.914 ± 0.017	0.800 ± 0.018
	Linear$_{mean}$	0.599 ± 0.030	0.489 ± 0.041	0.948 ± 0.043	0.826 ± 0.024
Informative	LPS$_{IM}$	0.720 ± 0.062	0.661 ± 0.069	0.937 ± 0.036	0.863 ± 0.032
	GAK$_{IM+LOCF}$	0.669 ± 0.015	0.586 ± 0.024	0.940 ± 0.021	0.846 ± 0.011
	GAK$_{IM+mean}$	0.696 ± 0.030	0.617 ± 0.033	0.945 ± 0.022	0.856 ±0.011
	Linear$_{IM+LOCF}$	0.628 ± 0.016	0.529 ± 0.030	0.945 ± 0.011	0.834 ± 0.005
	Linear$_{IM+mean}$	0.668 ± 0.037	0.568 ± 0.033	**0.951 ± 0.030**	0.850 ± 0.021
	TCK$_{IM}$	**0.802 ± 0.016**	**0.806 ± 0.027**	0.927 ± 0.017	**0.895 ± 0.010**



(a) LPS

(b) GAK$_{mean}$

(c) Linear$_{mean}$

(d) TCK

(e) TCK$_{IM}$

Figure 3: Plot of the two-dimensional KPCA representation of the colon rectal cancer surgery patients obtained using 5 kernels.

and GAK improves a bit by accounting for informative missingness, whereas the performance of LPS decreases. TCK$_{IM}$ performs similarly to the baselines in terms of specificity, but considerably better in terms of F1-score, sensitivity and accuracy. This demonstrates that the missing patterns in the blood test time series are informative and the TCK$_{IM}$ is capable of exploiting this information to improve performance on the task of detecting patients with infections.

Fig. 3 shows KPCA embeddings corresponding to the two largest eigenvalues obtained using 5 different kernels. While the representations obtained using GAK and the linear kernel are noisy and to a large degree mix the infected and non-infected patients, the two classes (SSI and non-SSI) are more separated in the representations obtained using TCK and LPS. The TCK$_{IM}$ is even better at forcing the SSI patients to stay in the same region or cluster while it at the same time spreads out the patients without infection, revealing the diversity among these patients.

7. Conclusions and future directions

In this work, we presented robust multivariate time series kernels capable of exploiting informative missing patterns and incomplete label information. In contrast to other frameworks that exploit informative missingness [6, 16], which need complete label information, the time series cluster kernels are specially designed for situations in which no labels or only a few labels are available. Lack of labels and large amounts of missing data are two challenges that characterize the medical domain, and therefore, we think the proposed kernels will be particularly useful in this domain, which we also demonstrated in this work through a case study of postoperative infections among colon rectal cancer patients. However, the kernels are not limited to this domain. We believe that these kernels could be useful tools in other application domains facing similar challenges.

A limitation of TCK$_{IM}$ is that if the missingness is by no means correlated with the outcome of interest, there will be limited gain in performance compared to the TCK, or might even a decrease in performance. For this reason it is important that the user has some domain knowledge and has some understanding about the process that led to missing values in the data, as illustrated in our case study from healthcare.

An other limitation of the time series cluster kernels is that they are designed for MTS of the same length. A possible next step would be to work on a formulation that can deal with varying length. In further work, we would also like to investigate the possibility of introducing a Bayesian formulation for the discrete modality in the mixed mode mixture models by putting informative priors over the parameters in the Bernoulli part of the model.

Conflict of interest

The authors have no conflict of interest related to this work.

Acknowledgement

This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the *Next Generation Learning Machines*. Cristina Soguero-Ruiz is partially supported by project TEC2016-75361-R from Spanish Government and by project DTS17/00158 from Institute of Health Carlos III (Spain).

The authors would like to thank Kristian Hindberg from UiT The Arctic University of Norway for his assistance on preprocessing and extracting the data from the EHR system. We would also like to thank Rolv-Ole Lindsetmo and Knut Magne Augestad from the University Hospital of North Norway, Fred Godtliebsen from UiT, together with Stein Olav Skrøvseth from the Norwegian Centre for E-health Research for helpful discussions throughout the study and manuscript preparation.

References

[1] D. B. Rubin, Inference and missing data, Biometrika 63 (3) (1976) 581–592.

[2] G. Molenberghs, Incomplete data in clinical studies: analysis, sensitivity, and sensitivity analysis, Drug Information Journal 43 (4) (2009) 409–429.

[3] G. Molenberghs, C. Beunckens, C. Sotto, M. G. Kenward, Every missingness not at random model has a missingness at random counterpart with equal fit, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70 (2) (2008) 371–388.

[4] A. S. Allen, P. J. Rathouz, G. A. Satten, Informative missingness in genetic association studies: case-parent designs, The American Journal of Human Genetics 72 (3) (2003) 671–680.

[5] C.-Y. Guo, J. Cui, L. A. Cupples, Impact of non-ignorable missingness on genetic tests of linkage and/or association using case-parent trios, BMC Genetics 6 (1) (2005) S90.

[6] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Scientific reports 8 (1) (2018) 6085.

[7] J. L. Schafer, J. W. Graham, Missing data: our view of the state of the art., Psychological methods 7 (2) (2002) 147.

[8] J. L. Schafer, Analysis of incomplete multivariate data, CRC press, 1997.

[9] R. J. Little, D. B. Rubin, Statistical analysis with missing data, John Wiley & Sons, 2014.

[10] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, Pattern classification with missing data: a review, Neural Computing and Applications 19 (2) (2010) 263–282.

[11] S. A. Rahman, Y. Huang, J. Claassen, N. Heintzman, S. Kleinberg, Combining Fourier and lagged k-nearest neighbor imputation for biomedical time series data, Journal of Biomedical Informatics 58 (2015) 198 – 207.

[12] J. M. Engels, P. Diehr, Imputation of missing longitudinal data: a comparison of methods, Journal of Clinical Epidemiology 56 (10) (2003) 968 – 976.

[13] I. R. White, P. Royston, A. M. Wood, Multiple imputation using chained equations: issues and guidance for practice, Statistics in medicine 30 (4) (2011) 377–399.

[14] F. M. Bianchi, L. Livi, A. Ferrante, J. Milosevic, M. Malek, Time series kernel similarities for predicting paroxysmal atrial fibrillation from ECGs, arXiv preprint arXiv:1801.06845.

[15] K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, S. O. Skrøvseth, R.-O. Lindsetmo, A. Revhaug, R. Jenssen, Learning similarities between irregularly sampled short multivariate time series from EHRs, 3rd ICPR International Workshop on Pattern Recognition for Healthcare Analytics, Cancun, Mexico, 2016.

[16] Z. C. Lipton, D. Kale, R. Wetzel, Directly modeling missing data in sequences with RNNs: Improved classification of clinical time series, in: Machine Learning for Healthcare Conference, Vol. 56, PMLR, 2016, pp. 253–270.

[17] F. M. Bianchi, L. Livi, K. Ø. Mikalsen, M. Kampffmeyer, R. Jenssen, Learning representations for multivariate time series with missing data using temporal kernelized autoencoders, arXiv preprint arXiv:1805.03473.

[18] B. M. Marlin, D. C. Kale, R. G. Khemani, R. C. Wetzel, Unsupervised pattern discovery in electronic health care data using probabilistic clustering models, in: Proc. of 2nd ACM SIGHIT Int. Health Informatics Symposium, 2012, pp. 389–398.

[19] M. Ghassemi, M. A. F. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits, M. Feng, A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data, in: Conference on Artificial Intelligence, AAAI, 2015, pp. 446–453.

[20] K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, R. Jenssen, Time series cluster kernel for learning similarities between multivariate time series with missing data, Pattern Recognition 76 (2018) 569–581.

[21] K. Ø. Mikalsen, C. Soguero-Ruiz, A. Revhaug, R.-O. Lindsetmo, R. Jenssen, et al., Using anchors from free text in electronic health records to diagnose postoperative delirium, Computer Methods and Programs in Biomedicine 152 (Supplement C) (2017) 105 – 114.

[22] R. Jenssen, Kernel entropy component analysis, IEEE Trans Pattern Anal Mach Intell 33 (5) (2010) 847–860.

[23] G. Camps-Valls, L. Bruzzone, Kernel methods for remote sensing data analysis, John Wiley & Sons, 2009.

[24] C. Soguero-Ruiz, A. Revhaug, R.-O. Lindsetmo, K. M. Augestad, R. Jenssen, et al., Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records, IEEE journal of biomedical and health informatics 20 (5) (2016) 1404–1415.

[25] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis, Cambridge university press, 2004.

[26] H. Chen, F. Tang, P. Tino, X. Yao, Model-based kernel for efficient time series analysis, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 392–400.

[27] D. J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1994, pp. 359–370.

[28] P.-F. Marteau, S. Gibet, On recursive edit distance kernels with application to time series classification, IEEE Transactions on Neural Networks and Learning Systems 26 (6) (2015) 1121–1133.

[29] M. Cuturi, J.-P. Vert, O. Birkenes, T. Matsui, A kernel for time series based on global alignments, in: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, Vol. 2, IEEE, 2007, pp. II–413.

[30] M. Cuturi, Fast global alignment kernels, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 929–936.

[31] M. G. Baydogan, G. Runger, Time series representation and similarity based on local autopatterns, Data Mining and Knowledge Discovery 30 (2) (2016) 476–509.

[32] A. Barla, F. Odone, A. Verri, Histogram intersection kernel for image classification, in: Proceedings of International Conference on Image Processing, Vol. 3, IEEE, 2003, pp. III–513.

[33] T. G. Dietterich, Ensemble methods in machine learning, in: International workshop on multiple classifier systems, Springer Berlin Heidelberg, 2000, pp. 1–15.

[34] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE transactions on pattern analysis and machine intelligence 12 (10) (1990) 993–1001.

[35] S. Vega-Pons, J. Ruiz-Shulcloper, A survey of clustering ensemble algorithms, International Journal of Pattern Recognition and Artificial Intelligence 25 (03) (2011) 337–372.

[36] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the royal statistical society. Series B (methodological) (1977) 1–38.

[37] G. McLachlan, T. Krishnan, The EM algorithm and extensions, Vol. 382, John Wiley & Sons, 2007.

[38] S. Kullback, R. A. Leibler, On information and sufficiency, The annals of mathematical statistics 22 (1) (1951) 79–86.

[39] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The ucr time series classification archive, `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/` (October 2018).

[40] M. Lichman, UCI machine learning repository, `http://archive.ics.uci.edu/ml`, accessed: 2018-08-29 (2013).

[41] R. T. Olszewski, Generalized feature extraction for structural pattern recognition in time-series data, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA (2001).

[42] L. Wang, Z. Wang, S. Liu, An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm, Expert Systems with Applications 43 (2016) 237 – 249.

[43] Fast global alignment kernel Matlab implementation, `http://www.marcocuturi.net/GA.html`, accessed: 2018-08-02.

[44] S. S. Lewis, R. W. Moehring, L. F. Chen, D. J. Sexton, D. J. Anderson, Assessing the relative burden of hospital-acquired infections in a network of community hospitals, Infection Control & Hospital Epidemiology 34 (11) (2013) 1229–1230.

[45] S. S. Magill, W. Hellinger, J. Cohen, R. Kay, et al., Prevalence of healthcare-associated infections in acute care hospitals in Jacksonville, Florida, Infection Control 33 (03) (2012) 283–291.

[46] G. de Lissovoy, K. Fraeman, V. Hutchins, D. Murphy, D. Song, B. B. Vaughn, Surgical site infection: incidence and impact on hospital utilization and treatment costs, American Journal of Infection Control 37 (5) (2009) 387–397.

[47] C. Soguero-Ruiz, A. Revhaug, R.-O. Lindsetmo, R. Jenssen, et al., Predicting colorectal surgical complications using heterogeneous clinical data and kernel methods, Journal of Biomedical Informatics 61 (2016) 87–96.

[48] A. S. Strauman, F. M. Bianchi, K. Ø. Mikalsen, M. Kampffmeyer, C. Soguero-Ruiz, R. Jenssen, Classification of postoperative surgical site infections from blood measurements with missing data using recurrent neural networks, in: 2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), 2018, pp. 307–310.

[49] C. Soguero-Ruiz, R. Jenssen, K. M. Augestad, S. O. Skrøvseth, et al., Data-driven temporal prediction of surgical site infection, in: AMIA Annual Symposium Proceedings, Vol. 2015, American Medical Informatics Association, 2015, p. 1164.

[50] K. Jensen, C. Soguero-Ruiz, K. Ø. Mikalsen, R.-O. Lindsetmo, I. Kouskoumvekaki, M. Girolami, S. O. Skrovseth, K. M. Augestad, Analysis of free text in electronic health records for identification of cancer patient trajectories, Scientific Reports 7 (2017) 46226.

[51] J. Silvestre, J. Rebanda, C. Lourenço, P. Póvoa, Diagnostic accuracy of C-reactive protein and procalcitonin in the early detection of infection after elective colorectal surgery–a pilot study, BMC infectious diseases 14 (1) (2014) 444.

[52] F. J. Medina-Fernández, D. J. Garcilazo-Arismendi, R. García-Martín, L. Rodríguez-Ortiz, J. Gómez-Barbadillo, et al., Validation in colorectal procedures of a useful novel approach for the use of C-reactive protein in postoperative infectious complications, Colorectal Disease 18 (3) (2016) O111–O118.

[53] M. R. Angiolini, F. Gavazzi, C. Ridolfi, M. Moro, P. Morelli, M. Montorsi, A. Zerbi, Role of C-reactive protein assessment as early predictor of surgical site infections development after pancreaticoduodenectomy, Digestive surgery 33 (4) (2016) 267–275.

[54] S. Liu, J. Miao, G. Wang, M. Wang, X. Wu, K. Guo, M. Feng, W. Guan, J. Ren, Risk factors for postoperative surgical site infections in patients with crohn's disease receiving definitive bowel resection, Scientific Reports 7 (1) (2017) 9828.

[55] E. Mujagic, W. R. Marti, M. Coslovsky, J. Zeindler, et al., The role of preoperative blood parameters to predict the risk of surgical site infection, The American Journal of Surgery 215 (4) (2018) 651–657.

[56] A. Goulart, C. Ferreira, A. Estrada, F. Nogueira, S. Martins, A. Mesquita-Rodrigues, N. Sousa, P. Leao, Early inflammatory biomarkers as predictive factors for freedom from infection after colorectal cancer surgery: A prospective cohort study, Surgical infections 19 (4) (2018) 446–450.

[57] Z. Hu, G. B. Melton, E. G. Arsoniadis, Y. Wang, M. R. Kwaan, G. J. Simon, Strategies for handling missing clinical data for automated surgical site infection detection from the electronic health record, Journal of Biomedical Informatics 68 (2017) 112–120.

[58] S. L. Gans, J. J. Atema, S. Van Dieren, B. G. Koerkamp, M. A. Boermeester, Diagnostic value of C-reactive protein to rule out infectious complications after major abdominal surgery: a systematic review and meta-analysis, International journal of colorectal disease 30 (7) (2015) 861–873.

[59] P. C. Sanger, G. H. van Ramshorst, E. Mercan, et al., A prognostic model of surgical site infection using daily clinical wound assessment, Journal of the American College of Surgeons 223 (2) (2016) 259 – 270.e2.

[60] E. H. Lawson, C. Y. Ko, J. L. Adams, W. B. Chow, B. L. Hall, Reliability of evaluating hospital quality by colorectal surgical site infection type, Annals of surgery 258 (6) (2013) 994–1000.

Chapter 12

Paper IV

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Using anchors from free text in electronic health records to diagnose postoperative delirium

Karl Øyvind Mikalsen[a,b,*], Cristina Soguero-Ruiz[b,c], Kasper Jensen[d], Kristian Hindberg[a], Mads Gran[e], Arthur Revhaug[e,f,g], Rolv-Ole Lindsetmo[e,g], Stein Olav Skrøvseth[a,d], Fred Godtliebsen[a], Robert Jenssen[h,d,b]

[a] Department of Mathematics and Statistics, UiT The Arctic University of Norway, Tromsø, Norway
[b] UiT Machine Learning Group, Norway
[c] Department of Signal Theory and Comm., Telematics and Computing, Universidad Rey Juan Carlos, Fuenlabrada, Spain
[d] Norwegian Centre for E-health Research, University Hospital of North Norway (UNN), Tromsø, Norway
[e] Department of Gastrointestinal Surgery, UNN, Tromsø, Norway
[f] Clinic for Surgery, Cancer and Women's Health, UNN, Tromsø, Norway
[g] Institute of Clinical Medicine, UiT, Tromsø, Norway
[h] Department of Physics and Technology, UiT, Tromsø, Norway

ARTICLE INFO

Article history:
Received 9 July 2017
Revised 5 September 2017
Accepted 15 September 2017

Keywords:
Electronic health records
Semi-supervised learning
Learning with anchors framework
Postoperative delirium
Data-driven clinical decision support
Clustering

ABSTRACT

Objectives: Postoperative delirium is a common complication after major surgery among the elderly. Despite its potentially serious consequences, the complication often goes undetected and undiagnosed. In order to provide diagnosis support one could potentially exploit the information hidden in free text documents from electronic health records using data-driven clinical decision support tools. However, these tools depend on labeled training data and can be both time consuming and expensive to create.
Methods: The recent learning with anchors framework resolves this problem by transforming key observations (anchors) into labels. This is a promising framework, but it is heavily reliant on clinicians knowledge for specifying good anchor choices in order to perform well. In this paper we propose a novel method for specifying anchors from free text documents, following an exploratory data analysis approach based on clustering and data visualization techniques. We investigate the use of the new framework as a way to detect postoperative delirium.
Results: By applying the proposed method to medical data gathered from a Norwegian university hospital, we increase the area under the precision-recall curve from 0.51 to 0.96 compared to baselines.
Conclusions: The proposed approach can be used as a framework for clinical decision support for postoperative delirium.

1. Introduction

Complications after major surgery are unfortunately not uncommon. Central nervous system dysfunction, including postoperative delirium (PD), is often seen in geriatric patients undergoing major surgery [1]. Despite its potentially serious consequences, such as an increase in length of hospitalization, morbidity, mortality, and adverse events, it is often hard to detect PD [2]. Moreover, if the complication goes undiagnosed, it could have economical consequences for the care giver, as hospitals' reimbursement rates are dependent on correct coding.

For these reasons several works have investigated risk factors and prediction of PD. Bohner et al. predicted the risk for PD among patients undergoing aortic, carotid, and peripheral vascular surgery using multivariate linear logistic regression [3]. In [4,5], the authors predicted risk for PD after major abdominal surgery and found well-known predictors such as advanced age or ASA-score. Common for the previous studies is that only a few structured variables have been used as features for the prediction model. However, we believe that also the free text documents in the patients' electronic health records (EHRs) contain valuable information about PD that can be used for diagnosis support. In particular, nurses collect useful information about the patient health status since they observe the patients after the surgery and report about them three times every day.

[*] Corresponding author.
E-mail address: karl.o.mikalsen@uit.no (K.Ø. Mikalsen).

http://dx.doi.org/10.1016/j.cmpb.2017.09.014
0169-2607/© 2017 Elsevier B.V. All rights reserved.

Recent advances in machine learning for healthcare have shown great potential for exploiting the “hidden” information in the EHRs to provide data-driven clinical decision support, especially if large amounts of labeled data are available [6–9]. In the aforementioned studies, the patients were manually labeled with and without PD. However, the labeling task could be a time consuming and expensive process [10]. To overcome this drawback Halpern et al. proposed a very promising framework, with a large number of possible applications. In this framework, which we refer to as the anchor method (AM), one can learn phenotypes and predict clinical state variables from EHR unlabeled data only by specifying a few key observations called anchors [11,12]. An underlying assumption is that the presence of an anchor variable implies the presence of the latent label of interest. Thus, training examples for which the anchor variable is present are positive examples, while nothing can be said for the remaining examples.

If the data mainly consist of free text, a limitation with AM is that trustworthy anchors could be difficult to identify, even for clinicians. Moreover, in settings where the sample size is larger than the dimensionality $(N > d)$, the originally proposed (ridge) l_2-regularized logistic regression classifier within AM works well. It keeps all variables in the model and the coefficients of correlated variables are shrunken toward each other. However, when $d \gg N$ ridge regularization is not a good choice [13].

In this paper we investigate the use of AM as a way to develop models to detect PD, and thereby being able to diagnose and code it properly. To resolve the problem of specifying reliable anchors we develop a problem specific method based on domain knowledge and exploratory data analysis using clustering and visualization techniques. Furthermore, we propose to use a different classifier in the AM framework, namely the elastic net, which forces sparsity and has been shown to provide robustness in settings where the dimensionality is higher than the sample size [13,14]. We show that, by introducing this new methodology, AM can be successfully applied to problems where no obvious anchors exist. In particular, by applying it to clinical data gathered from a Norwegian university hospital, we show that it can be used to extract hidden information from unstructured free text and thereby provide diagnosis support for PD.

The rest of this paper is organized as follows. Section 2 describes methods, including the AM framework and our proposed anchor specification method. Experiments and results are presented in Section 4, we discuss the results and further work. Conclusions are drawn in Section 5.

2. Methods

2.1. Background on the learning with anchors framework

AM is particularly well suited for text documents where the features can be represented using e.g. bag-of-words or medical ontologies. In the method there are two different kinds of binary variables; *observed* and *latent*. An observed variable is a variable that can be observed directly from the EHR. It could for example be the answer to a question such as *does the word “confused” appear anywhere in some of the free text documents?* A latent variable cannot be extracted directly from the EHR and could be the answer to a higher level question such as *does the patient have postoperative delirium?* Formulating such questions is difficult since there are so many different ways to answer them, and it could also be that answers are not documented in the EHR.

An *anchor* variable is an observed variable that can be extracted directly from the EHR and contains valuable information about the *latent* variable one wants to uncover. The anchor should satisfy two properties, (1) given that the anchor is observed, then also its latent variable is on, and (2) it is independent of all other observations, conditioned on the latent variable. The latter property states that once the value of the latent variable is known, no other observed variables provide additional information about the anchor.

Given these definitions, a description of the steps in the original AM is as follows: (1) Select data source; (2) represent features using e.g. bag-of-words; (3) specify anchor (for this step our proposed method can be used); (4) extract the vector that represents the anchor from the feature matrix and use it as a label vector; (5) train a classifier to predict whether the anchor is on or not (elastic net can be used); (6) the trained model can be calibrated using a validation set [15]; and (7) for an unseen patient where the anchor is not observed, the model is used to predict the likelihood of the anchor being on. This scheme is illustrated in the upper part of Fig. 1.

In more detail the framework is as follows. Assume that there are N patients and p observed variables. Let Y be the latent variable we want to predict for each patient. Let $\mathbf{x}^{-}$ represent all observed variables except for the anchor A. Assuming that we have found an anchor, A, the last three steps are as follows:

(5) Learn $P(A=1\,|\,\mathbf{x}^{-})$ using a classifier that provides a probabilistic output.
(6) Using a validation set, K, compute $C=\sum_{k\in K} P(A=1\,|\,\mathbf{x}_k^{-})/|K|$, where $\mathbf{x}_k^{-}$ is the data for patient k with the anchor removed.
(7) For an unseen patient, t, with $A=0$, predict $P(Y_t=1)=P(A=1\,|\,\mathbf{x}_t^{-})/C$. If $A=1$, $P(Y_t=1)=1$ because of the first property of anchors.

2.2. Proposed anchor framework solution

Fig. 1 illustrates how the *learning with anchors framework* and the proposed *anchor specification method* work. In the following we explain how to specify anchors using an exploratory data analysis and review the proposed classifier.

2.2.1. Predictive anchors via exploratory analysis

The two properties that anchors are supposed to satisfy are very strict and therefore it often turns out that it is difficult to find such anchors. However, in practice, the conditional independence property does not have to be completely satisfied [12]. On the other hand, if property 1 is relaxed, the false positive rate will automatically increase. With our proposed method, it is possible to define an anchor from free text by first searching for a *predictive* anchor – an observed variable that originally does not satisfy property 1, but by adding a certainty measure we can define a true anchor from it. This makes the AM framework applicable for a larger variety of problems.

The proposed method consists of four steps, which are explained below and is as follows.

In *step 1* one has to *identify a subset of relevant document types*, which requires domain knowledge, and create a feature representation.

In *step* 2, we *define a predictive anchor, B*, as a feature that is a surrogate for the latent variable of interest, and whose semantic meaning could vary in different settings in general, but restricted to the subset of relevant document types, it has a clearer meaning. We propose to use clustering to suggest predictive anchor candidates, B. For this reason it is important that the clustering method is robust and not sensitive to parameter choices. We therefore use the *kNN mode seeking consensus clustering* algorithm [16] (Appendix A), which has been shown to be robust on a variety of datasets. The idea with the clustering is to identify groups of patients of different health status. The visualization method t-SNE (Appendix B) is used, in combination with clinical knowledge, to further analyze the clustering results and thereby,

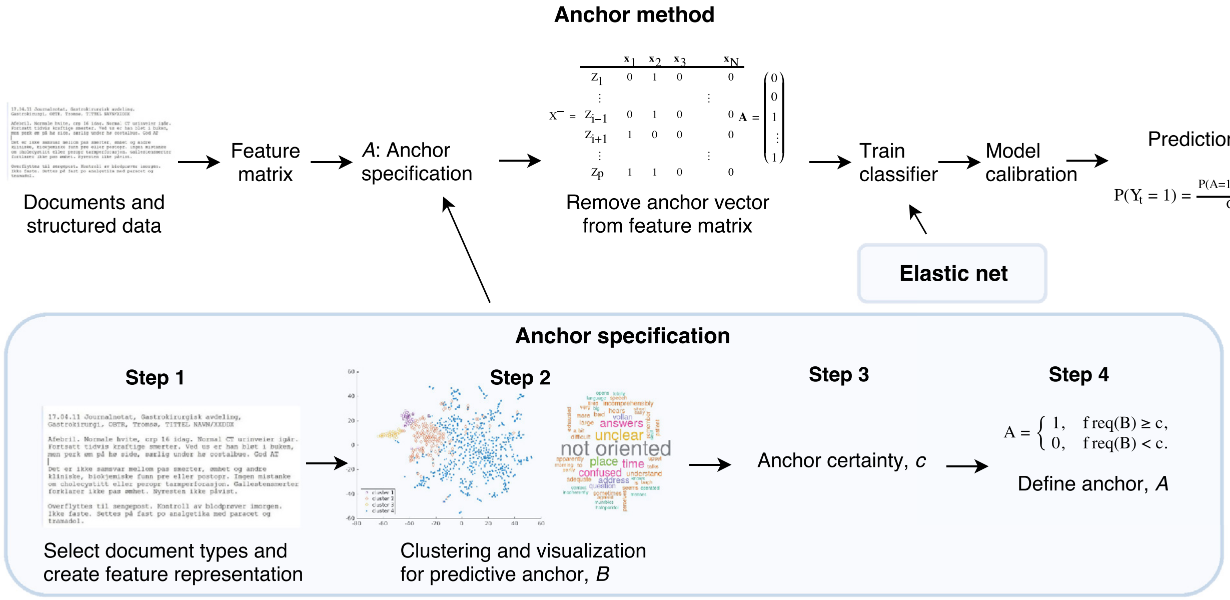

Anchor method
Documents and structured data
Feature matrix
A: Anchor specification
X^{-} =
x_1 x_2 x_3 x_N
Z_1 0 1 0 0
Z_{i-1} 0 1 0 0
Z_{i+1} 1 0 0 0
Z_P 1 1 0 0
$\mathbf{A} = (0, 0, 1, \vdots, 1)^T$
Remove anchor vector from feature matrix
Train classifier
Model calibration
Predictions
$P(Y_t = 1) = \frac{P(A=1 \mid x_t^{-})}{C}$
Elastic net
Anchor specification
Step 1
17.04.11 Journalnotat, Gastrokirurgisk avdeling, Gastrokirurgi, OBTR, Tromsø, TITTEL NAVN/XXDDX
Afebril. Normale hvite, crp 16 idag. Normal CT urinveier igår. Fortsatt tidvis kraftige smerter. Ved us er han bløt i buken, men perk øm på hø side, særlig under hø costalbue. God AT
Det er ikke samsvar mellom pas smerter, ømhet og andre kliniske, biokjemiske funn pre eller postopr. Ingen mistanke om cholecystitt eller peropr tarmperforasjon. Gallestensmerter forklarer ikke pas ømhet. Nyresten ikke påvist.
Overflyttes til sengepost. Kontroll av blodprøver imorgen. Ikke faste. Settes på fast po analgetika med paracet og tramadol.
Select document types and create feature representation
Step 2
60
40
20
0
-20
-40
-60
-80 -60 -40 -20 0 20 40 60
cluster 1
cluster 2
cluster 3
cluster 4
not oriented
unclear
answers
confused
place
time
address
question
understand
hears
vollan
incomprehensibly
sometimes
Clustering and visualization for predictive anchor, B
Step 3
Anchor certainty, c
Step 4
$A = \begin{cases} 1, & freq(B) \geq c, \\ 0, & freq(B) < c. \end{cases}$
Define anchor, A

Fig. 1. Schematic diagram of the method. The upper part of the figure explains how the learning with anchors framework works and the lower part illustrates the proposed anchor specification method.

identify groups containing patients with normal outcomes and groups of patients in worse condition. An example of a helpful tool for this task is to plot wordclouds of the most informative words for each cluster and then let the domain experts identify predictive anchor candidates from the wordclouds.

In *step 3*, we define the *certainty, c*, of the predictive anchor, *B*, as the lowest frequency that makes the predictive anchor trustworthy. Frequency in this setting means the frequency across the set of documents associated with a specific patient. We note that applying a global threshold of 1 basically corresponds to saying that the predictive anchor is an anchor. If one wants to make more conservative anchors, one can use a higher global threshold to reduce the probability of obtaining false positives. However, this definition also enables the opportunity to use a locally varying certainty. For example one could apply the proposed clustering and visualization techniques to stratify the data into groups with varying certainty.

Step 4 consists of using the term frequency restricted to the subset of relevant document types of the predictive anchor candidate *B* and the certainty measure *c* to *define the anchor A* as

$$A = \begin{cases} 1, & freq(B) \geq c, \\ 0, & freq(B) < c. \end{cases} \tag{1}$$

The idea behind the procedure is that, in general, some words are not anchors when they are written in a random document, but in certain documents it could be that the words are used in special settings and therefore are more trustworthy. It is also possible that some words, that in themselves cannot be trusted as anchors, could become more certain when they appear more than once.

2.2.2. Elastic net

In AM, a classifier that provides a probabilistic output is required. Halpern et al. applied l_2-regularized logistic regression. We propose to use the *elastic net* instead since it is robust in settings where the dimension is higher than the sample size [14]. A review is given here.

For a data point, $\mathbf{x}$, with an unknown label $y \in \{0, 1\}$, the logarithm of the ratio of the posterior probabilities $P(y = 0 \mid \mathbf{x})$ and $P(y = 1 \mid \mathbf{x})$ is modeled via a linear function, $w_0 + \mathbf{w}^T\mathbf{x}$. Given a training set, $\{(\mathbf{x}_k, y_k)\}$, the parameters $w = (w_0, \mathbf{w})$ are found by maximizing a regularized log-likelihood,

$$l(w) = \sum_{k=1}^{N_0} \log P(y_k = 0 \mid \mathbf{x}_k^{(0)}, w) + \sum_{k=1}^{N_1} \log P(y_k = 1 \mid \mathbf{x}_k^{(1)}, w) \\ - \lambda\Big((1-\alpha)\|\mathbf{w}\|_2^2 + \alpha\|\mathbf{w}\|_1\Big), \tag{2}$$

where $\lambda > 0$, $\alpha \in [0, 1]$, $\| \cdot \|_p$ is the l_p−norm and N_j is the number of data points in the jth class.

A “side-effect” of the elastic net is that it provides a ranked list of the most important features. The list can be used together with clinical knowledge to suggest new predictive anchors. One can then create a composite anchor out of the union of the individual anchors. Using multiple anchors could often be beneficial because it gives more positive examples for training.

3. Experiments and results

3.1. Data description

We wanted to use AM to detect whether a patient had developed PD or not. Hence, the latent variable of interest was $Y =$ *hasPD*. For this particular task we explored a data set extracted from the Department of Gastrointestinal Surgery (DGS) at the University Hospital of North Norway (UNN) from 2004 to 2012. In particular, we extracted EHRs for 7741 patients. The data include structured data such as ICD-10 codes describing the main diagnosis, age, sex, length of surgery, blood tests and health status, as well as free text from documents such as doctor’s notes, radiology reports and semi-structured nurses notes. The nurses notes are semi-structured since they are formulated as questionnaires with 12 bullet points and the nurses answer the questions using free text. For each patient the nurses write at least three notes every day; morning, afternoon and evening.

A clinician (author M.G.) made a list of surgeries of interest, basically consisting of major abdominal surgeries requiring general anesthesia. Based on this, 1138 patients who potentially could suffer from PD were selected into a cohort. In AM no labels are

Table 1
Summary of clustering results. The table shows the number of patients belonging to each cluster, the marker and color representing the cluster in the *t*-SNE map and certain keywords describing the different clusters.

Cluster	# of patients	Marker/color in Fig. 2	Keywords
1	34	Purple squares	*Disoriented* and *confused*
2	134	Red diamonds	*Adequate* and *communicates*
3	31	Yellow circles	*Sedated*
4	631	Blue dots	*Good mood* and *nothing to report*

needed, but to test the learning system, the clinician manually read the EHR for a subset consisting of 308 patients and found that 24 of them had PD after the surgery. Hence, the training set consisted of the remaining 830 unlabeled patients.

The remainder of this section is divided into two main subsections. In Subsection 3.2 we apply the proposed methodology to specify the first anchor. For the clarity of this exposition we leave some of the details for Appendix C, for example the specification of the other anchors. In Subsection 3.3 we apply AM and demonstrate the results of the methodology we have proposed.

3.2. Anchor specification using proposed method

As a first step, our clinicians suggested some words that potentially can be used as anchors; *delirium, delir, postoperative*. However, these words rarely or never occur in the EHR and cannot be used as anchors. We therefore employed our proposed method to specify anchors.

Step 1. Identification of relevant types of text documents. It was hypothesized by our clinicians that since the nurses take care of the patients continuously after the surgery, most likely information about PD would be discovered and reported by them. In particular, the bullet points in the semi-structured nurses notes related to communication/senses and knowledge/ development/ psychological are important descriptors of the mental status for the patient. Following this clinical knowledge, we chose to search for anchors in the free text only from the first two bullet points in the nurses notes.

A *term frequency - inverse patient frequency* (tf-ipf) representation was used instead of the more common *inverse document frequency* (idf) since we did not have access to each document for each patient [17]. However, the effect of the tf-ipf is the same, the value of the tf-ipf is proportional to the number of times a word appears for each patient, and is reduced by the frequency of the word for all patients. To further compensate for the redundancy in the features because of a lack of preprocessing (correlation between misspelled and correctly spelled words, etc.) principal component analysis (PCA) [18] was used to reduce the dimensionality. Based on a plot of the eigenvalues, we decided to use the 20 dimensions corresponding to the 20 top ranked eigenvalues. We notice that it is possible to compute both the tf-ipf and PCA feature representation also for new unseen patients.

Step 2. Identification of a predictive anchor. The kNN mode seeking consensus clustering algorithm was run for the 830 patients in the training set. Based on the dendrogram [19], the number of clusters was automatically chosen to 4. A low dimensional embedding of the data was created using t-SNE and the resulting mapping is shown in Fig. 2. The different colors and markers represent the different clusters. This figure verifies that the clustering results are reasonable; nearby points in the two dimensional space are clustered together. Table 1 provides a summary of the clustering results and more details are provided in Appendix C. Cluster 4 seems to contain patients with normal, positive outcomes. In cluster 1 words like *confused, unclear, disoriented* dominate, whereas in cluster 3 the theme is sedation and sedation drugs. In cluster 2, many of the most frequent words are related to speech and communication.

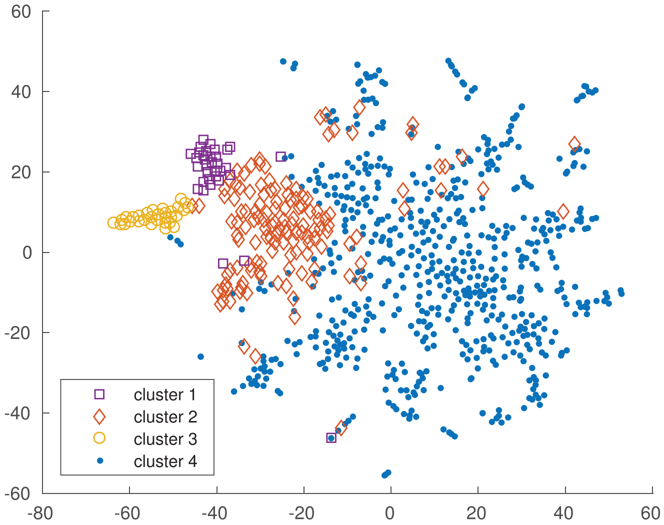
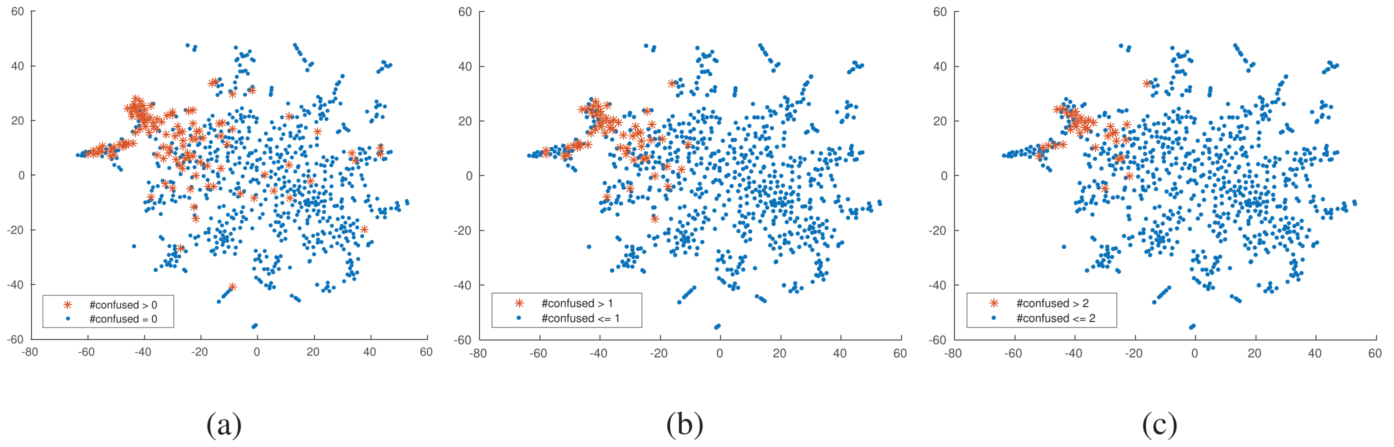


Fig. 2. Locations of the four clusters in the *t*-SNE map, obtained using the kNN mode seeking consensus clustering algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The fact that most of the high-frequent words in cluster 1 are words describing a patient's mental status, e.g. *disoriented, unclear, confused, messes* (see Fig. C.4 in Appendix C), indicates that it is natural to search for anchor candidates in this cluster. Clinicians suggested to use *confused* as the most evident word. Hence, we considered it as our first predictive anchor.

Step 3. Certainty assessment. Figs. 3a–c show the location in the two dimensional *t*-SNE map of the patients with different frequencies of the word *confused* in their nurses notes. We see that *confused* also appears for some patients in the cluster containing “normal” patients (cluster 4), but for many of these patients only once. Fig. 3c shows that patients that have a frequency of at least three for *confused* are concentrated around cluster 1 and 3. Higher frequency probably means that several nurses made the same observation more times. Hence, it is reasonable to assume that higher frequency means higher certainty. An underlying cluster assumption is that patients that belong to the same cluster are similar, and therefore one could argue that if *confused* appears for a patient that belong to cluster 1 or 3 only once, then it is probably not noise since the patient is supposed to be similar to patients for whom the word appear with a higher frequency.

Cluster 2 and 4 are larger and have higher variance. Some observations of *confused* in these clusters could be treated as noise and we therefore following clinicians' input defined the certainty



(a)

(b)

(c)

Fig. 3. The red stars show the location in the *t*-SNE map of the patients for whom the word *confused* appear in their nurses notes. In (a) it appears at least once, in (b) at least twice and in (c) at least three times. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

measure, c_1, as

$$c_1 = \begin{cases} 2, & \text{if the patient belongs to cluster 1 or 3,} \\ 3, & \text{if the patient belongs to cluster 2 or 4.} \end{cases}$$

Step 4. Definition of anchor. The anchor $A_1 = confused^*$ was then defined according to Eq. (1). The * means that the certainty measure is considered to define the anchor. Note that we probably could have chosen $c_1 = 1$ for patients in cluster 1 and 3 as well, but we rather want false negatives than false positives, and therefore make a more conservative choice for the certainty.

3.3. Classification based on specified anchors

3.3.1. Feature representation for classifier

A bag-of-words (BoW) model was used to represent the presence or absence of each different word that appeared in the clinical narrative [20]. Stop words and words that appeared for fewer than five patients were removed. The structured data, gender, type of surgery and some ICD-10 codes, were represented as booleans. Age was discretized into two intervals; older or younger than 65 years, since the literature emphasizes that especially patients older than 65 years have higher risk of getting PD [1,2,21]. For the American Society of Anesthesiologists (ASA) physical state grade, Scholz et al. [22] showed that for a score of at least three is a risk factor for PD. We therefore made a boolean by putting a threshold at three. In total this resulted in 20,949 different features.

3.3.2. Evaluation of proposed method

The R-package *glmnet* [23] was used to run the elastic net logistic regression. The regularization parameter λ was chosen using 10 fold cross-validation. We could also choose the other regularization parameter α using cross-validation. However, to ensure that we did not see the effect of different types of regularizations when comparing to baselines we chose $\alpha = 0.5$. To incorporate that our prior belief is that each variable is equally important, we ensured that the penalty applied equally to all variables by standardizing the binary variables to zero mean and standard deviation one [13]. We chose to measure performance using the area under the precision-recall curve (AUC-PR) because it captures the performance over the entire operating range and has been shown to work well on imbalanced data [24]. For this measure only the ordering of the scores is needed and therefore it was not necessary to tune the calibration coefficient. 95% confidence intervals (CIs) were evaluated using 100 bootstrap samples from the test set [25].

Table 2
Area under the PR-curve (AUC-PR) for three baselines and the proposed method. The two first anchors are chosen from all documents, whereas the two last one are chosen from the nurses notes (D). 95% confidence intervals are shown in parenthesis.

Anchor	*Confused*	*Confused* ×3	*Confused*+	A_1
AUC-PR 95% CI	0.507 (0.351, 0.652)	0.707 (0.541, 0.856)	0.503 (0.360, 0.637)	**0.803** (0.633, 0.918)

3.3.3. Demonstrating the effect of exploratory anchor selection

Section 3.2 introduced a text-based method for exploring anchors from EHRs using clinical knowledge, basically creating labels for a classifier (see Fig. 1). Here we demonstrate the effect of this exploratory anchor selection procedure by comparing to baselines where we applied AM with anchors not specified using the proposed method. To isolate the effect of the proposed anchor specification method we used the elastic net with $\alpha = 0.5$ also for the baseline. The effect of the classifier choice will be demonstrated in a later subsection.

The first baseline we compared to was AM with the anchor *confused*, where *confused* was specified by naively letting all patients where the word confused appeared in some of their documents have an anchor. We also applied AM to the anchor *confused* ×*3*, which was defined such that it is on if confused appeared at least three times in any of the documents. To demonstrate that it is not only a matter of choosing the correct document types, we compared to yet another baseline; we applied AM to the anchor *confused* + , which is on only if confused is observed in the free text only from the first two bullet points in the nurses notes.

Table 2 shows AUC-PR values and 95%-CIs obtained using the baselines and AM with the anchor A_1. We see that with the anchor A_1 an AUC-PR value of 0.803 was obtained, which is a considerable increase compared to the baselines.

By comparing to different baselines, we have now isolated the effects of (1) specifying the anchor only from the free text only from the first two bullet points in the nurses notes, and (2) specifying the anchor using our proposed methodology. We have shown that both steps are necessary to obtain a reasonably good performance.

3.3.4. Demonstrating the effect of document selection in feature representation for classifier

Clinical knowledge was used to suggest that anchor selection should come from the first two bullet points in the nurses notes.

Table 3
Lists of the top ranked features obtained using elastic net logistic regression with the anchors A_1 =*confused**, $A_2 = \{A_1$, *disoriented**}, $A_3 = \{A_1, A_2$, *unclear**} and $A_4 = \{A_1, A_2, A_3$, *haloperidol**}, respectively, as labels.

Rank	A_1	A_2	A_3	A_4
1	Disoriented	Unclear	Haloperidol	Perceive
2	Unclear	Eye contact	Messes	Messes
3	Clear	Responds	Responds	Responds
4	Case	Picking	Perceive	Picking
5	Bed	Hands	Indistinct	Indistinct
6	Messy	Indistinct	Agitated	Understand
7	Visions	Sleep	Remembers	Agitated
8	Eyes	Messy	Understand	Opens
9	Called	Messes	Hallucinated	Hallucinated
10	Fall	Bring	Messy	Messy
⋮				
24	Haloperidol	ASA score[1]	Forgets	Incomprehensible

[1] The only structured variable that appeared among the 25 top ranked variables.

Table 4
AUC-PR values obtained by adding more anchors. In the columns to the right we have also shown AUC-PR values obtained using l_2 regularized logistic regression as the classifier in AM.

	Elastic net		l_2-regularization	
	AUC-PR	95% CI	AUC-PR	95% CI
A_1	0.838	(0.694, 0.930)	0.692	(0.555, 0.844)
A_2	0.925	(0.851, 0.975)	0.815	(0.658, 0.916)
A_3	0.964	(0.911, 0.993)	0.910	(0.817, 0.975)
A_4	0.962	(0.923, 0.996)	0.915	(0.827, 0.998)
Supervised baseline	0.770	(0.652, 0.888)	0.580	(0.469, 0.691)

However, it was also hypothesized that the nurses notes likely is the most important data source for identifying information about PD. Surgical operation notes, doctor's notes, radiology reports, etc., will probably introduce more noise than relevant information. We therefore used clinical knowledge to reduce the number of data sources for the classifier to only structured data and free text from the nurses notes, which reduced the number of features to 2008. With this approach the AUC-PR value increased from 0.803 to 0.838, 95% CI (0.694, 0.930), with the anchor A_1. The CI is wide, but at least we see that the AUC-PR did not decrease.

3.3.5. Demonstrating the effect of adding more anchors and classifier choice

The elastic net outputs a ranked list of the most important features, which potentially could contain suggestions of new predictive anchors. Table 3 shows the ranked features provided by AM when A_1 was used as anchor (second column). Based on the ranking and clinical knowledge, we added the word *disoriented* as a predictive anchor.

Using the same certainty measure as for *confused* we defined the anchor *disoriented** according to Eq. (1) and created a composite anchor, A_2, as the union of *confused** and *disoriented**. Table 4 shows that AM with the anchor A_2 gave an AUC-PR value of 0.925, which is a considerable improvement.

Based on the ranking in the third column in Table 3 and clinical knowledge we added the word *unclear* as a predictive anchor. We defined the composite anchor, A_3, as the union of *confused**, *disoriented** and *unclear**. Table 4 shows that using A_3 we obtained an AUC-PR value of 0.964, which is a large improvement. Similarly, we created the anchor A_4 using the predictive anchor *haloperidol*. However, the AUC-PR value of 0.962 is very similar to the result obtained using the anchor A_3.

We see that the list of the top ranked features obtained using four anchors contains words like *messes, picking, indistinct, understand, agitated, hallucinated, visions* and *incomprehensible*. These words are definitely related to the mental status and potentially we could continue to add more anchors. However, we decided to not add more anchors because these candidates were not predictive enough and/or ambiguous.

As we mentioned above, since the sample size is lower than the dimensionality, we chose to use the elastic net. We compared to l_2 regularization by computing AUC-PR values and 95% CIs using the anchors A_1, A_2, A_3 and A_4. Table 4 shows that the elastic net is clearly beneficial. For example, for the anchor A_1 using l_2 regularization an AUC-PR value of 0.692 was obtained, whereas using the elastic net we got 0.838.

We also compared to a supervised baseline where we trained a classifier (elastic net) on the test set using 5-fold cross-validation. Mean AUC-PR and standard errors were calculated using bootstrap (creating 100 different 5-folds). Table 4 shows that with this a approach an AUC-PR value of 0.770 was obtained, considerably lower than for AM with two or more anchors.

4. Discussion

The proposed method is not fully automatic, it still requires some manual work. Therefore a natural question to ask is whether one actually gains something in terms of reduced labor intensity compared to manual label annotation. However, then one should keep in mind that while the latter must be done individually (e.g. by retrospectively reading the EHR for each patient one wants to label), in the former the manual work is done once and for all. Hence, the time spent on anchor annotation is actually not comparable to manual label annotation, and the difference becomes larger the larger the dataset is. We also want to emphasize that the proposed method is not fully generalizable to all diagnostic challenges. That being stated, it is easy to find other clinically interesting problems, both in retro- and prospective settings, where the method is applicable. One example is to use this method to pre-operatively identify malnourished patients [26]. In this case the notes regarding nutritional status would be particularly relevant. We also believe that the method is transferable to *predicting* patients at risk for post-operative complications. Potentially the method can be used in more general text-based settings, not necessarily in a clinical application.

4.1. Limitations and further work

AM falls into the classical PU-learning setting where one assumes that only the unlabeled dataset, *U*, is contaminated, whereas the positive set, *P*, is assumed to not contain false positives. In our approach we adapted the way of choosing the set, *P*, such that this assumption is not broken. However, recently, approaches where one assumes that also *P* can be contaminated, have been proposed [27,28]. The main ingredient in these methods is to use resampling on *P* to provide robustness against false positives. In [29] Claesen et al. showed that this approach can be used to predict whether a patient will start glucose-lowering pharmacotherapy. It will be interesting to use the anchors as proposed by Halpern et al. such that *P* is contaminated and thereafter applying an approach similar to the robust ensemble SVM, proposed in [28], in further work.

There are of course many challenges related to the unstructured text we have available [30,31]. Often the time spent on entering text into the EHRs is limited. A document could for example be a dictate of a conversation during a consultation. In other cases information could be recorded on an audio-recorder and then transcribed by a secretary at a later time. For these reasons incomplete sentences and typos are more common in medical text than in usual published text. In addition, there are words that contain digits, medical short forms and acronyms. Another challenge, special to Norwegian medical text, is related to the fact that there are two official languages in Norway and that a relatively large fraction of the employees at UNN are from other countries in Scandinavia. Some of them write in their own language, others have learned some Norwegian and therefore text written by them could be a mixture of several languages. We could have done more natural language processing to compensate for these challenges, but would have required a lot of effort since all the text mining software that is developed for English language does not exist for Norwegian language. However, there is ongoing work in our group trying to introduce less noisy conceptual features based on medical ontologies [32]. Since the AM framework do not make any assumption on how the features are represented, these can be included in further work.

Another limitation of our work is the quality of the gold standards. The clinicians created the gold standard of PD based on actual information in the EHR. Diagnosing PD was in part based on a consciousness assessment tool, the Observational Scale of Level of Arousal (OSLA) [33,34], as the EHR lacked sufficient data to use standardized delirium screening instruments. Hence, there is a risk that the gold standard could be biased.

Finally, we want to mention that in this work we have demonstrated the effects of the proposed methodology on a medium-sized dataset. The focus has been on diagnosing PD. However, in future work we would like to even more investigate the generalization abilities on bigger datasets and other problems. In particular, we will look at the problem of pre-operatively identifying and predicting malnourished patients at UNN.

5. Conclusion

We have adapted the learning with anchors framework to medical data gathered from a Norwegian university hospital. We introduced a new method for specifying anchors, providing the opportunity to obtain a labeled training set without manual label annotation. The importance of the proposed method was demonstrated on task where the aim was to detect postoperative delirium. By creating the labels in naive way we got an area under the PR-curve (AUC-PR) of 0.51, whereas by introducing our suggested improvements and adaptations we got an AUC-PR value of 0.96. We believe that the method potentially can be used in other clinical problems as well as in a more general text-based settings, not necessarily related with healthcare.

Conflict of interest

The authors have no conflict of interest regarding the study.

Acknowledgments

This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the *Next Generation Learning Machines* and IKTPLUSS grant no. 270738 *Deep Learning for Health*. Cristina Soguero-Ruiz is partially supported by projects TEC2016-75361-R from Spanish Government and by DTS17/00158 from Institute of Health Carlos III (Spain).

Appendix A. kNN mode seeking consensus clustering

In this section we give a brief description of the clustering method used for anchor specification. The clustering method belongs to the consensus framework, meaning that the same kNN-mode seeking algorithm is applied many times with a random k-parameter to a resampled version of the dataset each time. The kNN mode seeking algorithm [35,36] is a density based algorithm, similar to mean-shift [37,38], but the kernel density estimates are replaced by k-nearest neighbors (kNN) density estimates. This algorithm is used in each iteration in the consensus clustering. A detailed description of the framework is given in Algorithm 1. An advantage with this method is that there are no critical parameter choices such as number of clusters, bandwidth parameters, etc.

Algorithm 1 Consensus clustering using kNN mode seeking.

```
Input  Dataset $X$, range of $k$-values $K$, subsampling rate $p$ and number of clustering trials $M$.
Initialize $I$ and $S$ as $\mathbf{0}_{N\times N}$
for each clustering trial do
    Draw a random $k^*$ from $K$.
    Draw a random sample of size $pN$, $X^*$, from $X$.
    For each pair of data points in $X^*$ update the counter matrix $I$ by $I_{ij} = I_{ij} + 1$, where $(i, j)$ are the indices of the data points in $X$.
    Use kNN mode seeking with parameter $k^*$ to obtain a clustering of $X^*$.
    For each pair of data points in $X^*$, $(i, j)$, that belong to the same cluster, update $S$ by $S_{ij} = S_{ij} + 1$.
end for
Normalize the consensus matrix, $\mathcal{S}$, by dividing element-wise by the counter matrix; $\mathcal{S}_{ij} = \frac{S_{ij}}{I_{ij}}$
Create a dendrogram using average linkage.
Obtain the final clustering by selecting the cluster configuration with the longest lifetime.
Output  Clustering $C$ of $X$.
```

To assign cluster labels to new patients cannot be done using the kNN mode seeking consensus clustering algorithm since there exist no out-of-sample mapping. However, since the clustering algorithm is based on a k-nearest neighbors search, one could assign cluster labels to new data points using a kNN classifier [13].

Appendix B. *t*-distributed stochastic neighbor embedding (*t*-SNE)

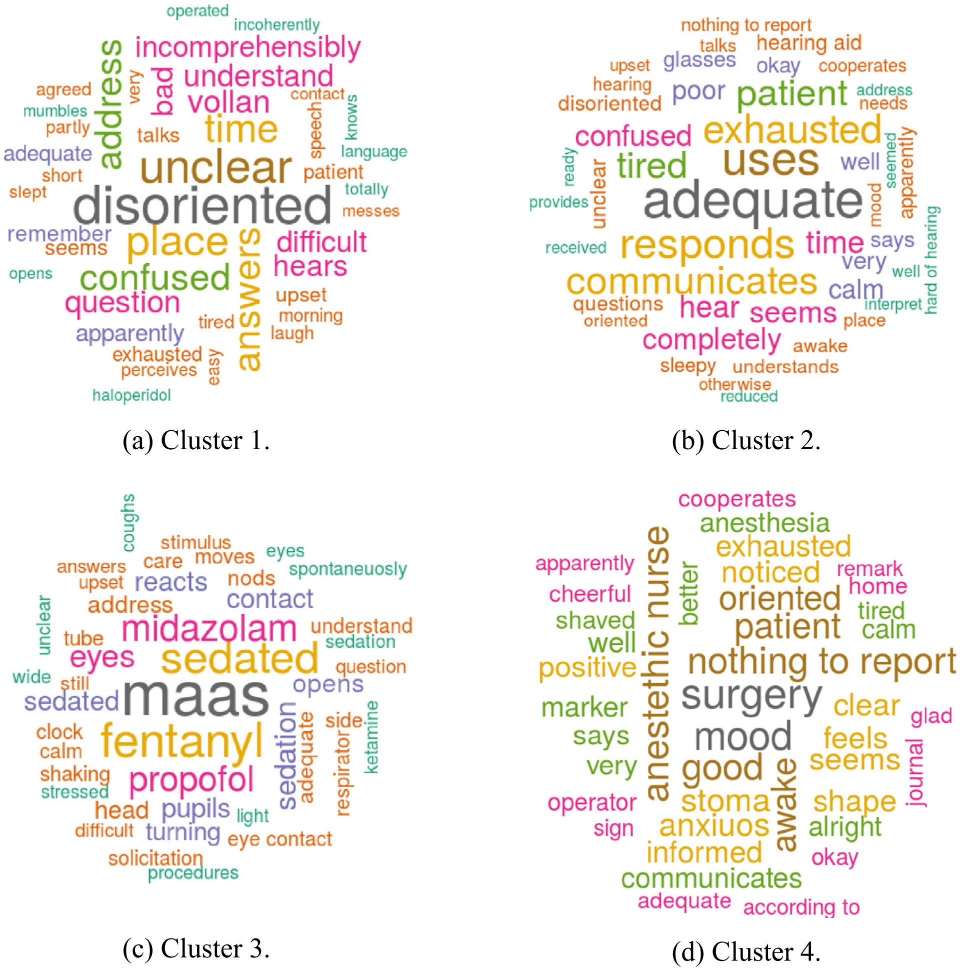
The t-SNE algorithm is one of the most well-established techniques for visualizing high-dimensional data in two or three dimensions. It has shown robustness and has become the state-of-the-art visualization method for many different data types [39]. The algorithm has the property that it creates a single map that reveals structure in the data at many different scales. The objective in this algorithm, which consists of two main stages, is to map points, $\mathbf{x} \in \mathbb{R}^p$, in a high dimension, p, to a low dimension d, $\mathbf{v} \in \mathbb{R}^d$ [39]. Firstly, one estimates a joint probability distribution, $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2N}$, in the original, high-dimensional space over each pair of data points using a Gaussian kernel

$$p_{j|i} = \frac{e^{-\frac{1}{2\sigma_i^2}||\mathbf{x}_i-\mathbf{x}_j||^2}}{\sum_{k\neq i} e^{-\frac{1}{2\sigma_i^2}||\mathbf{x}_i-\mathbf{x}_k||^2}}. \tag{B.1}$$

Hence, p_{ij} represents the similarity between the data points $\mathbf{x}_i$ and $\mathbf{x}_j$. Secondly, the heavy-tailed Student t-distribution with one degree of freedom is used to model similarities in the low-dimensional space as

$$q_{ij} = \frac{(1+||\mathbf{v}_i-\mathbf{v}_j||^2)^{-1}}{\sum_{k\neq l}(1+||\mathbf{v}_k-\mathbf{v}_l||^2)^{-1}}. \tag{B.2}$$

Then, the locations of the points $\mathbf{v}_i$ are found by minimizing the Kullback–Leibler divergence, $KL(P\,||\,Q) = \sum_{i\neq j} p_{ij}\log(p_{ij}q_{ij}^{-1})$, using gradient descent. P and Q are the joint probability distributions over all data points in the high- and low-dimensional space, respectively.



(a) Cluster 1.

(b) Cluster 2.

(c) Cluster 3.

(d) Cluster 4.

Fig. C4. By applying the clustering procedure as described in Section 3.2 to the training data four clusters were obtained. In this figure we have shown the most important features in each cluster. The size of each word corresponds to their relative tf-ipf values.

Table C5
Fraction of patients in each cluster for whom the word *confused* appeared at least 1,2,3 and 4 times, respectively, in their nurses notes.

Frequency	1	2	3	4
Cluster 1	0.7059	0.5000	0.3529	0.2647
Cluster 2	0.3507	0.2090	0.1343	0.0970
Cluster 3	0.4839	0.2903	0.1290	0.0323
Cluster 4	0.0349	0.0063	0.0016	0
Overall	0.1301	0.0699	0.0422	0.0277

Table C6
Fraction of patients in each cluster for whom the word *disoriented* appeared at least 1,2,3 and 4 times, respectively, in their nurses notes.

Frequency	1	2	3	4
Cluster 1	0.9117	0.7941	0.6470	0.5882
Cluster 2	0.2910	0.2089	0.0970	0.0522
Cluster 3	0.5161	0.2903	0.2258	0.1935
Cluster 4	0.0285	0.0031	0	0
Overall	0.1253	0.0795	0.0506	0.0397

Appendix C. Anchor specification

In addition to the wordclouds (Fig. C.4) and the *t*-SNE map shown in Fig. 3a–c, Table C.5 contains information related to the word *confused* that was used to assess the certainty of this predictive anchor. For example Table C.5 shows that for 35% of the patients in cluster 1 the frequency of *confused* is at least three, whereas for 71% of the patients the frequency is at least one.

C1. Adding more anchors

As we described in Section 3.3.5, we used the ranking provided by AM with the anchor A_1 and clinical knowledge, to add the word *disoriented* as a predictive anchor. By looking at the wordcloud in Fig. C.4 and Table C.6, we observe that the top ranked word *disoriented* also is very frequent in cluster 1. This is another reason for using *disoriented* as the next predictive anchor.

The semantic meanings of disoriented and confused are quite similar. Moreover, the *t*-SNE plot (not shown here) of the patients with the word *disoriented* in their nurses notes is very similar to the *t*-SNE plot corresponding to *confused* shown in Fig. 3. Therefore we decided to use (almost) same certainty measure for these two predictive anchors,

$$c_2 = \begin{cases} 2, & \text{if the patient belongs to cluster 1 or 3, or other} \\ & \text{predictive anchors appear at least twice.} \\ 3, & \text{otherwise.} \end{cases}$$

We defined the anchor *disoriented** according to Eq. (1) and created a composite anchor, A_2, as the union of *confused** and *disoriented**.

The two other anchors, *unclear** and *haloperidol**, were added in a very similar fashion. From them we defined the composite

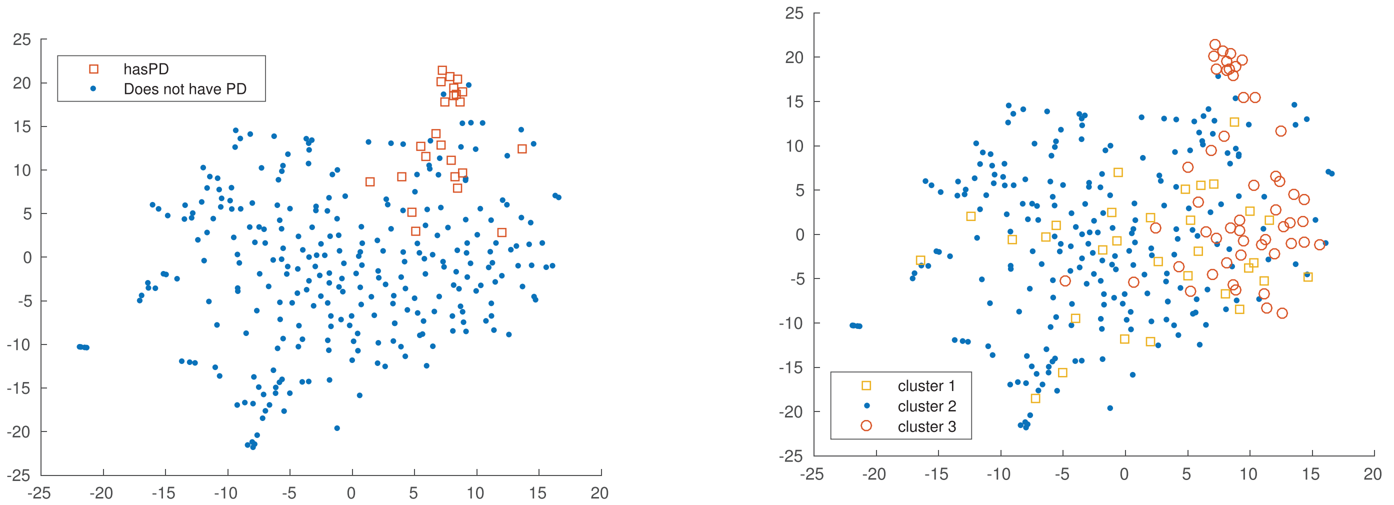


Fig. D5. Plots of the *t*-SNE mapping of the test set. (a) Locations of the patients with PD in a two dimensional t-SNE map. The red squares correspond to patients that have PD, and the blue dots to patients that do not have PD. (b) Locations of the three clusters in a two dimensional t-SNE map. Yellow squares correspond to patients that belong to cluster 1, red circles to patients in cluster 2, blue dots to patients in cluster 3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

anchor, A_3, as the union of *confused**, *disoriented** and *unclear** and A_4 as the union of all four.

Appendix D. Classification based on clustering of test set

By looking at the clustering results shown as word clouds in Fig. C.4 it seems like cluster 1 contains many words related to PD and this might indicate that doing classification only based on the clustering results could solve the problem we have considered in this paper. We investigate this further here.

We applied the clustering algorithm to the labeled test set alone and obtained three clusters. A *t*-SNE mapping of the data in two dimensions is shown in Fig. D.5. By looking at the high-frequent words in the different clusters, we also in this case found a cluster containing many words related to the mental status of the patient. Based on these results we classified all patients in cluster 3 as *has PD* and got an AUC-PR value of 0.456 with a 95% CI (0.436, 0.483). These results are not very convincing and we conclude that it is meaningful to apply the AM for this problem.

References

[1] S. Deiner, J.H. Silverstein, Postoperative delirium and cognitive dysfunction, BJA: Br. J. Anaesth. 103, suppl. 1 (2009) i41–i46, doi:10.1093/bja/aep291.

[2] S.K. Inouye, T. Robinson, C. Blaum, J. Busby-Whitehead, M. Boustani, A. Chalian, S. Deiner, D. Fick, L. Hutchison, J. Johanning, M. Katlic, J. Kempton, M. Kennedy, E. Kimchi, C. Ko, J. Leung, M. Mattison, S. Mohanty, A. Nana, D. Needham, K. Neufeld, H. Richter, Postoperative delirium in older adults: best practice statement from the american geriatrics society, J. Am. Coll. Surg. 220 (2) (2015) 136–148, doi:10.1016/j.jamcollsurg.2014.10.019.

[3] H. Böhner, T.C. Hummel, U. Habel, C. Miller, S. Reinbott, Q. Yang, A. Gabriel, R. Friedrichs, E.E. Müller, C. Ohmann, et al., Predicting delirium after vascular surgery, Ann. Surg. 238 (2003) 149–156.

[4] Y. Morimoto, M. Yoshimura, K. Utada, K. Setoyama, M. Matsumoto, T. Sakabe, Prediction of postoperative delirium after abdominal surgery in the elderly, J. Anesth. 23 (1) (2009) 51–56, doi:10.1007/s00540-008-0688-1.

[5] J.W. Raats, W.A. van Eijsden, R.M.P.H. Crolla, E.W. Steyerberg, L. van der Laan, Risk factors and outcomes for postoperative delirium after major surgery in elderly patients, PLOS ONE 10 (8) (2015) 1–12, doi:10.1371/journal.pone.0136071.

[6] C. Soguero-Ruiz, K. Hindberg, I. Mora-Jiménez, J.L. Rojo-Álvarez, S.O. Skrovseth, F. Godtliebsen, K. Mortensen, A. Revhaug, R.-O. Lindsetmo, K.M. Augestad, R. Jenssen, Predicting colorectal surgical complications using heterogeneous clinical data and kernel methods, J. Biomed. Inf. 61 (2016) 87–96, doi:10.1016/j.jbi.2016.03.008.

[7] S. Huang, P. LePendu, S. Iyer, M. Tai-Seale, D. Carrell, N.H. Shah, Toward personalizing treatment for depression: predicting diagnosis and severity, JAMIA 21 (6) (2014) 1069–1075, doi:10.1136/amiajnl-2014-002733.

[8] S. Dua, U.R. Acharya, P. Dua, Machine Learning in Healthcare Informatics, 56, Springer, 2014.

[9] P. Yadav, M. Steinbach, V. Kumar, G. Simon, Mining electronic health records (EHR): a survey, Tech. Rep. (2015).

[10] C. Shivade, P. Raghavan, E. Fosler-Lussier, P.J. Embi, N. Elhadad, S.B. Johnson, A.M. Lai, A review of approaches to identifying patient phenotype cohorts using electronic health records, J. Am. Med. Inf. Assoc. 21 (2) (2013) 221–230.

[11] Y. Halpern, Y. Choi, H. Steven, D. Sontag, Using anchors to estimate clinical state without labeled data, in: AMIA Annual Symposium Proceedings, 2014, pp. 606–615.

[12] Y. Halpern, S. Horng, Y. Choi, D. Sontag, Electronic medical record phenotyping using the anchor and learn framework, J. Am. Med. Inf. Assoc. (2016), doi:10.1093/jamia/ocw011.

[13] T.J. Hastie, R.J. Tibshirani, J.H. Friedman, The Elements of Statistical Learning : Data Mining, Inference, and Prediction, Springer series in statistics, Springer, New York, 2009. Autres impressions : 2011 (corr.), 2013 (7e corr.)

[14] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc., Series B 67 (2005) 301–320, doi:10.1111/j.1467-9868.2005.00503.x.

[15] C. Elkan, K. Noto, Learning classifiers from only positive and unlabeled data, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2008, 2008, pp. 213–220, doi:10.1145/1401890.1401920.

[16] J.N. Myhre, K.Ø. Mikalsen, S. Løkse, R. Jenssen, Consensus clustering using kNN mode seeking, in: Image Analysis, Springer, 2015, pp. 175–186, doi:10.1007/978-3-319-19665-7_15.

[17] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manag. 24 (5) (1988) 513–523.

[18] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2002.

[19] J.C. Gower, G.J.S. Ross, Minimum spanning trees and single linkage cluster analysis, J. R. Stat. Soc. Series C (Appl. Stat.) 18 (1) (1969), doi:10.2307/2346439.

[20] C. Soguero-Ruiz, K. Hindberg, J. Rojo-Alvarez, S.O. Skrovseth, F. Godtliebsen, K.E. Mortensen, A. Revhaug, R.-O. Lindsetmo, K.M. Augestad, R. Jenssen, Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records., IEEE J Biomed. Health Inform. 20 (5) (2016) 1404–1415, doi:10.1109/JBHI.2014.2361688.

[21] T.N. Robinson, B. Eiseman, Postoperative delirium in the elderly: diagnosis and management., Clin. Interv. Aging 3 (2) (2008) 351–355.

[22] A.F.M. Scholz, C. Oldroyd, K. McCarthy, T.J. Quinn, J. Hewitt, Systematic review and meta-analysis of risk factors for postoperative delirium among older patients undergoing gastrointestinal surgery, Br. J. Surg. 103 (2) (2016) e21–e28, doi:10.1002/bjs.10062.

[23] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, J. Stat. Softw. 33 (1) (2010) 1–22, doi:10.1145/1273496.1273501.

[24] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proceedings of the 23rd International Conference on Machine Learning, in: ICML '06, ACM, 2006, pp. 233–240, doi:10.1145/1143844.1143874.

[25] B. Efron, The bootstrap and modern statistics, J. Am. Stat. Assoc. 95 (452) (2000) 1293–1296.

[26] N. Ward, Nutrition support to patients undergoing gastrointestinal surgery, Nutr. J. 2 (1) (2003) 1–5, doi:10.1186/1475-2891-2-18.

[27] F. Mordelet, J.-P. Vert, A bagging SVM to learn from positive and unlabeled examples, Pattern Recognit. Lett. 37 (2014) 201–209, doi:10.1016/j.patrec.2013.06.010.

[28] M. Claesen, F.D. Smet, J.A. Suykens, B.D. Moor, A robust ensemble approach to learn from positive and unlabeled data using SVM base models, Neurocomputing 160 (2015) 73–84, doi:10.1016/j.neucom.2014.10.081.

[29] M. Claesen, F.D. Smet, P. Gillard, C. Mathieu, B.D. Moor, Building classifiers to predict the start of glucose-lowering pharmacotherapy using Belgian health expenditure data, CoRR abs/1504.07389 (2015).

[30] S.M. Meystre, G.K. Savova, K.C. Kipper-Schuler, J.F. Hurdle, Extracting information from textual documents in the electronic health record: a review of recent research., Yearb. Med. Inf. (2008) 128–144.

[31] P. Jensen, L. Jensen, S. Brunak, Mining electronic health records: towards better research applications and clinical care., Nat. Rev. Genet. 13 (6) (2012) 395–405, doi:10.1038/nrg3208.

[32] C. Soguero-Ruiz, L. Lechuga-Suarez, I. Mora-Jiménez, J. Ramos-Lopez, O. Barquero-Perez, A. Garcia-Alberola, J.L. Rojo-Alvarez, Ontology for heart rate turbulence domain from the conceptual model of snomed-ct, IEEE Trans. Biomed. Eng. 60 (7) (2013) 1825–1833.

[33] B. Neerland, M. Ahmed, L. Watne, K. Hov, T. Wyller, New consciousness scale for delirium., Tidsskrift for den Norske lægeforening: tidsskrift for praktisk medicin, ny række 134 (2) (2014) 150.

[34] Z. Tieges, A. McGrath, R.J. Hall, A.M. MacLullich, Abnormal level of arousal as a predictor of delirium and inattention: an exploratory study, Am. J. Geriatr. Psychiatry 21 (12) (2013) 1244–1253.

[35] W.L. Koontz, P.M. Narendra, K. Fukunaga, A graph-theoretic approach to nonparametric cluster analysis, Comput., IEEE Trans. 100 (9) (1976) 936–944. 710.1109/TC.1976.1674719

[36] R.P. Duin, A.L. Fred, M. Loog, E. Pekalska, Mode Seeking Clustering by kNN and Mean Shift Evaluated, in: Structural, Syntactic, and Statistical Pattern Recognition, Springer, 2012, pp. 51–59.

[37] Y. Cheng, Mean shift, mode seeking, and clustering, Pattern Analysis and Machine Intelligence, IEEE Transactions on 17 (8) (1995) 790–799, doi:10.1109/34.400568.

[38] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, Pattern Anal. Mach. Intell., IEEE Trans. 24 (5) (2002) 603–619, doi:10.1109/34.1000236.

[39] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.

Appendix A

Missing data

Missing data are common in many real-world applications, and might appear for various reasons. For example, a sensor may stop working after some time and therefore the measurements are censored, a participant in a survey may forget to answer a question in a questionnaire, or a physician might not order all possible blood tests for a patient because she thinks that some of them are less relevant.

These few selected examples mentioned have in common that they lead to *incomplete datasets*, but the *missing data patterns* may vary. The description of which values in an incomplete dataset are missing and which are not, is commonly referred to as the missing data pattern. In the sensor example, all data that should have been measured after the component stopped working will be missing, whereas the order of unanswered questions in the questionnaire might be completely random, and therefore the missing data patterns are different in these two examples.

From a statistical perspective, an incomplete dataset is generated by a composite random process, where there is one distribution from which complete data are sampled and one missing data process that determines which values are missing, i.e. the missing data pattern. Traditionally, the underlying (random) process that generates the particular patterns that are observed in the data and the relationship between missing data, is referred to as the *missing data mechanism* (Little and Rubin, 2014).

The missing data mechanism is obviously highly dependent on the problem at hand, but according to the theory of Rubin (1976), there is one fundamental distinction to make when categorizing missing data mechanisms, namely whether data are *missing at random*, or equivalently, whether the probability that a variable is missing is dependent on the value of the variable. Next, we briefly review this theory introduced by Rubin.

Missing data mechanisms

Notation. Let $X = \{x_n \mid x_n \in \mathbb{R}^d\}_{n=1}^N$ denote a dataset consisting of N d-dimensional vectors, and X_{obs} and X_{miss} denote the observed and missing entries of the data, respectively. Further, we define a binary missing data indicator as $R = \{r_n \mid r_n \in \{0,1\}^d\}_{n=1}^N$, where $r_{ni} = 1$ if the entry x_{ni} is missing and $r_{ni} = 0$ otherwise. The missing data mechanism is assumed be described by a conditional distribution, parametrized by θ, $p(R \mid X,\, \theta)$.

There are three types of missing data mechanisms, namely Missing Completely At Random (MCAR), Missing At Random (MAR), Missing Not At Random (MNAR).

If the missing data mechanism is MCAR, then the probability of elements being missing does not depend on their values, i.e. given any $X \in \mathbb{R}^{N\times d}$ and set of parameters θ,

$$p(R \mid X,\, \theta) = p(R \mid \theta). \tag{A.1}$$

If data are MAR the distribution of R depends only on the observed part of X, namely X_{obs}, i.e.

$$p(R \mid X,\, \theta) = p(R \mid X_{obs},\, \theta) \tag{A.2}$$

for all possible configurations of X_{miss} and θ. On the other hand, the MNAR mechanism appears if $p(R \mid X,\, \theta)$ depends on the values of the missing elements (X_{miss}).

Methods for dealing with missing data

Missing data handling methods have been subject to extensive research since Rubin published his famous paper in 1976. The amount of published work on this topic is therefore tremendous. For this reason it is not possible to provide a complete overview of the field, but we will in the following section describe some common methods for dealing with incomplete data.

Complete-case analysis Complete case analysis, also known as listwise or casewise deletion, is probably the most straightforward and simple way to treat missing data. In this approach, all cases (data points) that contain missing values are simply discarded from the dataset. By doing so, one obtains a new complete, rectangular dataset that is smaller than the original incomplete dataset in terms of number of data points.

This approach could work if a large dataset with a small fraction of missing data is available, but even in this scenario one could end up with biased results if the missingness mechanism is not MCAR. Under other circumstances, complete-case analysis leads to a lot of discarded information and models with low statistical power (Rubin, 1976; Schafer, 1997). Moreover, if test data points are not completely observed, the method cannot be applied. Despite these shortcomings, complete-case analysis is the most commonly applied method for dealing with missing data in clinical trials (Bell et al., 2014).

As an alternative, one can employ available case analysis, which is also commonly referred to as pairwise deletion. This strategy can be applied in methods where not all variables are analyzed at the same time. Hence, it cannot directly be applied in most standard classifiers, but can for example be used if one wants to estimate the covariance between to variables. In this case one would use all data points where these two variables are present in order to do the estimation. For more details on complete-case analysis we point the interested reader to (Schafer and Graham, 2002).

Single imputation Imputation methods impute the missing values with estimated values. In the so-called *single imputation* methods, in contrast to *multiple imputation*, each missing value is imputed only once. One of the most well-known such methods is mean imputation. Given multivariate vectorial data and a missing value in a variable, one simply just fill in the mean of that variable across all data points where that variable is not missing. There are several different variations of this method. For example, in a classification setting, one can restrict oneself to the mean in each class, whereas for multivariate time series, instead of considering the population mean, one can fill in the mean of the previous known values or the mean of all known values for the variate.

Other related imputation methods include filling in other statistics such as the median, maximum, minimum, or one could even fill in zeros for all missing data. The latter obviously introduces a bias into the data, but can sometimes prove efficient (Hu et al., 2017). Moreover, these imputation do not account for the uncertainty and variability in the data since all missing values are replaced with identical values and therefore e.g. distances and standard errors could be underestimated (Little and Rubin, 2014). An example of a situation where single imputation could prove more efficient is when more or less all data points in the dataset have a few missing elements, but the overall fraction of missing data is low (Schafer and Graham, 2002).

The methods described so far utilize global properties of the dataset (except row mean imputation). However, a broad class of the single imputation methods are only based on local information. In this regard, the most extreme methods are those where missing values in a data point are filled in only based on the observed values in that single data point. An example of such a method is the row mean imputation method discussed in the previous paragraph. For time series data, other examples include smoothing and interpolation techniques such as Kalman filters, linear interpolation, and the well-known last observation carried forward scheme that imputes the last non-missing value for the following missing values. Within this category, one also finds Fourier transform based imputation (Rahman et al., 2015). Other methods that are based on local information are those that impute based on the neighboring data points. kNN based methods use the observed variables to identify the k most similar completely observed data points and thereafter find an estimate for the missing value by taking a weighted average of the corresponding value among the neighbors (García-Laencina et al., 2010). The weights can be computed using e.g. different types of distances to the neighbors (Troyanskaya et al., 2001).

For more detailed overview of these imputation methods we refer the interested reader to Donders et al. (2006).

Multiple imputation Multiple imputation (Rubin, 2004) is the only imputation method that can account for uncertainty in the estimated replacement values for the missing data. As the name of the method suggests, the imputation is done by, for each missing element, finding a set of multiple (M) possible values to fill in. Hence, M different complete datasets are created. In order to capture the uncertainty about the correct values to impute for the missing values an appropriate model that accounts for random variation is chosen.

Bibliography

Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* John Wiley & Sons, Inc., New York, NY, USA.

Aczon, M., Ledbetter, D., Ho, L., Gunny, A., Flynn, A., Williams, J., and Wetzel, R. (2017). Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks. *arXiv preprint arXiv:1701.06675.*

Adavanne, S. and Virtanen, T. (2017). Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network. *arXiv preprint arXiv:1710.02998.*

Agarwal, V., Podchiyska, T., Banda, J. M., et al. (2016). Learning statistical models of phenotypes using noisy labeled training data. *Jour. American Medical Informatics Ass.*, 23(6):1166–1173.

Albers, D., Elhadad, N., Claassen, J., et al. (2018). Estimating summary statistics for electronic health record laboratory data for use in high-throughput phenotyping algorithms. *Journal of Biomedical Informatics*, 78:87 – 101.

Albers, D. J., Elhadad, N., Tabak, E., Perotte, A., and Hripcsak, G. (2014). Dynamical phenotyping: using temporal analysis of clinically collected physiologic data to stratify populations. *PloS one*, 9(6):e96443.

Albers, D. J., Levine, M., Gluckman, B., Ginsberg, H., Hripcsak, G., and Mamykina, L. (2017). Personalized glucose forecasting for type 2 diabetes using data assimilation. *PLoS computational biology*, 13(4):e1005232.

AMA (2007). *Current procedural terminology: CPT.* American Medical Association.

Anderson, L., Aubourg, É., Bailey, S., Beutler, F., Bhardwaj, V., et al. (2014). The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: baryon acoustic oscillations in the Data Releases 10 and 11 Galaxy samples. *Monthly Notices of the Royal Astronomical Society*, 441(1):24–62.

Angelidis, S. and Lapata, M. (2018). Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association of Computational Linguistics*, 6:17–31.

Angluin, D. and Laird, P. (1988). Learning from noisy examples. *Machine Learning*, 2(4):343–370.

Arias-Castro, E., Mason, D., and Pelletier, B. (2016). On the Estimation of the Gradient Lines of a Density and the Consistency of the Mean-Shift Algorithm. *Journal of Machine Learning Research*, 17(43):1–28.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404.

Aslam, J. A. and Decatur, S. E. (1996). On the sample complexity of noise-tolerant learning. *Information Processing Letters*, 57(4):189–195.

Bai, T., Zhang, S., Egleston, B. L., and Vucetic, S. (2018). Interpretable representation learning for healthcare via capturing disease progression through time. In *Proc. of the 24th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '18, pages 43–51, New York, NY, USA.

Bair, E. (2013). Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(5):349–361.

Bajor, J. M., Mesa, D. A., Osterman, T. J., and Lasko, T. A. (2018). Embedding complexity in the data representation instead of in the model: A case study using heterogeneous medical data. *arXiv preprint arXiv:1802.04233*.

Banda, J. M., Halpern, Y., Sontag, D., and Shah, N. H. (2017). Electronic phenotyping with aphrodite and the observational health sciences and informatics (ohdsi) data network. *AMIA Summits on Translational Science Proceedings*, 2017:48.

Banda, J. M., Seneviratne, M., Hernandez-Boussard, T., and Shah, N. H. (2018). Advances in electronic phenotyping: From rule-based definitions to machine learning models. *Annual Review of Biomedical Data Science*, 1(1):53–68.

Basu, S., Bilenko, M., and Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68. ACM.

Basu, S., Davidson, I., and Wagstaff, K. (2008). *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press.

Baydogan, M. G. and Runger, G. (2016). Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery*, 30(2):476–509.

Baytas, I. M., Xiao, C., Zhang, X., Wang, F., Jain, A. K., and Zhou, J. (2017). Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 65–74, New York, NY, USA. ACM.

Beaulieu-Jones, B. K., Greene, C. S., et al. (2016). Semi-supervised learning of the electronic health record for phenotype stratification. *Journal of biomedical informatics*, 64:168–178.

Belkin, M. and Niyogi, P. (2003). Using manifold stucture for partially labeled classification. In *Advances in neural information processing systems*, pages 953–960.

Bell, M. L., Fiero, M., Horton, N. J., and Hsu, C.-H. (2014). Handling missing data in RCTs; a review of the top medical journals. *BMC medical research methodology*, 14(1):118.

Bellazzi, R. and Zupan, B. (2008). Predictive data mining in clinical medicine: current issues and guidelines. *International journal of medical informatics*, 77(2):81–97.

Ben-Hur, A. and Noble, W. S. (2005). Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(suppl_1):i38–i46.

Bengio, Y., Delalleau, O., and Le Roux, N. (2006). *Label Propagation and Quadratic Criterion*. The MIT Press.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bennett, C. C. (2012). Utilizing rxnorm to support practical computing applications: Capturing medication history in live electronic health records. *Journal of Biomedical Informatics*, 45(4):634 – 641. Translating Standards into Practice: Experiences and Lessons Learned in Biomedicine and Health Care.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 359–370. AAAI Press.

Bhardwaj, G. (2018). How AI is transforming the future of healthcare. *Forbes Technology Council (https://www.forbes.com/sites/forbestechcouncil/2018/01/30/how-ai-is-transforming-the-future-of-healthcare)*.

Bi, Y. and Jeske, D. R. (2010). The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise. *Journal of Multivariate Analysis*, 101(7):1622–1637.

Bilmes, J. A. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, 4(510):126.

Birant, D. and Kut, A. (2007). St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221.

Birkhead, G. S., Klompas, M., and Shah, N. R. (2015). Uses of electronic health records for public health surveillance to advance public health. *Annual review of public health*, 36:345–359.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Boag, W., Suresh, H., Celi, L. A., Szolovits, P., and Ghassemi, M. (2018). Racial disparities and mistrust in end-of-life care. *arXiv preprint arXiv:1808.03827*.

Bogojeska, J., Stöckel, D., Zazzi, M., Kaiser, R., Incardona, F., Rosen-Zvi, M., and Lengauer, T. (2012). History-alignment models for bias-aware prediction of virological response to hiv combination therapy. In *Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 118–126. Journal of Machine Learning Research.

Boland, M. R., Tatonetti, N. P., and Hripcsak, G. (2015). Development and validation of a classification approach for extracting severity automatically from electronic health records. *Journal of Biomedical Semantics*, 6(1):14.

Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6(Sep):1579–1619.

Botsis, T., Hartvigsen, G., Chen, F., and Weng, C. (2010). Secondary use of ehr: data quality issues and informatics opportunities. *Summit on Translational Bioinformatics*, 2010:1.

Bouveyron, C. and Girard, S. (2009). Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recognition*, 42(11):2649–2658.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Brodley, C. E. and Friedl, M. A. (1999). Identifying mislabeled training data. *J. Artif. Int. Res.*, 11(1):131–167.

Brown, S.-A. (2016). Patient similarity: Emerging concepts in systems and precision medicine. *Frontiers in Physiology*, 7:561.

Burges, C. J. et al. (2010). Dimension reduction: A guided tour. *Foundations and Trends® in Machine Learning*, 2(4):275–365.

Busse, R., Geissler, A., and Quentin, W. (2011). *Diagnosis-Related Groups in Europe: Moving towards transparency, efficiency and quality in hospitals.* McGraw-Hill Education (UK).

Caballero Barajas, K. L. and Akella, R. (2015). Dynamically modeling patient's health state from electronic medical records: A time series approach. In *Proceedings of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 69–78. ACM.

Cahan, A. and Cimino, J. J. (2016). Visual assessment of the similarity between a patient and trial population. *Applied clinical informatics*, 7(02):477–488.

Cai, D., He, X., and Han, J. (2007). Semi-supervised discriminant analysis. In *Proc. Int. Conf. Computer Vision (ICCV'07)*.

Camps-Valls, G. and Bruzzone, L. (2009). *Kernel methods for remote sensing data analysis.* John Wiley & Sons.

Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308.

Carroll, R. J., Eyler, A. E., and Denny, J. C. (2011). Naïve electronic health record phenotype identification for rheumatoid arthritis. In *AMIA annual symposium proceedings*, volume 2011, page 189. American Medical Informatics Association.

Cerulo, L., Elkan, C., and Ceccarelli, M. (2010). Learning gene regulatory networks from only positive and unlabeled data. *BMC bioinformatics*, 11(1):228.

Chacón, J. E. (2012). Clusters and water flows: a novel approach to modal clustering through morse theory. *arXiv preprint arXiv:1212.1384*.

Chang, S., Zhang, Y., Tang, J., Yin, D., Chang, Y., Hasegawa-Johnson, M. A., and Huang, T. S. (2016). Positive-unlabeled learning in streaming networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 755–764. ACM.

Chapelle, O., Scholkopf, B., and Zien, A. (2006). *Semi-supervised learning.* Cambridge, Mass.: MIT Press.

Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64. Citeseer.

Chaudhuri, K., Dasgupta, S., Kpotufe, S., and von Luxburg, U. (2014). Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912.

Che, C., Xiao, C., Liang, J., Jin, B., Zho, J., and Wang, F. (2017). An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 198–206. SIAM.

Che, Z., Kale, D., Li, W., Bahadori, M. T., and Liu, Y. (2015). Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 507–516, New York, NY, USA. ACM.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Nature Scientific Reports*, 8(1):6085.

Chen, D. P., Weber, S. C., Constantinou, P. S., Ferris, T. A., Lowe, H. J., and Butte, A. J. (2007). Clinical arrays of laboratory measures, or "clinarrays", built from an electronic health record enable disease subtyping by severity. In *AMIA Annual Symposium Proceedings*, volume 2007, page 115. American Medical Informatics Association.

Chen, R., Sun, J., Dittus, R. S., Fabbri, D., et al. (2016a). Patient stratification using electronic health records from a chronic disease management program. *IEEE journal of biomedical and health informatics*.

Chen, W.-J., Shao, Y.-H., Li, C.-N., and Deng, N.-Y. (2016b). Mltsvm: A novel twin support vector machine to multi-label learning. *Pattern Recognition*, 52:61 – 74.

Chen, Y., Ghosh, J., Bejan, C. A., Gunter, C. A., Gupta, S., et al. (2015). Building bridges across electronic health record systems through inferred phenotypic topics. *Journal of Biomedical Informatics*, 55:82 – 93.

Chen, Y., Rege, M., Dong, M., and Hua, J. (2008). Non-negative matrix factorization for semi-supervised data clustering. *Knowledge and Information Systems*, 17(3):355–379.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799.

Cheng, Y., Wang, F., Zhang, P., and Hu, J. (2016). Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 432–440.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141).

Chiu, P.-H. and Hripcsak, G. (2017). EHR-based phenotyping: Bulk learning and evaluation. *Journal of Biomedical Informatics*, 70:35 – 51.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J. (2016a). Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318.

Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. (2016b). Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM.

Choi, E., Bahadori, M. T., Song, L., Stewart, W. F., and Sun, J. (2017). Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 787–795. ACM.

Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. (2016c). Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512.

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016d). Medical concept representation learning from electronic health records and its application on heart failure prediction. *arXiv preprint arXiv:1602.03686*.

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016e). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24(2):361–370.

Choi, Y., Chiu, C. Y.-I., and Sontag, D. (2016f). Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cole, T. S., Frankovich, J., Iyer, S., LePendu, P., Bauer-Mehren, A., and Shah, N. H. (2013). Profiling risk factors for chronic uveitis in juvenile idiopathic arthritis: a new model for ehr-based research. *Pediatric Rheumatology*, 11(1):45.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Comstock, J. (2017). Scanadu cofounders' new project, doc.ai, is a conversational robot that explains lab results. *Healthcare IT news (https://www.healthcareitnews.com/news/scanadu-launches-new-project-using-conversational-robot-explain-lab-results)*.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. S. (2002). On kernel-target alignment. In *Advances in neural information processing systems*, pages 367–373.

Cuevas, A., Febrero, M., and Fraiman, R. (2001). Cluster analysis: a further approach based on density estimation. *Computational Statistics & Data Analysis*, 36(4):441–459.

Cunningham, J. P. and Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900.

Cuturi, M. (2011). Fast global alignment kernels. In *Proceedings of the 28th International Conference on Machine Learning*, pages 929–936.

Cuturi, M. and Doucet, A. (2011). Autoregressive kernels for time series. *arXiv preprint arXiv:1101.0673*.

Dagliati, A., Marini, S., Sacchi, L., Cogni, G., Teliti, M., Tibollo, V., Cata, P. D., Chiovato, L., and Bellazzi, R. (2018). Machine learning methods to predict diabetes complications. *Journal of Diabetes Science and Technology*, 12(2):295–302.

Dagliati, A., Sacchi, L., Zambelli, A., Tibollo, V., Pavesi, L., Holmes, J., and Bellazzi, R. (2017). Temporal electronic phenotyping by mining careflows of breast cancer patients. *Journal of Biomedical Informatics*, 66:136 – 147.

Dai, W., Brisimi, T. S., Adams, W. G., Mela, T., Saligrama, V., and Paschalidis, I. C. (2015). Prediction of hospitalization due to heart diseases by supervised learning methods. *International Journal of Medical Informatics*, 84(3):189 – 197.

Dai, Y., Lokhandwala, S., Long, W., Mark, R., and Li-wei, H. L. (2017). Phenotyping hypotensive patients in critical care using hospital discharge summaries. In *Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on*, pages 401–404. IEEE.

Dalianis, H., Hassel, M., Henriksson, A., and Skeppstedt, M. (2012). Stockholm epr corpus: A clinical database used to improve health care. In *The Fourth Swedish Language Technology Conference (SLTC 2012), Lund, Sweden*, pages 17–18.

Davis, N. A. and LaCour, M. (2016). *Foundations of Health Information Management-E-Book*. Elsevier Health Sciences.

De Vine, L., Zuccon, G., Koopman, B., Sitbon, L., and Bruza, P. (2014). Medical semantic similarity with a neural language model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1819–1822, New York, NY, USA. ACM.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: Spectral clustering and normalized cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 551–556, New York, NY, USA. ACM.

Dinov, I. D. (2016). Methodological challenges and analytic opportunities for modeling and interpreting big healthcare data. *Gigascience*, 5(1):12.

Donders, A. R. T., van der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). Review: a gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091.

Doshi-Velez, F., Ge, Y., and Kohane, I. (2014). Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133:e54–e63.

Doshi-Velez, F. and Kim, B. (2018). Considerations for evaluation and generalization in interpretable machine learning. In Escalante, H., Escalera, S., et al., editors, *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Springer International Publishing, 1st edition.

Downs, G. M. and Barnard, J. M. (2002). Clustering methods and their uses in computational chemistry. *Reviews in computational chemistry*, 18:1–40.

Du Plessis, M., Niu, G., and Sugiyama, M. (2015). Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pages 1386–1394.

Du Plessis, M. C., Niu, G., and Sugiyama, M. (2014). Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems*, pages 703–711.

Dubois, S., Kale, D., Shah, N., and Jung, K. (2017). Learning effective representations from clinical notes. *arXiv preprint arXiv:1705.07025*.

Duin, R. P., Fred, A. L., Loog, M., and Pekalska, E. (2012). Mode seeking clustering by knn and mean shift evaluated. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 51–59. Springer.

Ebadollahi, S., Sun, J., Gotz, D., Hu, J., Sow, D., and Neti, C. (2010). Predicting patient's trajectory of physiological data using temporal trends in similar patients: a system for near-term prognostics. In *AMIA annual symposium proceedings*, volume 2010, page 192. American Medical Informatics Association.

Elakkia, K. and Narendran, P. (2016). Survey of medical image segmentation using removal of gaussian noise in medical image. *International Journal of Engineering Science*, 7593.

Elibol, M., Nguyen, V., Linderman, S., Johnson, M., Hashmi, A., and Doshi-Velez, F. (2016). Cross-corpora unsupervised learning of trajectories in autism spectrum disorders. *Journal of Machine Learning Research*, 17(1):4597–4634.

Elkan, C. and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220. ACM.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115.

Farhan, W., Wang, Z., Huang, Y., Wang, S., Wang, F., and Jiang, X. (2016). A predictive model for medical events based on contextual embedding of temporal sequences. *JMIR medical informatics*, 4(4).

Fazakis, N., Karlos, S., Kotsiantis, S., and Sgarbas, K. (2016). Self-trained lmt for semisupervised learning. *Computational intelligence and neuroscience*, 2016:10.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190.

Findley, G. (2011). *Knowing the risk: a review of the peri-operative care of surgical patients: summary.* National Confidential Enquiry into Patient Outcome and Death (NCEPOD).

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Florek, K., Łukaszewicz, J., Perkal, J., Steinhaus, H., and Zubrzycki, S. (1951). Sur la liaison et la division des points d'un ensemble fini. In *Colloquium Mathematicae*, volume 2, pages 282–285. Institute of Mathematics Polish Academy of Sciences.

Forrey, A. W., Mcdonald, C. J., DeMoor, G., Huff, S. M., Leavelle, D., Leland, D., Fiers, T., Charles, L., Griffin, B., Stalling, F., et al. (1996). Logical observation identifier names and codes (loinc) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. *Clinical Chemistry*, 42(1):81–90.

Foulds, J. and Frank, E. (2010). A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631.

Fred, A. L. N. (2001). Finding consistent clusters in data partitions. In *In Proc. 3d Int. Workshop on Multiple Classifier*, pages 309–318. Springer.

Fred, A. L. N. and Jain, A. K. (2002). Evidence accumulation clustering based on the k-means algorithm. In *Structural, Syntactic, and Statistical Pattern Recognition, LNCS 2396:442–451*, pages 442–451. Springer-Verlag.

Fred, A. L. N. and Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):835–850.

Frenay, B. and Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

Friedman, J. H. (1997). On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77.

Fröhlich, H., Balling, R., Beerenwinkel, N., Kohlbacher, O., Kumar, S., et al. (2018). From hype to reality: data science enabling personalized medicine. *BMC Medicine*, 16(1):150.

Fujino, A., Ueda, N., and Saito, K. (2005). A hybrid generative/discriminative approach to semi-supervised classifier design. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, pages 764–769. AAAI Press.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40.

Futoma, J., Hariharan, S., and Heller, K. (2017a). Learning to detect sepsis with a multitask gaussian process RNN classifier. *arXiv preprint arXiv:1706.04152*.

Futoma, J., Hariharan, S., Heller, K., et al. (2017b). An improved multi-output gaussian process rnn with real-time validation for early sepsis detection. In Doshi-Velez, F. et al., editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 243–254, Boston, Massachusetts. PMLR.

Futoma, J., Sendak, M., Cameron, B., and Heller, K. (2016a). Predicting disease progression with a model for multivariate longitudinal clinical data. In *Machine Learning for Healthcare Conference*, pages 42–54.

Futoma, J., Sendak, M., Cameron, C. B., and Heller, K. (2016b). Scalable joint modeling of longitudinal and point process data for disease trajectory prediction and improving management of chronic kidney disease. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, UAI'16, pages 222–231.

Gallego, B., Walter, S. R., Day, R. O., Dunn, A. G., Sivaraman, V., Shah, N., Longhurst, C. A., and Coiera, E. (2015). Bringing cohort studies to the bedside: framework for a 'green button'to support clinical decision-making. *Journal of comparative effectiveness research*, 4(3):191–197.

García-Laencina, P. J., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In Schölkopf, B. and Warmuth, M. K., editors, *Learning Theory and Kernel Machines*, pages 129–143, Berlin, Heidelberg. Springer Berlin Heidelberg.

Georgescu, B., Shimshoni, I., and Meer, P. (2003). Mean shift based clustering in high dimensions: A texture classification example. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 456–463. IEEE.

Gerhard, G. S., Carey, D. J., and Steele, G. D. (2013). Chapter 24 - Electronic Health Records in Genomic Medicine. In Ginsburg, G. S. and Willard, H. F., editors, *Genomic and Personalized Medicine (Second Edition)*, pages 287 – 294. Academic Press.

Gershman, S. J. and Blei, D. M. (2012). A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1 – 12.

Ghalwash, M., Li, Y., Zhang, P., and Hu, J. (2017). Exploiting electronic health records to mine drug effects on laboratory test results. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1837–1846, New York, NY, USA. ACM.

Ghassemi, M., Naumann, T., Doshi-Velez, F., Brimmer, N., Joshi, R., Rumshisky, A., and Szolovits, P. (2014a). Unfolding physiological state: Mortality modelling in intensive care units. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 75–84. ACM.

Ghassemi, M., Pimentel, M. A., Naumann, T., Brennan, T., Clifton, D. A., Szolovits, P., and Feng, M. (2015). A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *AAAI Conference on Artificial Intelligence*.

Ghassemi, M., Stan, J. H. V., Mehta, D. D., Zanartu, M., II, H. A. C., Hillman, R. E., and Guttag, J. V. (2014b). Learning to detect vocal hyperfunction from ambulatory neck-surface acceleration features: Initial results for vocal fold nodules. *IEEE Trans. Biomed. Engineering*, 61(6):1668–1675.

Ghassemi, M., Syed, Z., Mehta, D., et al. (2016). Uncovering voice misuse using symbolic mismatch. In Doshi-Velez, F. et al., editors, *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 239–252, Children's Hospital LA, Los Angeles, CA, USA. PMLR.

Ghassemi, M., Wu, M., Hughes, M. C., Szolovits, P., and Doshi-Velez, F. (2017). Predicting intervention onset in the icu with switching state space models. *AMIA Summits on Translational Science Proceedings*, 2017:82.

Ghosh, A., Manwani, N., and Sastry, P. (2015). Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93 – 107.

Glodek, M., Schels, M., and Schwenker, F. (2013). Ensemble gaussian mixture models for probability density estimation. *Computational Statistics*, 28(1):127–138.

Glueck, M., Naeini, M. P., Doshi-Velez, F., Chevalier, F., Khan, A., Wigdor, D., and Brudno, M. (2018). Phenolines: Phenotype comparison visualizations for disease subtyping via topic models. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):371–381.

Gómez-Chova, L., Jenssen, R., and Camps-Valls, G. (2012). Kernel entropy component analysis for remote sensing image clustering. *IEEE Geoscience and Remote Sensing Letters*, 9(2):312–316.

Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *Journal of machine learning research*, 12(Jul):2211–2268.

González, D., Aguado, J. V., Cueto, E., Abisset-Chavanne, E., and Chinesta, F. (2018). kpca-based parametric solutions within the pgd framework. *Archives of Computational Methods in Engineering*, 25(1):69–86.

Goodman, B. and Flaxman, S. (2016). European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*.

Gopalakrishnan, K., Balakrishnan, H., and Jordan, R. (2016). Clusters and communities in air traffic delay networks. In *2016 American Control Conference (ACC)*, pages 3782–3788. IEEE.

Gottlieb, A., Stein, G. Y., Ruppin, E., Altman, R. B., and Sharan, R. (2013). A method for inferring medical diagnoses from patient similarities. *BMC Medicine*, 11(1):194.

Gower, J. C. and Ross, G. (1969). Minimum spanning trees and single linkage cluster analysis. *Applied statistics*, pages 54–64.

Groves, P., Kayyali, B., Knott, D., and Van Kuiken, S. (2013). The 'big data'revolution in healthcare. *McKinsey Quarterly*, 2(3).

Gu, B. and Sheng, V. S. (2017). A robust regularization path algorithm for ν-support vector classification. *IEEE Transactions on neural networks and learning systems*, 28(5):1241–1248.

Gunasekar, S., Ho, J. C., Ghosh, J., Kreml, S., Kho, A. N., Denny, J. C., Malin, B. A., and Sun, J. (2016). Phenotyping using structured collective matrix factorization of multi–source ehr data. *arXiv preprint arXiv:1609.04466*.

Gunter, T. D. and Terry, N. P. (2005). The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions. *Journal of medical Internet research*, 7(1).

Guo, L., Jin, B., Yao, C., Yang, H., Huang, D., and Wang, F. (2016). Which doctor to trust: a recommender system for identifying the right doctors. *Journal of medical Internet research*, 18(7).

Guyon, I., Von Luxburg, U., and Williamson, R. C. (2009). Clustering: Science or art. In *NIPS 2009 workshop on clustering theory*, pages 1–11.

Halpern, Y., Choi, Y., Steven, H., and Sontag, D. (2014). Using anchors to estimate clinical state without labeled data. *AMIA Annual Symposium Proceedings*, pages 606–615.

Halpern, Y., Horng, S., Choi, Y., and Sontag, D. (2016). Electronic medical record phenotyping using the anchor and learn framework. *Journal of the American Medical Informatics Association.*

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001.

Hao, T., Bi, C., Xing, G., Chan, R., and Tu, L. (2017). Mindfulwatch: A smartwatch-based system for real-time respiration monitoring during meditation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(3):57:1–57:19.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer.

Häyrinen, K., Saranto, K., and Nykänen, P. (2008). Definition, structure, content, use and impacts of electronic health records: a review of the research literature. *International journal of medical informatics*, 77(5):291–304.

He, W., Goodkind, D., and Kowal, P. R. (2016). *An aging world: 2015.* United States Census Bureau Washington, DC.

Hein, M. and Bousquet, O. (2005). Hilbertian metrics and positive definite kernels on probability measures. In *AISTATS*, pages 136–143.

Henderson, J., Malin, B. A., Ho, J. C., and Ghosh, J. (2018). Piveted-granite: Computational phenotypes through constrained tensor factorization. *arXiv preprint arXiv:1808.02602*.

Henriksson, A., Kvist, M., Dalianis, H., and Duneld, M. (2015). Identifying adverse drug event information in clinical notes with distributional semantic representations of context. *Journal of biomedical informatics*, 57:333–349.

Henriksson, A., Moen, H., Skeppstedt, M., Daudaravičius, V., and Duneld, M. (2014). Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of biomedical semantics*, 5(1):6.

Hensley, A., Doboli, A., Mangoubi, R., and Doboli, S. (2015). Generalized label propagation. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Hersh, W. R., Weiner, M. G., Embi, P. J., Logan, J. R., et al. (2013). Caveats for the use of operational electronic health record data in comparative effectiveness research. *Medical care*, 51(8 0 3):S30.

Hille, E. and Phillips, R. S. (1996). *Functional analysis and semi-groups*, volume 31. American Mathematical Soc.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. (2010). *Bayesian nonparametrics*, volume 28. Cambridge University Press.

Ho, J. C., Ghosh, J., Steinhubl, S. R., Stewart, W. F., Denny, J. C., Malin, B. A., and Sun, J. (2014a). Limestone: High-throughput candidate phenotype generation via tensor factorization. *Journal of biomedical informatics*, 52:199–211.

Ho, J. C., Ghosh, J., and Sun, J. (2014b). Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 115–124. ACM.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220.

Hripcsak, G. and Albers, D. J. (2012). Next-generation phenotyping of electronic health records. *Journal of the American Medical Informatics Association*, 20(1):117–121.

Hripcsak, G. and Albers, D. J. (2018). High-fidelity phenotyping: richness and freedom from bias. *Journal of the American Medical Informatics Association*, 25(3):289–294.

Hripcsak, G., Albers, D. J., and Perotte, A. (2015). Parameterizing time in electronic health record studies. *Journal of the American Medical Informatics Association*, 22(4):794–804.

Hripcsak, G., Ryan, P. B., Duke, J. D., Shah, N. H., et al. (2016). Characterizing treatment pathways at scale using the OHDSI network. *Proceedings of the National Academy of Sciences*, 113(27):7329–7336.

Hsieh, C.-J., Natarajan, N., and Dhillon, I. S. (2015). Pu learning for matrix completion. In *ICML*, pages 2445–2453.

Hu, J., Perer, A., and Wang, F. (2016). Data driven analytics for personalized healthcare. In Weaver, C. A. et al., editors, *Healthcare Information Management Systems: Cases, Strategies, and Solutions*, pages 529–554. Springer International Publishing, Cham.

Hu, Z., Melton, G. B., Arsoniadis, E. G., Wang, Y., Kwaan, M. R., and Simon, G. J. (2017). Strategies for handling missing clinical data for automated surgical site infection detection from the electronic health record. *Journal of Biomedical Informatics*, 68:112–120.

Huang, Y., McCullagh, P., Black, N., and Harper, R. (2007). Feature selection and classification model construction on type 2 diabetic patients' data. *Artificial Intelligence in Medicine*, 41(3):251 – 262.

Huang, Z., Dong, W., Duan, H., and Li, H. (2014). Similarity measure between patient traces for clinical pathway analysis: problem, method, and applications. *IEEE journal of biomedical and health informatics*, 18(1):4–14.

Hughes, J. S., Averill, R. F., et al. (2004). Clinical Risk Groups (CRGs): a classification system for risk-adjusted capitation-based payment and health care management. *Medical care*, 42(1):81–90.

Izquierdo-Verdiguier, E., Jenssen, R., Gómez-Chova, L., and Camps-Valls, G. (2015). Spectral clustering with the probabilistic cluster kernel. *Neurocomputing*, 149:1299–1304.

Jacobson, O. and Dalianis, H. (2016). Applying deep learning on electronic health records in swedish to predict healthcare-associated infections. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 191–195.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

Jain, S., White, M., and Radivojac, P. (2016). Estimating the class prior and posterior from noisy positives and unlabeled data. In *Advances in Neural Information Processing Systems*, pages 2693–2701.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. (2013). Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jensen, P. B., Jensen, L. J., and Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395.

Jenssen, R. (2010). Kernel entropy component analysis. *IEEE Trans Pattern Anal Mach Intell*, 33(5):847–860.

Jenssen, R. (2013). Entropy-relevant dimensions in the kernel feature space: Cluster-capturing dimensionality reduction. *IEEE Signal Processing Magazine*, 30(4):30–39.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209.

Johnson, A. and Mark, R. (2017). Real-time mortality prediction in the intensive care unit. In *AMIA... Annual Symposium proceedings. AMIA Symposium*, volume 2017, pages 994–1003. American Medical Informatics Association.

Johnson, A. E. W., Ghassemi, M. M., Nemati, S., Niehaus, K. E., Clifton, D. A., and Clifford, G. D. (2016). Machine learning and decision support in critical care. *Proceedings of the IEEE*, 104(2):444–466.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214.

Jung, K., Covington, S., Sen, C. K., Januszyk, M., Kirsner, R. S., Gurtner, G. C., and Shah, N. H. (2016). Rapid identification of slow healing wounds. *Wound Repair and Regeneration*, 24(1):181–188.

Jung, K., LePendu, P., Iyer, S., Bauer-Mehren, A., Percha, B., and Shah, N. H. (2015). Functional evaluation of out-of-the-box text-mining tools for data-mining tasks. *Journal of the American Medical Informatics Association*, 22(1):121–131.

Jung, K. and Shah, N. H. (2015). Implications of non-stationarity on predictive modeling using ehrs. *Journal of Biomedical Informatics*, 58:168 – 174.

Kale, D. C., Gong, D., Che, Z., Liu, Y., Medioni, G., Wetzel, R., and Ross, P. (2014). An examination of multivariate time series hashing with applications to health care. In *Data Mining (ICDM), 2014 IEEE international conference on*, pages 260–269. IEEE.

Kampffmeyer, M., Løkse, S., Bianchi, F. M., Jenssen, R., and Livi, L. (2018). The deep kernelized autoencoder. *Applied Soft Computing*, 71:816 – 825.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*.

Kanehira, A. and Harada, T. (2016). Multi-label ranking from positive and unlabeled data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kate, R. J., Perez, R. M., Mazumdar, D., Pasupathy, K. S., and Nilakantan, V. (2016). Prediction and detection models for acute kidney injury in hospitalized older adults. *BMC medical informatics and decision making*, 16(1):39.

Ke, C., Jin, Y., Evans, H., Lober, B., Qian, X., Liu, J., and Huang, S. (2017). Prognostics of surgical site infections using dynamic health data. *Journal of Biomedical Informatics*, 65:22–33.

Khakabimamaghani, S. and Ester, M. (2016). Bayesian biclustering for patient stratification. In *Biocomputing 2016: Proceedings of the Pacific Symposium*, pages 345–356. World Scientific.

Khalid, S., Khalil, T., and Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference*, pages 372–378.

Khardon, R. and Wachman, G. (2007). Noise tolerant variants of the perceptron algorithm. *Journal of Machine Learning Research*, 8(Feb):227–248.

Kierkegaard, P. (2011). Electronic health record: Wiring europe's healthcare. *Computer Law & Security Review*, 27(5):503 – 515.

Kim, Y., El-Kareh, R., Sun, J., Yu, H., and Jiang, X. (2017a). Discriminative and distinct phenotyping by constrained tensor factorization. *Scientific reports*, 7(1):1114.

Kim, Y., Sun, J., Yu, H., and Jiang, X. (2017b). Federated tensor factorization for computational phenotyping. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 887–895. ACM.

Kimeldorf, G. and Wahba, G. (1971). Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95.

King, G. and Zeng, L. (2001). Logistic regression in rare events data. *Political analysis*, 9(2):137–163.

Kiryo, R., Niu, G., du Plessis, M. C., and Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, pages 1675–1685.

Kisilevich, S., Mansmann, F., and Keim, D. (2010). P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *Proceedings of the 1st international conference and exhibition on computing for geospatial research & application*, page 38. ACM.

Krishnan, R. G., Shalit, U., and Sontag, D. (2017). Structured inference networks for nonlinear state space models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2101–2109.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kulis, B., Basu, S., Dhillon, I., and Mooney, R. (2009). Semi-supervised graph clustering: a kernel approach. *Machine Learning*, 74(1):1–22.

Kulis, B. and Grauman, K. (2012). Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1092–1104.

Kuo, M.-H., Sahama, T., Kushniruk, A. W., Borycki, E. M., and Grunwell, D. K. (2014). Health big data analytics: current perspectives, challenges and potential solutions. *International Journal of Big Data Intelligence*, 1(1-2):114–126.

Kwon, B. C., Eysenbach, B., Verma, J., Ng, K., Filippi, C. D., Stewart, W. F., and Perer, A. (2018). Clustervision: Visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):142–151.

Lacasa, L., Nicosia, V., and Latora, V. (2015). Network structure of multivariate time series. *Scientific reports*, 5:15508.

Lallich, S., Muhlenbach, F., and Zighed, D. A. (2002). Improving classification by removing or relabeling mislabeled instances. In *International Symposium on Methodologies for Intelligent Systems*, pages 5–15. Springer.

Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. (2004a). Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72.

Lanckriet, G. R. G., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004b). A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635.

Lasko, T. A., Denny, J. C., and Levy, M. A. (2013). Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS one*, 8(6):e66341.

Lawrence, N. D. and Schölkopf, B. (2001). Estimating a kernel fisher discriminant in the presence of label noise. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 306–313, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lee, C. H. and Yoon, H.-J. (2017). Medical big data: promise and challenges. *Kidney research and clinical practice*, 36(1):3.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562.

Lee, H., Yoo, J., and Choi, S. (2010). Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, 17(1):4–7.

Lee, J., Maslove, D. M., and Dubin, J. A. (2015). Personalized mortality prediction driven by electronic medical data and a patient similarity metric. *PloS one*, 10(5):e0127428.

Lewis, S., Foltynie, T., Blackwell, A. D., Robbins, T. W., Owen, A. M., and Barker, R. A. (2005). Heterogeneity of parkinson's disease in the early clinical stages using a data driven approach. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(3):343–348.

Li, J., Ray, S., and Lindsay, B. G. (2007a). A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(Aug):1687–1723.

Li, L., Cheng, W.-Y., Glicksberg, B. S., Gottesman, O., et al. (2015). Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science translational medicine*, 7(311).

Li, T., Ding, C., and Jordan, M. I. (2007b). Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 577–582. IEEE.

Li, Y.-F., Tsang, I. W., Kwok, J. T., and Zhou, Z.-H. (2013). Convex and scalable weakly labeled svms. *The Journal of Machine Learning Research*, 14(1):2151–2188.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, pages 2–11, New York, NY, USA. ACM.

Lindberg, D. A., Humphreys, B. L., and McCray, A. T. (1993). The unified medical language system. *Methods of information in medicine*, 32(04):281–291.

Lingren, T., Chen, P., Bochenek, J., Doshi-Velez, F., Manning-Courtney, P., et al. (2016). Electronic health record based algorithm to identify patients with autism spectrum disorder. *PLoS ONE 11(7): e0159621.*

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. C. (2015). Learning to diagnose with LSTM recurrent neural networks. *CoRR*, abs/1511.03677.

Lipton, Z. C., Kale, D. C., and Wetzel, R. C. (2016). Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *MLHC*, volume 56 of *JMLR Workshop and Conference Proceedings*, pages 253–270. JMLR.org.

Little, R. J. and Rubin, D. B. (2014). *Statistical analysis with missing data.* John Wiley & Sons.

Liu, B., Dai, Y., Li, X., Lee, W. S., and Yu, P. S. (2003). Building text classifiers using positive and unlabeled examples. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 179–186. IEEE.

Liu, B., Li, Y., Sun, Z., Ghosh, S., and Ng, K. (2018a). Early prediction of diabetes complications from electronic health records: A multi-task survival analysis approach. In *AAAI Conference on Artificial Intelligence.*

Liu, C., Wang, F., Hu, J., and Xiong, H. (2015). Temporal phenotyping from longitudinal electronic health records: A graph based framework. In *Proceedings of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 705–714. ACM.

Liu, T. and Tao, D. (2016a). Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461.

Liu, T. and Tao, D. (2016b). Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461.

Liu, Y., Wen, K., Gao, Q., Gao, X., and Nie, F. (2018b). Svm based multi-label learning with missing labels for image annotation. *Pattern Recognition*, 78:307 – 317.

Løkse, S., Bianchi, F. M., Salberg, A.-B., and Jenssen, R. (2017). Spectral clustering using pckid–a probabilistic cluster kernel for incomplete data. In *Scandinavian Conference on Image Analysis*, pages 431–442. Springer.

Lourenço, A., Bulò, S. R., Rebagliati, N., Fred, A. L., Figueiredo, M. A., and Pelillo, M. (2015). Probabilistic consensus clustering using evidence accumulation. *Machine Learning*, 98(1-2):331–357.

Ma, T., Xiao, C., and Wang, F. (2018). Health-atm: A deep architecture for multifaceted patient health record representation and risk prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 261–269. SIAM.

Madjarov, G., Kocev, D., Gjorgjevikj, D., and Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern recognition*, 45(9):3084–3104.

Mancisidor, R. A., Kampffmeyer, M., Aas, K., and Jenssen, R. (2018). Segment-based credit scoring using latent clusters in the variational autoencoder. *arXiv preprint arXiv:1806.02538*.

Mandel, J. C., Kreda, D. A., Mandl, K. D., Kohane, I. S., and Ramoni, R. B. (2016). Smart on fhir: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*, 23(5):899–908.

Manwani, N. and Sastry, P. (2013). Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151.

Marengoni, A., Angleman, S., Melis, R., Mangialasche, F., Karp, A., Garmen, A., Meinow, B., and Fratiglioni, L. (2011). Aging with multimorbidity: a systematic review of the literature. *Ageing research reviews*, 10(4):430–439.

Marlin, B. M., Kale, D. C., Khemani, R. G., and Wetzel, R. C. (2012). Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2Nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, pages 389–398, New York, NY, USA. ACM.

McDonald, R. A., Hand, D. J., and Eckley, I. A. (2003). An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *International Workshop on Multiple Classifier Systems*, pages 35–44. Springer.

McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: Inference and applications to clustering*, volume 84. Marcel Dekker.

Mehrabi, S., Sohn, S., Li, D., Pankratz, J. J., Therneau, T., Sauver, J. L. S., Liu, H., and Palakal, M. (2015). Temporal pattern and association discovery of diagnosis codes using deep learning. In *Healthcare Informatics (ICHI), 2015 International Conference on*, pages 408–416. IEEE.

Menardi, G. (2015). A review on modal clustering. *International Statistical Review.*

Menon, A., Van Rooyen, B., Ong, C. S., and Williamson, B. (2015). Learning from corrupted binary labels via class-probability estimation. In *International Conference on Machine Learning*, pages 125–134.

Mihailovic, N., Kocic, S., and Jakovljevic, M. (2016). Review of diagnosis-related group-based financing of hospital care. *Health services research and managerial epidemiology*, 3:2333392816647892.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K.-R. (1999). Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.*, pages 41–48. Ieee.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Miller, D. J. and Uyar, H. S. (1997). A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in neural information processing systems*, pages 571–577.

Minarro-Giménez, J. A., Marin-Alonso, O., and Samwald, M. (2014). Exploring the application of deep learning techniques on medical text corpora. *Studies in health technology and informatics*, 205:584–588.

Miotto, R., Li, L., Kidd, B. A., and Dudley, J. T. (2016). Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6:26094.

Miotto, R., Wang, F., Wang, S., Jiang, X., and Dudley, J. T. (2017). Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics.*

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, 52(1-2):91–118.

Mordelet, F. and Vert, J.-P. (2014). A bagging svm to learn from positive and unlabeled examples. *Pattern Recognition Letters*, 37:201–209.

Moskovitch, R., Polubriaginof, F., Weiss, A., Ryan, P., and Tatonetti, N. (2017). Procedure prediction from symbolic electronic health records via time intervals analytics. *Journal of Biomedical Informatics*, 75:70 – 82.

Mould, D. (2012). Models for disease progression: new approaches and uses. *Clinical Pharmacology & Therapeutics*, 92(1):125–131.

Mudelsee, M. (2013). *Climate time series analysis*. Springer.

Mukherjee, S. (2017). A.I. versus M.D. *The New Yorker (https://www.newyorker.com/magazine/2017/04/03/ai-versus-md)*.

Murdoch, T. B. and Detsky, A. S. (2013). The inevitable application of big data to health care. *JAMA*, 309(13):1351–1352.

Murgia, M. (2017). Babylon raises $60m to build AI doctor to diagnose illnesses. *Financial Times (https://www.ft.com/content/1f56997a-290f-11e7-bc4b-5528796fe35c)*.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204.

Nettleton, D. F., Orriols-Puig, A., and Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306.

Névéol, A., Dalianis, H., Velupillai, S., Savova, G., and Zweigenbaum, P. (2018). Clinical natural language processing in languages other than english: opportunities and challenges. *Journal of Biomedical Semantics*, 9(1):12.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856.

Ng, K., Sun, J., Hu, J., and Wang, F. (2015). Personalized predictive modeling and risk factor identification using patient similarity. *AMIA Summits on Translational Science Proceedings*, 2015:132.

Nguyen, P., Tran, T., Wickramasinghe, N., and Venkatesh, S. (2017). *deepr*: A convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30.

Nie, F., Xiang, S., Liu, Y., and Zhang, C. (2010a). A general graph-based semi-supervised learning with novel class discovery. *Neural Computing and Applications*, 19(4):549–555.

Nie, F., Xu, D., Tsang, I. W.-H., and Zhang, C. (2010b). Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134.

Nissim, N., Boland, M. R., Moskovitch, R., et al. (2015). An active learning framework for efficient condition severity classification. In Holmes, J. H. et al., editors, *Artificial Intelligence in Medicine*, pages 13–24. Springer.

Nissim, N., Shahar, Y., Elovici, Y., Hripcsak, G., and Moskovitch, R. (2017). Inter-labeler and intra-labeler variability of condition severity classification models using active and passive learning methods. *Artificial Intelligence in Medicine*, 81:12 – 32.

Niu, G., Du Plessis, M. C., Sakai, T., Ma, Y., and Sugiyama, M. (2016). Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *Advances in neural information processing systems*, pages 1199–1207.

Noble, W. S. et al. (2004). Support vector machine applications in computational biology. *Kernel methods in computational biology*, 71:92.

NOMESCO (2011). *NOMESCO classification of surgical procedures.* Nordic Medico-Statistical Committee, Copenhagen.

Orphanou, K., Dagliati, A., Sacchi, L., Stassopoulou, A., Keravnou, E., and Bellazzi, R. (2016). Combining naive bayes classifiers with temporal association rules for coronary heart disease diagnosis. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 81–92.

Orphanou, K., Dagliati, A., Sacchi, L., Stassopoulou, A., Keravnou, E., and Bellazzi, R. (2018). Incorporating repeating temporal association rules in naïve bayes classifiers for coronary heart disease diagnosis. *Journal of Biomedical Informatics*, 81:74 – 82.

Orphanou, K., Stassopoulou, A., and Keravnou, E. (2014). Temporal abstraction and temporal bayesian networks in clinical domains: A survey. *Artificial Intelligence in Medicine*, 60(3):133 – 149.

Pan, S. J., Yang, Q., et al. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Panuccio, A., Bicego, M., and Murino, V. (2002). A hidden markov model-based approach to sequential data clustering. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 734–743. Springer.

Parbhoo, S., Bogojeska, J., Zazzi, M., Roth, V., and Doshi-Velez, F. (2017). Combining kernel and model based learning for hiv therapy selection. In *AMIA Summits on Translational Science Proceedings*, volume 2017, page 239.

Parimbelli, E., Marini, S., Sacchi, L., and Bellazzi, R. (2018). Patient similarity for precision medicine: A systematic review. *Journal of Biomedical Informatics*, 83:87 – 96.

Patrini, G., Nielsen, F., Nock, R., and Carioni, M. (2016). Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning*, pages 708–717.

Pechenizkiy, M., Puuronen, S., Tsymbal, A., and Pechenizkiy, O. (2006). Class noise and supervised learning in medical domains: The effect of feature extraction. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)(CBMS)*, volume 00, pages 708–713.

Pejaver, V., Urresti, J., Lugo-Martinez, J., Pagel, K. A., Lin, G. N., Nam, H.-J., Mort, M., Cooper, D. N., Sebat, J., Iakoucheva, L. M., et al. (2017). Mutpred2: inferring the molecular and phenotypic impact of amino acid variants. *BioRxiv*, page 134981.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perer, A., Wang, F., and Hu, J. (2015). Mining and exploring care pathways from electronic medical records with visual analytics. *Journal of Biomedical Informatics*, 56:369 – 378.

Perez, A., Weegar, R., Casillas, A., Gojenola, K., Oronoz, M., and Dalianis, H. (2017). Semi-supervised medical entity recognition: A study on spanish and swedish clinical corpora. *Journal of biomedical informatics*, 71:16–30.

Perros, I., Papalexakis, E. E., Park, H., et al. (2018). Sustain: Scalable unsupervised scoring for tensors and its application to phenotyping. *CoRR*, abs/1803.05473.

Perros, I., Papalexakis, E. E., Wang, F., et al. (2017). Spartan: Scalable parafac2 for large & sparse data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 375–384. ACM.

Pham, T., Tran, T., Phung, D., and Venkatesh, S. (2016). Deepcare: A deep dynamic memory model for predictive medicine. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 30–41. Springer.

Pivovarov, R., Perotte, A. J., Grave, E., Angiolillo, J., Wiggins, C. H., and Elhadad, N. (2015). Learning probabilistic phenotypes from heterogeneous ehr data. *Journal of Biomedical Informatics*, 58:156 – 165.

Poole, S., Schroeder, L. F., and Shah, N. (2016). An unsupervised learning method to identify reference intervals from a clinical database. *Journal of Biomedical Informatics*, 59:276 – 284.

Quaglini, S., Stefanelli, M., Lanzola, G., Caporusso, V., and Panzarasa, S. (2001). Flexible guideline-based patient careflow systems. *Artificial Intelligence in Medicine*, 22(1):65 – 80. Workflow Management and Clinical Guidelines.

Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., and Ghassemi, M. (2017). Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. In *Machine Learning for Healthcare Conference*, pages 147–163.

Rahman, S. A., Huang, Y., Claassen, J., Heintzman, N., and Kleinberg, S. (2015). Combining fourier and lagged k-nearest neighbor imputation for biomedical time series data. *Journal of Biomedical Informatics*, 58:198 – 207.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., et al. (2018). Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1):18.

Ramos, J., Kockelkorn, T. T. J. P., Ramos, I., Ramos, R., Grutters, J., Viergever, M. A., van Ginneken, B., and Campilho, A. (2016). Content-based image retrieval by metric learning from radiology reports: Application to interstitial lung diseases. *IEEE Journal of Biomedical and Health Informatics*, 20(1):281–292.

Rasmussen, C. E. (2000). The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560.

Rasmussen, C. E. (2004). Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer.

Ravı, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., and Yang, G.-Z. (2017). Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1):4–21.

Razavian, N., Blecker, S., Schmidt, A. M., Smith-McLallen, A., Nigam, S., and Sontag, D. (2016a). Population-level prediction of type 2 diabetes using claims data and analysis of risk factors. *Big Data*, Data and Healthcare Special Issue.

Razavian, N., Marcus, J., and Sontag, D. (2016b). Multi-task prediction of disease onsets from longitudinal laboratory tests. In Doshi-Velez, F. et al., editors, *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56, pages 73–100. PMLR.

Ren, O., Johnson, A. E. W., Lehman, E. P., Komorowski, M., Aboab, J., et al. (2018). Predicting and understanding unexpected respiratory decompensation in critical care using sparse and heterogeneous clinical data. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 144–151.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. (2017). Right for the right reasons: Training differentiable models by constraining their explananations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Melbourne, Australia.

Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., and Sontag, D. (2017). Learning a health knowledge graph from electronic medical records. *Nature Scientific Reports*, 7(1):5994.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons.

Rumshisky, A., Ghassemi, M., Naumann, T., Szolovits, P., Castro, V., McCoy, T., and Perlis, R. (2016). Predicting early psychiatric readmission with natural language processing of narrative discharge summaries. *Translational psychiatry*, 6(10):e921.

Runkler, T. A. and Bezdek, J. C. (2003). Web mining with relational clustering. *International Journal of Approximate Reasoning*, 32(2-3):217–236.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach.* Malaysia; Pearson Education Limited,.

Sakai, T., Plessis, M. C., Niu, G., and Sugiyama, M. (2017). Semi-supervised classification based on classification from positive and unlabeled data. In *International Conference on Machine Learning*, pages 2998–3006.

Sandryhaila, A. and Moura, J. M. F. (2013). Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656.

Sanger, P. C., van Ramshorst, G. H., Mercan, E., Huang, S., et al. (2016). A prognostic model of surgical site infection using daily clinical wound assessment. *Journal of the American College of Surgeons*, 223(2):259 – 270.e2.

Savage, N. (2012). Better medicine through machine learning. *Communications of the ACM*, 55(1):17–19.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data.* CRC press.

Schafer, J. L. and Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, 7(2):147.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

Schölkopf, B., Smola, A. J., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schölkopf, B., Smola, A. J., and Müller, K.-R. (1999). Kernel principal component analysis. In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in kernel methods: support vector learning*, pages 327–352. MIT press.

Schroeder, L. F., Giacherio, D., Gianchandani, R., Engoren, M., and Shah, N. H. (2016). Postmarket surveillance of point-of-care glucose meters through analysis of electronic medical records. *Clinical Chemistry*, 62(5):716–724.

Schulam, P. and Saria, S. (2016). Integrative analysis using coupled latent variable models for individualizing prognoses. *Journal of Machine Learning Research*, 17(234):1–35.

Schulam, P. and Saria, S. (2017). Reliable decision support using counterfactual models. In Guyon, I. et al., editors, *Advances in Neural Information Processing Systems 30*, pages 1697–1708. Curran Associates, Inc.

Schulam, P., Wigley, F., and Saria, S. (2015). Clustering longitudinal clinical marker trajectories from electronic health data: Applications to phenotyping and endotype discovery. In *AAAI*, pages 2956–2964.

Scott, C., Blanchard, G., and Handy, G. (2013). Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference On Learning Theory*, pages 489–511.

Scutti, S. (2017). 'automated dermatologist' detects skin cancer with expert accuracy. *CNN (https://edition.cnn.com/2017/01/26/health/ai-system-detects-skin-cancer-study/index.html)*.

Seifoddini, H. K. (1989). Single linkage versus average linkage clustering in machine cells formation applications. *Computers & Industrial Engineering*, 16(3):419–426.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

Sha, Y., Venugopalan, J., and Wang, M. D. (2016). A novel temporal similarity measure for patients based on irregularly measured data in electronic health records. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '16, pages 337–344, New York, NY, USA. ACM.

Shah, N. H., LePendu, P., Bauer-Mehren, A., Ghebremariam, Y. T., Iyer, S. V., Marcus, J., Nead, K. T., Cooke, J. P., and Leeper, N. J. (2015). Proton pump inhibitor usage and the risk of myocardial infarction in the general population. *PLOS ONE*, 10(6):1–16.

Shameer, K., Johnson, K. W., Yahi, A., Miotto, R., et al. (2017). Predictive modeling of hospital readmission rates using electronic medical record-wide machine learning: a case-study using mount sinai heart failure cohort. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING*, pages 276–287. World Scientific.

Shankar, P. R., Kesari, A., Shalini, P., Kamalashree, N., Bharadwaj, C., Raj, N., Srinivas, S., Shivakumar, M., Ulle, A. R., and Tagadur, N. N. (2018). Predictive modeling of surgical site infections using sparse laboratory data. *International Journal of Big Data and Analytics in Healthcare (IJBDAH)*, 3(1):13–26.

Sharafoddini, A., Dubin, J. A., and Lee, J. (2017). Patient similarity in prediction models based on health data: a scoping review. *JMIR medical informatics*, 5(1).

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

Shickel, B., Tighe, P. J., Bihorac, A., and Rashidi, P. (2018). Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5):1589–1604.

Shivade, C., Raghavan, P., Fosler-Lussier, E., Embi, P. J., Elhadad, N., Johnson, S. B., and Lai, A. M. (2013). A review of approaches to identifying patient phenotype cohorts using electronic health records. *Journal of the American Medical Informatics Association*, 21(2):221–230.

Simon, H. U. (1996). General bounds on the number of examples needed for learning probabilistic concepts. *Journal of Computer and System Sciences*, 52(2):239 – 254.

Sinha, A., Hripcsak, G., and Markatou, M. (2009). Large datasets in biomedicine: a discussion of salient analytic issues. *Journal of the American Medical Informatics Association*, 16(6):759–767.

Skeppstedt, M., Dalianis, H., et al. (2014). Automatic recognition of disorders, findings, pharmaceuticals and body structures from clinical text: An annotation and machine learning study. *Journal of Biomedical Informatics*, 49:148 – 158.

Smistad, E., Falch, T. L., Bozorgi, M., Elster, A. C., and Lindseth, F. (2015). Medical image segmentation on gpus–a comprehensive review. *Medical image analysis*, 20(1):1–18.

Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 – 197.

Smyth, P. (1996). Bounds on the mean classification error rate of multiple experts. *Pattern Recognition Letters*, 17(12):1253–1257.

Soguero-Ruiz, C., Fei, W. M., Jenssen, R., Augestad, K. M., et al. (2015). Data-driven temporal prediction of surgical site infection. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1164. American Medical Informatics Association.

Soguero-Ruiz, C., Hindberg, K., Mora-Jiménez, I., Rojo-Álvarez, J. L., et al. (2016a). Predicting colorectal surgical complications using heterogeneous clinical data and kernel methods. *Journal of Biomedical Informatics*, 61:87–96.

Soguero-Ruiz, C., Hindberg, K., Rojo-Álvarez, J. L., Skrøvseth, S. O., et al. (2016b). Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records. *IEEE journal of biomedical and health informatics*, 20(5):1404–1415.

Soleimani, H., Hensman, J., and Saria, S. (2018). Scalable joint models for reliable uncertainty-aware event prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(8):1948–1963.

Soleimani, H., Subbaswamy, A., and Saria, S. (2017). Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions. *arXiv preprint arXiv:1704.02038*.

Song, Y., Nie, F., Zhang, C., and Xiang, S. (2008). A unified framework for semi-supervised dimensionality reduction. *Pattern recognition*, 41(9):2789–2799.

Stearns, M. Q., Price, C., Spackman, K. A., and Wang, A. Y. (2001). Snomed clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium*, page 662. American Medical Informatics Association.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Strehl, A. and Ghosh, J. (2003). Cluster Ensembles — a Knowledge Reuse Framework for Combining Multiple Partitions. *The Journal of Machine Learning Research*, 3:583–617.

Stuetzle, W. and Nugent, R. (2012). A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2014). Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

Sun, J., Sow, D., Hu, J., and Ebadollahi, S. (2010a). Localized supervised metric learning on temporal physiological data. In *2010 20th International Conference on Pattern Recognition*, pages 4149–4152.

Sun, J., Sow, D., Hu, J., and Ebadollahi, S. (2010b). A system for mining temporal physiological data streams for advanced prognostic decision support. In *2010 IEEE International Conference on Data Mining*, pages 1061–1066.

Sun, J., Wang, F., Hu, J., and Edabollahi, S. (2012). Supervised patient similarity measure of heterogeneous patient records. *SIGKDD Explor. Newsl.*, 14(1):16–24.

Suo, Q., Ma, F., Yuan, Y., Huai, M., Zhong, W., Zhang, A., and Gao, J. (2017). Personalized disease prediction using a cnn-based similarity learning method. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 811–816.

Suresh, H., Hunt, N., Johnson, A., Celi, L. A., Szolovits, P., and Ghassemi, M. (2017). Clinical intervention prediction and understanding with deep neural networks. In Doshi-Velez, F. et al., editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68, pages 322–337, Boston, Massachusetts. PMLR.

Sutton, R. S., Barto, A. G., Bach, F., et al. (1998). *Reinforcement learning: An introduction*. MIT press.

Sweeney, J. F. (2013). Postoperative complications and hospital readmissions in surgical patients: an important association. *Annals of surgery*, 258(1):19.

Tahir, M. A., Kittler, J., and Bouridane, A. (2012). Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recognition Letters*, 33(5):513–523.

Tamang, S., Milstein, A., Sørensen, H. T., Pedersen, L., Mackey, L., Betterton, J.-R., Janson, L., and Shah, N. (2017). Predicting patient 'cost blooms' in denmark: a longitudinal population-based study. *BMJ Open*, 7(1).

Tanha, J., van Someren, M., and Afsarmanesh, H. (2017). Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*, 8(1):355–370.

Taylor, R. A., Moore, C. L., Cheung, K.-H., and Brandt, C. (2018). Predicting urinary tract infections in the emergency department with machine learning. *PloS one*, 13(3):e0194085.

Teixeira, P. L., Wei, W.-Q., Cronin, R. M., Mo, H., VanHouten, J. P., et al. (2017). Evaluating electronic health record data sources and algorithmic approaches to identify hypertensive individuals. *Journal of the American Medical Informatics Association*, 24(1):162–171.

Topchy, A., Jain, A. K., and Punch, W. (2005). Clustering ensembles: Models of consensus and weak partitions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1866–1881.

Trajdos, P. and Kurzynski, M. (2015). An extension of multi-label binary relevance models based on randomized reference classifier and local fuzzy confusion matrix. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 69–76. Springer.

Trajdos, P. and Kurzynski, M. (2018). Weighting scheme for a pairwise multi-label classifier based on the fuzzy confusion matrix. *Pattern Recognition Letters*, 103:60 – 67.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative review. *J Mach Learn Res*, 10:66–71.

Vani, A., Jernite, Y., and Sontag, D. (2017). Grounded Recurrent Neural Networks. *ArXiv e-prints arXiv:1705.08557*.

Vapnik, V. N. and Chervonenkis, A. Y. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer.

Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372.

Velupillai, S., Mowery, D., South, B. R., Kvist, M., and Dalianis, H. (2015). Recent advances in clinical natural language processing in support of semantic analysis. *Yearbook of medical informatics*, 10(1):183.

Velupillai, S., Skeppstedt, M., Kvist, M., Mowery, D., et al. (2014). Cue-based assertion classification for swedish clinical text—developing a lexicon for pycontextswe. *Artificial Intelligence in Medicine*, 61(3):137 – 144.

Viani, N., Miller, T. A., Dligach, D., et al. (2017a). Recurrent neural network architectures for event extraction from italian medical reports. In ten Teije, A. et al., editors, *Artificial Intelligence in Medicine*, pages 198–202, Cham. Springer International Publishing.

Viani, N., Sacchi, L., Tibollo, V., et al. (2017b). Clinical timelines development from textual medical reports in italian. In *IEEE 3rd Int. Forum on Research and Technologies for Society and Industry (RTSI)*, pages 1–5.

Vincent, P., Larochelle, H., Lajoie, I., et al. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408.

von Luxburg, U. (2007). A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416.

Vranas, K. C., Jopling, J. K., Sweeney, T. E., Ramsey, M. C., Milstein, A. S., Slatore, C. G., Escobar, G. J., and Liu, V. X. (2017). Identifying distinct subgroups of icu patients: a machine learning approach. *Critical care medicine*, 45(10):1607–1615.

Wang, F. (2015). Adaptive semi-supervised recursive tree partitioning: the art towards large scale patient indexing in personalized healthcare. *Journal of biomedical informatics*, 55:41–54.

Wang, F., Li, T., and Zhang, C. (2008). Semi-supervised clustering via matrix factorization. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 1–12. SIAM.

Wang, F. and Sun, J. (2015a). Psf: a unified patient similarity evaluation framework through metric learning with weak supervision. *IEEE journal of biomedical and health informatics*, 19(3):1053–1060.

Wang, F. and Sun, J. (2015b). Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, 29(2):534–564.

Wang, F. and Zhang, C. (2007). Feature extraction by maximizing the average neighborhood margin. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

Wang, F. and Zhang, C. (2008). Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67.

Wang, S., Wang, J., Wang, Z., and Ji, Q. (2014a). Enhancing multi-label classification by modeling dependencies among labels. *Pattern Recognition*, 47(10):3405 – 3413.

Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., and MacNeille, P. (2017a). A bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning*, 18(70):1–37.

Wang, W. and Krishnan, E. (2014). Big data and clinicians: a review on the state of the science. *JMIR medical informatics*, 2(1).

Wang, X., Qian, B., and Davidson, I. (2014b). On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, 28(1):1–30.

Wang, X., Sontag, D., and Wang, F. (2014c). Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 85–94, New York, NY, USA. ACM.

Wang, Y., Chen, R., Ghosh, J., et al. (2015a). Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '15, pages 1265–1274, New York, NY, USA. ACM.

Wang, Y., Tian, Y., Tian, L.-L., Qian, Y.-M., and Li, J.-S. (2015b). An electronic medical record system with treatment recommendations based on patient similarity. *Journal of medical systems*, 39(5):55.

Wang, Z., Li, L., Glicksberg, B. S., Israel, A., Dudley, J. T., and Ma'ayan, A. (2017b). Predicting age by mining electronic medical records with deep learning characterizes differences between chronological and physiological age. *Journal of Biomedical Informatics*, 76:59 – 68.

Watanabe, T., Miyata, H., Konno, H., et al. (2017). Prediction model for complications after low anterior resection based on data from 33,411 japanese patients included in the national clinical database. *Surgery*, 161(6):1597 – 1608.

Weegar, R., Kvist, M., Sundström, K., et al. (2015). Finding cervical cancer symptoms in swedish clinical text using a machine learning approach and negex. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1296.

Wei, W.-Q. and Denny, J. C. (2015). Extracting research-quality phenotypes from electronic health records to support precision medicine. *Genome medicine*, 7(1):41.

Weiskopf, N. G. and Weng, C. (2013). Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association*, 20(1):144–151.

Wells, B. J., Chagin, K. M., Nowacki, A. S., and Kattan, M. W. (2013). Strategies for handling missing data in electronic health record derived data. *eGEMs*, 1(3).

WHO (2004). *International statistical classification of diseases and related health problems*, volume 1. World Health Organization.

WHO (2014). Global status report on noncommunicable diseases 2014: attaining the nine global noncommunicable diseases targets; a shared responsability. In *Global status report on noncommunicable diseases 2014: attaining the nine global noncommunicable diseases targets; a shared responsability*. World Health Organization.

WHO (2015). *World report on ageing and health*. World Health Organization.

WHO (2016). *Collaborating Centre for Drug Statistics Methodology, Guidelines for ATC classification and DDD assignment*. World Health Organization.

Wickstrøm, K., Kampffmeyer, M., and Jenssen, R. (2018). Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps. *arXiv preprint arXiv:1807.10584*.

Wilson, D. R. and Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine learning*, 38(3):257–286.

Wu, M., Ghassemi, M., Fend, M., Celi, L. A., Szolovits, P., and Doshi-Velez, F. (2017). Understanding vasopressor intervention and weaning: Risk prediction in a public heterogeneous clinical time series database. *Journal of the American Medical Informatics Association*, 24(3):488–495.

Xiao, C., Choi, E., and Sun, J. (2018). Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xing, X., Yu, Y., Jiang, H., and Du, S. (2013). A multi-manifold semi-supervised gaussian mixture model for pattern classification. *Pattern Recognition Letters*, 34(16):2118 – 2125.

Xu, J. (2013). Fast multi-label core vector machine. *Pattern Recognition*, 46(3):885 – 898.

Xu, Y., Xu, Y., and Saria, S. (2016). A bayesian nonparametric approach for estimating individualized treatment-response curves. In *Machine Learning for Healthcare Conference*, pages 282–300.

Xue, Y., Klabjan, D., and Luo, Y. (2018). Predicting icu readmission using grouped physiological and medication trends. *Artificial Intelligence in Medicine.*

Yadav, P., Steinbach, M., Kumar, V., and Simon, G. (2018). Mining electronic health records (ehrs): A survey. *ACM Comput. Surv.*, 50(6):85:1–85:40.

Yang, K., Li, X., Liu, H., Mei, J., Xie, G., et al. (2017). Tagited: Predictive task guided tensor decomposition for representation learning from electronic health records. In *Proc. of 31st AAAI Conference on Artificial Intelligence.*

Yang, X., Fu, H., Zha, H., and Barlow, J. (2006). Semi-supervised nonlinear dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 1065–1072, New York, NY, USA. ACM.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. (2016). Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 40–48. JMLR. org.

Ye, L. and Keogh, E. (2009). Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM.

Yousefi, L., Saachi, L., Bellazzi, R., Chiovato, L., and Tucker, A. (2017). Predicting comorbidities using resampling and dynamic bayesian networks with latent variables. In *IEEE 30th Int. Symp.on Computer-Based Medical Systems (CBMS)*, pages 205–206.

Yu, K.-H., Zhang, C., Berry, G. J., Altman, R. B., Ré, C., Rubin, D. L., and Snyder, M. (2016). Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nature communications*, 7:12474.

Yu, S., Chakrabortty, A., Liao, K. P., et al. (2017a). Surrogate-assisted feature extraction for high-throughput phenotyping. *Journal of the American Medical Informatics Association*, 24(e1):e143–e149.

Yu, S., Liao, K. P., Shaw, S. Y., et al. (2015). Toward high-throughput phenotyping: unbiased automated feature extraction and selection from knowledge sources. *Journal of the American Medical Informatics Association*, 22(5):993–1000.

Yu, Y., Wang, J., Tan, Q., Jia, L., and Yu, G. (2017b). Semi-supervised multilabel dimensionality reduction based on dependence maximization. *IEEE Access*, 5:21927–21940.

Zhan, M., Cao, S., Qian, B., Chang, S., and Wei, J. (2016). Low-rank sparse feature selection for patient similarity learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1335–1340.

Zhang, D., Zhou, Z.-H., and Chen, S. (2007). Semi-supervised dimensionality reduction. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 629–634. SIAM.

Zhang, J., Kowsari, K., Harrison, J. H., Lobo, J. M., and Barnes, L. E. (2018). Patient2Vec: A Personalized Interpretable Deep Representation of the Longitudinal Electronic Health Record. *ArXiv e-prints*.

Zhang, M.-L. and Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048.

Zhang, P., Wang, F., and Hu, J. (2014). Towards drug repositioning: a unified computational framework for integrating multiple aspects of drug similarity and disease similarity. In *AMIA Annual Symposium Proceedings*, volume 2014, page 1258. American Medical Informatics Association.

Zhang, P., Wang, F., Hu, J., and Sorrentino, R. (2015). Label propagation prediction of drug-drug interactions based on clinical side effects. *Scientific reports*, 5:12339.

Zhang, Y., Chen, R., Tang, J., Stewart, W. F., and Sun, J. (2017). Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1315–1324, New York, NY, USA. ACM.

Zhao, J., Henriksson, A., Asker, L., and Boström, H. (2014a). Detecting adverse drug events with multiple representations of clinical measurements. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pages 536–543. IEEE.

Zhao, J., Henriksson, A., Asker, L., and Boström, H. (2015a). Predictive modeling of structured electronic health records for adverse drug event detection. *BMC medical informatics and decision making*, 15(4):S1.

Zhao, J., Henriksson, A., and Bostrom, H. (2014b). Detecting adverse drug events using concept hierarchies of clinical codes. In *2014 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 285–293. IEEE.

Zhao, J., Henriksson, A., Kvist, M., Asker, L., and Boström, H. (2015b). Handling temporality of clinical events for drug safety surveillance. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1371. American Medical Informatics Association.

Zhao, J., Papapetrou, P., Asker, L., and Boström, H. (2017). Learning from heterogeneous temporal data in electronic health records. *Journal of biomedical informatics*, 65:105–119.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328.

Zhou, Z.-H. (2017). A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.

Zhu, T., Johnson, A. E. W., Yang, Y., Clifford, G. D., and Clifton, D. A. (2018). Bayesian fusion of physiological measurements using a signal quality extension. *Physiological Measurement*, 39(6):065008.

Zhu, X. (2005). *Semi-supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA. AAI3179046.

Zhu, X. (2006). Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003a). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.

Zhu, X., Wu, X., and Chen, Q. (2003b). Eliminating class noise in large datasets. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 920–927.

Zhu, Z., Yin, C., Qian, B., Cheng, Y., Wei, J., and Wang, F. (2016). Measuring patient similarities via a deep architecture with medical concept embedding. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 749–758. IEEE.

Zhuang, N., Yan, Y., Chen, S., Wang, H., and Shen, C. (2018). Multi-label learning based deep transfer neural network for facial attribute classification. *Pattern Recognition*, 80:225 – 240.