2018 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2018, AALBORG, DENMARK

REMOTE SENSING IMAGE REGRESSION FOR HETEROGENEOUS CHANGE DETECTION

Luigi T. Luppino[1]*, Filippo M. Bianchi*[1]*, Gabriele Moser*[2]*, Stian N. Anfinsen*[1]

Machine Learning Group, Department of Physics and Technology, UiT The Arctic University of Norway[1]
DITEN Department, University of Genoa, Italy[2]
luigi.t.luppino@uit.no

ABSTRACT

Change detection in heterogeneous multitemporal satellite images is an emerging topic in remote sensing. In this paper we propose a framework, based on image regression, to perform change detection in heterogeneous multitemporal satellite images, which has become a main topic in remote sensing. Our method learns a transformation to map the first image to the domain of the other image, and vice versa. Four regression methods are selected to carry out the transformation: Gaussian processes, support vector machines, random forests, and a recently proposed kernel regression method called homogeneous pixel transformation. To evaluate not only potentials and limitations of our framework, but also the pros and cons of each regression method, we perform experiments on two data sets. The results indicates that random forests achieve good performance, are fast and robust to hyperparameters, whereas the homogeneous pixel transformation method can achieve better accuracy at the cost of a higher complexity.

***Index Terms*—** Domain adaptation, heterogeneous image sources, change detection, regression.

1. INTRODUCTION

Change detection (CD) is a well known task in satellite remote sensing: the goal is to recognise changes in imagery acquired on the same location but at different times. The applications range from disaster assessment to long term trend monitoring [1, 2, 3]. Most of the past works on CD assume that the satellite images are homogeneous, i.e. the data were collected by the same kind of sensors and using the same configurations and modalities [1, 2, 3]. Even though there are techniques which mitigate the issues due to misalignments [2, 4, 5], co-registration is another fundamental assumption for CD: every pixel of the image at time one and its corresponding pixel of the image at time two are assumed to represent the exact same location on the earth.

The development of new sensors and the improvement of their capabilities has eventually brought the remote sensing community to consider the use of satellite images acquired under heterogeneous conditions [1, 3, 6, 7]. This has led to methods based on heterogeneous sources of data [1, 2, 6, 8, 9, 10], also referred to as multi-source [3, 11], multi-modal [4], multi-sensor or cross-sensor [5, 12, 13, 14] and information unbalanced data [15]. As already reviewed in [3], there is not a unique way to group CD methods. However, the distinction between techniques aimed at homogeneous and heterogeneous data is clear. In the latter case, the assumptions that the same physical quantities are measured, classes have always the same signatures, and data follow the same statistical behaviour are no longer valid [7]. Without any additional steps, traditional homogeneous CD techniques cannot handle this [7, 10]. To overcome the problem, a possible preliminary step is either project data from both times into a common domain [3, 11, 12, 13] or transfer data from the time one to the time two domain [8, 10]. These methodologies are related to topics such as domain adaptation, data transformation and transfer learning [4, 6, 10, 11, 16]. Post-classification comparison represents an exception: the best classifier for the pre-event data and the best one for the post-event data are selected, then the classification maps are compared to find the pixels which do not belong to the same class at both times. Clearly, the performance with this approach depends highly on the choice and the design of the two classifiers, as well as on the quality and the size of the training set [1]. The exponential increase of interest in deep learning has also lead to the development of novel methods based on deep learning architectures, both in the homogeneous [16, 17] and in the heterogeneous case [1, 2, 15]. Most of these methods are examples of feature learning, since they exploit the capability of e.g. convolutional neural networks (and especially stacked denoising autoencoders) to infer spatial information from the data and consequently to learn a new representation of it.

In this work, we suggest a simple, yet effective methodology to perform CD with heterogeneously acquired data. It consists of training a regression function to predict how every pixel at time one would have been if it was acquired by a second sensor at time two, and vice versa. We will refer to this methodology as *image regression*, a term which has on some occasions been used in the CD literature [18, 19, 20]. Once the predictions of the images are computed, homogenenous CD methods can be applied to obtain the map of changes. In particular, we consider three supervised methods to perform the regression: Gaussian processes, support vector machines, and random forests. Moreover, the homogeneous pixel transformation method recently proposed by Liu et al. [10] is chosen as a representative of the state-of-the-art. As main contribution, we evaluate the performance of the different regression methods in the proposed framework. By testing a selection of both well-established and more recent regression methods on two different data sets, we evaluate the consistency of their performance, as well as their pros and cons, helping the user to choose the most suitable approach according to requirements, such as best performance in detection, easiest tuning of the hyperparameters, shortest training and test time. The remainder of this article is the following: Section 2 introduces the reader to the methodology, the notation, and the regression methods listed above. Results on two data sets are presented in Section 3. Section 4 concludes the paper.

2. METHODOLOGY

We follow the notation adopted in [10]: the two images represent the same region but are acquired by different sensors at a different times

and are denoted as $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. A limited part of the image has changed between time one and time two. A training data set $\mathcal{T}$ of M corresponding pixel pairs, $\mathcal{T} = \{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$, is manually selected from areas not affected by changes in the two images. According to [10], this provision of training data is not a strong requirement, although it prompts user interaction. The training data $\mathcal{T}$ allows us to learn a regression function $f^{(1)}$ such that

$$\boldsymbol{y}_m = f^{(1)}(\boldsymbol{x}_m) = \hat{\boldsymbol{y}}_m + \epsilon_m^{(1)}, \quad m = 1, \ldots, M \tag{1}$$

where $\hat{\boldsymbol{y}}_m$ is the dependent variable, $\boldsymbol{x}_m$ is the regressor, and $\epsilon_m^{(1)}$ is the residual. We then train the reverse regression equation

$$\boldsymbol{x}_m = f^{(2)}(\boldsymbol{y}_m) = \hat{\boldsymbol{x}}_m + \epsilon_m^{(2)}, \quad m = 1, \ldots, M \tag{2}$$

in which $\hat{\boldsymbol{x}}_m$ is predicted starting from the regressor $\boldsymbol{y}_m$. With these two functions it is possible to predict $\hat{\boldsymbol{Y}}$, i.e. the image which would have been obtained if sensor $\mathcal{Y}$ had observed the reality at time one, and $\hat{\boldsymbol{X}}$, the image of the reality at time two which would have been acquired by sensor $\mathcal{X}$. Once the two predictions are computed, conventional change metrics such as image differences or ratios can be applied to highlight the differences between the original images and the corresponding predicted ones. There is a plethora of more complex and more effective homogeneous CD techniques which could be applied at this stage, but the main goal of this work is to compare the image regression methods applied to obtain the predicted images. The two-way regression can be referred to as an ensemble approach where two weaker results are combined to obtain a stronger and more reliable outcome.

Let the distance image be defined as

$$d(\cdot,\cdot) : \mathbf{R}^{n1\times n2\times P} \times \mathbf{R}^{n1\times n2\times P} \longrightarrow \mathbf{R}^{n1\times n2},$$

i.e. a pixel-wise distance between two images of size $n1 \times n2$ and P channels. When the distance images, $d(\boldsymbol{X}, \hat{\boldsymbol{X}})$ and $d(\boldsymbol{Y}, \hat{\boldsymbol{Y}})$, are normalised and combined, distances that are consistently high in both images will indicate high probability of change, whereas false alarms due to a spurious high value in one of the distances will be suppressed. We choose to combine the distances by a simple average. Before normalising, it is reasonable to clip the distances beyond some standard deviations of the mean value (e.g. $d_i > \bar{d} + 4\sigma_d$), so that outliers do not compromise such a step. At this stage, noise filtering can be applied if necessary. Finally, a change map can be achieved by thresholding. Fig. 1 illustrates the methodology. In the following we briefly describe the regression methods considered in this work to evaluate $f^{(1)}$ and $f^{(2)}$.

2.1. Gaussian Process Regression

A Gaussian process (GP) is a collection of random variables, any finite subset of which have a joint Gaussian distribution. It is completely specified by its mean function $\boldsymbol{m}(\boldsymbol{x})$ and covariance (kernel) function $k_{\boldsymbol{x}_i,\boldsymbol{x}_j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. For regression purposes, zero mean GPs are most often used [21]. Given a training set of M input vectors (arranged in rows) $\boldsymbol{X} \in \mathbf{R}^{M\times P}$, the corresponding set of target vectors $\boldsymbol{Y} \in \mathbf{R}^{M\times Q}$, and a set of N new observed vectors $\boldsymbol{X}_* \in \mathbf{R}^{N\times P}$, the joint distribution of the training vectors $\boldsymbol{Y}$ and the sought regressed vectors $\hat{\boldsymbol{Y}} \in \mathbf{R}^{N\times Q}$, conditioned on the input data $\boldsymbol{X}$ and $\boldsymbol{X}_*$, is

$$\begin{bmatrix} \boldsymbol{Y} \\ \hat{\boldsymbol{Y}} \end{bmatrix} | \boldsymbol{X}, \boldsymbol{X}_* \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}_*} \\ \boldsymbol{K}_{\boldsymbol{X}_*,\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}_*,\boldsymbol{X}_*} \end{bmatrix}\right). \tag{3}$$

where $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}_*}$ is the matrix whose (i,j)th entry is the value of the covariance on the ith row of $\boldsymbol{X}$ and the jth row of $\boldsymbol{X}_*$, and $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$, $\boldsymbol{K}_{\boldsymbol{X}_*,\boldsymbol{X}}$, and $\boldsymbol{K}_{\boldsymbol{X}_*,\boldsymbol{X}_*}$ have similar meanings. Thus, the following posterior distribution is derived (see [21] for details):

$$\begin{aligned} \hat{\boldsymbol{Y}} \,|\, \boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}\big(&\boldsymbol{K}_{\boldsymbol{X}_*,\boldsymbol{X}} \cdot \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}^{-1} \cdot \boldsymbol{Y}, \\ &\boldsymbol{K}_{\boldsymbol{X}_*,\boldsymbol{X}_*} - \boldsymbol{K}_{\boldsymbol{X}_*,\boldsymbol{X}} \cdot \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}^{-1} \cdot \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}_*}\big) \end{aligned} \tag{4}$$

The two main factors affecting the quality of the regression are the choice of kernel function and its hyperparameters. In this work, we opted for the commonly used radial basis function (RBF)

$$k_{\boldsymbol{x}_i,\boldsymbol{x}_j} = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T L (\boldsymbol{x}_i - \boldsymbol{x}_j)\right), \tag{5}$$

where $\boldsymbol{\theta} = \{L, \sigma_f^2\}$ is the set of hyperparameters, with signal variance σ_f^2 and $L = l^{-2}I$, if the length-scale parameter l is a scalar (isotropic kernel), or $L = \mathrm{diag}(\boldsymbol{l}^{-2})$, if $\boldsymbol{l}$ is a vector (anisotropic kernel) [21]. Concerning the optimisation of $\boldsymbol{\theta}$, a gradient ascent is performed to maximise the marginal likelihood $\mathcal{P}(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{\theta})$. A weak point of this optimisation is that it might lead to a local maximum instead of the global one, so it is recommended to iterate the procedure several times starting from different random points in the hyperparameter space $\Omega_{\boldsymbol{\theta}}$.

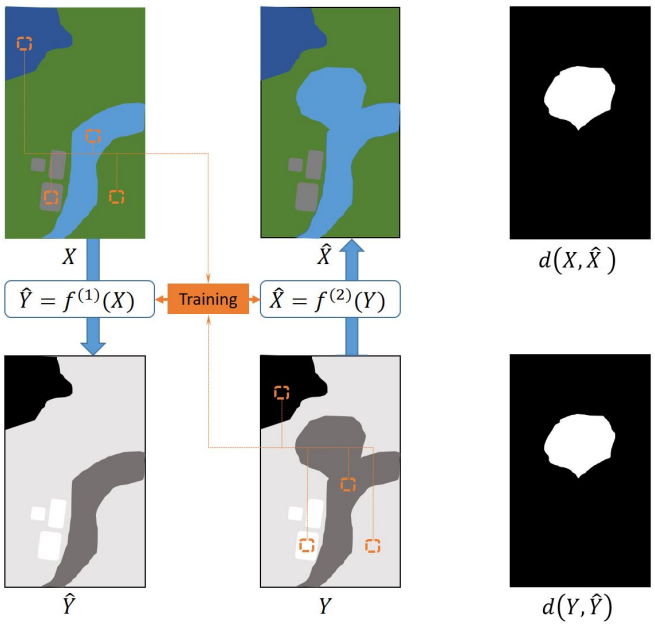


Fig. 1: Image regression: the two functions $f^{(1)}$ and $f^{(2)}$ are trained starting from the same data points, two predicted images are obtained, and finally two difference images are achieved.

2.2. Multi-output Support Vector Regression

Support vector machines (SVMs) are a very well known machine learning approach used for classification and regression. By solving the so-called dual problem, it is possible to find the best separating or fitting curve with respect to a loss function that accounts for misclassification or reconstruction error and with respect to a regularisation parameter which defines the width of a soft margin around such a curve. In addition, the support vectors, i.e. the training points that define the margin, are highlighted from the rest of the training set.

Instead of coping with multi-output problems all at once, the solution usually adopted is to tune a different SVM for each regressand

variable. Therefore, the standard implementations of support vector regression (SVR) are designed to predict a single output feature, ignoring the potentially nonlinear relations across the target features [22]. Tuia et al. [22] proposed a multi-input multi-output (MIMO) SVR method to overcome this limitation. During the training phase, it aims to minimise the cost function

$$L_p\left(\boldsymbol{W}, \boldsymbol{b}\right) = \frac{1}{2}\sum_{q=1}^{Q} \parallel \boldsymbol{w}_q \parallel^2 + C\sum_{m=1}^{M} L\left(\mu_m\right) \tag{6}$$

where

$$L\left(\mu_m\right) = \begin{cases} 0 & \mu_m < \epsilon \\ \mu_m^2 - 2\mu_m\epsilon + \epsilon^2 & \mu_m \geq \epsilon \end{cases}, \tag{7}$$

$$\mu_m = \parallel \boldsymbol{e}_m \parallel = \sqrt{\boldsymbol{e}_m^T \boldsymbol{e}_m}\,, \tag{8}$$

$$\boldsymbol{e}_m^T = \boldsymbol{y}_m^T - \phi(\boldsymbol{x}_m)^T\,\boldsymbol{W} - \boldsymbol{b}^T. \tag{9}$$

Here, $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_Q]$, $\boldsymbol{w_q} \in \mathbf{R}^P$ and $\boldsymbol{b} = [q_1, \ldots, q_Q]$ are the coefficients and the bias of the linear combination of the data points $\boldsymbol{x}_m$ transferred in the Hilbert space by the kernel function ϕ. The penalty factor C sets the trade-off between the regularisation term and the sum of the error terms $L\left(\mu_m\right)$. If it is too large, nonseparable points would highly penalise the cost function and too many data points would turn into support vectors, causing overfitting. Vice versa, a small C would lead to underfitting. ϵ is half the width of the insensitivity zone. This zone delimits a "tube" around the approximated function and all the training data points within the insensitivity zone do not contribute to the cost function (see Eq. 7). For too small values of ϵ, too many data points would be considered as support vectors (overfitting), the generalisation performance would be affected and the variance of the fitted curve would be too large. On the contrary, a too large ϵ would cause underfitting and the overall accuracy would be low. Another critical hyperparameter is the width σ of the RBF kernel ϕ. To select the right combination of hyperparameters $\boldsymbol{\theta} = \{C, \epsilon, \sigma\}$, a grid search for the smallest cross-validation error or the minimization of an error bound can be applied.

2.3. Random Forest Regression

Random forests (RF) were proposed by Breiman in [23] to perform both classification and regression, by exploiting the simplicity of random decision trees and the robustness of ensemble methods. From now on, only regression will be considered, but for classification purposes the approach is similar.

A RF consists on T trees, at whose nodes m randomly selected features are compared to random thresholds (e.g. $\text{feat}_1 > \text{thr}_1$ & $\ldots$ & $\text{feat}_m > \text{thr}_m$). In each tree, the training data points are divided over the branches according to these conditions, and the trees expand until only one data point is contained in each of the final nodes (leaves). Thus, the corresponding training vectors $\boldsymbol{y}_m$ are assigned to the leaves. During the test phase an input vector $\boldsymbol{x_*}$ goes through each tree and reaches one of the leaves, giving as output the assigned training vector $\boldsymbol{y}_m$. Finally, the average of the T outputs is computed, thereby obtaining the final regressed vector $\hat{\boldsymbol{y}}$.

To generalise better, every tree is trained on a bootstrap sample drawn from the training set, and a randomly drawn subset of features (of fixed cardinality) is used on each node of each tree. The validation is carried out through out-of-bag estimation [23]. Moreover, the behaviour of a RF can be controlled by tuning three parameters: the size of the forest (i.e. the number of trees T), the number of features m considered in every node, and the depth of the trees. A common remedy against overfitting is to prune the trees by leaving $p > 1$ data points in each leaf node, which will give in output the average of their corresponding training vectors $\boldsymbol{y}_m$. Concerning the number of features considered at every node, in [23] it is suggested by empirical results to set $m = \lfloor \frac{\log P}{\log 2} \rfloor$, where P is the dimension of the vectors $\boldsymbol{y}$. It is common practice to follow the rule of thumb: $m = \lfloor P/3 \rfloor$ [23]. However, there are no practical rules to choose the size of the forest. One may think that for a larger number of trees the outcomes become better, but [23] proved that at some point the overall accuracy saturates due to the rise of a strong correlation between the trees. Therefore, a compromise between gained accuracy and computational load must be found.

2.4. Homogeneous Pixel Transformation

The homogeneous pixel transformation (HPT) method proposed by Liu et al. [10] is a kernel regression based on the K-nearest neighbours (KNN) of each data point. This technique recalls the distance weighted averaging or locally weighted regression previously presented in [24], where many related aspects are also studied: possible kernels, distance measures, choices of the bandwidth, denoising techniques, and outlier detection. For every data point in the first image $\boldsymbol{x}_i$, the K nearest neighbours among the training vectors $\boldsymbol{x}_m \in \mathcal{T}$ are sought for. The regression consists of the weighted sum

$$\hat{\boldsymbol{y}}_i = \sum_{k=1}^{K} w_{i,k} \cdot \boldsymbol{y}_{i,k}\,, \tag{10}$$

where

$$w_{i,k} = w\left(\boldsymbol{x}_i, \boldsymbol{x}_k\right) = e^{-\gamma d_{i,k}}\,. \tag{11}$$

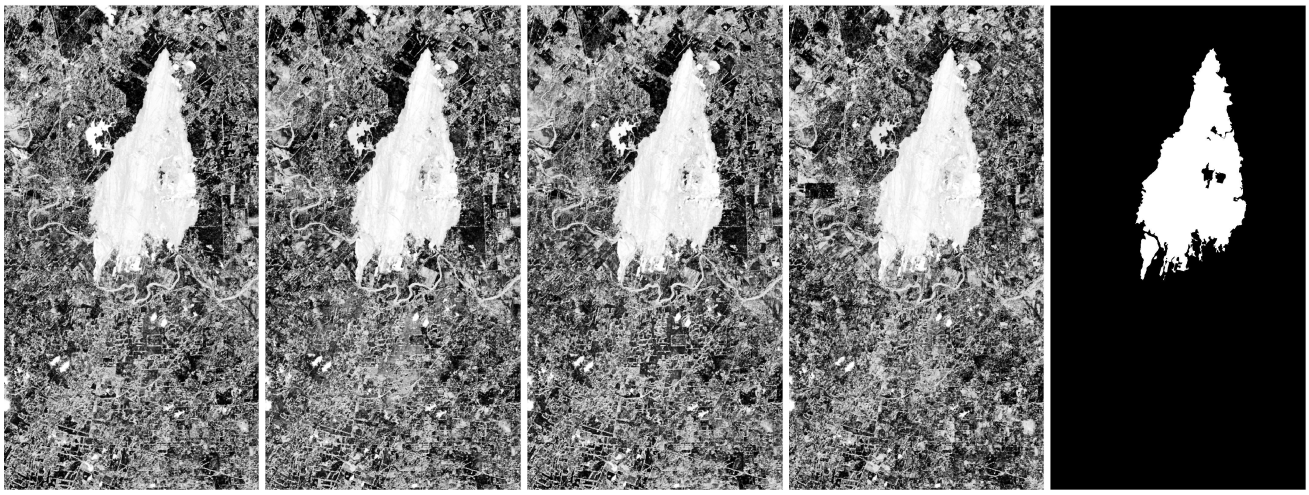
$d_{i,k}$ is the Euclidean distance between $\boldsymbol{x}_i$ and its k^{th} nearest neighbour $\boldsymbol{x}_k$, $\boldsymbol{y}_{i,k}$ is the corresponding vector of $\boldsymbol{x}_k$ in $\mathcal{T}$, whereas the kernel width γ regulates how strongly the farthest neighbours are penalised. If γ is too small, the addends tend to be equally weighted and the sum is close to an average, if γ is too large, few main addends contribute to the sum whilst the rest are heavily penalised. Before computing the weights, a relative normalisation of the distances is applied:

$$d_{i,k} = \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_k\|}{\max_k \|\boldsymbol{x}_i - \boldsymbol{x}_k\|}\,. \tag{12}$$

The normalisation in [10] is defined as relative, because it considers the maximum among the distances between the data point $\boldsymbol{x}_i$ and its neighbours. However, while testing our implementation, we found that it is better to perform an absolute normalisation, thus seeking the maximum among all the computed distances.

3. EXPERIMENTAL RESULTS

The performance of the CD framework, configured with the proposed regression methods, is evaluated on two different data sets in terms of accuracy and computational speed. Accuracy is measured in terms of *area under the curve* (AUC), a value between 0.5 (poor) and 1 (optimal) which indicates the area below the receiver operating characteristic curve, which plots the false positive rate against the true positive rate. The measured speed is the elapsed time during computation of the regressions in both the directions, starting from the training phase and ending after the test phase. It must be pointed out that two of the methods are implemented in Python libraries (GP and RF), whereas the code provided by [22] for the



(a) GP (b) SVR (c) RF (d) HPT (e) Ground Truth

Fig. 2: Distance images obtained by applying the proposed approach with different regression methods: (a) Gaussian process regression, (b) MIMO support vector regression, (c) random forest regression, (d) homogeneous pixel transformation [10], (e) ground truth.

MIMO SVR method is written in MATLAB, and so is our implementation of the HPT method. Therefore, an exact comparison of execution time of each algorithm is not possible, even though the two programming languages yield similar performance. Nevertheless, the running times are indicators that can help us rank the four algorithms in terms of speed.

3.1. Forest fire in Texas

The first data set is composed of a multispectral image acquired by Landsat 5 TM before a forest fire in Bastrop County, Texas, during September-October, 2011. An EO-1 ALI multispectral acquisition after the event completes the data set[1]. Both images are optical with 7 and 10 channels, respectively, some of which cover the same spectral bands, so the signatures of the classes involved are very similar. Among the possible heterogeneous CD scenarios, this is one of the easiest. The ground truth of the event (see Fig. 2e) is provided by Volpi et al. [13]. The training set (roughly 2% of the total data points) is selected manually with several rectangular patches taken from areas not affected by the fire event. In a preliminary study phase, the hyperparameters of the GPs are set after only one iteration of the gradient ascent. For the SVR, $C = 1$, $\epsilon = 0.1$, and $\sigma = 1$ are set following [22]. Concerning the RF, $T = 128$, $m = [p/3]$, $p = 5$ are chosen after a coarse grid search on T and p. Last, the HPT is tuned by setting $K = 300$ and $\gamma = 100$, as empirically found in [10].

After the combination of the two image differences, a 3×3 median filter is applied to remove salt and pepper noise. In Fig. 2, the outcomes of the median filter for the four methods are depicted, showing how well they all behave. The only exception is the HPT, which tends to overfit, as can be noticed in Fig. 2d. One example is the black rectangular patch on the left of the area interested by the event, which actually corresponds to one of the selected patches of the training set. However, all the AUCs reach values above 98%, showing how well all four methods tackle the image regression task. On one hand, this image pair is not especially challenging for the proposed approaches. On the other hand, conventional CD methods designed for homogeneous data would be unfeasible here. Hence, this result demonstrate the effectiveness of the proposed regression-based approach to heterogeneous CD.

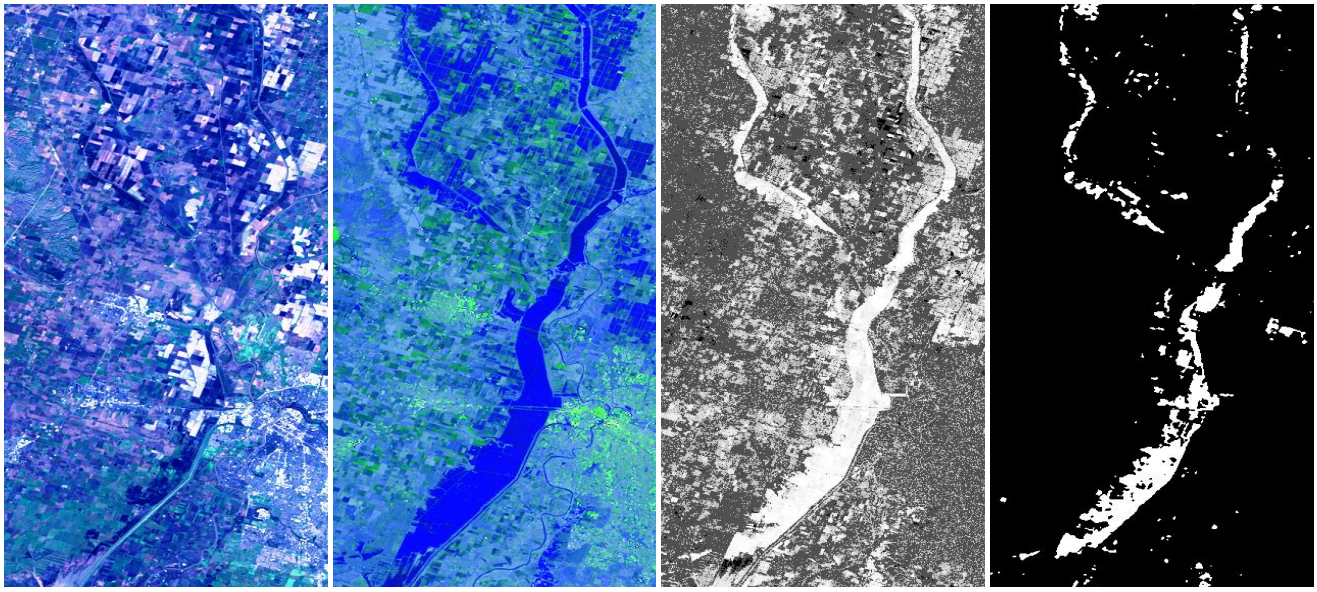
3.2. Flood in California

The second data set represents a more challenging scenario, as it involves an optical image and a synthetic aperture radar (SAR) image. The image at time 1 is a Landsat 8 acquisition covering Sutter County, California, on 5 January 2017[1]. It is composed of 9 channels covering the spectrum from deep blue to short-wave infrared, plus two long-wave infrared channels (Fig. 3a shows the RGB channels). Fig. 3b shows the image acquired on 18 February 2017 by Sentinel-1A over the same scene, after the occurrence of a flood[2]. The sensor uses two different polarisations (VV and VH), and the ratio between the two intensities completes the set of 3 channels. To obtain a reasonable ground truth without recourse to manual selection, we considered two other single-polarisation SAR images acquired approximately at the same times as the previous ones. The normalised ratio between these images is depicted in Fig. 3c, and the ground truth obtained by thresholding it at 0.5 can be seen in Fig. 3d. The selection of an appropriate training set is not trivial. There are many different kind of terrain involved, and excluding any of them might lead to poor results. Therefore, the training set is drawn randomly from the parts of the images which are clearly distinguished as unchanged areas, to avoid to inadvertently exclude classes by a manual selection. Again, the size of the set $\mathcal{T}$ is 2% of the total image size.

The optimisation of the hyperparameters is carried out differently for every method. Concerning the GP method, five iterations of the gradient ascent from random starting points are performed. For the SVR method, a grid search on $\Omega_{\boldsymbol{\theta}}$ leads to selection of the best combination of C,ϵ and σ after cross-validation. A grid search for the RF hyperparameters investigates the values $T = \{32, 64, 128, 256, 512\}$ and $p = \{5, 10, 15, 20\}$, while m is kept equal to $\lfloor p/3 \rfloor$. The same procedure is applied for the HPT over the values $K = \{16, 32, 64, 128\}$ and $\gamma = \{10^{-2}, \ldots, 10^{3}\}$. A series of 100 runs is performed, each of these with a different

[1]Distributed by LP DAAC, http://lpdaac.usgs.gov

[2]Data processed by ESA, http://www.copernicus.eu/



(a) Landsat 8 ($t1$) (b) Sentinel-1A ($t2$) (c) Ratio $^{SAR_{t_1}}/_{SAR_{t_2}}$ (d) Ground Truth

Fig. 3: Flood in California: (a) Landsat 8 ($t1$), (b) Sentinel-1A ($t2$), (c) Ratio between SAR intensities at $t1$ and $t2$, (d) ground truth.

training set, to evaluate the mean and standard deviation of the AUC and the training and test times for the four methods. The results are summarised in Table 1.

Table 1: Mean and standard deviation of the AUC and the elapsed time for the four methods applied on the second data set.

	m_{AUC}	σ_{AUC}	m_t	σ_t
GP	0.74692	0.00043	257.11	1.47016
RF	0.81680	0.00541	**132.00**	0.77075
SVR	0.81299	0.05455	2024.58	396.86244
HPT	**0.84001**	0.01450	924.91	8.99086

Similar results can be achieved by RF, SVR, and HPT. On the contrary, the GP method produce worse results. It could be thought that more iterations of the gradient ascent might lead to better solutions, but the consistency of the results throughout the whole series of runs suggests the opposite. Instead, the main drawback of the SVR algorithm is the computing time. Even a coarse grid search over the three hyper-parameters implicates a long training and validation phase. Although it is capable of reaching peaks of 0.89 for the AUC, it is also sensitive to the selection of the hyper-parameters. This brings to a larger σ_{AUC}, and examples of low AUCs. On the other hand, both the HPT and the RF method have their strong suit. The former can reach better results, whereas the latter is computationally faster. In Fig. 4, the elapsed time vs AUC scatter plot for the series of runs supports the previous comments.

Focusing on the RF, the scatter plot in Fig. 5 shows that the computational time grows linearly with respect to the number of trees. However, a larger T does not necessarily provide benefits, as this example demonstrates. Instead, smaller values of m brings on average to better results, at the cost of small additional times. Consequently, it can be recommended to not exaggerate with the pruning. We also recall that, in many formulations of the RF approach, trees are not pruned at all. About the HPT method, no significant trends

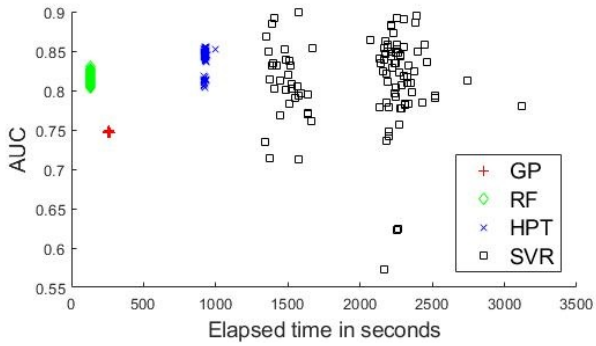


Fig. 4: Elapsed time vs AUC scatter plot for GP (red), SVR (black), RF (green), and HPT (blue)

are recognised in its scatter plot (which is not reported), suggesting that the best option is to select a small number of neighbours K to reduce the amount of computations, and to perform a log-scale search on γ to find the most suitable.

4. CONCLUSIONS

In this paper, we proposed a CD framework based on image regression and evaluated the performance obtained using four different regression methods. The experiments on two data sets proved the effectiveness of the methodology, especially for two of the regression algorithms. Although the HPT method achieved the best results, RF regression proved capable of reaching close results with a shorter computation time. A future work would be to investigate further the role of the hyperparameters, also on other data sets, and to experiment with smaller or more difficult training sets. Another important subject of future research will be the development of an unsupervised version of the methodology.

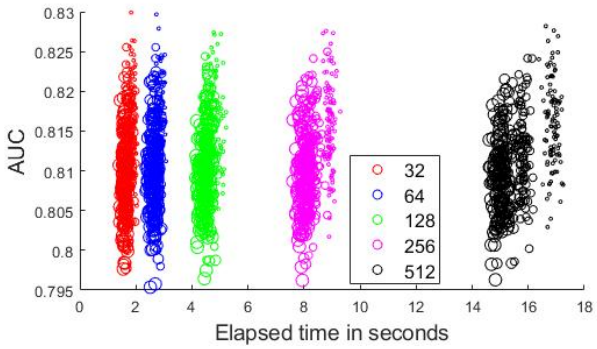


Fig. 5: Elapsed time vs AUC scatter plot for the RF: colors refer to different values of T (see legend), whereas a bigger marker size denotes a larger number of considered features $m \in [5, 10, 15, 20]$.

5. REFERENCES

[1] Wei Zhao, Zhirui Wang, Maoguo Gong, and Jia Liu, "Discriminative feature learning for unsupervised change detection in heterogeneous images based on a coupled neural network," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[2] Puzhao Zhang, Maoguo Gong, Linzhi Su, Jia Liu, and Zhizhou Li, "Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 116, pp. 24–41, 06 2016.

[3] Maoguo Gong, Puzhao Zhang, Linzhi Su, and Jia Liu, "Coupled dictionary learning for change detection from multisource data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7077–7091, 2016.

[4] Diego Marcos, Raffay Hamid, and Devis Tuia, "Geospatial correspondences for multimodal registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5091–5100.

[5] Gang Liu, Julie Delon, Yann Gousseau, and Florence Tupin, "Unsupervised change detection between multi-sensor high resolution satellite images," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 2435–2439.

[6] Zhun-ga Liu, Li Zhang, Gang Li, and You He, "Change detection in heterogeneous remote sensing images based on the fusion of pixel transformation," in *Information Fusion (Fusion), 2017 20th International Conference on*. IEEE, 2017, pp. 1–6.

[7] Luigi Tommaso Luppino, Stian Normann Anfinsen, Gabriele Moser, Robert Jenssen, Filippo Maria Bianchi, Sebastiano Serpico, and Gregoire Mercier, "A clustering approach to heterogeneous change detection," in *Scandinavian Conference on Image Analysis*. Springer, 2017, pp. 181–192.

[8] G. Mercier, G. Moser, and S. B. Serpico, "Conditional copulas for change detection in heterogeneous remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 5, pp. 1428–1441, May 2008.

[9] Jorge Prendes, Marie Chabert, Frédéric Pascal, Alain Giros, and Jean-Yves Tourneret, "A new multivariate statistical model for change detection in images acquired by homogeneous and heterogeneous sensors," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 799–812, 2015.

[10] Zhunga Liu, Gang Li, Gregoire Mercier, You He, and Quan Pan, "Change detection in heterogenous remote sensing images via homogeneous pixel transformation," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1822–1834, 2018.

[11] Devis Tuia, Diego Marcos, and Gustau Camps-Valls, "Multitemporal and multi-source remote sensing image classification by nonlinear relative normalization," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 120, pp. 1–12, 2016.

[12] Bård Storvik, Geir Storvik, and Roger Fjortoft, "On the combination of multisensor data using meta-gaussian distributions," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2372–2379, 2009.

[13] Michele Volpi, Gustau Camps-Valls, and Devis Tuia, "Spectral alignment of multi-temporal cross-sensor images with automated kernel canonical correlation analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 107, pp. 50–63, 2015.

[14] Redha Touati and Max Mignotte, "An energy-based model encoding nonlocal pairwise pixel interactions for multisensor change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1046–1058, 2018.

[15] Linzhi Su, Maoguo Gong, Puzhao Zhang, Mingyang Zhang, Jia Liu, and Hailun Yang, "Deep learning and mapping based ternary change detection for information unbalanced images," *Pattern Recognition*, vol. 66, pp. 213–228, 2017.

[16] Salman H Khan, Xuming He, Fatih Porikli, and Mohammed Bennamoun, "Forest change detection in incomplete satellite images with deep neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5407–5423, 2017.

[17] Haobo Lyu, Hui Lu, and Lichao Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sensing*, vol. 8, no. 6, pp. 506, 2016.

[18] Ross S Lunetta and Christopher D Elvidge, *Remote sensing change detection*, vol. 310, Taylor & Francis, 1999.

[19] Ashbindu Singh, "Review article digital change detection techniques using remotely-sensed data," *International journal of remote sensing*, vol. 10, no. 6, pp. 989–1003, 1989.

[20] J-F Mas, "Monitoring land-cover changes: a comparison of change detection techniques," *International journal of remote sensing*, vol. 20, no. 1, pp. 139–152, 1999.

[21] Carl Edward Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*, pp. 63–71. Springer, 2004.

[22] Devis Tuia, Jochem Verrelst, Luis Alonso, Fernando Pérez-Cruz, and Gustavo Camps-Valls, "Multioutput support vector regression for remote sensing biophysical parameter estimation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 804–808, 2011.

[23] Leo Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] Christopher G Atkeson, Andrew W Moore, and Stefan Schaal, "Locally weighted learning for control," in *Lazy learning*, pp. 75–113. Springer, 1997.