pyndl: Naïve Discriminative Learning in Python

Konstantin Sering[1], **Marc Weitz**[1,2], **Elnaz Shafaei-Bajestan**[1], **and David-Elias Künstle**[1,3]

1 University of Tübingen **2** UiT The Arctic University of Norway **3** International Max Planck Research School for Intelligent Systems

DOI: 10.21105/joss.04515

Software

- Review
- Repository
- Archive

Editor: Øystein Sørensen

Reviewers:

- @frankier
- @jinhangjiang
- @VenkteshV

Submitted: 23 June 2022
Published: 15 December 2022

Summary

The *pyndl* package implements Naïve Discriminative Learning (NDL) in Python. NDL is an incremental learning algorithm grounded in the principles of discrimination learning (Rescorla & Wagner, 1972; Widrow & Hoff, 1960) and motivated by animal and human learning research (e.g. Baayen et al., 2011; Rescorla, 1988). Lately, NDL has become a popular tool in language research to examine large corpora and vocabularies, with 750,000 spoken word tokens (Shafaei-Bajestan et al., 2022) and a vocabulary size of 52,402 word types (Sering et al., 2018). In contrast to previous implementations, *pyndl* allows for a broader range of analysis, including non-English languages, adds further learning rules and provides better maintainability while having the same fast processing speed. As of today, it supports multiple research groups in their work and led to several scientific publications.

Statement of need

Naïve Discriminative Learning (NDL) is a computational modelling framework that phrases language comprehension as a multiple label classification problem (Baayen et al., 2011; Sering et al., 2018). It is grounded in the simple but powerful principles of discrimination learning implemented on a 2-layer symbolic network combined with the Rescorla-Wagner learning rule (Rescorla & Wagner, 1972). This learning rule explains phenomena, where animals associate co-occurring cues and outcomes, e.g. a flashing light and food, only if the cue is effective in predicting the outcome (see Baayen & Ramscar, 2015, for an introduction). In linguistics, NDL models are trained on large corpora to investigate the language's structure and to provide insights into the learning process of life long language acquisition (please find an extensive list of examples in Baayen & Ramscar, 2015, sec. 5).

Several implementations of NDL have been created over time but are struggling with modern challenges of linguistic research like multi-language support, increasing model sizes, and open science principles. The first implementation was the R package *ndl* (Arppe et al., 2018), which could solve the Danks equilibrium equations (Danks, 2003), but did not provide an exact iterative solver. An iterative solver was added to the R package *ndl2* (Shaoul et al., 2014). *ndl* and *ndl2* made efficient implementations to learning algorithms available to language researchers.

However, the code of *ndl2* was only available upon request until end of the year 2022. One reason for this has been that it only runs on Linux and CRAN's guidelines make it difficult to publish single platform packages (R Core Team, 2022). Another reason is the limited maintainability through low level C and C++ code next to high level R code. A severe limitation of *ndl* and *ndl2* is that both packages have difficulties with non-ASCII input, causing problems in the analysis of non-English text corpora due to special characters or non-Latin alphabets. An example would be the processing of Arabic or Slavic languages; even German umlauts are inconvenient to use in *ndl* and *ndl2*. Furthermore, in *ndl2*, it is impossible to conveniently access huge weight matrices due to a size limitation of arrays in the R programming language (R Core Team, 2022). This limit does not allow for more than 46,340 word types in a word-type to word-type model, which is too small to capture the full lexicon in most languages.

Implementation and use in research

pyndl reimplements the learning rule of NDL mainly in Python with small code chunks outsourced to Cython to speed up the processing. This allows the processing of UTF-8 encoded corpora enabling the analysis of many non-European languages and alphabets (e.g. Mandarin or Cyrillic, Milin et al., 2020). Using the Python ecosystem, the size of weight matrices in *pyndl* is only limited by the memory available. Computed weights using the *xarray* format (Hoyer & Hamman, 2017) can be easily integrated into down-stream tasks like analyzing the association strength between grapheme clusters a word types.

The input to *pyndl* is agnostic to the actual domain as long as it is tokenized as Unicode character strings. Input sequences can consist of multiple tokens separated by underscores which is together with the tab-character the only special character in *pyndl*. While *pyndl* provides some basic preprocessing for grapheme tokenization, the preprocessing of ideograms, pictograms, logograms, and speech is possible using custom preprocessing. For example, word classification using tokenized speech audio recordings was investigated in Arnold et al. (2017). Inputs in this work consisted of around 50 tokens per time slice, where each token encoded the pitch, loudness, and variability into a string.

The input format is based on previous implementations of NDL. In contrast to previous implementations, *pyndl* was open-source software from the beginning and developed with usability and maintainability in mind. The better maintainability of *pyndl* does not come at the cost of performance: The benchmark results in Figure 1, described in detail in our package's documentation, shows that *pyndl* is faster than *ndl* and *ndl2*. Memory is used efficiently by storing data records in compressed form and loading data points as they are needed during learning. *pyndl* provides the same core functionality as the previous R packages in Python. After installation, *pyndl* can be called from R or Julia scripts by convenient bridges, like any Python library. An example on how to use *pyndl* from R can be found in our documentation.

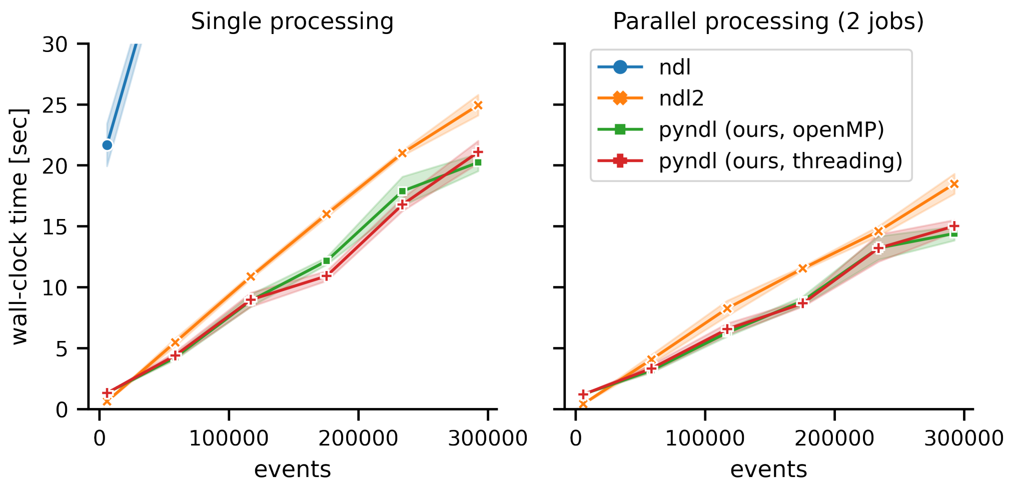

Figure 1: Execution wall time for different implementations of the Rescorla-Wagner learning rule for different number of learning events. The mean and standard-error (n=10) for the wall time show that our implementation, *pyndl*, is the fastest for larger numbers of events.

The improved maintainability of *pyndl* also allows for an easier addition of new features. For example, the NDL learner was extended to a learner for continuous inputs as cues, outcomes or both. When both cues and outcomes are continuous, the Rescorla-Wagner learning rule changes to the Widrow-Hoff learning rule. This extension is added by keeping the API to the learner comparable to NDL and computationally exploiting the structure of the multi-hot encoded features in the symbolic representation of language.

pyndl is used by several research groups to analyse language data and is regularly used in scholarly work. These works use *pyndl* for models in a wide range of linguistic subfields and explicitly use the easy extensibility and UTF-8 support of *pyndl*. Tomaschek et al. (2019) and Baayen & Smolka (2020) investigate morphological effects of the context of German and English languages, while Shafaei-Bajestan & Baayen (2018) and Sering et al. (2018) show auditory comprehension based on simple acoustic features, and Romain et al. (2022) model the learning of tenses. Milin et al. (2020) profited from *pyndl*'s UTF-8 support when using Cyrillic cues to show different language phenomena with the simple Widrow-Hoff learning rule as a special case of the Rescorla-Wagner. Tomaschek & Duran (2019) used *pyndl* to model the McGurk effect across different languages. Shafaei-Bajestan et al. (2021) presented a linearized version of the Rescorla-Wagner rule that they could add to *pyndl* and compare with the classic version. Divjak et al. (2020) showed the benefits of learning language models over probabilistic and rule-based models.

Acknowledgements

The authors thank R. Harald Baayen for his support in creating and maintaining *pyndl* as a scientific software package in the Python ecosystem. Furthermore, the authors thank Lennart Schneider for his major contributions, as well as all other contributors on GitHub.

This research was supported by an ERC advanced Grant (no. 742545) and by the Alexander von Humboldt Professorship awarded to R. Harald Baayen and by the University of Tübingen.

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting David-Elias Künstle, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645 and the High North Population Studies at UiT The Arctic University of Norway for supporting Marc Weitz.

Finally, this paper and the associated package benefited from constructive and conscientious peer review. We would like to thank our three reviewers, Venktesh V, Jinhang Jiang, and especially Frankie Robertson, for their constructive and in-depth feedback and their suggestions on how to make the package more user-friendly.

References

Arnold, D., Tomaschek, F., Sering, K., Lopez, F., & Baayen, R. H. (2017). Words from spontaneous conversational speech can be recognized with human-like accuracy by an error-driven learning algorithm that discriminates between meanings straight from smart acoustic features, bypassing the phoneme as recognition unit. *PLOS ONE*, *12*(4), e0174623. https://doi.org/10.1371/journal.pone.0174623

Arppe, A., Hendrix, P., Milin, P., Baayen, R. H., Sering, T., & Shaoul, C. (2018). *Package "ndl"*.

Baayen, R. H., Milin, P., Đurđević, D. F., Hendrix, P., & Marelli, M. (2011). An amorphous model for morphological processing in visual comprehension based on naive discriminative learning. *Psychological Review*, *118*(3), 438–481. https://doi.org/10.1037/a0023851

Baayen, R. H., & Ramscar, M. (2015). 5. Abstraction, storage and naive discriminative learning. In *Handbook of cognitive linguistics* (pp. 100–120). DE GRUYTER. https:

//doi.org/10.1515/9783110292022-006

Baayen, R. H., & Smolka, E. (2020). Modeling morphological priming in german with naive discriminative learning. *Frontiers in Communication*, *5*. https://doi.org/10.3389/fcomm.2020.00017

Danks, D. (2003). Equilibria of the rescorla–wagner model. *Journal of Mathematical Psychology*, *47*(2), 109–121. https://doi.org/10.1016/s0022-2496(02)00016-0

Divjak, D., Milin, P., Ez-zizi, A., Józefowski, J., & Adam, C. (2020). What is learned from exposure: An error-driven approach to productivity in language. *Language, Cognition and Neuroscience*, *36*(1), 60–83. https://doi.org/10.1080/23273798.2020.1815813

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Milin, P., Madabushi, H. T., Croucher, M., & Divjak, D. (2020). Keeping it simple: Implementation and performance of the proto-principle of adaptation and learning in the language sciences. *arXiv Preprint arXiv:2003.03813*.

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org

Rescorla, R. A. (1988). Pavlovian conditioning: It's not what you think it is. *The American Psychologist*, *43*(3), 151–160. https://doi.org/10.1037/0003-066x.43.3.151

Rescorla, R. A., & Wagner, A. R. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In *Classical conditioning: Current research and theory*.

Romain, L., Ez-zizi, A., Milin, P., & Divjak, D. (2022). What makes the past perfect and the future progressive? Experiential coordinates for a learnable, context-based model of tense and aspect. *Cognitive Linguistics*, *0*(0). https://doi.org/10.1515/cog-2021-0006

Sering, K., Milin, P., & Baayen, R. H. (2018). Language comprehension as a multi-label classification problem. *Statistica Neerlandica*, *72*(3), 339–353. https://doi.org/10.1111/stan.12134

Shafaei-Bajestan, E., & Baayen, R. H. (2018, September). Wide learning for auditory comprehension. *Interspeech 2018*. https://doi.org/10.21437/interspeech.2018-2420

Shafaei-Bajestan, E., Moradipour-Tari, M., Uhrig, P., & Baayen, R. H. (2021). LDL-AURIS: A computational model, grounded in error-driven learning, for the comprehension of single spoken words. *Language, Cognition and Neuroscience*, 1–28. https://doi.org/10.1080/23273798.2021.1954207

Shafaei-Bajestan, E., Uhrig, P., & Baayen, R. H. (2022). *Making sense of spoken plurals*.

Shaoul, C., Schilling, N., Bitschnau, S., Arppe, A., Hendrix, P., & Baayen, R. H. (2014). Ndl2: Naïve discriminative learning. *R Package Version*, *1*. https://github.com/quantling/ndl2

Tomaschek, F., & Duran, D. (2019). *Modelling multi-modal integration–the case of the McGurk effect*. https://doi.org/10.31234/osf.io/prvzq

Tomaschek, F., Plag, I., Ernestus, M., & Baayen, R. H. (2019). Phonetic effects of morphology and context: Modeling the duration of word-final s in english with naïve discriminative learning. *Journal of Linguistics*, *57*(1), 123–161. https://doi.org/10.1017/s0022226719000203

Widrow, B., & Hoff, M. E. (1960). *Adaptive switching circuits*. Stanford Univ Ca Stanford Electronics Labs. https://doi.org/10.21236/ad0241531