Learning binary classification rules for sequential data

Marine Collery[1,2,*], Remy Kusters[2,3]

[1]IBM France Lab, Orsay, France
[2]Inria Saclay Ile-de-France, Palaiseau, France
[3]IBM Research, Orsay, France

Abstract

Discovering patterns for classification of sequential data is of key importance for a variety of fields, ranging from genomics to fraud detection. In this short paper, we propose a differentiable method to discover both local and global patterns for rule-based binary classification. Key to this end-to-end differentiable approach is that the patterns used in the rules are learned alongside the rules themselves.

Sequence classification using rules composed of "classification relevant patterns" is a problem that has received considerable attention in statistical machine learning and data mining due to its applications in e.g. speech processing, fraud detection or genomics. There exists a wide literature that combines (un)supervised pattern mining techniques with sequence classification. Sequential pattern mining has focused to a great extent on mining frequent symbolic subsequences [1]. In feature-based classification, there are two approaches for capturing the sequential nature of features available for classification: either through preprocessing or learned simultaneously with the classification task itself. The present work extends an existing literature that learns classification rules over sequential data [2, 3] with a differentiable approach that builds on top of similar methods for binary tabular data [4, 5]. The novelty of our work lies in using *learned patterns* as atoms in a rule-based classifier for sequential data.

In this paper, we propose a differentiable rule learning classification model for sequential data where the conditions are composed of sequence-dependent patterns that are discovered *alongside* the classification task itself. More precisely, we aim at learning a rule of the following structure: *if* pattern *then class* = 1 *else class* = 0. In particular we consider two types of patterns: local and global patterns as introduced in [6]. A local pattern describes a subsequence at a specific position in the sequence while a global pattern is invariant to the location in the sequence (see Figure 1 for an example).

Model The base rule model (Figure 1) we invoke is composed of two consecutive layers that respectively mimic logical AND and OR operators (inspired by the rule learning modules in [4, 5]. The AND layer takes binary features (which are atomic boolean formulae) as input and its output is used as input by the OR layer. The output of the OR layer is mapped to the classification label y_i. These two layers are defined with binary weights that select the nodes which are included in the respective boolean expression (conjunction or disjunction). In other words, this network implements the evaluation of a DNF. This model has a direct equivalence with a binary classification rule like *if* $(A \wedge B) \vee C$ *then class* = 1 *else class* = 0, where A, B and C are binary input features (atoms in logical terms) [1].

In this paper, we apply the base rule model as a 1D-convolutional window of fixed length over a sequence and retrieve all outputs as input for an additional disjunctive layer which we refer to as the Conv-OR layer. The base rule model learns a boolean expression over the window size length and the Conv-OR layer indicates where along the sequence that logical expression is valid. If the evaluation of the logical expression is valid all along the sequence then it can be described as a *global pattern*, otherwise the learned pattern represents a *local pattern*.

Expressivity With this approach, different sequence-dependent expressions can be extracted and their nature depends on the learned weights of the Conv-OR layer (Figure 1).

- If all the weights of the Conv-OR layer are activated (i.e. equal to 1), the logical expression learned by the base model is valid in all the sequence: a *global* pattern is learned.
- If only some of the weight of the Conv-OR layer are activated, the logical expression learned by the base model is valid only in the window associated to that weight: a *local* pattern is learned. The

STRL'22: First International Workshop on Spatio-Temporal Reasoning and Learning, July 24, 2022, Vienna, Austria
*Corresponding author.
marine.collery@ibm.com (M. Collery)

[1] A natural extension of this architecture for sequential data would be to extend this base rule model with an explicit recursion of the base rule model, similar to a RNN. This approach was tested but faced the same limitations as any classical RNNs, i.e., vanishing gradients and only captures short-term dependencies.

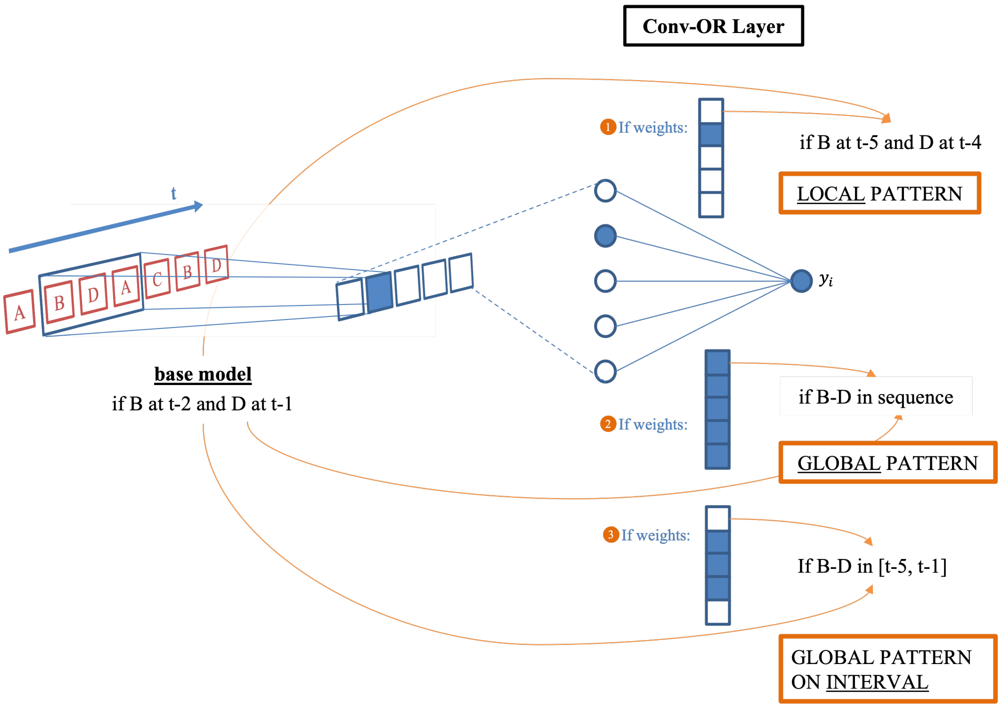

Figure 1: Architecture of the model with an example on a sequence of letters. The base rule model is applied as 1D-convolutional window over the sequence. The resulting boolean values are given as input of the Conv-OR layer which indicates through its activated weights where along the sequence the expression learned by the base model is true. The output of the Conv-OR layer is mapped to the label of the sequence y_i. For local patterns, the base model expression needs to be shifted accordingly to the Conv-OR Layer weights.

base model logical expression is modified accordingly to match that shift (see example in Figure 1 with a shift of 3 sequential steps).

The obtained weights thus translate to rules with the following grammar:

rule → *if* expr *then class* = 1 *else class* = 0
expr → local pattern | global pattern | sequence prop

We introduce t, the position when the last observation in a sequence was made. With t being our reference, in a sequence of size $N \in \mathbb{N}$, $t - i$ refers to the moment of the i^{th} observation before t ($0 \leq i \leq N - 1$). A, B, C and D are toy binary input features, for simplicity of the example those features can not be activated simultaneously at the same position t in the sequence but the method is still valid for complex symbolic sequences [7].

With those definitions, we list below examples of different sequence-dependent atoms that can be expressed with the proposed architecture (See Figure 1):

local pattern is a boolean expression composed of atoms that are TRUE at a specific position, for example `A at t-15`.

global pattern is an expression describing the presence of a pattern anywhere in the sequence, for example `B-D in sequence`, where "$-$" sign refers to "followed by" in global patterns.

global pattern over window `B-*-D in window [t-6; t-3]`

condition on sequence property is a condition on the sequence length for example `4` $\leq$ `length of sequence` $\leq$ `6` (not shown on the figure but it corresponds to a specific case where the base model has learned an empty rule.)

Those expressions could then be used as input to another AND/OR layer model for instance, for extending the rule complexity.

Sparsity Requirements In order to learn those expressions, especially the global ones, the model needs to generalize without observing *all* possible instances at training time. The first requirement for that matter is sparsity in the base model. The approach taken follows a sparsify-during-training method [8] and dynamically enforce sparsity in weights from 0% to 100% [9]. The model with the highest prediction accuracy on validation dataset and the highest sparsity is kept.

Preliminary results Experiments with synthetic toy examples containing ground-truth patterns as those expressed in Figure 1 can be discovered from moderately small samples sizes (See Appendix A for details). Variations and extensions of these pattern have shown promising results and further tests need to be pursued to scope the range of discoverable patterns with this approach.

Limitations There are limitations to this architecture. The main one being that the window size is fixed and that it limits the size of the patterns that can be found. There is a trade-off between the maximum size of patterns and the training complexity that has to be further investigated.

Conclusion To conclude, we presented a 1D-convolutional neural architecture to discover local and global patterns in sequences while learning binary classification rules. This architecture is fully differentiable and requires sparsity that is enforced dynamically. Its main limitation is its dependence to the window size parameter. Further work will consist in integrating this block into more complex architectures to augment the expressivity of the learned rules. Moreover, the algorithm will be tested on concrete datasets such as UCI splice dataset or E. Coli promoter gene sequences dataset [10] to demonstrate its ability to discover rules with *learned* non-trivial patterns.

Acknowledgments

This work has been partially funded by the French government as part of project PSPC AIDA 2019-PSPC-09, in the framework of the “Programme d’Investissement d’Avenir”.

References

[1] J. Han, M. Kamber, J. Pei, 13 - Data Mining Trends and Research Frontiers, in: J. Han, M. Kamber, J. Pei (Eds.), Data Mining (Third Edition), The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Boston, 2012, pp. 585–631. URL: https://www.sciencedirect.com/science/article/pii/B9780123814791000137. doi:10.1016/B978-0-12-381479-1.00013-7.

[2] C. Zhou, B. Cule, B. Goethals, Pattern Based Sequence Classification, IEEE Transactions on Knowledge and Data Engineering 28 (2015) 1–1. doi:10.1109/TKDE.2015.2510010.

[3] E. Egho, D. Gay, R. Trinquart, M. Boullé, N. Voisine, F. Clérot, MiSeRe-Hadoop: A Large-Scale Robust Sequential Classification Rules Mining Framework, in: L. Bellatreche, S. Chakravarthy (Eds.), Big Data Analytics and Knowledge Discovery, volume 10440, Springer International Publishing, Cham, 2017, pp. 105–119. URL: http://link.springer.com/10.1007/978-3-319-64283-3_8. doi:10.1007/978-3-319-64283-3_8, series Title: Lecture Notes in Computer Science.

[4] L. Qiao, W. Wang, B. Lin, Learning Accurate and Interpretable Decision Rule Sets from Neural Networks, arXiv:2103.02826 [cs] (2021). URL: http://arxiv.org/abs/2103.02826, arXiv: 2103.02826.

[5] R. Kusters, Y. Kim, M. Collery, C. d. S. Marie, S. Gupta, Differentiable Rule Induction with Learned Relational Features, arXiv:2201.06515 [cs, stat] (2022). URL: http://arxiv.org/abs/2201.06515, arXiv: 2201.06515.

[6] C. C. Aggarwal, On effective classification of strings with wavelets, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD ’02, Association for Computing Machinery, New York, NY, USA, 2002, pp. 163–172. URL: https://doi.org/10.1145/775047.775071. doi:10.1145/775047.775071.

[7] Z. Xing, J. Pei, E. Keogh, A brief survey on sequence classification, ACM SIGKDD Explorations Newsletter 12 (2010) 40–48. URL: https://dl.acm.org/doi/10.1145/1882471.1882478. doi:10.1145/1882471.1882478.

[8] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, A. Peste, Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks, arXiv:2102.00554 [cs] (2021). URL: http://arxiv.org/abs/2102.00554, arXiv: 2102.00554.

[9] T. Lin, S. U. Stich, L. Barba, D. Dmitriev, M. Jaggi, Dynamic Model Pruning with Feedback, arXiv:2006.07253 [cs, stat] (2020). URL: http://arxiv.org/abs/2006.07253, arXiv: 2006.07253.

[10] D. Dua, C. Graff, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2017. URL: http://archive.ics.uci.edu/ml.

A. Experiments with toy examples

The model is tested on toy synthetic datasets that are generated to fit the rules presented in Figure 3. Datasets are composed of 10000 sequences from which 10% is used for validation and an other 10% for testing; the rest being used for training (batch of 100). We used Adam optimizer with a learning rate set to 0.1. In this setup, we obtain 100% accuracy in 8 out of 10 training of 200 epochs.