Towards Detection of Unknown Polymorphic Patterns Using Prior Knowledge

Przemysław Kucharski[0000−0001−6051−2962],
Krzysztof Ślot[0000−0003−1228−0970]

Lodz University of Technology
Institute of Applied Computer Science
Stefanowskiego 18/22, 90-537 Łódź, Poland
pkuchars@iis.p.lodz.pl

DOI:10.34658/9788366741928.19

Abstract. *The presented paper proposes a novel approach for detecting unknown polymorphic patterns in sequences composed of random symbols and of known polymorphic patterns. We propose to represent rules that drive pattern generation as regular expressions. To detect unknown patterns, we first incorporate knowledge on known rules into a Convolutional Autoencoder (CAE), then we train the CAE with additional objective to prevent weights from learning the already known patterns. Analysis of training results provides statistically significant information on presence or absence of polymorphic patterns that were not previously known.*
Keywords: *polymorphic pattern detection, knowledge and learning integration, Convolutional Autoencoder.*

1. Introduction and Related Work

Recent trends in machine learning indicate an emerging need for methods and models capable of incorporating explicitly formulated knowledge. This is especially important when training data are scarce and shortage of available information needs to be compensated with other means. Motif detection is one of the most challenging tasks in data analysis, due to polymorphic nature of patterns that can encode a given information and scarcity of data, which impairs learning feasibility.

The presented paper is concerned with research on detection of polymorphic patterns in sequences, by utilizing prior knowledge to facilitate the considered, difficult task. Polymorphic sequences considered in the paper are sequences generated by regular expressions with flexible rules allowing high diversity of valid, alternative sequence instances.

A significant amount of research have been done so far on handling the problem of detecting patterns (motifs) in sequences, among which we can name methods that make the patterns recognized by convolutional neural networks more disentangled. Liang et al. [1] propose a training method for classification models that make the convolutional filters class-specific.

Koo et al. [2] examine representation of genomic motifs in CNNs. They searched for motifs in first layer convolutional filters, transforming them into position-weight matrices basing on the response of the filter to specific samples.

Zhang et al. [3] propose a method of updating a specific subfilter cascade chosen dynamically during training to produce more diverse convolutional filters and reduce overlap in representation. More general examination of this problem founds the solutions like structuring the network to resemble the knowledge base, which can be done either manually [4] or generated in the training process [5].

2. The proposed methodology

An objective of the proposed approach is to train a network that analyzes input using a cascade of convolutional filters, where a part of this structure is preset to encode knowledge on known polymorphic sequences (we refer to it as a Fixed Convolutional Module – FCM) and the remaining part is expected to learn any previously unknown regularities that exist in data (we refer to it as the Learnable Convolutional Module – LCM). We adopt a Convolutional Autoencoder (CAE) to be a framework for filter weight training, as it enables monitoring of a pattern learning process.

Convolutional filters can be seen as pattern-detecting operators that produce the maximum output for inputs that match filters' weights. To provide flexible representation of polymorphic patterns of arbitrary length it is reasonable to use a cascade of simple filters, arranged in a conventional convolutional layers. The first layer filters could be designed to capture different short patterns that comply to local rules defined by a given regular expression, whereas the purpose of subsequent layers could be to combine these short chunks into longer strings, that are compatible with the considered expressions. An ease in defining filter cascades that specialize in detecting specific input patterns enables simple incorporation of initial knowledge into a structure of convolutional neural networks.

The second layer filters that are to merge short segments detected by first-layer filters into the longer ones, need to have a depth that enables integration of all relevant first-layer detectors.

The proposed knowledge injection method can be considered universal, albeit in the presented experimental scenario several constraints were introduced in the filters. The method can be scaled both in terms of the number and size of patterns, as well as in terms of rule-complexity, by adding new layers of filters, and be used

in numerous applications, such as bioinformatics or anomaly detection. It should be emphasised that the size of filters does not define the exact length of the pattern, but only constraints its maximum length – as the proposed methodology allows for patterns that contain any character at each position, including the ends. Therefore, shorter patterns can be injected or detected by filling the remaining positions with expression allowing any character.

To search for unknown polymorphic patterns that are embedded in sequences comprising runs of random symbols together with known, possibly polymorphic strings, we initialize our algorithm with knowledge provided in a form of a cascade of appropriately preset filters.

The reconstruction follows the scheme provided in Equation 1, which involves weight normalization, aimed at converting learned weights into probabilities (transformation of R into R"), followed by thresholding that is expected to produce unambiguous basis for reconstruction of the detected regular expression. Here, the symbols U and W denote position-wise minimum of R and position-wise sum of R', respectively.

$$\begin{aligned} R' &= R + 2 * |U| \quad \text{where} \quad U_i = \min_{j \in J} R_{ij},\ i = 1, 2, \ldots, n \\ R'' &= R'/W \quad \text{where} \quad W_i = \sum_{j \in J} R'_{ij},\ i = 1, 2, \ldots, n \\ R''' &= f(R'') \quad \text{where} \quad f(r_{ij}) = \begin{cases} 1 & r_{ij} > th_{upper} \\ -1 & r_{ij} < th_{lower} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

The reconstructed regular expression RE is defined as a tree, build of shorter chunks RS, encoded by the first layer filters:

$$\begin{gathered} \forall_{s_i \in S} \quad s_i \in R_i \\ R \in RE \\ RE = \{RS\} \diamond \{\vee, \wedge\} \end{gathered} \tag{2}$$

where $\diamond$ denotes a recursive tree operator, s_i is a symbol at i^{th} position of a string S that is admissible at this position of the regular expression R (i.e. R_i), which is a specific instance of an expression RE.

We expect that throughout learning, any new polymorphic, unknown patterns present in input sequences, will get learned by learnable weights of the convolutional module. Learning of rules that underlie new polymorphic patterns might be impaired by influence of patterns that are already known to the network. Therefore, we consider an additional learning scenario, where learnable convolutional

filters are discouraged to discover knowledge that has already been injected to the network via FCM filters. To measure similarity of rules that generate patterns, we use a mean Levenshtein distance applied to pairs of sequences produced by the considered regular expressions.

3. Experiments

The proposed polymorphic pattern detection procedure has been trained on a datasets made up of 100 40-element long sequences. This small number of samples is motivated by the scarcity of genomic data representing rare patterns. Human genome data (GRCh38.p14 assembly) was used in the experiment. Short samples with set of 2 or 3 patterns in each were extracted, Labeling was performed with pattern detection using full matching.

Convolutional Autoencoder has been used as a computational framework for polymorphic pattern detection experiments.

Every network was trained for 500 epochs with the use of Mean Square Error as loss function and RMSProp as optimizer with learning rate of 0.001

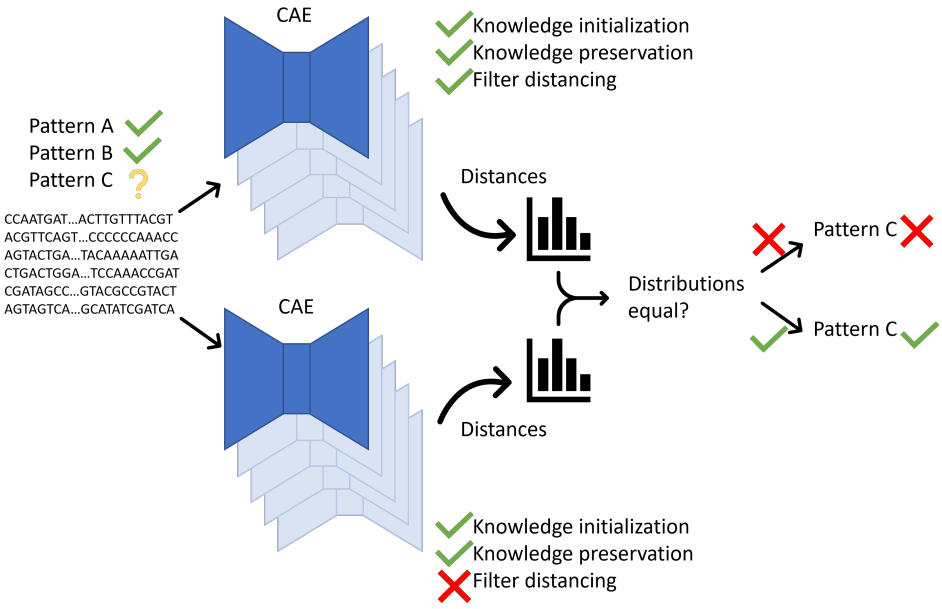

Figure 1. The flowchart of the proposed testing method. The purpose is detection of existence of unknown pattern C. Source: own work.

4. Results

The target characteristics to be quantified at the completion of training was a mean distance between the learned filter cascades and the fixed, knowledge-representing cascades. We were interested, whether there exist statistically significant differences among results produced for four different experimental scenarios (Figure 1):

1. Unknown pattern present and filter similarity discouragement turned on.
2. Unknown pattern not present and filter similarity discouragement turned on.
3. Unknown pattern present, no filter similarity discouragement.
4. Unknown pattern not present, no filter similarity discouragement.

For each of the scenarios, the resulting mean distances between pairs of regular expressions represented by LCM filters and FCM filters were evaluated. The results of Levene test and Fligner tests, for which the null hypothesis is that samples are drawn from the same distribution, show significant outcome when group 2 is tested against group 4, For groups 1 and 3, test results give no basis for rejecting the null hypothesis – in both training regimes, rules that are similarly distant from the preset ones are learned (Table 1).

Table 1. Results of statistical tests.

Unknown regex present in data	Levene test		Fligner test	
	Statistics	p-value	Statistics	p-value
True	1.89	0.09	3.74	0.24
False	3.56	0.02	9.57	0.01

5. Conclusions

The proposed method for unknown polymorphic pattern detection introduces several novel elements. Firstly, we show how prior knowledge on rules, which generate some of the patterns that could be found in input sequences, can be incorporated into a network and preserved during training. Another contribution is concerned with the proposal of measuring a distance between pattern-generating rules by evaluation of Levenshtein distances between sequences generated using these rules. The proposed network architecture is designed in a way that enables injection of complex knowledge. It is also worth noting that the presented problem

is complex and difficult to be solved by traditional approaches. As it can be seen from the results, the proposed data processing pipeline built upon the introduced methodology is capable of answering the question whether new, unknown patterns are present in the data.

References

[1] Liang H., Ouyang Z., Zeng Y., Su H., He Z., Xia S.T., Zhu J., Zhang B., *Training Interpretable Convolutional Neural Networks by Differentiating Class-Specific Filters*, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12347 LNCS, pp. 622–638, doi: 10.1007/978-3-030-58536-5-37.

[2] Koo P.K., Eddy S.R., *Representation learning of genomic sequence motifs with convolutional neural networks*, *PLoS Computational Biology*, 2019, vol. 15, no 12, pp. 1–17, doi: 10.1371/journal.pcbi.1007560.

[3] Zhang D., He L., Luo M., Xu Z., He F., *Weight asynchronous update: Improving the diversity of filters in a deep convolutional network*, *Computational Visual Media*, 2020, vol. 6, no 4, pp. 455–466, doi: 10.1007/s41095-020-0185-5.

[4] Towell G.G., Shavlik J.W., *Knowledge-based artificial neural networks*, *Artificial Intelligence*, 1994, vol. 70, no 1-2, pp. 119–165, ISSN 00043702, doi: 10.1016/0004-3702(94)90105-8.

[5] Gaier A., Ha D., *Weight agnostic neural networks*, [In:] H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (eds.), *Advances in Neural Information Processing Systems*, pp. 5364–5378, doi: 10.48550/arXiv.1906.04358.