proceedings

MDPI

Proceedings

Concept Lattice-Based Classification in NLP †

László Kovács

Department of Information Science, University of Miskolc, 3515 Miskolc, Hungary; kovacs@iit.uni-miskolc.hu; Tel.: +36-20-3319765

† Presented at the 14th International Conference on Interdisciplinarity in Engineering—INTER-ENG 2020, Târgu Mureș, Romania, 8–9 October 2020.

Published: 24 December 2020

Abstract: Classification in discrete object space is a widely used machine learning technique. In this case, we can construct a rule set using attribute level implication rules. In this paper, we apply the technique of formal concept analysis to generate the rule base of the classification. This approach is suitable for cases where the number of possible attribute subsets is limited. For testing of this approach, we investigated the problem of the part of speech prediction in natural language texts. The proposed model provides a better accuracy and execution cost than the baseline back-propagation neural network method.

Keywords: machine learning; morphological classification; formal concept analysis; natural language processing

1. Introduction

In the area of machine learning, classification is the most widely used technology. In the case of classification, we have an object space O, where every object $o \in O$ is given by a set of attribute–value pairs, and the objects are assigned to a category/class value:

$$O = \{(\{(a, v)\}, c) \big| a \in A, v \in Dom_a, c \in C\}, \tag{1}$$

where A denotes the attribute set and C is the category set. The main goal of the classifier is to generate a prediction function:

$$f_C : \{\{(a, v)\}\} \rightarrow C, \tag{2}$$

where f_c assigns a category value for every attribute–value pair set of the problem domain.

A special case of the classification problem is when the Dom_a sets contain only discrete values. Such discrete domains can be found among others in assignment problems, permutation to perform efficient classification in discrete domain space. Among the most widely known candidates, we can mention the decision tree classifier and the neural network classifier.

Neural network (NN) [1] classifiers are based on composition of elementary binary perceptron classifiers using separation hyperplanes. The elementary classification nodes are structured into a network where the nodes are connected by weighted directed edges. In the simplest case, the output of a node is given by:

$$sigmoid\left\{\sum_i w_i{\cdot}s_i + b\right\} \tag{3}$$

where w_i denotes the edge weight values, s_i is the notation for signal strength and b is a bias value. During the training process, the weight values are adjusted to produce a minimal misclassification error. In the case of Back-propagation NN, the input is given by a feature vector and the output is a category vector. The network contains only one hidden layer. Besides this simple classification

network, many other special complex network structures were developed in recent years. For the classification of multi-dimensional objects, the convolutional NN models provide an efficient solution. For the handling of sequences such as words or transaction events, the family of recurrent networks is the suitable tool. The LSTM (long-term short-term memory) NN are very popular in the domain of natural language processing.

In the case of decision trees [2], each child node corresponds to a distinct attribute value. As the child node set is discrete, thus also the attribute value set is assumed to be discrete. In the ID3 decision tree construction methods, the attribute with the smallest weighted entropy is selected:

$$argmin_a\{entropy(\{p_{(i,a)}\})\} \tag{4}$$

where the entropy refers to the homogeneity level of the category (class label) distribution.

The decision process of a decision tree can be represented by a set of logical formulas of the form:

$$If\ a_1 = v_1 \wedge a_2 = v_2 \wedge \ldots \wedge a_m = v_m\ then\ category = c \tag{5}$$

If we consider all formulas related to a decision tree, we can see that attribute a_1 is contained in all formulas, while the last attribute is used only in few formulas. The decision tree constructs a priority order of the attributes.

A more general ruleset can be constructed if we consider not a decision tree, but a decision lattice. The theory of Formal Concept Analysis (FCA) can be used to construct a decision lattice as generalization of the decision tree.

In the next section, the basic concepts of formal concept analysis are presented. This formalism is used to generate the frequent attribute patterns in the training set. Section 3 introduces a morphological classification model based on the concept lattice approach. The results of the experimental tests on efficiency comparison are analyzed in Section 4.

2. Classification with FCA

The theory of Formal Concept Analysis (FCA) [3] provides a tool for conceptualization in an object–attribute relational context. A formal concept corresponds to a pair of related closed sets. The first component containing objects is called the extent part, while the second component containing attributes is the intent part. Formal concepts created from the context can be structured into a concept lattice based on the set containment relationship. The ordering among the lattice elements can be considered as a specialization and generalization relationship among the concepts. The roots of FCA originate in the theory of Galois connections [4] and in the applied lattice and order theory developed later by Birkhoff [5]. The terminology and theoretical foundation of FCA was introduced and built up in the 1980s by Rudolf Wille and Bernhard Ganter [6].

In FCA, two special, partially ordered sets are considered, namely, G is the set of objects and M is the set of attributes. The corresponding fixpoints are called formal concepts, which are built up from a matching pair of object and attribute sets. Using the notations defined in [7], the terminology of FCA can be summarized in the following way:

A formal context is defined as a triplet $\langle G, M, I\rangle$, where I is a binary relation between G and M. The condition $(x, y) \in I$ is met if and only if the attribute y is true for the object x. Two derivation operators are introduced as mappings between the powersets of G and M. For $A \subseteq G, B \subseteq M$:

$$\begin{aligned} f(A) &= A^I = \{m \in M \vee \forall g \in A : (g, m) \in I\} \\ f(A) &= A^I = \{m \in M \vee \forall g \in A : (g, m) \in I\} \end{aligned} \tag{6}$$

For a context $\langle G, M, I\rangle$, a formal concept is defined as a pair (A, B), where $A \subseteq G, B \subseteq M, A = B^I$, $B = A^I$ are met. The composition of these derivations is closure operator. Regarding this derivation,

the components of a formal concept satisfy the $A = A^{II}, B = B^{II}$ conditions, too. The A component is called the extent of the concept, while B is the intent part.

On the set of formal concepts, C, generated from the context $\langle G, M, I \rangle$, a partial ordering relation is defined in the following way:

$$(A_1, B_1) \leqslant (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \tag{7}$$

FCA can be used also as a classification tool in Machine Learning [8,9]. In classification problems, the context is extended with a category attribute and the lattice is used similarly to a decision tree, where the tree leaves correspond to the maximal consistent concepts of the lattice. A pioneer work on the application of concept lattices in retrieval of semantic information is presented in [10]. The area of FCA-based classification is nowadays an active research problem domain [11,12].

Taking an object $(\{(a, v)\}, c)$ from the training set, we can convert the information on the object into the following implication rule:

$$a_1 = v_1 \wedge a_2 = v_2 \wedge \ldots \wedge a_m = v_m \Rightarrow c \tag{8}$$

Thus, the training set can be considered as a ruleset. The objects usually correspond to the atomic nodes in the FCA lattice. Every node in the lattice can be constructed as the intersection of some object-level nodes of the lattice, resulting a node of the following form:

$$a_1 = v_1 \wedge a_2 = v_2 \wedge \ldots \wedge a_m = v_m \Rightarrow c_1 \vee c_2 \vee \ldots \vee c_k \tag{9}$$

where $a_1 = v_1, a_2 = v_2, \ldots, a_m = v_m$ are present in every operand node of the intersection. If the right side of the rule contains only one class label, then this node is called the consistent node.

$$a_1 = v_1 \wedge a_2 = v_2 \wedge \ldots \wedge a_m = v_m \Rightarrow c \tag{10}$$

Among the consistent nodes of the lattice, a special role is assigned to the maximal consistent nodes having a minimal attribute set. A consistent node

$$a_1 = v_1 \wedge a_2 = v_2 \wedge \ldots \wedge a_m = v_m \Rightarrow c \tag{11}$$

is maximal if none of the attribute–value pairs can be removed without breaking the rule.

The set of maximal consistent nodes is considered as the core ruleset for the classification process. The concept lattice structure can be used to generate the elements of the core ruleset. The naive way to generate the lattice is to perform all intersections on the lattice elements. The main bottleneck of this approach is that the size of the lattice and the number of required intersections can be very huge. In the survey paper of [13], the efficiency of two popular methods, CbO and the NextClosure, is investigated. It is shown that the most efficient lattice construction methods have a theoretical complexity of $O\left(N^2MC\right)$ (see Figure 1). Theoretically, the value of C is bound by 2^M, but usually the density of the training set is lower, yielding a smaller lattice. A sharper theoretical upper bound was shown by Prisner in [14] and by Albano Chornomaz in [15]. They investigated a special type of context, the contranomial scale free context involving both theoretical analysis and test experiments. They could show that the execution cost is in general of polynomial shape and in the worst case of exponential shape.

Based on these cost properties, the processing of all possible intersections is implausible in acceptable time. Thus, to apply an FCA-based classification for problem domains with several millions of objects, we need an optimized and simplified method to determine the maximal consistent elements.

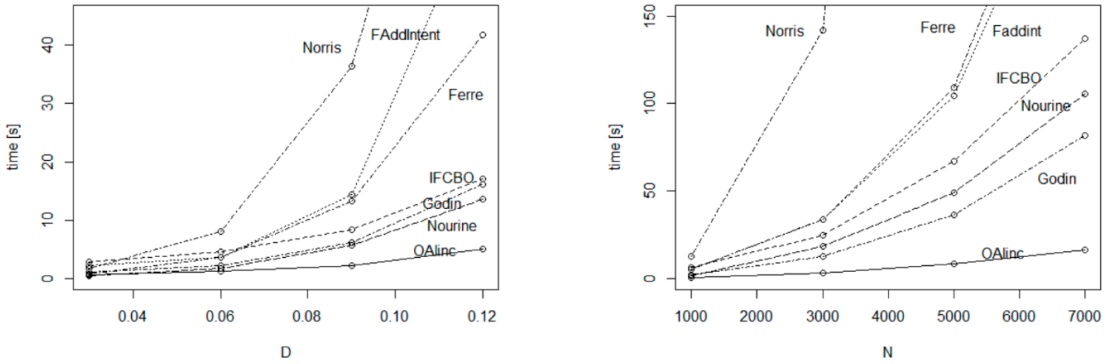

Figure 1. Execution cost function (N = 1000, M = 100) for left side; (M = 150, density level = 7) for right side.

3. Cost Reduction Methods for FCA-Based NLP POS Classifiers

In FCA, the main goal is to construct all possible formal concepts from the input object set. In the prediction task, we do not need all implication rules—one matching rule may be enough to determine the related class label. Thus, in the construction of FCA-based classifiers, we require only a subset of the all possible concept nodes. The relevance of a node can be given with the following three parameters:

- Support level (the size of the extension component, the number of matching objects);
- Heterogeneity level (regarding the class labels);
- Generalization ability (length of the generalization path).

In the classic approach, the engine builds up first the whole concept lattice. The maximal consistent nodes are selected from this lattice using a top-down traversing approach. In the lattice, the top node is usually inconsistent while the atomic nodes are usually consistent. The standard way of lattice construction is to process the atomic nodes first and generate the more general concepts later. This process corresponds to a bottom-up approach.

In the proposed method, the support level is also an important factor, as we need such rules that can cover a larger set of objects. Using rules with large support values results in a reduced ruleset.

We allow also inconsistent nodes in the lattice as in some cases only the atoms are consistent. Thus, the method yields a probability distribution of the different classes.

In the case of part of speech classifiers (NLP POS) [16], the classifier engine determines the grammatical category of the input words. We have analyzed the POS prediction for the Hungarian language, where we use eight basic POS units (noun, verb, adjective, adverb, article, conjunction, number, date) and 42 compound POS units. The composite POS labels are coming from the fact that some word forms have different meaning with different POS labels. For example, the word *vár* can be a verb (*to wait*) or a noun (*castle*).

For the representation of the words, we have used the character sequence format, thus for the word *almákat*, the corresponding letters (*a*, *l*, *m*, *á*, *k*, *a*, *t*) are the description attributes. A special constraint is used in our model, namely, we allow only a subset of the word subsequences as attribute tuples used in the generalization process. For example, for the word *almákat*, only the following strings are valid attribute subsets:

t#, at#, kat#, ákat#, mákat#, lmákat#, almákat#

where symbol # denotes the terminal symbol.

With this kind of simplification, for a word with a length of n characters, the number of possible subsets is equal to $O(n)$, instead of the usual subset count $O(2^n)$. Thus, the number of concept nodes is linear with total number of characters in the word set. This means that we can process all

candidate nodes in a linear time to generate the maximal consistent nodes. Based on this consideration, we constructed the following algorithm to generate all maximal consistent nodes.

```
SDict = dictionary()
For all word (w) in the training set:
    S = generate all substrings of w
    for each s in S:
        h = class_homogeneity (s)
        f = support(s)
    merge (s,h,f) into SDict
```

In the merge process, the cost frequency distribution of s is aggregated to the existing (*h*,*f*) values in the dictionary.

In the purging process, we can eliminate nodes having a weaker relevance value.

```
for all (s,h,f) in SDict:
    w = w(h,f)
    if w < w_limit:
        remote (s,h,f,) from SDict
```

The reduced dictionary can be used as rule space, every node as separate role of the form

$$f_c : \{\{(a, v)\}\} \rightarrow C \tag{12}$$

In the prediction phase, we perform a nearest neighbor classification process. In the first step, the set of subset stings is generated for the query word on the standard way mentioned previously. Then, for each substring s, the entry for s in the dictionary is located. Every entry contains also a class distribution vector; thus, we can calculate an aggregated class distribution for the query word. The winner category is the class with the highest probability.

4. Experimental Results

For the tests, we have used a training set containing 2,200,000 word-entries with POS value. For the baseline method, we have selected the back-propagation neural network and the LSTM neural network engine. In the tests, first we used a character-level representation of the words to compare to baseline neural network classifiers: standard back-propagation network and the LSTM recurrent network. Both NN units were implemented in Python Keras framework. The test results were a bit surprising as both NN engines provided very similar accuracy, near 70% for the case when the size of training set was 250,000. For the test, we have used a disjoint sample of 250,000 items. This experiment has shown that the sequence approach is not superior to the standard classification approach for our problem domain.

Later, we have introduced a different representation approach, where an extended character representation form was used. In this approach, the feature vector also contained the phonetical attributes of the characters. Using this approach, the NN engines could achieve 86% accuracy, while the combined pattern matching, the NN method, could reach a 96% accuracy. The test results are summarized in Table 1. In the table, PM denotes our proposed method and symbol NN is for the neural network approach.

Table 1. Efficiency comparison of the proposed method (PM) with neural network (NN)classifiers.

Method	NN *t*-Time	PM *t*-Time	NN Accuracy	PM Accuracy
90% train, 10% test	130 min	6 min	86%	96%
60% train, 10% test	80 min	4 min	86%	96%
3% train, 10% test	2 min	0.7 min	86%	95%
1% train, 10% test	1.5 min	0.5 min	85%	93%
0.3% train, 10% test	1 min	0.4 min	84%	91%
0.1% train, 10% test	0.6 min	0.3 min	82%	87%

In the tests, we have investigated two factors:

- Accuracy;
- Model construction time.

Based on the performed test experiments, we experienced two surprising results:

- The proposed model provides a significantly better classification accuracy;
- The model's construction in the proposed method is significantly faster than in the NN approach.

5. Conclusions

The FCA concept–lattice-based structure can be used also in classification problems where the maximal consistent nodes are basic rules for the prediction process. The FCA-based approach is suitable for cases where the number of possible attribute subsets is limited. In this paper, we introduced the adaption of the FCA-based classifier to solve an NLP-POS classification problem. Based on the test experiments, the proposed model provides a better accuracy and execution cost than the baseline back-propagation neural network method.

Funding: This research received no external funding.

Acknowledgments: The described article/presentation/study was carried out as part of the EFOP-3.6.1-16-2016-00011 "Younger and Renewing University—Innovative Knowledge City—institutional development of the University of Miskolc aiming at intelligent specialisation" project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Féraud, R.; Clérot, F. A methodology to explain neural network classification. *Neural Netw.* **2002**, *15*, 237–246. [CrossRef]
2. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [CrossRef]
3. Ganter, B.; Wille, R. *Formal Concept Analysis: Mathematical Foundation*; Springer: Heidelberg, Germany; New York, NY, USA, 1999.
4. Oystein, O. Galois Connexions. *Trans. Am. Math. Soc.* **1944**, *55*, 493–513.
5. Birkhoff, G. *Lattice Theory*; American Mathematical Society: Providence, RI, USA, 1940; Volume 25.
6. Wille, R. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Ordered Sets*; Springer: Dordrecht, The Netherlands, 1982; Volume 83.
7. Piskova, L.; Horvath, T. Comparing Performance of Formal Concept Analysis and Closed Frequent Itemset Mining Algorithms on Real Data. In Proceedings of the CLA 2013, La Rochelle, France, 15–18 October 2013; pp. 299–308.
8. Zhao, Y.; Yao, Y. Classification based on logical concept analysis. In Proceedings of the Computational Studies of Intelligence, Quebec, QC, Canada, 7–9 June 2006; pp. 419–430.
9. Makhalova, T.; Kuznetsov, S. On Overfitting of Classifiers Making a Lattice. In Proceedings of the ICFCA 2017, Rennes, France, 13–16 June 2017; pp. 184–197.
10. Carpineto, C.; Romano, G. A Lattice Conceptual Clustering System and Its application to Browsing Re-trieval. *Mach. Learn.* **1996**, *24*, 95–122. [CrossRef]
11. Hao, J.; Zheng, Y.; Xu, C.; Yan, Z.; Li, H. Feature assessment and classification of diabetes employing concept lattice. In Proceedings of the International Conference on Computer Supported Cooperative Work in Design, Porto, Portugal, 6–8 May 2019; pp. 333–338.
12. Telcian, A.S.; Cristea, D.M.; Sima, I. Formal concept analysis for amino acids classification and visualization. *Acta Univ. Sapientiae Inform.* **2020**, *12*, 22–38. [CrossRef]
13. Ignatov, D. Introduction to formal concept analysis and its applications in information retrieval and related fields. In *Russian Summer School in Information Retrieval*; Springer: Cham, Switzerland, 2014; pp. 42–141.
14. Prisner, E. Bicliques in graphs I: Bounds on their number. *Combinatorica* **2000**, *20*, 109–117. [CrossRef]

15. Albano, A.; Chornomaz, B. Why concept lattices are large. In Proceedings of the CLA 2015, Clermont-Ferrand, France, 13–16 October 2015; pp. 73–91.
16. Morton, T.S. Coreference for NLP applications. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, 1–8 October 2000.