Interval Pattern Concept Lattice as a Classifier Ensemble

Yury Kashnitsky, Sergei O. Kuznetsov

National Research University Higher School of Economics, Moscow, Russia
{ykashnitsky,skuznetsov}@hse.ru

Abstract. Decision tree learning is one of the most popular classification techniques. However, by its nature it is a greedy approach to finding a classification hypothesis that optimizes some information-based criterion. It is very fast but may lead to finding suboptimal classification hypotheses. Moreover, in spite of decision trees being easily interpretable, ensembles of trees (random forests and gradient-boosted trees) are not, which is crucial in some domains, like medical diagnostics or bank credit scoring. In case of such "small, but important-data" problems one is not obliged to perform a greedy search for classification hypotheses, and therefore alternatives to decision tree learning techniques may be considered. In this paper, we propose an FCA-based classification technique where each test instance is classified with a set of the best (in terms of some information-based criterion) classification rules. In a set of benchmarking experiments, the proposed strategy is compared with decision tree and nearest neighbor learning.

Keywords: machine learning, classification, decision tree learning, formal concept analysis, pattern structures

1 Introduction

The classification task in machine learning aims to use some historical data (a training set) to predict unknown discrete variables in unknown data (a test set). While there are dozens of popular methods for solving the classification problem, usually there is an accuracy-interpretability trade-off when choosing a method for a particular task. Neural networks, random forests and ensemble techniques (boosting, bagging, stacking etc.) are known to outperform simple methods in difficult tasks. Kaggle competitions also bear testimony for that – usually, winners resort to ensemble techniques, mainly to gradient boosting [13]. The mentioned algorithms are widely spread in those application scenarios where classification performance is the main objective. In Optical Character Recognition, voice recognition, information retrieval and many other tasks typically we are satisfied with a trained model if it has a low generalization error.

However, in lots of applications we need a model to be interpretable as well as accurate. Some classification rules, built from data and examined by experts, may be justified or proved. In medical diagnostics, when making highly responsible

decisions (e.g., predicting whether a patient has cancer), experts prefer to extract readable rules from a machine learning model in order to "understand" it and justify the decision. In credit scoring, for instance, applying ensemble techniques can be very effective, but the model is often obliged to have "sound business logic", that is, to be interpretable [10].

In what follows, we introduce some notions from Formal Concept Analysis (FCA) [5] and provide a technique to express decision tree learning in terms of a search for a hypothesis in a concept lattice (section 3). In section 4, we propose an algorithm which by its design guarantees that each test object is classified with a better (in terms of some criterion such as information gain or Gini impurity) rule than in case of applying a decision tree. Finally, we discuss the results of the experiments with several popular datasets (section 5), make conclusions and directions of further work on developing the performed ideas.

2 Pattern Structures and Projections

Pattern structures are natural extension of Formal Concept Analysis to objects with arbitrary partially-ordered descriptions [4].

Definition 1. *Let G be a set (of objects), let $(D, \sqcap)$ be a meet-semi-lattice (of all possible object descriptions) and let $\delta : G \to D$ be a mapping between objects and descriptions. Set $\delta(G) := \{\delta(g) | g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$, if every subset X of $\delta(G)$ has infimum $\sqcap X$ in $(D, \sqcap)$.* ***Pattern structure*** *is a triple $(G, \underline{D}, \delta)$, where $\underline{D} = (D, \sqcap)$, provided that the set $\delta(G) := \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ [4, 9].*

Definition 2. ***Patterns*** *are elements of D. Patterns are naturally ordered by subsumption relation $\sqsubseteq$: given $c, d \in D$ one has $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$. Operation $\sqcap$ is also called a* ***similarity*** *operation. A pattern structure $(G, \underline{D}, \delta)$ gives rise to the following* ***derivation operators*** *$(\cdot)^\diamond$:*

$$A^\diamond = \bigsqcap_{g \in A} \delta(g) \qquad \text{for } A \in G,$$
$$d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap).$$

Pairs (A, d) satisfying $A \subseteq G$, $d \in \underline{D}$, $A^\diamond = d$, and $A = d^\diamond$ are called ***pattern concepts*** *of $(G, \underline{D}, \delta)$.*

As in classical FCA, pattern concepts form a pattern concept lattice. In case it is too computationally demanding to build the whole lattice, projections are used to simplify object descriptions and boost the formation of a pattern concept lattice.

Definition 3. *A projection [4] of a semilattice $(D, \sqcap)$ is a kernel function $\psi : D \to D$, i.e $\forall x, y \in D$:*

- *$x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$ (monotonicity)*
- *$\psi(x) \sqsubseteq x$ (contractivity)*
- *$\psi(\psi(x)) = \psi(x)$ (idempotence)*

3 Interval Pattern Structure Projections

The theoretical part of the proposed approach is based on Formal Concept Analysis and pattern structures, in particular, on Interval Pattern Structures [6] that provide a way to apply FCA techniques to data with numeric attributes. Unfortunately, the size of the concept lattice is usually too large to be used efficiently in learning [11]. Hence, we introduce a so-called discretizing projection on interval pattern structures which helps to build more general object descriptions based on numeric attributes.

Definition 4. *Let $(G, (D, \sqcap), \delta)$ be an interval pattern structure.*
Let $T_i = \{\tau_{i1}, \ldots, \tau_{it_i}\}, i = 1, \ldots, m$ be m sets of real numbers where m is a cardinality of each $d \in D$. Then, $\psi(\langle [a_i, b_i] \rangle_{i \in [1,m]}) =$
$\langle [max\{\tau \mid \tau \in T_i \cup \{-\infty, +\infty\}, \tau \leq a_i\}, min\{\tau \mid \tau \in T_i \cup \{-\infty, +\infty\}, \tau \geq b_i\}] \rangle$
*is called a **discretizing projection** of a semilattice $(D, \sqcap)$.*

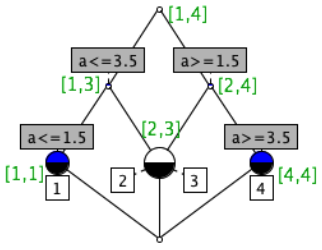
The discretizing projection, as defined in Def. 4, is a projection according to the definition Def. 3.

Example 1. Consider a toy dataset with only 4 objects and 1 numeric attribute as shown in Fig. 1 (left). In order to apply decision tree learning for some classification task with this dataset, one would apply some discretization method to produce binary attributes from attribute a. Consider the discretization shown in Fig. 1 (middle). The corresponding concept lattice is shown in the same figure on the right-hand side.

Fig. 1. A toy many-valued context, it's discretization and the corresponding concept lattice.

	a
1	1
2	2
3	3
4	4

	$a \leq 1.5$	$a \leq 3.5$	$a \geq 1.5$	$a \geq 3.5$
1	×	×		
2		×	×	
3		×	×	
4			×	×



$\psi([a, b]) = [max\{\tau \mid \tau \in T^+, \tau \leq a\}, min\{\tau \mid \tau \in T^+, \tau \geq b\}]$ with $T^+ = \{-\infty, 1.5, 3.5, +\infty\}$ is a projection of the semilattice built for a context that arises from the interordinal scaling of the initial many-valued numerical context. Address to [8] for more details on the link between interordinal scaling

and interval pattern structures. The pattern concept lattice corresponding to the discretizing projection $\psi([a, b])$ is isomorphic to the concept lattice of the discretized context shown in Fig. 1 (middle).

Introducing a discretizing projection is a general way to express any discretizing procedure (essential part of decision tree learning algorithms) in terms of FCA.

4 Learning with Pattern Concept Lattices

For classification tasks with complex data we propose Algorithm 1. The main idea is to find the classification rule for each test instance that maximizes some information criterion (Gini index, pairwise mutual information etc.). In case of interval pattern structures, by its design, the algorithm guarantees to classify each test instance with at least as good rule (in terms of an information criterion) as a decision tree. We apply a modification of the CloseByOne algorithm [7] to build all pattern concepts – the search space for classification rules.

Let $PS_{train} = (G_{train}, ((D, \sqcap), c_{train}), \delta_{train})$ and $PS_{test} = (G_{test}, (D, \sqcap), \delta_{test})$ be two pattern structures corresponding to a train and a test set in a classification task. Let $CbO_{PS}(PS, min\_supp)$ be the algorithm used to find all pattern concepts of a pattern structure PS with support greater or equal to min_supp. Let $inf : D \cup c_{train} \to \mathbb{R}$ be an information criterion used to rate classification rules (we use Gini impurity by default). Finally, let min_supp and n_rules be the parameters of the algorithm (the minimal support of each classification rule's premise and the number of rules to be used for prediction of each test instance's class attribute).

With this designations, the main steps of the proposed algorithm are the following:

1. Initialize a list of predicted labels for test instances c_{test} and a dictionary of classification rules r_{test} for each test instance.
2. Calculate the proportion of positive objects in the training set: $f_{pos} = \frac{|c'_{train}|}{|G_{train}|}$
3. With the CbO_{PS} algorithm, find $\mathcal{S}$ – a dictionary of all pattern concepts (with support greater or equal to min_supp) of a pattern structure PS_{train} Meanwhile, calculate the value of the criterion inf (values in the dictionary $\mathcal{S}$) for each concept intent (keys in the dictionary $\mathcal{S}$).
4. Sort $\mathcal{S}$ by its values.
5. For each test instance $g_t \in G_{test}$:
   - Find first n_{rules} concept intents from $\mathcal{S}$ such that $(A_i, d_i) \in \mathcal{S}, g_t^{\diamond} \sqsubseteq d_i, i = 1, \ldots, n_{rules}$
   - For each "top-ranked" concept intent d_i determine c_i – the proportion of positive objects among $d_i^{\diamond}$: $f_i^+ = \frac{|d_i^{\diamond} \cap c'_{train}|}{|d_i^{\diamond}|}$.
   - Thus, form $\{d_i \to f_i^+\}_{i \in [1, n\_rules]}$ – a set of classification rules for g_t. Set $r_{test}[t]$ be equal to this set of rules.

– Predict the value of the class attribute for g_t as an indicator of the average antecedent of $r_{test}[t]$ being greater or equal to the proportion of positive objects in the training set:

$$c_{test}[i] = [\sum_{i=1}^{n\_rules} f_i^+ \geq f_{pos} * n\_rules]$$

Algorithm 1 Concept Lattice-Based Rule-learner (CoLiBRi)

Input: $PS_{train} = (G_{train}, ((D, \sqcap), c_{train}), \delta_{train})$
$PS_{test} = (G_{test}, (D, \sqcap), \delta_{test})$
$min\_supp \in \mathbb{R}^+$, $n_{rules} \in \mathbb{N}$;
$CbO_{PS}(PS, min\_supp) : PS \to \mathcal{S}$;
$inf : D \times c_{train} \to \mathbb{R}$;
$sort(\mathcal{S}, inf) : \mathcal{S} \to \mathcal{S}$

Output: c_{test}, r_{test}

$c_{test} = \emptyset, r_{test} = \emptyset$
$f_{pos} = \frac{|c'_{train}|}{|G_{train}|}$
$\mathcal{S} = \{(A, d) : inf(d, c_{train}) \mid A \subseteq G_{train}, d \in D, A^\diamond = d, d^\diamond = A, |A| \geq min\_supp\} = CbO_{PS}(PS_{train}, min\_supp)$
$\mathcal{S} = sort(\mathcal{S}, inf)$
for $g_t \in G_{test}$ **do**
  $\{d_i\}_{i \in [1, n_{rules}]} = \{d \mid (A, d) \in \mathcal{S}, g_t^\diamond \sqsubseteq d\}$
  $f_i^+ = \frac{|d_i^\diamond \cap c'_{train}|}{|d_i^\diamond|}$
  $r_{test}[i] = \{d_i \to f_i^+\}_{i \in [1, n_{rules}]}$
  $c_{test}[i] = [\sum_{i=1}^{n\_rules} f_i^+ \geq f_{pos} * n\_rules]$
end for

In case of a classification task with numeric attributes we apply the same Algorithm 1 for interval pattern structures. To make it tractable, we apply it to projections $\psi(PS_{train})$ and $\psi(PS_{test})$ of a training and a test interval pattern structure. Here $\psi(PS)$, is a discretizing pattern structure projection as defined in Def. 4.

5 Experiments

We compare the proposed classification algorithm (denoted as "CoLiBRi" for "Concept Lattice-Based Rule-learner") with Scikit-learn [12] implementations of CART [2], Random Forest [1] and kNN on several datasets from the UCI machine learning repository.[1]

[1] http://repository.seasr.org/Datasets/UCI/csv/

dataset	DT acc	RF acc	kNN acc	CoLiBRi acc	DT F1	RF F1	kNN F1	CoLiBRi F1
audiology	0.75	**0.8**	0.63	0.79*	0.71	**0.74**	0.58	**0.74**
balance-scale	0.63	0.66	**0.76**	0.65	0.58	0.63	**0.75**	0.61
breast_cancer	0.7	0.74	0.73	**0.76**	**0.45**	0.42	0.38	0.44*
car	0.75	0.78*	0.71	**0.79**	0.75	**0.76**	0.71	**0.76**
hayses-roth	0.84*	0.83*	0.49	**0.86**	0.84*	0.82	0.49	**0.85**
lymph	0.8	0.83	**0.86**	0.83	0.77	**0.85**	0.84*	0.84*
mol_bio_prom	0.78	**0.83**	**0.83**	0.82*	0.78	**0.84**	0.8	0.83*
nursery	0.64	0.65	**0.72**	0.65	0.62	0.62	**0.7**	0.62
primary_tumor	0.41	**0.46**	0.41	0.45*	0.37	**0.41**	0.37	0.4*
solar_flare	0.7*	0.7*	0.63	**0.72**	0.67	0.69*	0.6	**0.71**
soybean	0.91*	0.91*	**0.92**	0.91*	0.91*	**0.93**	0.92*	0.91*
spect_train	0.61	**0.69**	0.68*	0.7	0.34	0.36*	0.23	**0.38**
tic-tac-toe	0.79	0.79	**0.85**	0.78	0.82	0.86	**0.89**	0.85

Table 1. Accuracies and F1-scores in classification experiments with the UCI machine learning datasets. "DT acc" and "DT F1" stand for average 5-run 5-fold CV accuracy and F1 score of the CART algorithm, ..., "CoLiBRi F1" stands for average 5-run 5-fold CV F1 score of the proposed CoLiBRi algorithm.

We used Gini impurity as a criterion for rule selection and MDL [3] for continuous feature discretization. CART, Random Forest and kNN parameters ($min\_samples\_leaf \in [1, 10]$ for tree-based algorithms and $k \in \{1, 2, 5, 15, 30, 50\}$ for kNN) were chosen in stratified 5-fold cross-validation. We built 10 trees for each instance of Random Forest classifier.

Parameter min_supp for "CoLiBRi" was taken equal to CART's $min\_samples\_leaf$ for each dataset. We used $n = 10$ classification rules to vote for a test instance label. The described algorithms were implemented in Python 2.7.3 and run on a 4-CPU machine with 4 GB RAM.

The results are presented in Table 1. Each entry stands for the average metric (accuracy or F1-score) in 5 runs of 5-fold cross-validation. In the table, the algorithm with the best performance on each metric is boldfaced. Other algorithm's whose performance is not statistically distinguishable from the best algorithm at p = 0.05 using paired t-tests on the 5 runs are *'ed. The best parameters for each algorithm are mentioned in Table 2.

As it can be seen, the proposed approach performs better than CART and is statistically indistinguishable from RF on most of the datasets. Surprising enough, kNN seems to be the best-performer (on average over all datasets) in terms of accuracy but not F1-score.

Conclusions and further work

In this paper, we have shown how searching for classification hypotheses in a formal concept lattice may yield accurate results while providing interpretable classification rules.

Further we plan to test the proposed strategy in classification tasks such as predicting biological activity (toxicology, mutagenicity, etc.) and telecom client

dataset	DT min_samples_leaf	RF min_samples_leaf	kNN k
audiology	1	1	2
balance-scale	6	1	50
breast cancer	4	3	5
car	3	2	5
hayses-roth	3	1	15
lymph	1	1	5
mol-bio-prom	3	3	5
nursery	3	4	50
primary tumor	4	4	30
solar flare	3	1	30
soybean	1	1	2
spect train	9	5	10
tic-tac-toe	10	3	10

Table 2. Best parameters in classification experiments with the UCI machine learning datasets. CoLiBRi's *min_supp* is taken equal to CART's *min_samples_leaf* for each dataset.

satisfaction where objects have complex descriptions (graphs and sequences correspondingly).

We also plan to introduce some randomization in mining rules for each test instance (as it is done with random forests) in order to further improve the classification quality.

References

[1] L. Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 0885-6125.

[2] L. Breiman et al. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

[3] U. M. Fayyad and K. B. Irani. "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning." In: *IJCAI*. 1993, pp. 1022–1029.

[4] B. Ganter and S. O. Kuznetsov. "Pattern Structures and Their Projections". In: *Conceptual Structures: Broadening the Base*. Ed. by Harry Delugach and Gerd Stumme. Vol. 2120. Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 2001, pp. 129–142.

[5] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. 1st. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[6] M. Kaytoue et al. "Mining gene expression data with pattern structures in formal concept analysis". en. In: *Information Sciences* 181.10 (May 2011), pp. 1989–2001.

[7] S. O. Kuznetsov. "A fast algorithm for computing all intersections of objects from an arbitrary semilattice". In: *Nauchno-Tekhnicheskaya Informatsiya Seriya 2-Informatsionnye Protsessy I Sistemy* 1 (1993), pp. 17–20.

[8] S. O. Kuznetsov. "Fitting Pattern Structures to Knowledge Discovery in Big Data". In: *Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings*. Vol. 7880. Lecture Notes in Computer Science. Springer, 2013, pp. 254–266.

[9] S. O. Kuznetsov. "Scalable Knowledge Discovery in Complex Data with Pattern Structures". In: *PReMI*. Ed. by Pradipta Maji et al. Vol. 8251. Lecture Notes in Computer Science. Springer, 2013, pp. 30–39.

[10] X. Li and Y. Zhong. "An Overview of Personal Credit Scoring: Techniques and Future Work". In: *International Journal of Intelligence Science* 2.4A (2012), pp. 181–189.

[11] A. Masyutin, Y. Kashnitsky, and S. O. Kuznetsov. "Lazy classification with interval pattern structures: Application to credit scoring". In: *Proceedings of the 4th International Workshop "What can FCA do for Artificial Intelligence?", FCA4AI 2015, co-located with the International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 25, 2015*. Vol. 1430. 2015, pp. 43–54.

[12] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[13] G. Tsoumakas et al. "WISE 2014 Challenge: Multi-label Classification of Print Media Articles to Topics". eng. In: *15th International Conference on Web Information Systems Engineering (WISE 2014). Proceedings Part II*. Vol. 8787. Lecture Notes in Computer Science. Springer, Oct. 2014, pp. 541–548.