Summation of Decision Trees

Egor Dudyrev[1][0000−0002−2144−3308]
Sergei O. Kuznetsov[1][0000−0003−3284−9001]

National Research University Higher School of Economics, Moscow, Russia

Abstract. Ensembles of decision trees, like Random Forests are efficient machine learning models with state-of-the-art prediction quality. However, their predictions are much less transparent than those of a single decision tree. In this paper, we describe a prediction model based on a single decision tree in terms of Formal Concept Analysis. We define a differential way to describing a decision rule. We conclude by presenting an approach to summing an ensemble of decision trees into a single decision semilattice with the same predictions.

Keywords: Ensembles of Decision Trees · Formal Concept Analysis · Supervised Machine Learning.

1 Introduction

A decision tree [4] is a popular machine learning model. It can help face the challenge of interpretable machine learning. However, usually it is too simplistic to show good learning performance. Ensembles of decision trees show better learning quality. Some of them – such as random forest [3] and gradient boosting [7] – are considered state-of-the-art. However, ensembles miss the high interpretability of a single decision tree.

Formal Concept Analysis (FCA) [8] is a mathematically well-founded theory aimed at data analysis. In [1], [2], [9], [10], researchers show the connection between decision trees and FCA.

This paper continues our study on the connection between FCA and decision trees started in [6]. In that paper, we have presented the following pipeline. First, we convert a decision tree into a concept lattice. Second, we fuse an ensemble of concept lattices into a single concept lattice. Third, we convert a concept lattice into a decision (semi)lattice: a supervised machine learning model with prediction quality non-inferior to that of ensembles of decision trees.

In what follows, we present a method for constructing a decision semilattice that outputs *the same* predictions as an ensemble of decision trees. We propose a differential way for describing a decision rule and, consequently, a decision tree and a decision semilattice. We finish by summing the ensemble of decision trees into a single decision semilattice.

2 Basic definitions

For standard definitions of FCA and decision trees, we refer the reader to [8] and [4], respectively.

Here we use binary attributes to describe the algorithms. In the experimental section, we extend the algorithm to processing numerical data with interval pattern structures [11].

The standard FCA framework operates with a set M of binary attributes. In what follows we often replace a set of attributes M by a set $M^\star$ that consists both of attributes $m \in M$ and their complements $\overline{m}$ ("not m"):

$$M^\star = M \cup \{\overline{m} \mid \forall m \in M\} \tag{1}$$

3 The proposed approach

3.1 Decision tree and decision semilattice

Definition 1. *A* decision rule (p, t) *is a pair of a subset of attributes* $p \subseteq M^\star$ *called a* premise *and a real number* $t \in \mathbb{R}$ *called a* target. *The attributes in the premise* p *are non-complementary, i.e.* $\forall m \in M^\star$: *if* $m \in p$ *then* $\overline{m} \notin p$.

Given a description $x \subseteq M^\star$, a decision rule can be expressed as "if x contains p: $p \subseteq x$ then predict t".

We order decision rules $(p, t), (\tilde{p}, \tilde{t})$ by the reverse inclusion of their premises:

$$(p, t) < (\tilde{p}, \tilde{t}) \Leftrightarrow p \supset \tilde{p} \tag{2}$$

We cannot apply a single decision rule to any possible description $x \subseteq M^\star$. Therefore, we should use a set of decision rules. A popular means of structuring decision rules in a set is a decision tree DT.

Definition 2. *Decision tree* DT *is an ordered set of decision rules satisfying the following properties: (a) each premise in* DT *is unique, (b)* DT *contains a root decision rule with the empty premise, (c) each non-root decision rule in* DT *has exactly one direct bigger neighbour ("parent"), and one direct smaller neighbour of a parent ("sibling") which differ by one complementary attribute:*

$$a)\ \forall (p, t) \in DT \quad \nexists \tilde{t} \in \mathbb{R}, \tilde{t} \neq t : (p, \tilde{t}) \in DT \tag{3}$$

$$b)\ \exists t \in \mathbb{R} : (\emptyset, t) \in DT \tag{4}$$

$$c)\ \forall (p, t) \in DT, p \neq \emptyset, \quad \exists! (p_{par}, t_{par}), (p_{sib}, t_{sib}) \in DT, m \in p : \tag{5}$$
$$(p_{par}, t_{par}) \succ (p, t), \quad (p_{par}, t_{par}) \succ (p_{sib}, t_{sib}), p_{sib} \neq p$$
$$p_{par} = p \setminus \{m\}, \qquad p_{sib} = p \setminus \{m\} \cup \{\overline{m}\}$$

We propose a more general type of the ordered set of decision rules: a decision semilattice DSL. To define it, we relax the property "c" of a decision tree DT.

Definition 3. *Decision semilattice DSL is an ordered set of decision rules satisfying properties a-b (eq. 3-4) from Definition 2.*

A decision tree DT is a special case of a decision semilattice DSL. Thus, any operation defined for a decision semilattice can also be applied to a decision tree.

We define a "prediction" function $\phi(DSL, x)$ as a function outputting a single target prediction for a description $x \subseteq M^\star$ based on a decision semilattice DSL:

$$\phi(DSL, x) = \frac{1}{|DSL^x_{min}|} \sum_{(p,t) \in DSL^x_{min}} t \tag{6}$$

$$\text{where } DSL^x_{min} = \{(p,t) \in DSL^x \mid \nexists (\tilde{p}, \tilde{t}) \in DSL^x : (\tilde{p}, \tilde{t}) < (p,t)\} \tag{7}$$

$$DSL^x = \{(p,t) \in DSL \mid p \subseteq x\} \tag{8}$$

3.2 Differential decision tree

In this subsection we define a "differential" way for describing a decision rule: (given a prior prediction $\hat{y} \in \mathbb{R}$) "if x contains $p : p \subseteq x$ then add t to the prediction $\hat{y}$".

We define a function $\phi^\Delta(DSL, x)$ which outputs a single target prediction for a description $x \subseteq M^\star$ based on a decision semilattice DSL and differential approach:

$$\phi^\Delta(DSL, x) = \sum_{(p,t) \in DSL^x} t \tag{9}$$

It is unclear how to construct "differential" decision trees and semilattices. We suggest a solution to the former task. To construct a differential decision tree, one can construct a decision tree DT and then "differentiate" it with a function δ:

$$\delta(DT) = \{(p, t - \tilde{t}) \mid (p,t), (\tilde{p}, \tilde{t}) \in DT : (p,t) \prec (\tilde{p}, \tilde{t})\} \cup \{(\emptyset, t) \in DT\} \tag{10}$$

Proposition 1. *For a decision tree DT a prediction $\phi(DT, x)$ matches the prediction $\phi^\Delta(\delta(DT), x)$ for any x.*

Proof. The proof is derived from two facts: (i) a decision tree DT always uses only one decision rule to make a final prediction: $|DT^x_{min}| = 1, \forall x \subseteq M^\star$ (ii) each target of a decision rule in $\delta(DT)$ represents the difference between the target of the corresponding decision rule in DT and the target of its parent.

3.3 Summation of differential decision semilattices

We define an addition operation on decision semilattices in the following way:

$$\begin{aligned} DSL_1 + DSL_2 = &\{(p, t_1 + t_2) \mid \forall (p, t_1) \in DSL_1, t_2 \in \mathbb{R} : (p, t_2) \in DSL_2\} \\ &\cup \{(p, t_1) \in DSL_1 \mid \forall t_2 \in \mathbb{R} : (p, t_2) \notin DSL_2\} \\ &\cup \{(p, t_2) \in DSL_2 \mid \forall t_1 \in \mathbb{R} : (p, t_1) \notin DSL_1\} \end{aligned} \tag{11}$$

The addition operation leads to an important proposition:

Proposition 2. *Given a set of n decision semilattices* $\{DSL_i\}_{i=1}^n$*, the "differential" prediction of the sum of decision semilattices matches the sum of "differential" predictions of the summand decision semilattices :*

$$\phi^{\Delta}(\sum_{i=1}^{n} DSL_i, x) = \sum_{i=1}^{n} \phi^{\Delta}(DSL_i, x), \quad \forall x \subseteq M^{\star} \tag{12}$$

Proof. The proof follows from the definitions of the addition operation (eq. 11) and the function ϕ^{Δ} (eq. 9).

The summation of several identical decision semilattices can be represented as multiplication by a real number:

$$DSL * k = \sum_{i=1}^{k} DSL = \{(p, t * k) \mid (p, t) \in DSL\}, \quad \forall k \in \mathbb{R} \tag{13}$$

3.4 Ensembles of decision trees as decision semilattices

Random forest RF and gradient boosting GB are state-of-the-art ensembles of decision trees. They both operate with a set of decision trees $\{DT_i\}_{i=1}^n$ and, optionally, real-valued hyperparameters. Although the ensembles construct the set of decision trees differently, their prediction functions ϕ^{RF} and ϕ^{GB} are similar as they both sum the predictions of the underlying decision trees:

$$\phi^{RF}(\{DT\}_{i=1}^n, x) = \frac{1}{n} \sum_{i=1}^{n} \phi(DT_i, x) \tag{14}$$

$$\phi^{GB}\left((\{DT\}_{i=1}^n, \alpha, \lambda), x\right) = \alpha + \lambda \sum_{i=1}^{n} \phi(DT_i, x), \quad \alpha, \lambda \in \mathbb{R} \tag{15}$$

Proposition 3. *Given a set of n decision trees* $\{DT_i\}_{i=1}^n$ *and real numbers* $\alpha, \lambda \in \mathbb{R}$*, there is (i) a decision semilattice* DSL_{RF} *such that the prediction* $\phi^{\Delta}(DSL_{RF}, x)$ *matches the prediction* $\phi^{RF}(\{DT_i\}_{i=1}^n, x)$ *for any description* $x \subseteq M^{\star}$*; (ii) a decision semilattice* DSL_{GB} *such that the prediction* $\phi^{\Delta}(DSL_{GB}, x)$ *matches the prediction* $\phi^{GB}(\{DT_i\}_{i=1}^n, x)$ *for any description* $x \subseteq M^{\star}$*:*

$$\textit{1)}\ \forall x \subseteq M^{\star} \quad \phi^{\Delta}(DSL_{RF}, x) = \phi^{RF}(\{DT_i\}_{i=1}^n, x) \tag{16}$$

$$DSL_{RF} = \frac{1}{n} \sum_{i=1}^{n} \delta(DT_i) \tag{17}$$

$$\textit{2)}\ \forall x \subseteq M^{\star} \quad \phi^{\Delta}(DSL_{GB}, x) = \phi^{GB}\big((\{DT_i\}_{i=1}^n, \alpha, \lambda), x\big) \tag{18}$$

$$DSL_{GB} = \{(\emptyset, \alpha)\} + \lambda \sum_{i=1}^{n} \delta(DT_i) \tag{19}$$

Proof. (i) By proposition 1, for any decision tree DT_i, there is a differential decision tree $\delta(DT_i) : \phi(DT_i, x) = \phi^{\Delta}(\delta(DT_i), x)$, $\forall x \subseteq M^{\star}$, (ii) By proposition 2, one can sum a set of differential decision trees into a single differential decision semilattice keeping predictions unchanged.

4 Experiments

This section presents an empirical proof that a decision semilattice can produce the same predictions as ensembles of decision trees. The experiments are run via FCApy[1] python package.

The experimental setup is as follows. First, we construct the "base" models: a decision tree, a random forest, a gradient boosting from sci-kit learn package [12], and a gradient boosting from XGBoost package [5]. Then we convert each decision tree of these models into a unified decision tree format used in FCApy. Finally, we aggregate the unified decision trees of ensemble models into a decision semilattice as defined in equations 17, 19.

We use three real-world datasets for regression to compare the models. They are: Boston Housing Data[2]("Bost."), California Housing dataset[3]("Cal."), Diabetes Data[4]("Diab.").

To construct each decision semilattice in less than a minute (on average), we limit each ensemble model by only ten decision trees with a maximum depth of six. The sole decision tree models are limited by a maximal depth of ten.

Table 1 shows the weighted average percentage error (WAPE) of the decision semilattices copying the predictions of the base models on both train and test parts of a dataset. The error does not exceed 1.9%.

The slight difference in the errors comes from the real-valued nature of the datasets. The premises of decision trees built on such data are of the form either "is $m \leq \theta$" or "is $m > \theta$" where m is a real-valued attribute and $\theta \in \mathbb{R}$. These premises are sensitive to the precision of θ. They also use both closed and open intervals, while our FCA-based implementation operates only the former ones. We replace each premise of the form "is $m > \theta$" by the premise "is $m \geq \theta + 10^{-9}$".

Base model	DecisionTree			GradientBoosting			RandomForest			XGBoost		
Dataset	Bost.	Cal.	Diab.	Bost.	Cal.	Diab.	Bost.	Cal.	Diab.	Bost.	Cal.	Diab.
Train error	0.00	0.00	0.00	0.44	0.00	0.35	0.88	1.75	0.10	0.00	0.00	0.00
Test error	0.02	0.01	0.25	0.63	0.00	0.31	0.84	1.88	0.30	0.22	0.03	0.59

Table 1. WAPE (in %) of the decision semilattices copying the predictions of the base models

5 Conclusion

In this paper, we have introduced a method for summing an ensemble of decision trees into a single decision semilattice model with the same predictions. To do so,

[1] https://github.com/EgorDudyrev/FCApy
[2] https://archive.ics.uci.edu/ml/machine-learning-databases/housing
[3] https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset
[4] https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

we have presented a "differential" way to describe decision rules and a function for differentiating a single decision tree.

In the future work, we plan to extend this approach to decision semilattices. We also plan to study the application of decision semilattice to improving interpretability of ensembles of decision trees.

Acknowledgments

The work of Sergei O. Kuznetsov on the paper was carried out at St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Science and supported by the Russian Science Foundation grant no. 17-11-01276

References

1. Assaghir, Z., Kaytoue, M., Jr., W.M., Villerd, J.: Extracting decision trees from interval pattern concept lattices. In: Napoli, A., Vychodil, V. (eds.) Proc. 8th Int. Conf. Concept Lattices and Their Applications, Nancy, France, October 17-20, 2011. CEUR Workshop Proc., vol. 959, pp. 319–332. CEUR-WS.org (2011)
2. Belohlávek, R., Baets, B.D., Outrata, J., Vychodil, V.: Inducing decision trees via concept lattices. Int. J. of General Systems **38**(4), 455–467 (2009)
3. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (10 2001)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
5. Chen, T., Guestrin, C.: Xgboost. Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (08 2016)
6. Dudyrev, E., Kuznetsov, S.O.: Decision concept lattice vs. decision trees and random forests. In: Braud, A., Buzmakov, A., Hanika, T., Ber, F.L. (eds.) Formal Concept Analysis - 16th Int. Conf., ICFCA 2021, Strasbourg, France, June 29 - July 2, 2021, Proc. LNCS, vol. 12733, pp. 252–260. Springer (2021)
7. Friedman, J.: Greedy function approximation: A gradient boosting machine. Annals of Statistics **29**, 1189–1232 (10 2001)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer Berlin Heidelberg (1999)
9. Krause, T., Lumpe, L., Schmidt, S.E.: A link between pattern structures and random forests. In: Valverde-Albacete, F.J., Trnecka, M. (eds.) Proc. 15th Int. Conf. Concept Lattices and Their Applications, Tallinn, Estonia, June 29-July 1, 2020. CEUR Workshop Proc., vol. 2668, pp. 131–143. CEUR-WS.org (2020)
10. Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Eklund, P.W. (ed.) Concept Lattices, 2nd Int. Conf. on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proc. LNCS, vol. 2961, pp. 287–312. Springer (2004)
11. Kuznetsov, S.O.: Pattern structures for analyzing complex data. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Slezak, D., Zhu, W. (eds.) Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, 12th Int. Conf., RSFDGrC 2009, Delhi, India, December 15-18, 2009. Proc. LNCS, vol. 5908, pp. 33–44. Springer (12 2009)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. J. of machine learning research **12**(Oct), 2825–2830 (2011)