

Continuous Target Variable Prediction with Augmented Interval Pattern Structures: Lazy Algorithm

Alexey Masyutin and Sergei O. Kuznetsov

National Research University Higher School of Economics
Moscow, Russia

`alexey.masyutin@gmail.com, skuznetsov@hse.ru`

Abstract. Pattern structures are known to provide a tool for predictive modeling and classification. However, in order to generate classification rules concept lattice should be built. This procedure may take much time and resources. In previous work it was shown that it is possible to escape the problem with so-called lazy associative classification algorithm. It does not require lattice construction and it is applicable to classification problems such as credit scoring. In this paper we adjust this method to the case of continuous target variable, i.e. regression problem, and apply it to recovery rates forecasting. We perform parameters tuning, assess the accuracy of the algorithm based on the bank data and compare it to the models adopted in the bank system and other benchmarks.

1 Introduction

Banks and financial institutions take and mitigate credit risk on daily basis. Credit risk commonly has the biggest contribution to the bank losses compared to other types of risks such as market, operational and liquidity risks [10]. The key to successful risk-management is to adequately assess the possibility of credit losses and potential amount of the loans that is going to be recovered in case of default. The problem of accurate risk assessment is not only important for an individual bank, but it is also crucial for the banking system as a whole. The problem is so vital that banking industry is strictly regulated by central banks and Basel supervising committee, which even pose certain requirements for predictive models that are used by banks [17]. Some predictive models, so-called "black-box" models provide good results that are hard to interpret. So, the major feature of risk management practice is that, regardless of the model accuracy, it must not be the black box. That is why methods such as neural networks and SVM classifiers did not earn much trust within the banking community [4]. At the same time, the more accurate the model is, the less capital charge the bank is going to have. So, banks prefer accurate models that provide interpretable decision-making. Therefore, FCA-based algorithms seem to be helpful since they rely on concepts that have obvious interpretation. The intent of a concept can be interpreted as a set of rules that is supported by the

extent of the concept. In previous work it was shown that FCA-based interval pattern structures methods are applicable to credit scoring which represents the classification problem with binary target variable [14,16]. Classifying credit applicants into good and potentially delinquent clients is the first part of credit risk assessment. The second part is to estimate *recovery rate* in case of default, i.e. the proportion of the loan that is going to be collected by the bank [10]. As far as recovery rates prediction is concerned, it implies continuous target variable. In this paper, we will adopt the lazy classification algorithm based on interval pattern structures to the case of continuous target variable, i.e. we will introduce modified lazy *regression* algorithm (MLRA). The paper is structured as follows: Section 2 provides basic formal concept analysis definitions. Section 3 describes the architecture of MLRA and its parameters. Section 4 describes the data used for algorithm accuracy evaluation and comparison with benchmarks such as random forests. Section 5 concludes the paper. Finally, we attach a pseudo-code for the algorithm in Appendix.

2 Main Definitions

First, we recall some standard definitions related to FCA, see e.g. [1,2].

Let G be a set (of objects), let (D, \sqcap) be a meet-semi-lattice (of all possible object descriptions) and let $\delta: G \rightarrow D$ be a mapping. Then $(G, \underline{D}, \delta)$, where $\underline{D} = (D, \sqcap)$, is called a *pattern structure* [1], provided that the set $\delta(G) := \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice (D_δ, \sqcap) of (D, \sqcap) , i.e., every subset X of $\delta(G)$ has an infimum $\sqcap X$ in (D, \sqcap) . Elements of D are called *patterns* and 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* $(\cdot)^\circ$:

$$A^\circ = \bigsqcap_{g \in A} \delta(g) \quad \text{for } A \in G,$$

$$d^\circ = \{g \in G \mid d \sqsubseteq \delta(g)\} \quad \text{for } d \in (D, \sqcap).$$

These operators form a Galois connection between the powerset of G and (D, \sqcap) . The pairs (A, d) satisfying $A \subseteq G$, $d \in D$, $A^\circ = d$, and $A = d^\circ$ are called *pattern concepts* of $(G, \underline{D}, \delta)$, with *pattern extent* A and *pattern intent* d . Operator $(\cdot)^\circ$ is an algebraical closure operator on patterns, since it is idempotent, extensive, and monotone [1]. In case of credit scoring we work with pattern structures on intervals as soon as a typical object-attribute data table is not binary, but has many-valued attributes. Instead of binarizing (scaling) data, one can directly work with many-valued attributes by applying interval pattern structure. For two intervals $[a_1, b_1]$ and $[a_2, b_2]$, with $a_1, b_1, a_2, b_2 \in \mathbb{R}$ the *meet operation* is defined as [14]:

$$[a_1, b_1] \sqcap [a_2, b_2] = [\min(a_1, a_2), \max(b_1, b_2)]$$

The concept-based learning model for standard object-attribute representation (i.e., formal contexts) is naturally extended to pattern structures, when we have a binary target attribute, i.e. a set of positive examples G_+ and a set of negative examples G_- [16].

However, what should we do when the target attribute is not a class label but a continuous variable? For that case we augment the definition of interval pattern structure by equipping it with additional feature h .

Augmented interval pattern structures

Let us define an *augmented interval pattern structure* as a quadruple $(G, \underline{D}, \delta, h)$, where the description d consists of two elements d_x and d_y (d_y is an interval for target attribute $y \in \mathbb{R}$ and d_x is a vector of intervals for explanatory attributes x which are supposed to predict the target attribute y), $\delta : G \rightarrow \underline{D}$ and $h \in H$, where H is a family of density distribution functions for target attribute y , i.e. $\int_{-\infty}^{+\infty} h(s)ds = 1$. We will also use notation δ_x and δ_y to distinguish between descriptions containing explanatory attributes and target attribute correspondingly. The meet operation definition is left unchanged.

Suppose, we have an arbitrary set of objects $A_0 \subseteq G$, i.e. $A_0 = \{g_1, g_2, \dots, g_J\}$, $\delta(g_j) = \{\delta_x, \delta_y\} = \{[x_{1j}; x_{1j}], \dots, [x_{Mj}; x_{Mj}], [y_j; y_j]\}$, for $j = 1, \dots, J$, where M is number of explanatory attributes. Then we define the derivation operator in the following way

$$A_0^\diamond = (d_0, h_0)$$

where $d_0 = \{d_{x0}, d_{y0}\}$, and $d_{x0} = \delta_x(g_1) \sqcap \dots \sqcap \delta_x(g_J)$ and target attribute description $d_{y0} = \delta_y(g_1) \sqcap \dots \sqcap \delta_y(g_J)$ which is in fact a single interval $[y_{min}, y_{max}]$ and $h_0 : d_{y0} \rightarrow [0; 1]$. The h_0 is in effect a target attribute density distribution function based on observations of A_0 , which we describe below. Let τ_0, \dots, τ_K be a partition of d_{y0} and $\tau_0 = y_{min}, \tau_K = y_{max}$ and $\Delta\tau_i = \frac{y_{max} - y_{min}}{K} = \tau_i - \tau_{i-1}, i = 1, \dots, K$. Then:

$$h([\tau_{i-1}, \tau_i]) = \frac{|\{g \in A | [\tau_{i-1}, \tau_i] \sqsubseteq \delta_y(g)\}|}{|A|}, \forall i = 1, \dots, K$$

Thus, h is a density function of target attribute y values of objects in A . The derivation operator on descriptions returns the set of objects with description subsuming the description d_{x0} whatever target description d_{y0} and density function h are:

$$A_0^{\diamond\diamond} = (d_0, h_0)^\diamond \stackrel{\text{def}}{=} d_{x0}^\diamond = A_1$$

where $A_1 = \{g \subseteq G | d_{x0} \sqsubseteq \delta_x(g)\}$. Finally, $A_1^\diamond = (d_1, h_1)$. Note, that $d_1 = \{d_{x0}, d_{y1}\}$, i.e. only target attribute description d_y is updated, so does h density function, while the explanatory variables description d_{x0} remains the same.

In order to approach target attribute prediction problem it will be useful to define α -weak premises with allowed dropout. An h -augmented interval pattern $d \in \underline{D}$ is called an α -weak premise with allowed ω -dropout iff:

$$1 - \frac{|\{g \in A | d_y^{min} - \omega(m - d_y^{min}) \leq \delta_y(g) \leq d_y^{max} + \omega(d_y^{max} - m)\}|}{|A|} \leq \alpha$$

where $d = (d_x, d_y)$, d_y is a single interval $[d_y^{min}; d_y^{max}]$ for target attribute y $A = d_x^\diamond$, and m is a median of density function h which reflects the distribution of target attribute within the interval d_y based on objects from A . Parameter α controls the frequency of hypothesis falsifications and parameter ω controls the magnitude of falsification, i.e. how dramatically it is falsificated. In our case the magnitude is evaluated as the times the $\delta_y(g) - d_y^{max}$ is larger than $d_y^{max} - m$ if $\delta_y(g) > d_y^{max}$ or the times the $\delta_y(g) - d_y^{min}$ is larger than $m - d_y^{min}$ if $\delta_y(g) < d_y^{min}$. Note, that in case when $\omega = 0$ we apply the strictest criterion to consider a hypothesis as falsificated:

$$\begin{aligned} 1 - \frac{|\{g \in A | d_y^{min} \leq \delta_y(g) \leq d_y^{max}\}|}{|A|} \leq \alpha &\Leftrightarrow 1 - \frac{|\{g \in A | d_y \sqsubseteq \delta_y(g)\}|}{|A|} \leq \alpha \Leftrightarrow \\ &\Leftrightarrow \frac{|\{g \in A | d_y \not\sqsubseteq \delta_y(g)\}|}{|A|} \leq \alpha \end{aligned}$$

3 Lazy predictive algorithm with continuous target attribute

Assume we have a set of objects G and numerical context with a set of explanatory attributes x_1, \dots, x_M and target attribute y . In contrast to classification problem the context is not divided into positive and negative examples as soon as y take numerical values. Now, suppose we receive a test object g_t with observable attributes x , but with unknown value of target attribute y . Is there a way to predict y using interval pattern structures approach? Indeed, there is, and we are going to describe it below and compare the accuracy results with some benchmarks.

The first stage of algorithm is mining α -weak premises with allowed ω -dropout, the second is to perform prediction for test object g_t based on the mined premises. Let us start by choosing *subsample size* parameter which is the number of objects being randomly extracted from G . Then we specify α and ω parameters that control for "anti-support" in terms of both frequency and magnitude. Upon randomly extracting some objects $A_0 = \{g_1, \dots, g_K\}$ we compute following pattern $d_0 = \delta(g_1) \sqcap \dots \sqcap \delta(g_K) \sqcap \delta(g_t)$ and density distribution function h_0 for target attribute values. If d_0 is an α -weak premise with allowed ω -dropout then it is added to the set of premises that will be used for prediction later. Together with the pattern it is necessary to store the density function h . But which of h_0, h_1 or other we have to use?

Here we introduce another parameter of the algorithm which is called "capped". Capped is a boolean value, and if true then the range for target attribute d_{y1} in d_0^\diamond is truncated to d_{y0} and corresponding density function is h_1 calculated on

the truncated set of target values. If capped parameter is false, then we add d_{y1} and calculate the density function based on all target values that fell into d_{y1} based on objects from d_0° . The whole procedure is repeated many times and the *number of iterations* parameter controls for that.

Having finished with premises mining, we move on to the next stage which is building up a prediction for target attribute based on mined premises. In our case, the resulting prediction was defined by mixture of distributions from all premises. In practice all target attribute values stored within premises were put together to form a final distribution. Finally, we tried both an average and a median of that distribution as the prediction for target attribute. Such approach takes into account different *support* of the premises as soon as premises with greater number of objects will contribute more.

However, one can argue that premises are different in sense of *anti-support* and *deviation* in target attribute values. Indeed, we would put more weight to the prediction based on premises with narrow range of target attribute values and the ones with less falsifying examples from set G . Therefore, we added target values to the final distributions with different weights, thus both weighted average and weighted median were used as forecast.

We introduced two boolean parameters which controlled the weightening schemes. The first parameter is *account for anti-support* and the second is *penalty for high deviation*. When account for anti-support parameter is true, then the target values $\delta_y(g)$ of objects $g \in A$ with the premise d are given weight according to the anti-support of that premise:

$$w_a = \frac{|\{g \in A | d_y^{min} - \omega(m - d_y^{min}) \leq \delta_y(g) \leq d_y^{max} + \omega(d_y^{max} - m)\}|}{|A|}$$

When penalty for high deviation is true, then the weight is decreased with the higher deviation in the target attribute values:

$$w_{pen} = \frac{1}{\sigma(\delta_y(g))}$$

where $\sigma(\delta_y(g))$ is standard deviation of target attribute values. If the parameters values are false then the weights are equal to one. The final weight for the target attribute value of the object g , which will be contributed to aggregate distribution used for prediction, is defined as product of the two weights:

$$w(g) = w_a \cdot w_{pen}$$

Finally, suppose that P is a set of mined α -weak premises with allowed ω -dropout. The prediction for target attribute y of a test object g_t can be based on weighted average:

$$\widehat{\delta_y(g_t)} = \frac{\sum_{p \in P} \sum_{g \in A_p} \delta_y(g) \cdot w(g)}{\sum_{p \in P} \sum_{g \in A_p} w(g)}$$

or on the weighted median:

$$\widehat{\delta}_y(g_t) = \text{median}\left(\bigcup_{g \in \cup_p A_p} \bigcup_{p \in P} \bigcup_{g \in A_p} (\delta_y(g), w(g))\right)$$

In case where P is an empty set, the prediction is average or median of all target attribute values in G , i.e. the prediction is based on "naive" model.

4 Data and experiments

The data we used for the computation represent a pool of delinquent corporate clients loans, which were expected to be restructured. The process of restructuring is started at the early stage when the client shows the first signs of insolvency. At that very moment a bank chooses either to execute *default strategy*, when the court processes are launched and any disposable collateral is displayed for sale, or to execute *restructuring strategy*, when the funding conditions are being revisited usually resulting in a longer credit period. In case of corporate clients banks usually do not want to go to extremes right from the start as soon as court launch and collateral sales imply costs and spending time resources. Also, the bank would prefer to maintain relations with the client if financial distress is temporary. So, the decision whether to launch default strategy or not is based to the greater extent on the recovery expectations. This makes the problem of recovery prediction crucial for banking decision making. Recovery rate is a number between zero and one which reflects the share of the current exposure which the client is going to payback on some time horizon. If recovery rate expectation is at high level, the bank would prefer restructuring and court launch otherwise.

In this paper we use financial data from balance sheets and profit and loss statements of 612 corporate clients of a top-10 Russian bank. Among others factors we used assets-to-liabilities ratio, debt-to-equity ratio, earnings before taxes and interest payments, return on assets etc, resulting. These clients were assessed at the time of early insolvency signals and the resulting recovery rate was collected.

The data was randomly divided into two parts with 70% of observations in one part and 30% in the other. The bigger part was used as a context with known target attribute for the lazy algorithm and 30% was used as a test set to evaluate predictions and their accuracy. The same data partition was used to run random forests with different tunings with 70% part used as a training set and the other as test set. For random forests there were three parameters tuned by grid search which are *minimum nodesize*, *number of trees* and *number of feasible variables*.

The accuracy of predictions were evaluated in terms of mean absolute deviation (MAD):

$$MAD = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{N}$$

where y_i is a target attribute (recovery rate) for i -th client in the test set and \hat{y}_i is prediction.

The random forests were run with following parameters grid: minimum node-size ranging from 30 to 100 with increment 10, number of trees ranging took values 10, 30, 50 and 100, and number of feasible variables from ranging from 5 to 45 with increment of 5.

As far as lazy algorithm is concerned, we tuned seven parameters, four of them were continuous and three were boolean. *Subsample size* took following values: 0.01, 0.02, 0.03, 0.04, 0.05, 0.1. *Number of iterations*: 100, 500, 1000, 2000. *Alpha threshold*: 0, 0.05, 0.01, 0.015, 0.02. *Allowed dropout*: 0, 0.1, 0.5, 1, 1.5.

For each combination of parameters we calculated MAD for the test set and in fact that produced metadata for the analysis. Effectively we obtained MAD distributions, which at the first step helped us to choose in favour of forecast based on weighted median forecast rather than weighted average as soon as MAD distributions for the latter took dramatically higher values which are, of course, undesirable.

When building new algorithm one has some intuition about it mechanism and we performed regression analysis of algorithm accuracy versus parameters values to check that intuition. Also, the analysis was important to determine better parameters tuning and explain variation in accuracy of the predictions. The results of regression are presented below:

Table 1. Regression analysis for dependency between MAD and algorithm parameters

Coefficients	Estimate	Std.Error	t	p-value
(Intercept)	0,3288	0,0006	519,4	0,0000
Subsample size	0,0155	0,0031	4,940	0,0000
Number of iterations	-0,0004	0,0000	-18,05	0,0000
Alpha-threshold	-0,0457	0,0270	-1,695	0,0903
Allowed dropout	-0,0011	0,0004	-2,975	0,0030
Capped	-0,0022	0,0004	-5,401	0,0000
Account for anti-support	0,0002	0,0004	0,624	0,5329
Penalty for high deviation	0,0010	0,0004	2,433	0,0150

We see that increasing number of iterations, allowing dropouts and using capped improve algorithm performance as soon as the coefficients are negative and significant: overall error of prediction decreases as those factors increase. Surprisingly, adjusting account for anti-support and penalty for high deviation parameters do not show significant improvement in accuracy. Also, we expected that there are some non-linear dependencies between MAD and parameter values as soon as, intuitively, there has to be an optimal subsample size of randomly extracted objects. Therefore, we support the regression output with one-factor scatter plots with average MAD across all other iterations versus each parameter:

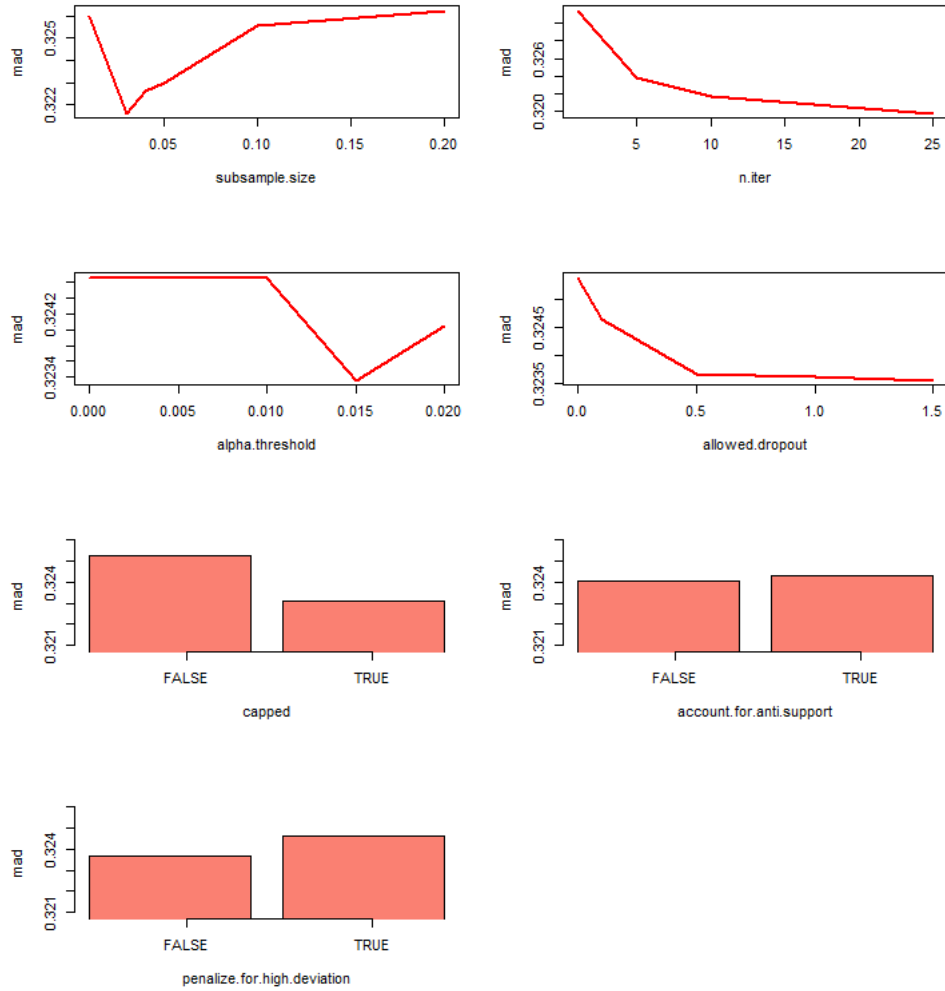


Fig. 1. Single-factor analysis of average MAD versus parameter value: continuous and boolean parameters

As expected, there is a local minimum for the subsample size being extracted from G . It is quite natural because as the subsample size grows, the intersection of the subsample with a test object results in a generic description, which is very likely to be falsified by objects with target attribute value out of the premise description target range.

According to performed grid search the range with the lowest MAD (0.247 - 0.290) on the test sample is achieved in following parameter area: alpha-threshold

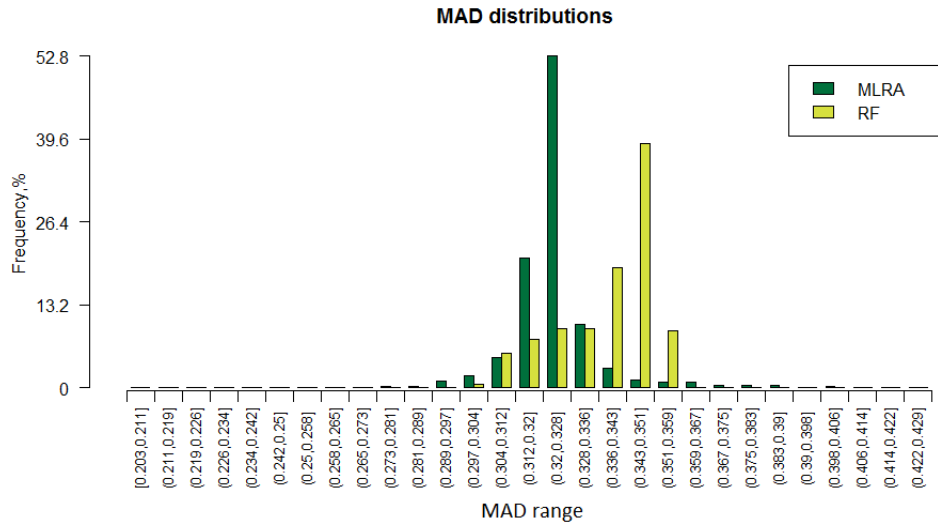


Fig. 2. MAD distribution shows that lazy algorithm allows one to obtain prediction error relatively lower than the one with random forest tunings

= 1.5%, number of iterations = 10, subsample size = 1%, allowed dropout = 0.1. The result was compared to benchmarks represented by random forest tunings.

5 Conclusion

Formal concept analysis offers attractive instruments to extract knowledge from data as soon as intents of concepts can be considered as associative rules. FCA-based algorithms are suitable for predictive modeling in areas where model interpretation clarity is of great priority. However, in previous work only classification problems were considered, while continuous target attribute prediction, i.e. regression problem, was out of focus. In this paper, we adjusted the lazy algorithm [3,16], so that it can perform continuous predictions. The adjustment required a new definition of an augmented interval pattern structure. In effect, the adjusted algorithm mines the premises (with target attribute expected distribution) that are relevant to test object and then prediction is performed based on the target attribute distribution, e.g. based on the median of the distribution.

We applied the algorithm to delinquent corporate clients loans in order to predict the recovery rate for each loan. The data we used comes from the pilot project with one of the top-10 banks in Russia. Mean absolute deviation was chosen as accuracy metric of the algorithm. We performed simple grid search by running the algorithm with different parameter values and chose the tuning with the lowest value of the metric. The classification accuracy of the algorithm was compared to some benchmarks represented by random forests, as soon as their

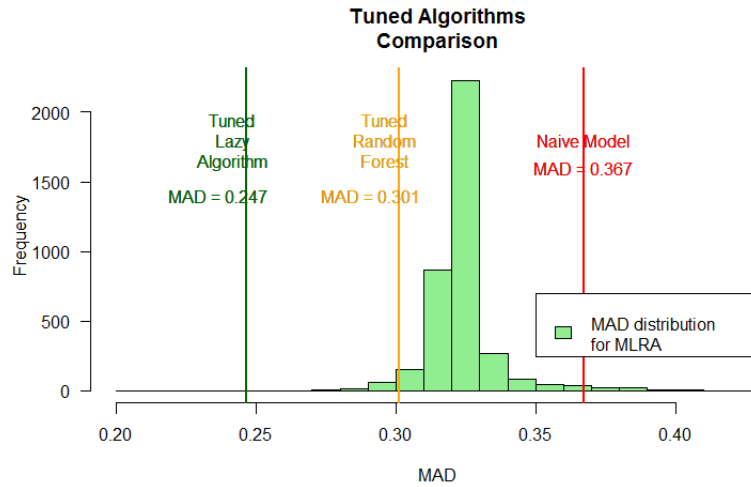


Fig. 3. MAD distribution of the lazy algorithm versus best tuning for random forest and naive model MAD

predictions are based on combination of simple rules, too. The proposed modified lazy regression algorithm showed comparable quality in the greater number of runs and in certain parameters area it outperformed random forests. However, it has to be mentioned that the number of parameters is greater in our algorithm what, in effect, results in greater algorithm complexity and greater degrees of freedom. As an area for further research, one can consider keeping the density function h not only for target attribute in premises, but also make use of those density functions for explanatory attributes as well. It can be expected, that if the premises are mined not only based on allowed dropout and alpha-threshold parameters, but also based on some properties of attributes distribution, then the premises will be more relevant for the test objects and will produce more accurate predictions for target attribute.

References

1. B. Ganter and S.O. Kuznetsov: Pattern structures and their projections, in *Conceptual Structures: Broadening the Base*, Harry Delugach and Gerd Stumme, Eds., vol. 2120 of *Lecture Notes in Computer Science*, pp. 129–142. Springer, Berlin/Heidelberg, 2001.
2. B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin, 1999.
3. S.O. Kuznetsov: Scalable knowledge discovery in complex data with pattern structures, in *PREMI*, Pradipta Maji, Ashish Ghosh, M. Narasimha Murty, Kuntal Ghosh, and Sankar K. Pal, Eds., vol. 8251 of *Lecture Notes in Computer Science*, pp. 30–39, Springer, 2013.

4. L.Thomas, D. Edelman, J.Crook: Credit Scoring and Its Applications, Monographs on Mathematical Modeling and Computation, SIAM: Philadelphia, pp. 107–117, 2002.
5. N. Siddiqi: Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring, WILEY, ISBN: 978-0-471-75451-0, 2005.
6. B. Baesens, T.V. Gestel, S. Viaene, M. Stepanova, J. Suykens: Benchmarking state-of-the-art classification algorithms for credit scoring, Journal of the Operational Research Society 54 (6), 627-635, 2003.
7. A. Ghodselahi: A Hybrid Support Vector Machine Ensemble Model for Credit Scoring, International Journal of Computer Applications (0975 – 8887), Volume 17–No.5, 2011.
8. L. Yu, S.Wang and K.K. Lai: An intelligent agent-based fuzzy group decision making model for financial multicriteria decision support: the case of credit scoring. European journal of operational research. vol. 195. pp.942-959, 2009.
9. T.V. Gestel, B. Baesens, J.A. Suykens, Poel V.D.V., B.: Bayesian kernel based classification for financial distress detection. European journal of operational research. vol. 172. pp. 979-1003, 2006.
10. P. Ravi and V. Ravi: Bankruptcy Prediction in Banks and Firms via Statistical and Intelligent Techniques-A Review, European Journal of Operational Research, Vol. 180, No. 1, pp. 1-28, 2007
11. S.O. Kuznetsov and M.V. Samokhin: Learning closed sets of labeled graphs for chemical applications., in ILP, Stefan Kramer and Bernhard Pfahringer, Eds., vol. 3625 of Lecture Notes in Computer Science, pp. 190– 208, Springer, 2005
12. SAS Institute Inc., Developing Credit Scorecards Using Credit Scoring for SAS® Enterprise Miner™ 12.1, Cary, NC: SAS Institute Inc., 2012.
13. R.R. Hocking: The Analysis and Selection of Variables in Linear Regression, Biometrics, 32, 1976.
14. M. Kaytoue, S.O. Kuznetsov, A. Napoli, and S. Duplessis: Mining gene expression data with pattern structures in formal concept analysis, Information Sciences, vol. 181, no. 10, pp. 1989–2001, 2011.
15. A. Veloso and W. Meira, Jr.: Demand-Driven Associative Classification., Springer, 2011.
16. A. Masyutin, Y. Kashnitsky, S.O. Kuznetsov: Lazy Classification with Interval Pattern Structures: Application to Credit Scoring, In: Proc. 4th International Workshop "What can FCA do for Artificial Intelligence?" (FCA4AI 2015), pp. 43-54, 2015.
17. E.I. Altman, A. Resti and A. Sironi: The link between default and recovery rates: effects on the procyclicality of regulatory capital ratios", BIS Working Papers No 113, Monetary and Economic Department, 2002.

Appendix

Algorithm 1 Lazy Regression by Sub-Samples with Continuous Target Attribute

Input: $\{G\}$ – numerical context with explanatory variables x and single target attribute y .

M – number of explanatory attributes.

$sub.smpl$ – percentage of the context randomly used for intersection with the test object (parameter).

$num.iter$ – number of iterations (resamplings) during the premise mining (parameter).

$alpha.threshold$ is the maximum allowable percentage of the context G which represents the objects which falsify the premise (parameter).

g_t – test object.

Output: $\delta_y(\widehat{g_t})$ – prediction that is produced by the voting rule.

P – a set of premises, i.e. associative rules produced for the test object g_t . P can be empty.

for $iter$ from 1 to $num.iter$ **do**

$A_0 = \text{random.sample}(G, \text{size} = sub.smpl \cdot |G|)$ — mine α - weak premises with ω -allowed dropout.

$d_0 = \delta_x(g_1) \sqcap \dots \sqcap \delta_x(g_s) \sqcap \delta_x(g_t)$, $g_s \in A_0 \forall s$

Compute empirical density function h_0 for d_0 .

$A_1 = d_0^c$

if $1 - \frac{|\{g \in A_1 \mid d_0 y^{min} - \omega(m - d_0 y^{min}) \leq \delta_y(g) \leq d_0 y^{max} + \omega(d_0 y^{max} - m)\}|}{|A_1|} \leq \alpha$ **then**

Update empirical density function h_0 to h_1 based on new values of target attribute in A_1 .

Add (d_0, h_1) to the set P of α - weak premises with ω -allowed dropout.

else

Do nothing

end if

end for

Define weighting scheme w_a, w_{pen} .

Calculate the median for mixture of distribution functions h_p based on $d_{py}, \forall p \in P$.

$$\delta_y(\widehat{g_t}) = \frac{\sum_{p \in P} \sum_{g \in A_p} \delta_y(g) \cdot w(g)}{\sum_{p \in P} \sum_{g \in A_p} w(g)}$$

If P is empty, then calculate the median for target attributes of all $g \in G$ (naive prediction).
