Training Set Selection for Monotonic Ordinal Classification

José-Ramón Cano
Dept. of Computer Science
University of Jaén
Linares, Spain
jrcano@ujaen.es

Salvador García
Dept. of Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
salvagl@decsai.ugr.es

***Abstract*—This is a summary of our article published in Data & Knowledge Engineering [1] to be part of the MultiConference CAEPIA'18 KeyWorks.**

***Index Terms*—Monotonic Classification, Ordinal Classification, Training Set Selection, Data Preprocessing, Machine Learning**

I. Summary

Learning with ordinal data sets has increased the attention of the machine learning community in recent years. These data sets are characterized by the presence of an ordinal output and they are commonly found in real life.

Monotonic classification is an ordinal classification problem where monotonic constraints are present in the sense that a higher value of a feature in an instance, fixing the other values, should not decrease its class assignment [2]. Monotonicity is a property commonly found in many environments of our lives like economics, natural language or game theory [4]. A classical example of monotonicity is in the case of bankruptcy prediction in companies, where appropriate actions can be taken in time, considering the information based on financial indicators taken from their annual reports. The comparison of two companies where one dominates the other on all financial indicators shows clearly where the monotonicity is present, which supposes that the overall evaluation of the second cannot be higher than the evaluation of the first. This strategy could be applied to the credit rating score used by banks as well as for the bankruptcy prediction strategy .

In the specialized literature we can find multiple monotonic classifiers proposed. As a restriction, some of them require the training set to be purely monotone to work properly. Other classifiers can handle non-monotonic data sets, but they do not guarantee monotone predictions.

In addition, real-life data sets are likely to have noise, which obscures the relationship between features and the class. This fact affects the prediction capabilities of the learning algorithms which learn models from those data sets.

This work was supported by TIN2014-57251-P, by the Spanish "Ministerio de Economía y Competitividad" and by "Fondo Europeo de Desarrollo Regional" (FEDER) under Project TEC2015-69496-R and the Foundation BBVA project 75/2016 BigDaPTOOLS.

In order to address these shortcomings and to test the prediction competences of the monotonic classifiers, the usual trend is to generate data sets which completely satisfy the monotonicity conditions. The intuitive idea behind this is that the models trained on monotonic data sets should offer better predictive performance than the models trained on the original data. In the specialized literature, we find two possible techniques to generate monotonic data sets. Monotonic data sets can be created by generating artificial data [5] and by relabeling the real data [6]. The latter restores the monotonicity of the data set by changing the class labels in those instances which violate the monotonicity constraints. Class relabeling is the only approach which can be applied in real life data sets, and has shown promising results in the literature.

As an alternative to relabel, Training Set Selection (TSS) is known as an application of instance selection methods [3] over the training set used to build any predictive model. The effects produced by TSS are: reduction in space complexity, decrease in computational cost and the selection of the most representative instances by discarding noisy ones.

In this paper we propose a TSS algorithm to manage monotonic classification problems, called Monotonic Training Set Selection (MonTSS). MonTSS can be considered as the first in the literature for performing TSS in monotonic classification problems. It is a data preprocessing technique which, by means of a suitable TSS process for monotonic domains, offers an alternative without modifying the class labels of the data set, it instead removes harmful instances. MonTSS incorporates proper measurements to identify and select the most suitable instances in the training set to enhance both the accuracy and the monotonic nature of the models produced by different classifiers.

The whole process is presented in Figure 1, and as can be seen it is composed of three stages:

1) The MonTSS process starts with a preprocessing step where MonTSS analyzes the original data set by quantifying the relationship between each input feature and the output class. This relation is estimated with a metric called Rank Mutual Information (RMI). With it, we

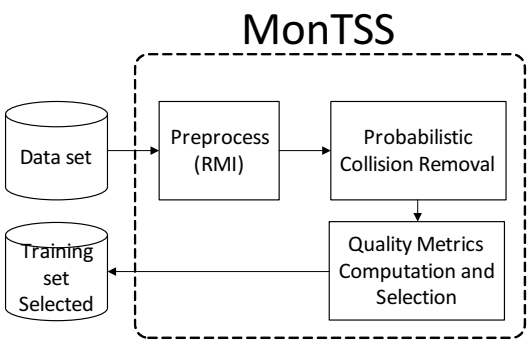


Fig. 1. MonTSS process.

know the features which have a real direct or inverse monotonic relation with the class or no relation as well (including unordered categorical features). The RMI value is evaluated in the training data set to decide which features are used in the computation of collisions between instances.

In essence, rank mutual information can be considered as the degree of monotonicity between features A_1,...,A_f and the feature class Y. Given any feature A_j and feature class Y, the value of RMI for the feature A_j is calculated as follows:

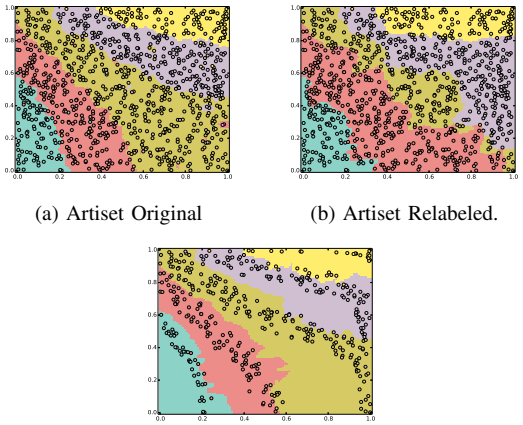
$$\text{RMI}(A_j, Y) = -\frac{1}{n}\sum_{i=1}^{n} log \frac{\#[x_i]^{\leq}_{A_j} \cdot \#[x_i]^{\leq}_{Y}}{n \cdot \#([x_i]^{\leq}_{A_j} \cap [x_i]^{\leq}_{Y})} \quad (1)$$

where n is the number of instances in data set D, $[x_i]^{\leq}_{A_j}$ is the set formed by all the instances of the set D whose feature A_j is less or equal than feature A_j of instance x_i, and $[x_i]^{\leq}_{Y}$ is the set composed of the instances of the set D whose feature class Y is less or equal than feature class Y of instance x_i.

2) In the second stage, the probabilistic collision removal mechanism is applied, which eliminates most of the instances which produce collisions. The remaining instances are used as input in the last stage.
3) Here, the quality metrics are computed and based on them, the selection procedure is developed considering the following rule:

$$\text{Select } x_i = \begin{cases} \text{true} & \textbf{if } \text{Del}(x_i) < \text{Infl}(x_i) \\ & \text{or } \text{Del}(x_i) \geq 0.9 \\ \text{false} & \text{otherwise.} \end{cases} \quad (2)$$

The rationale behind this rule is to retain the instances which are closer to the class boundaries, using a straightforward threshold of 0.9 which is independent from the diversity of their neighborhood. Furthermore, a relationship between $\text{Del}(x_i)$ and $\text{Infl}(x_i)$ can be easily established as they represent a measurement in the same range of the relative rate of the situation and the neighborhood variety of every instance. In this respect, the rule is built as a function of both measures. As a result, for instance, the rule preserves the instances belonging to central areas if there are instances of other classes around.



(a) Artiset Original (b) Artiset Relabeled.

(c) Artiset MonTSS.

Fig. 2. Artificial data set (Artiset) preprocessed by Relabeling and MonTSS with the borders calculated by MkNN with 3 neighbors.

We have compared the results offered by well-known classical monotonic classifiers over 30 data sets with and without the use of MonTSS as a data preprocessing stage. As graphical example of use we present the Fig. 2.

The results show that MonTSS is able to select the most representative instances, which leads monotonic classifiers to always offer equal or better results than without preprocessing.

MonTSS is able to select the most representative instances independently of the classifier to be applied later. This leads monotonic classifiers to always offer equal or better results than without preprocessing. Furthermore, data related metrics are notably improved, fully satisfying the monotonicity restrictions without affecting or modifying the nature of the original data. At the same time, it reduces the number of non-comparable pairs of instances and the size of the training data sets before the learning stage starts.

References

[1] J.-R. Cano and S. García, "Training set selection for monotonic ordinal classification," Data & Knowledge Engineering, vol. 112, pp. 94–105, 2017.
[2] H. Daniels and M. Velikova,"Monotone and partially monotone neural networks," IEEE Transactions on Neural Networks, vol. 21, num. 6, pp. 906–917, 2010.
[3] J.-R. Cano, N.R. Aljohani, R.A. Abbasi, J.S. Alowidbi and S. García, "Prototype selection to improve monotonic nearest neighbor," Engineering Applications of Artificial Intelligence, vol. 60, pp. 128–135, 2017.
[4] W. Kotłowski and R. Słowiński, "On nonparametric ordinal classification with monotonicity constraints," IEEE Transactions on Knowledge and Data Engineering, vol. 25, num. 11, pp. 2576–2589, 2013.
[5] R. Potharst, A. Ben-David and M.C. van Wezel, "Two algorithms for generating structured and unstructured monotone ordinal data sets," Engineering Applications of Artificial Intelligence, vol. 22, num. 4-5, pp- 491–496, 2009.
[6] M. Rademaker, B. De Baets and H. De Meyer, "Optimal monotone relabelling of partially non-monotone ordinal data," Optimization Methods and Software, vol. 27, num. 1, pp. 17–31, 2012.