Instance Selection for Regression: adapting DROP

Álvar Arnaiz-González, José F. Diez-Pastor, César García-Osorio, and Juan J. Rodríguez

Universidad de Burgos, España
{alvarag,jfdpastor,cgosorio,jjrodriguez}@ubu.es

Resumen This is a summary of our article published in Neurocomputing [1], presented to the Multi-Conference CAEPIA'16 KeyWorks.

Keywords: Machine Learning, regression, instance selection, DROP, noise filtering

1 Summary

Nowadays, a major challenge of Machine Learning algorithms is their application to increasingly massive datasets, such as insurance company data, banking transactions, telecommunications companies, financial markets and digital image processing... and more recently those needed in fields such as Bioinformatics. One way to facilitate the learning process when applied to these huge datasets is the reduction of training dataset size by applying some instance selection techniques. Instance selection methods try to find a subset of instances that, if it is used as a set to train a regressor, its predictive capability will be similar to the original [2]. The reduction of the size of the training set accelerates the learning process, or the testing process in the case of instance-based learning (IBL) [3].

The family of methods known as DROPn (DROP1...5) [5] contains some of the methods that yield the best results in classification [4]. They belong to the category of decremental methods and the removal criteria is defined in terms of nearest neighbours and associates (the inverse relation of nearest neighbours). Using these relations, two variables are computed: '`with`' and '`without`'. For each instance, the variable `with` counts how many of its associates are correctly classified when the instance is kept in the dataset, while the variable `without` records the count of correctly classified associates when the instance is removed from the dataset. If the value of `without` is greater than or equal to the value of `with` the instance is not helping in the classification of its associates, and hence, it can be removed from the dataset.

In classification, the calculation of `with` and `without` is straightforward. To cope with regression tasks, i.e. numeric class, we proposed two ideas: to compare the accumulated error that occurs when an instance is included and when it is discarded, and to use in regression an approximation to the concept of class used in classification (something like a *soft* class):

© Ediciones Universidad de Salamanca

– Using error accumulation: variables accumulate the error produced by the regressor on each of the instances, and therefore, they have real values.
– Using thresholding: variables are counters that increase their values depending on whether the difference of the regressor prediction and the actual value exceeds a threshold θ_D.

Both ideas were used to adapt DROP2 and DROP3 to regression and the four resultant algorithms were subjected to benchmark testing, where they were compared against each other, against RegCNN, one of the few methods for this kind of Machine Learning task, and against the result obtained from a regressor trained with the original dataset. The regressors used were: a method based on instances (kNN), a multilayer perceptron, and a method for constructing regression trees (REPTree).

To sum up, the best algorithms are DROPx-RE in most cases if accuracy is to be maximised, as these methods are designed to minimize the loss of predictive capability of the resulting dataset. On the one hand, if the aim is to minimize the size of the datasets after selection, the best method are DROPx-RT both in the experimentation with the original and noisy datasets.

A remarkable property of the proposed adaptations of DROP for regression is their robustness in the presence of noise. The experiments carried out with noise levels of 10 %, 20 % and 30 % have shown that the proposed instance selection algorithms are not only able to reduce the size of the training dataset, but they are also able to reduce the noise and significantly improve the accuracy achieved by different regressors.

Acknowledgments

This work was partially supported by the Spanish Ministry of Economy and Competitiveness through project TIN2011-24046.

Referencias

1. Álvar Arnaiz-González, José F. Díez-Pastor, Juan J. Rodríguez, and César Ignacio García-Osorio. Instance selection for regression: adapting DROP. *Neurocomputing*, pages–, 2016.
2. E. Leyva, A. Gonzalez, and R. Perez. A set of complexity measures designed for applying meta-learning to instance selection. *Knowledge and Data Engineering, IEEE Transactions on*, 27(2):354–367, Feb 2015.
3. Elena Marchiori. Hit miss networks with applications to instance selection. *J. Mach. Learn. Res.*, 9:997–1017, June 2008.
4. J. A. Olvera-López, J. Fco. Martínez-Trinidad, J. A. Carrasco-Ochoa, and J. Kittler. Prototype selection based on sequential search. *Intell. Data Anal.*, 13(4):599–631, December 2009.
5. D. Randall Wilson and Tony R. Martinez. Instance pruning techniques. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, pages 404–411. Morgan Kaufmann, 1997.

© Ediciones Universidad de Salamanca