WebDR: A Web Workbench for Data Reduction

Stefanos Ougiaroglou* and Georgios Evangelidis

Department of Applied Informatics, School of Information Sciences,
University of Macedonia, 156 Egnatia str, GR-54006 Thessaloniki, Greece
{stoug,gevan}@uom.gr

Abstract. Data reduction is a common preprocessing task in the context of the k nearest neighbour classification. This paper presents WebDR, a web-based application where several data reduction techniques have been integrated and can be executed on-line. WebDR allows the performance evaluation of the classification process through a web interface. Therefore, it can be used by the academia for educational and experimental purposes.

Keywords: k-NN classification, data reduction, web-based application.

1 Introduction

The k Nearest Neighbour (k-NN) classifier [3] is an effective classifier but has some weaknesses that may render its use inappropriate. The first one is the high computational cost involved (all distances between each unseen item and all training data must be computed). In cases of large datasets, this drawback renders the classification a time-consuming procedure. Another weakness is that the k-NN classifier must maintain all the training data always available. Thus, it involves high storage requirements. Moreover, the accuracy achieved by the classifier depends on the quality of the training set (TS). Noise and mislabelled data, as well as outliers and overlaps between data regions of different classes may mislead the algorithm and affect the accuracy.

Data Reduction Techniques (DRTs) can cope with all the weaknesses. They can be grouped into two main categories: (i) prototype selection algorithms (PS) [6], and, (ii) prototype abstraction algorithms (PA) [17]. PS algorithms select representative items (or prototypes) from the initial training set, whereas PA algorithms generate items by summarizing on similar training items.

PS algorithms are divided into two subcategories. They can be either condensing or editing algorithms. PA and PS-condensing algorithms have the same motivation. They aim to build a small representative set of the TS. This set is called the condensing set (CS). Usage of the CS has the benefits of low computational cost and storage requirements, while accuracy is not affected. Editing algorithms aim to improve accuracy rather than achieve high reduction rates. For that purpose, they try to improve the quality of the TS by removing outliers, noise and by smoothing the class decision boundaries.

* S. Ougiaroglou is supported by the State Scholarships Foundation of Greece (I.K.Y.)

T. Calders et al. (Eds.): ECML PKDD 2014, Part III, LNCS 8726, pp. 464–467, 2014.

Several papers have been published that present DRTs with the corresponding experimental results. Some of them have been implemented under KEEL [2], an open-source java-based framework. However, to the best of our knowledge there is no software that allows experimentations over the web. This observation is behind the motivation of this work. We introduce WebDR[1] (Web-based Data Reduction), a web-based application that allows the execution and the performance evaluation of several DRTs over the web.

Section 2 outlines k-NN classification through data reduction. Section 3 presents WebDR. Finally, Section 3 concludes the paper and presents our future plans.

2 *k*-NN Classification through Data Reduction

The reduction rates achieved by many PA and PS-condensing algorithms depend on the level of noise in the TS. The higher the level of noise is, the lower reduction rates are achieved. Hence, their effective application implies the removal of noise, i.e., execution of editing beforehand [4]. Therefore, an editing algorithm should be run in order to either improve accuracy or make more effective the application of a PA or PS-condensing algorithm.

k-NN classification through data reduction is summarized in Figure 1. The process has two stages, preprocessing (optionally) and classification. There are four possible preprocessing types: (i) **No-preprocessing:** If the TS is small and noise-free, no preprocessing is required. (ii) **Only editing:** If the TS is small but contains noise, only editing should be executed during preprocessing. (iii) **Only condensing:** In cases of large and noise-free TSs, data reduction without editing should be executed (i.e., a PA or PS-condensing algorithm). (iv) **Both editing and PA or PS-condensing:** In cases of large TSs that contain noise, both types of preprocessing algorithms must be run.

The goal of a complete data reduction preprocessing procedure is to build a noise-free CS by keeping or generating for each class a sufficient number of prototypes that are essential for the k-NN classification.

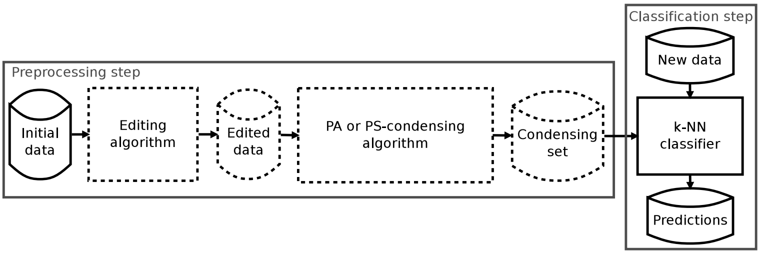

Fig. 1. k-NN classification through data reduction

[1] http://dbtech.uom.gr/webdr

3 WebDR

WebDR offers several DRTs available on-line. The user can plan and run experiments and measure the classification performance through an interactive web interface over several known datasets distributed by the KEEL[2] or/and the UCI[3] dataset repositories and time-series datasets distributed by the UCR time-series classification/clustering website[4]. All the available datasets can be explored in detail using the "dataset explorer" tool that is available in WebDR.

WebDR allows the performance evaluation of the DRTs by measuring three criteria, namely, (i) **Reduction rate:** the ratio of the discarded items over the initial items of the TS. The higher the reduction rate, the faster the k-NN classification (fewer distances are computed); (ii) **Accuracy** achieved by the k-NN classifier when it runs over the CS; (iii) **Preprocessing cost:** the computational cost required for the construction of the CS.

The preprocessing costs are estimated by counting the distances computed by the corresponding DRTs. WebDR adopts the Euclidean distance as the distance metric. The reported performance measurements are averages obtained via five-fold cross-validation. It is worth mentioning that all datasets built during preprocessing are available to the users in a five-fold form (five pairs of training and testing sets). They can be downloaded and used by the user locally. Of course, the number of the nearest neighbours and the DRT specific parameters (if any) can be adjusted through the interface.

All the possible preprocessing types can be executed by WebDR. Its main page offers four links. Each one leads to the corresponding type of preprocessing. Currently, the following DRTs have been integrated in WebDR:

– **Editing algorithms:** ENN-rule [18], All-k-NN [16], Multiedit [5], EHC [14]
– **Condensing algorithms:** CNN-rule [7], IB2 [1], PSC [8]
– **PA algorithms:** RSP3 [15], RHC [11,10], dRHC [10], ERHC [9], AIB2 [13], RkM [12]

WebDR is hosted on a Debian GNU/Linux server with two 64-bit Quad-Core CPUs and 2GB of main memory. All algorithms were coded in C. The web interface was developed using PHP (server-side programming) and html/CSS and javascript (client-side programming). The executable binaries of the implemented algorithms are located and executed on the server.

4 Conclusions and Future Work

The paper presented WebDR, a web-based application that allows the performance evaluation of several DRTs over the web. It aspires to support teaching and research on data reduction. We plan to integrate more DRTs and datasets in WebDR. Moreover, we will develop a mechanism that will allow users to run experiments on their own datasets.

[2] http://sci2s.ugr.es/keel/datasets.php

[3] http://archive.ics.uci.edu/ml/

[4] http://www.cs.ucr.edu/~eamonn/time_series_data/

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. 6(1), 37–66 (1991), http://dx.doi.org/10.1023/A:1022689900470
2. Alcala-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernandez, J.C., Herrera, F.: Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput. 13(3), 307–318 (2008)
3. Dasarathy, B.V.: Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press (1991)
4. Dasarathy, B.V., Sánchez, J.S., Townsend, S.: Nearest neighbour editing and condensing tools synergy exploitation. Pattern Analysis & Applications 3(1), 19–30 (2000)
5. Devijver, P.A., Kittler, J.: On the edited nearest neighbor rule. In: Proceedings of the Fifth International Conference on Pattern Recognition. The Institute of Electrical and Electronics Engineers (1980)
6. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell. 34(3), 417–435 (2012)
7. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14(3), 515–516 (1968)
8. Olvera-Lopez, J.A., Carrasco-Ochoa, J.A., Trinidad, J.F.M.: A new fast prototype selection method based on clustering. Pattern Anal. Appl. 13(2), 131–141 (2010)
9. Ougiaroglou, S., Evangelidis, G.: Efficient editing and data abstraction by finding homogeneous clusters. In: Submitted, under review
10. Ougiaroglou, S., Evangelidis, G.: RHC: Non-parametric cluster-based data reduction for efficient k-nn classification. Pattern Analysis and Applications pp. (accepted, to appear)
11. Ougiaroglou, S., Evangelidis, G.: Efficient dataset size reduction by finding homogeneous clusters. In: Proceedings of the Fifth Balkan Conference in Informatics, BCI 2012, pp. 168–173. ACM Press, New York (2012)
12. Ougiaroglou, S., Evangelidis, G.: A simple noise-tolerant abstraction algorithm for fast *k*-NN classification. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part II. LNCS, vol. 7209, pp. 210–221. Springer, Heidelberg (2012)
13. Ougiaroglou, S., Evangelidis, G.: AIB2: An abstraction data reduction technique based on ib2. In: Proceedings of the 6th Balkan Conference in Informatics, BCI 2013, pp. 13–16. ACM, New York (2013)
14. Ougiaroglou, S., Evangelidis, G.: EHC: Non-parametric editing by finding homogeneous clusters. In: Beierle, C., Meghini, C. (eds.) FoIKS 2014. LNCS, vol. 8367, pp. 290–304. Springer, Heidelberg (2014)
15. Sánchez, J.S.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recognition 37(7), 1561–1564 (2004)
16. Tomek, I.: An experiment with the edited nearest-neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics 6, 448–452 (1976)
17. Triguero, I., Derrac, J., Garcia, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. Trans. Sys. Man Cyber Part C 42(1), 86–100 (2012)
18. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. on Systems, Man, and Cybernetics 2(3), 408–421 (1972)