Neurocomputing 280 (2018) 101–110

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Exploring the effect of data reduction on Neural Network and Support Vector Machine classification☆

Stefanos Ougiaroglou[a,b,*], Konstantinos I. Diamantaras[b], Georgios Evangelidis[a]

[a] Department of Applied Informatics, School of Information Sciences, University of Macedonia, Thessaloniki 54006, Greece
[b] Department of Information Technology, Alexander TEI of Thessaloniki, Sindos 57400, Greece

ARTICLE INFO

Article history:
Received 20 January 2017
Revised 23 July 2017
Accepted 16 August 2017
Available online 20 November 2017

MSC:
00-01
99-00

Keywords:
Neural Networks
Support Vector Machines
k-NN classification
Data reduction
Prototype selection
Prototype generation
Condensing

ABSTRACT

Neural Networks and Support Vector Machines (SVMs) are two of the most popular and efficient supervised classification models. However, in the context of large datasets many complexity issues arise due to high memory requirements and high computational cost. In the context of the application of Data Mining algorithms, data reduction techniques attempt to reduce the size of training datasets in terms of the number of instances by selecting some of the existing instances or by generating new training instances. The idea is to speed up the application of the data mining algorithm with minimum or no sacrifice in performance. Data reduction techniques have been extensively used in the context of *k*-Nearest Neighbor classification, a lazy classifier that works by directly using a training dataset rather than building a model. This paper explores the application of data reduction techniques as a preprocessing step before the training step of Neural Networks and SVMs. Furthermore, the paper proposes a new data reduction technique that is based on k-median clustering algorithm. Our experimental results illustrate that, in the case of SVMs, data reduction techniques can effectively reduce the dataset size incurring small performance degradation. In the case of Neural Networks, the performance loss is somewhat greater, for the same data reduction rate, but both SVM and Neural Network models outperform the *k*-NN approach that is typically used in Data Mining applications.

1. Introduction

In the recent years, problems involving high volumes of data challenge the effectiveness, efficiency and scalability of machine learning and data mining algorithms. Most popular proposed algorithms cannot be applied in such big data analysis scenarios, thus, new research issues have attracted the attention of both the industry and academia. For those algorithms data reduction is a necessary preprocessing step. Data Reduction can be thought of as item reduction or dimensionality reduction. In this paper, we deal with item reduction. More specifically, we consider classification tasks where data reduction processes are guided by the class labels.

Most of the Data Reduction Techniques (DRTs) were proposed in the context of dealing with the drawbacks of *k*-Nearest Neighbors (*k*-NN) classifier [2]. This classifier has high computational cost during classification, high memory requirements and is noise sensitive. This is because, the *k*-NN classifier works by using the training set as a model to classify new instances. Usually, the larger or more detailed the training set, the more accurate the classification.

A DRT can be either a Prototype Selection algorithm (PS) [3] that selects representative instances from the training set, or a Prototype Generation algorithm (PG) [4] that generates representatives by summarizing similar training instances. These chosen/generated representatives are called Prototypes. PS algorithms can be either editing or condensing. Editing attempts to improve accuracy by removing noise, outliers and mislabeled instances and by smoothing the decision boundaries between classes. PG and PS-condensing algorithms build a small condensing dataset that represents the initial training data, thus, allowing the classification step to achieve low cost while accuracy remains almost as high as that achieved by using the original data. Some PG and PS-condensing algorithms are called hybrid because they integrate the concept of editing.

Dimensionality reduction and traditional sampling techniques have been applied to speed up the training times of eager classifiers. However, to the best of our knowledge, item reduction techniques have been used in the context of *k*-NN classification, but

☆ This is an extended version of the work presented in [1].

* Corresponding author at: Department of Applied Informatics, School of Information Sciences, University of Macedonia, 54006 Thessaloniki, Greece.
E-mail addresses: stoug@uom.gr, stoug@it.teithe.gr (S. Ougiaroglou), kdiamant@it.teithe.gr (K.I. Diamantaras), gevan@uom.gr (G. Evangelidis).

https://doi.org/10.1016/j.neucom.2017.08.076
0925-2312/

not in the context of large datasets in order to render the usage of Neural Networks and SVMs applicable on them. This is the key observation behind the motivation of this paper. Another motive is to check whether PG algorithms we proposed in the past can aid the development of fast and accurate Neural Networks and/or SVM based classifiers.

This paper contributes an experimental study on several datasets where Neural Networks and SVM based classifiers, which are trained by the original training data and the corresponding condensing sets built by state-of-the-art DRTs, are compared to each other and against the corresponding *k*-NN classifiers. Our study reviews in detail the algorithms that are used in the experimental study and reveals that the usage of DRTs leads to fast and accurate SVM-based classifiers.

Moreover, this paper presents a new variation of a PG algorithm we proposed in the past. The previously proposed algorithm is called Editing and Reduction through Homogeneous Clusters (ERHC) [5] and utilizes *k*-means clustering [6,7] in order to build its condensing set. The main motivation behind the development of the new variation is to examine whether the use of *k*-medians [8] instead of *k*-means is a better choice in the case of datasets with outliers and noise. The new variation is called ERHC-MD and is expected to be more tolerant to the existence of outliers.

Although dimensionality reduction can be combined with PS-condensing and PG algorithms to obtain even faster *k*-NN classifiers or to speed-up the training of eager classifiers, this is not the objective of the present paper.

The rest of this paper is organized as follows. Section 2 briefly reviews Neural Networks, SVMs and the *k*-NN classifier. Section 3 presents in detail the PG and PS-condensing algorithms that we use in our experimental setup. The ERHC-MD algorithm is presented in Section 3.5.4. Section 4 presents the experimental study and the obtained results. Finally, Section 5 concludes the paper and gives direction for feature work.

2. Neural Networks, Support Vector Machines and the *k*-NN classifier

2.1. Feed-forward Neural Networks

Multilayer feed-forward neural networks [9] are some of the most popular learning models used in both classification and regression applications. The Multilayer Perceptron (MLP) with a single sigmoid hidden layer is known to have the *Universal Approximator* property [10], ie. it is capable of approximating any continuous function in the unit hypercube with arbitrary degree of accuracy, provided that there is no limit on the number of available hidden units. Different variations of the classical Back-Propagation algorithm [11] are typically used to train an MLP.

One of the most popular and efficient BP variations is the Conjugate Gradient method [12,13]. Although networks with one hidden layer have been criticized for their relative inefficiency in capturing complex data representations compared to deep networks [14] they are still considered as yardsticks against which other learning methods are compared. One advantage of shallow networks is the fact that they have fewer tunable parameters to adjust, typically only the number of hidden units. Also the learning algorithm can have lower computational complexity compared to a deep learning model. Since our work focuses on the usefulness of data reduction as a preprocessing step for pattern classification and 2-layer neural networks are very popular classifiers we use them as one of the benchmark methods for our study.

2.2. Support Vector Machines

SVMs are supervised learning models introduced in 1995 by Cortes and Vapnik [15], but the original idea lies in the theory of statistical learning introduced by Vapnik [16] almost two decades earlier. They are suitable for pattern classification but can be easily extended to handle nonlinear regression problems in which case they are known as Support Vector Regressors (SVRs). The separating surface offered by an SVM classifier maximizes the *margin*, i.e., the distance of the closest patterns to it. This helps the generalization performance of the model and in fact it is related to the idea of Structural Risk Minimization [17,18] which avoids overfitting. With the use of nonlinear kernel functions such as Gaussian (RBF) or *n*th order polynomials, SVM models can produce nonlinear separating surfaces achieving very good performance in complex problems.

Due to their good generalization performance these models have become very popular with a wide range of applications, including document classification, image classification, bioinformatics, handwritten character recognition, etc. One of the major drawbacks of these models is the memory and the computational complexity requirements for large datasets. The reason is that the separating surface is obtained by solving a quadratic programming problem involving an $N \times N$ matrix, where N is the number of items in the dataset. Although there are techniques that can reduce the complexity to $O(N^2)$ [19], the problem remains hard and the size of the problem can easily become prohibitively large calling for methods for data reduction such as the ones discussed in the following sections.

2.3. k-Nearest Neighbor classifier

The *k*-NN classifier [2] is an extensively used lazy (or instance-based) classification algorithm. Lazy classifiers do not build any classification model like eager classifiers do. The *k*-NN classifier is a quite simple and easy to implement algorithm, the predictions it makes are easy to understand/explain, it is analytically tractable, and, for $k = 1$ and unlimited instances the error rate is asymptotically never worse than twice the minimum possible, which is the Bayes rate [20].

A new instance is classified by retrieving the *k* nearest instances or neighbors to it from the training set. Then, the new instance is assigned to the most common class among the classes of the *k* nearest neighbors. This class is called the major class. The Euclidean distance is the commonly used distance metric, although any distance metric can be used. Despite not spending any time to train a model, the classification step is time consuming because in the worst case the algorithm must compute all distances between the new instance and all the training instances.

Another issue is that the selection of the value of *k* affects the accuracy of the classifier. The value of *k* that gives the highest accuracy depends on the dataset at hand and needs to be tuned in advance. Usually, large *k* values are appropriate for noisy datasets since they examine larger neighborhoods. In binary problems, an odd value for *k* should be used so that possible ties during nearest neighbors voting are avoided. In problems with more than two classes, ties are resolved by choosing a random “most common” class or the class voted by the nearest neighbor. The later is the approach adopted in the experimental study of this paper.

3. Prototype generation and condensing algorithms

Although, there are numerous PG and PS-condensing algorithms available in the literature, here, we review only the ones used in our experimental study. For the interested reader, abstraction and