Pattern Recognition Letters 62 (2015) 61–67

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Parallel selective sampling method for imbalanced and large data classification☆

Annarita D'Addabbo, Rosalia Maglietta*

Institute of Intelligent Systems for Automation - National Research Council, via Amendola 122/D-O, Bari 70126, Italy

ARTICLE INFO

Article history:
Received 10 December 2014
Available online 5 June 2015

Keywords:
Imbalanced learning
Classification
Support vector machine
Selective sampling methods

ABSTRACT

Several applications aim to identify rare events from very large data sets. Classification algorithms may present great limitations on large data sets and show a performance degradation due to class imbalance. Many solutions have been presented in literature to deal with the problem of huge amount of data or imbalancing separately. In this paper we assessed the performances of a novel method, Parallel Selective Sampling (PSS), able to select data from the majority class to reduce imbalance in large data sets. PSS was combined with the Support Vector Machine (SVM) classification. PSS-SVM showed excellent performances on synthetic data sets, much better than SVM. Moreover, we showed that on real data sets PSS-SVM classifiers had performances slightly better than those of SVM and RUSBoost classifiers with reduced processing times. In fact, the proposed strategy was conceived and designed for parallel and distributed computing. In conclusion, PSS-SVM is a valuable alternative to SVM and RUSBoost for the problem of classification by huge and imbalanced data, due to its accurate statistical predictions and low computational complexity.

1. Introduction

Many real-word applications of machine learning classifiers have to identify rare events from very large data sets. For example, in the studies on the automated segmentation from magnetic resonance images [19–21], the number of training examples is very huge (up to millions), the classes are strongly imbalanced, and generating accurate statistical solution is not trivial. In addition, data imbalance in huge data sets is also reported in other applicative domains, such as marketing data [22], oil spill detection or land cover changes from remote sensing images [16,27], text classification [18] and scene classification [35]. In these areas, very large data sets have to be handled and the minority class is the one of interest, consequently two problematic issues add on: the computational complexity dependent on the size of the data set and the need to pursue a fairly high rate of correct detections in the minority class.

Many classification algorithms present great limitations on large data sets and show a performance degradation due to class imbalance [14]. For example, Support Vector Machines (SVM) [33], that are employed in many applicative domains [3,4,13,24], really become intractable and computationally too expensive when huge data sets are handled [32]. In fact, the training complexity of SVM is highly dependent on the size of the data set. Moreover, SVM classification performance can be hindered by class imbalance [1,30]. Compared with other standard classifiers, it is more accurate on moderately imbalanced data. The reason is that only SVs are used for classification and many majority samples far from the decision boundary can be removed without affecting the classification. However, an SVM classifier can be sensitive to high class imbalance, resulting in a drop of the classification performance on the minority class. In fact, it is prone to generate classifier that has a strong estimation bias toward the majority class: since the number of majority class patterns exceeds that of the minority class, the class boundary becomes vulnerable to be distorted [15]. Nevertheless, these limitations are common to many other classification schemes such as Multi-Layer Perceptron (MLP) [7] and Logistic Regression (LR) [23].

To overcome these problems, a selection of examples has to be performed sampling a small number of patterns from the majority class to reduce both the number of data and the imbalance. Such a procedure is well known in literature as "undersampling" method [12]. It generally improves the classification performance and reduces the computational complexity, however it presents a potential disadvantage of distorting the distribution of the majority class. If the sampled patterns from the majority class do not represent the original distribution, it may degrade the classification performance. This potential drawback comes dramatically true when the number of minority class patterns is very small [15]. However, other techniques are

☆ This paper has been recommended for acceptance by Y. Liu.

* Corresponding author. Tel.: +39 80 5929454; fax: +39 80 5929460.
E-mail address: maglietta@ba.issia.cnr.it (R. Maglietta).

http://dx.doi.org/10.1016/j.patrec.2015.05.008

not feasible with very large data set because they work: (1) by modifying cost for misclassified patterns belonging to the minority class, without changing the number of original data [7], (2) by increasing the total number of examples by copying patterns from the minority class to balance the ratio of classes ("oversampling" method) [9], (3) by combining oversampling and undersampling techniques [34].

Several methods to select examples in a classification problem are presented in literature, using two different approaches: the example-selection method can be embedded within the learning algorithm or the examples can be filtered before passing them to the classification scheme [2,26]. It is worth noting that the first type of selection methods generally work by preserving the original ratio between classes [6,11]: if there is a great skew in the data, it continues to be. To overcome this problem, filtered methods are more suited for preprocessing data before the classification step. Numerous algorithms can be used taking into account the class-membership of samples to solve the imbalance in the data [2]. In this framework, a very interesting method has been developed by Evgeniou and Pontil in [10]. They present a preprocessing algorithm that computes clusters of points in each class, based on Euclidean distance, and substitute each cluster with the mass center of the points in the cluster. The algorithm tends to produce large (small) clusters of data points which are far (near) from the boundary between the two classes. These strategies did not focus on both large and imbalanced data learning. More recently, a method focused on both big and class imbalanced data classification was proposed [29]. It is a cost-sensitive support vector machine using randomized dual coordinate descent method (CSVM-RDCD) and it belongs to the class of embedded methods, i.e. the examples selection is integrated in the learning algorithm and classifier dependent. This method was tested on three data sets with relative class imbalance and three data sets with severe class imbalance, of which only one of them with a large number of examples. In all the experiments the recognition rates of the minority class, computed by CSVM-RDCD and SVM, are roughly comparable, with an improvement of about 1%. New studies are required in order to examine in more depth the case study of imbalanced and big data.

A valuable alternative is given by filter methods which are attractive because they adjust only the distribution of the original training set, independently of the given classifier. In this paper, we describe a novel approach, named Parallel Selective Sampling (PSS), that selects data from the majority class to reduce imbalance in big data sets. PSS is a filter method which can be combined with a variety of classification strategies. It is based on the idea (usually used in SVM) that only training data, near the separating boundary (for classification), are relevant. In this way the core information from the training data - i.e. the training data points near the separating boundary - is preserved while the size of the training set is effectively decreased. Relevant examples from the majority class are selected and used in the successive classification step using SVM. Due to the complex computational requirements, PSS is conceived and designed for parallel and distributed computing. Finally, PSS–SVM allows accurate statistical predictions keeping down the computational times.

The paper is organized as follows: in Section 2 we describe in details the proposed sampling method and we introduce the main properties of SVM for classification of large and imbalanced data sets. In Section 3 we discuss the experimental results obtained in the analysis of real and simulated data sets. Section 4 concludes the paper and summarizes the main results.

2. Methods

2.1. PSS

The PSS method can be used to preprocess very large training data with significant skew between classes. It is an undersampling method because it acts by reducing examples belonging only to the majority class. It is based on the computation of Tomek links [31], defined as a pair of nearest neighbors of opposite classes. Given $\{E_1, \ldots, E_n\} \in R^k$, a pair (E_i, E_j) is called a Tomek link if E_i and E_j have different labels, and there is not an E_l such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$, where $d(\cdot, \cdot)$ is the Euclidean distance. Here Tomek links are used to remove samples of majority class staying in areas of input space dense of data belonging to the same class.

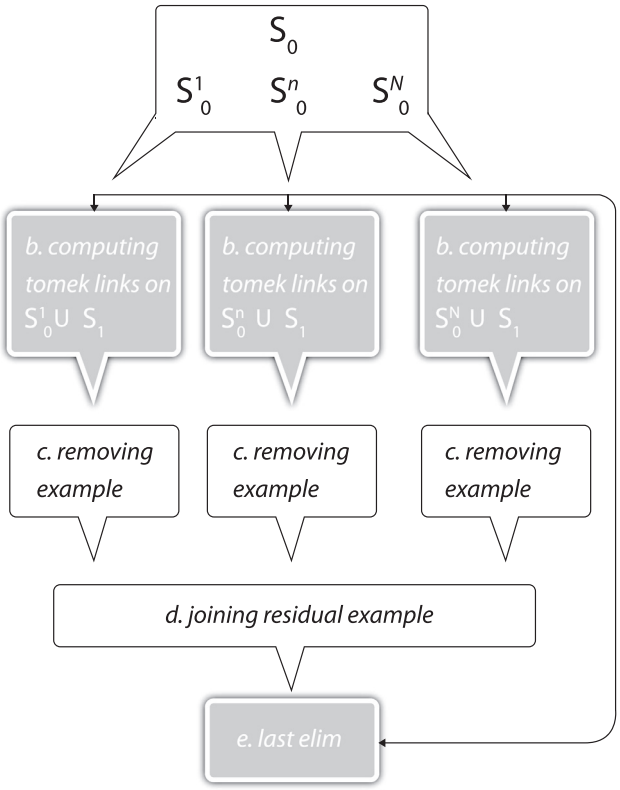

Fig. 1. Block diagram of PSS.

Let $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ be the training set, where $x_i \in R^k$ and $y_i \in \{0, 1\}$, $\forall i = 1, \ldots, \ell$. We define S_0 the set of ℓ_0 training data belonging to class $y = 0$ and S_1 the set of ℓ_1 training data belonging to class $y = 1$, with $\ell_0 \gg \ell_1$. PSS achieves a reduced training set whose percentage M% of the minority class on the total number of examples is chosen by the user.

a: **data partitioning.** The S_0 set is divided into N subset S_0^n with $n = 1, 2, \ldots,$N, with N set by the user. In this way, N different undersampling procedures are performed in parallel computation (see Fig. 1).
For each S_0^n, with $n = 1, 2, \ldots,$N, the following steps are performed:

b: **computing Tomek links.** Let us define the set T^n of all examples in the majority class S_0^n that are first neighbors of one sample in S_1, that is $T^n = \{x \in S_0^n | (x, z)$ is Tomek link on $S_1 \cup S_0^n, z \in S_1\}$.

c: **removing examples.** Let us randomly select $\bar{x} \in D^n = S_0^n \backslash T^n$; the following steps are performed (see Fig. 2):

- the Tomek link $(\bar{x}, \bar{z})$ is computed over the data set $\bar{x} \cup S_1$, with $\bar{z} \in S_1$;
- the Euclidean distances $d(\bar{x}, x)$ are computed for each $x \in S_0^n$;
- let us define the subset $L = \{x \in S_0^n | d(\bar{x}, x) < d(\bar{x}, \bar{z})\}$, (see the red circumference in Fig. 2a). The Tomek link $(x^*, \bar{z})$ in $\bar{z} \cup L$ is computed, i.e. x^* is defined as the first neighbor in L of $\bar{z}$;
- let us define the set $R = \{x \in L | d(\bar{x}, x) < [d(\bar{x}, \bar{z}) - d(x^*, \bar{z})]\}$ (see the blue circumference in Fig. 2a). Let us delete all the points in R that are not Tomek links, i.e. each $x \in R'$ with