Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

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

EPRENNID: An evolutionary prototype reduction based ensemble for nearest neighbor classification of imbalanced data

Sarah Vluymans [a,b,c,*], Isaac Triguero [b,d,e], Chris Cornelis [a,c], Yvan Saeys [b,d]

[a] Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Belgium
[b] Data Mining and Modeling for Biomedicine, VIB Inflammation Research Center, Ghent, Belgium
[c] Department of Computer Science and Artificial Intelligence, University of Granada, Spain
[d] Department of Internal Medicine, Ghent University, Belgium
[e] School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, United Kingdom

ARTICLE INFO

Article history:
Received 27 November 2015
Received in revised form
7 July 2016
Accepted 4 August 2016
Communicated by Swagatam Das

Keywords:
Imbalanced data
Prototype selection
Prototype generation
Differential evolution
Nearest neighbor

ABSTRACT

Classification problems with an imbalanced class distribution have received an increased amount of attention within the machine learning community over the last decade. They are encountered in a growing number of real-world situations and pose a challenge to standard machine learning techniques. We propose a new hybrid method specifically tailored to handle class imbalance, called EPRENNID. It performs an evolutionary prototype reduction focused on providing diverse solutions to prevent the method from overfitting the training set. It also allows us to explicitly reduce the underrepresented class, which the most common preprocessing solutions handling class imbalance usually protect. As part of the experimental study, we show that the proposed prototype reduction method outperforms state-of-the-art preprocessing techniques. The preprocessing step yields multiple prototype sets that are later used in an ensemble, performing a weighted voting scheme with the nearest neighbor classifier. EPRENNID is experimentally shown to significantly outperform previous proposals.

1. Introduction

Class imbalance is present in a dataset when its instances are unevenly distributed among the classes. It is encountered in many real-world situations such as medical diagnosis [1], microarray data analysis [2] or software quality evaluation [3]. Many applications are inherently prone to class imbalance, motivating the increased amount of attention to this issue within the machine learning community [4].

The class imbalance problem [5] refers to the fact that the performance of learning algorithms can be severely hampered by data imbalance. In this work, we focus on two-class imbalanced classification, where the elements of the *majority* class outnumber those of the *minority* class. Traditionally, the majority elements are denoted as *negative*, whereas the minority elements are referred to as *positive*. Standard classification techniques may not perform well in this context, as they internally assume equal class distributions. Consequently, over the last decade, a considerable amount of work has been proposed in the specialized literature to alleviate the imbalance problem [6–8]. Some approaches work at the data level, while others develop custom classification processes. At the data level, the so-called data sampling methods modify the training dataset to produce a better balance between classes [9,10]. Solutions at the algorithm level are modifications of existing methods and internally deal with the intrinsic challenges of imbalanced classification [11,12].

Prototype reduction techniques [13] were originally developed to simplify large training datasets in order to improve the noise tolerance, the speed and the storage requirements of learning models [14,15]. They can be applied to imbalanced datasets [16–18] as a data level approach, balancing majority and minority classes. Two main families of prototype reduction techniques exist in the literature: prototype selection (PS) [19] and prototype generation (PG) [20]. The former is limited to selecting a subset of instances from the original training data, while the latter can create new artificial instances to better adjust the decision boundaries of the classes. However, PG methods are known to be susceptible to overfitting [20,21]. The best performing models are evolutionary-based techniques, such as differential evolution [22]. In [23], the authors showed that a hybrid setting of PS and PG can significantly improve the classification process in a balanced class setting. To the best of our knowledge, no hybrid PS-PG techniques

* Corresponding author at: Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Belgium.
E-mail address: Sarah.Vluymans@UGent.be (S. Vluymans).

http://dx.doi.org/10.1016/j.neucom.2016.08.026
0925-2312/

have been developed to deal with imbalanced classification problems so far.

In this paper, we propose a combined model for the classification of two-class imbalanced data, integrating both a hybrid preprocessing and a classification step. We extend the framework of [23] for use in the presence of class imbalance, considerably modifying both PS and PG stages. We also aim at introducing diversity in the process. The multiple prototype sets resulting from the preprocessing step are further combined in a custom ensemble for classification. The classification step is an extension of the k nearest neighbor classifier (kNN [24]). We call our method EPRENNID, an Evolutionary Prototype Reduction based Ensemble for Nearest Neighbor classification of Imbalanced Data.

The main contributions of this work are as follows:

- We first introduce a new evolutionary PS method specifically tuned to handle class imbalance. Although it is related to undersampling methods, it takes a step away from them by allowing the removal of minority elements from the dataset, as in [25]. Most existing methods do not allow such kind of reduction of non-representative or noisy elements from the positive class.
- To alleviate the overfitting issues of prototype reduction models, we take advantage of the evolutionary nature of the proposed method. Instead of yielding a single reduced set, EPRENNID provides several well-performing and diverse ones.
- The evolutionary PG method used in this work [23] has been modified to handle the class imbalance problem.
- Finally, the optimized prototype sets are used in a classifier ensemble, using an adaptive scheme selecting the most suitable prototype sets to classify each single target instance with kNN.

To analyze the performance of our proposal, we carry out an extensive experimental study on 35 two-class imbalanced datasets, categorized into different groups corresponding to the difficulty of identifying minority elements. We compare our model with state-of-the-art models and apply non-parametric statistical tests to check whether there are significant differences among them.

The remainder of this paper is structured as follows. In Section 2, we review the PS and PG schemes and provide more details on related work in imbalanced classification. Section 3 introduces the proposed model, with a detailed explanation of the separate preprocessing and classification phases. We have conducted a comprehensive experimental study. Its setup is described in Section 4, while Section 5 lists and discusses our results. Finally, Section 6 formulates the conclusions of this work and outlines future research directions.

2. Preliminaries and related work

This section provides the necessary background for the remainder of the paper. Section 2.1 presents prototype selection and generation techniques, focusing on the methods on which our model is based. Section 2.2 introduces the problem of classification with imbalanced datasets and its evaluation is recalled in Section 2.3.

2.1. Prototype reduction

Prototype reduction techniques aim to reduce the available training set $T = \{x_1, x_2, \ldots, x_n\}$ of labeled instances to a smaller set of prototypes $S = \{y_1, y_2, \ldots, y_r\}$, with $r < n$ and each y_i either drawn from T or artificially constructed. The set S, rather than the entire set T, is used afterwards to train the classifier.

These methods are commonly combined and designed to be used with the kNN classifier. This lazy learning algorithm [26] assigns new input instances to the class to which the majority of their k nearest neighbors in the training set belongs. Despite its performance, it suffers from several drawbacks such as low efficiency, high storage requirements and sensitivity to noise. PS and PG techniques can be beneficial to alleviate these issues. To that end, the instances contained in S should form a good representation of the original class distributions. Furthermore, their size relative to that of T should be small enough in order to considerably reduce the storage and execution time requirements of kNN.

A PS method reduces T to S by selecting a subset of its instances. This implies that for every instance $y_i \in S$ there exists an element $x_j \in T$ such that $y_i = x_j$. In [19], a taxonomy for PS methods was proposed and an extensive experimental study was conducted. The main difference between PG and PS is that the former can either select elements from T or construct artificial ones, while the latter is restricted to selecting elements from T. Therefore, a set S constructed by a PG method is not necessarily a subset of T, allowing for a larger flexibility in the construction of S. For PG methods, a related taxonomy has been proposed in [20]. In what follows, we describe the PS and PG methods on which we base our proposal.

2.1.1. Steady state memetic algorithm for instance selection

The Steady State Memetic Algorithm (SSMA) is a genetic algorithm for PS. In several experimental studies (e.g. [19,23]), it has been shown to be one of the best-performing PS methods, which is due to its optimization procedure performed in each iteration. As a genetic algorithm, it evolves a population of I individuals, the chromosomes, over a number of generations G. Each individual corresponds to a candidate subset and is encoded as a bitstring, where a 0 in the ith position means that the ith element of T is not included in the subset, while a 1 means that it is. The quality of an individual, that is, how good a solution it is, is evaluated by a so-called fitness function. To calculate the fitness of a candidate subset S, SSMA uses a combined criterion, namely the accuracy of the kNN classifier on the entire training set T using S as prototype set and the reduction in size of S relative to T.

The population is optimized over the subsequent generations, such that the final fittest individual corresponds to an optimal solution. To guide the evolution, it uses two genetic operators: crossover and mutation. In each generation, two parents are selected to produce two new individuals by means of the Half Uniform Crossover (HUX) procedure: positions in which the parents take on the same value are simply copied to the children, while for the remaining ones, each child randomly copies half of each parent.

Afterwards, random mutation is applied to the children. This procedure changes the value of a randomly selected position with probability p. The most defining aspect of the SSMA method is its use of an optimization procedure, the so-called meme. This is an iterative optimization process that pursues a double objective to improve individuals of the population: the reduction of the number of selected prototypes and the enhancement of the classification accuracy. The meme is applied on a generated child when its fitness value is higher than the current lowest fitness in the population. When its fitness is lower, the optimization is only executed with a small probability. We refer to the original proposal [27] for a detailed description.

2.1.2. Scale factor local search in differential evolution

Scale Factor Local Search in Differential Evolution (SFLSDE) [28] was shown to be one of the top performing PG methods in the experimental study of [23]. It is a positioning adjustment algorithm, optimizing the positions of the instances in the dataset. The method uses differential evolution (DE [29,22]), which follows the

Please cite this article as: S. Vluymans, et al., EPRENNID: An evolutionary prototype reduction based ensemble for nearest neighbor classification of imbalanced data, Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2016.08.026