J. Vis. Commun. Image R. 38 (2016) 340–350

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Low false positive learning with support vector machines ☆

Daniel Moraes *, Jacques Wainer, Anderson Rocha

Institute of Computing, University of Campinas, Av. Albert Einstein, 1251, 13083-852 Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 30 July 2015
Revised 12 February 2016
Accepted 7 March 2016
Available online 15 March 2016

Keywords:
Support vector machines
k-nearest neighbors
Low false positive learning

ABSTRACT

Most machine learning systems for binary classification are trained using algorithms that maximize the accuracy and assume that false positives and false negatives are equally bad. However, in many applications, these two types of errors may have very different costs. In this paper, we consider the problem of controlling the false positive rate on SVMs, since its traditional formulation does not offer such assurance. To solve this problem, we define a feature space sensitive area, where the probability of having false positives is higher, and use a second classifier (unanimity *k*-NN) in this area to better filter errors and improve the decision-making process. We call this method Risk Area SVM (RA-SVM). We compare the RA-SVM to other state-of-the-art methods for low false positive classification using 33 standard datasets in the literature. The solution we propose shows better performance in the vast majority of the cases using the standard Neyman–Pearson measure.

1. Introduction

There are several applications that are sensitive to false positives, such as spam filtering, face recognition and computer-aided diagnosis. In these applications, the errors from one class are much more costly than errors from the other class, and keeping the false positive rate under a maximal tolerance is usually more important than achieving a high classification accuracy. For example, in computer-based diagnosis, especially if the automated system is being used for triage of patients, falsely determining that a case is normal is much more serious than falsely determining that the case is abnormal. If a case is flagged as abnormal, it will usually proceed to a more costly diagnostics, but a case flagged as normal will not be further investigated. Thus a case falsely determined as normal will remain with the wrong diagnostics. In the case of a patient, the wrong diagnostics will cause the patient or the physician to believe he does not have the condition, and leave it untreated, which may have serious consequences. If the computer diagnostic flags the patient as having the disease, a more complex (and potentially more costly) procedure will be performed, which will likely determine that the patient does not have the condition. We will call the situation of wrongly flagging a case as normal (higher cost) as a *false positive*[1], also called in the literature as a *false alarm*. We will also say that the positive class is more *sensitive*. Formally, for a given classifier f and a new data $\mathbf{x}_i \in \mathbb{R}^d$ where d is the feature space dimension, the data class is denoted by y_i while $f(\mathbf{x}_i)$ denotes the predicted class of $\mathbf{x}_i$ by f. A false positive is a data point $\mathbf{x}_i$ such that $f(\mathbf{x}_i) = +1$, but $y_i = -1$. A false negative is a point $\mathbf{x}_j$ such that $f(\mathbf{x}_j) = -1$, but $y_j = +1$. The **false positive rate** of the classifier f is then:

$$\mathrm{FP}(f) = \frac{|\{\mathbf{x}_i \mid f(\mathbf{x}_i) = +1 \text{ and } y_i = -1\}|}{|\{\mathbf{x}_i \mid y_i = -1\}|}.$$

Similarly, the false negative rate $\mathrm{FN}(f)$ is the ratio of the number of false negatives divided by the number of positive cases.

Support Vector Machine (SVM) is a powerful algorithm for binary classification, which is known by its ability to handle high dimensional data efficiently. It has been widely used in many applications providing state-of-the-art accuracy to many classification problems. However, in the context of low false positive learning, a drawback of the traditional support vector classifier formulation is that it penalizes errors in both classes equally, and offers no assurance regarding the false positive rate. Thus, in problems such as spam filtering, for which a false positive rate constraint must be complied, the traditional SVM can be useless.

Nevertheless, observing the aforementioned limitations, some extensions to SVM have been proposed aiming at controling errors in an asymmetric way. The most common techniques for that are the Bias-Shifting (BS) and the Cost-Sensitive SVM (CS-SVM). While the former tries to control the false alarms by shifting the SVM's decision boundary toward the sensitive class (positive in our case), the latter tries to adjust the SVM's formulation in order to make

☆ This paper has been recommended for acceptance by Dacheng Tao.

* Corresponding author.
E-mail addresses: daniel.moraes@students.ic.unicamp.br (D. Moraes), wainer@ic.unicamp.br (J. Wainer), anderson.rocha@ic.unicamp.br (A. Rocha).

[1] This is maybe confusing because the medical literature treats the normal case as *negative* and thus in the medical literature one would like to limit the false negative to a very low value. We will follow the computer science literature that prefers to call the costly mistake as false positive.

http://dx.doi.org/10.1016/j.jvcir.2016.03.007
1047-3203/

misclassifications from the sensitive class more costly than the other class. The CS-SVM offers state-of-the-art results on the problem of low false positive classification, and several studies have been made in this direction. The BS, on the other hand, gives results that are close to the CS-SVM and is as efficient as the traditional SVM.

In this work, we propose the Risk Area SVM (RA-SVM), a novel method to efficiently solve the low false positive classification problem. It is an extension of the traditional support vector machine classifier and is able to control the false positive rate given a user-specified maximum allowed threshold. The RA-SVM selects a sensitive region close to the SVM's decision boundary with a high incidence of false positives. Within that region, which we call *risk area*, the decision to classify a sample as positive is based on inspecting its k nearest neighbors (k-NN), and a new data inside this region will be classified as positive only if all its k-nearest neighbors are also positive.

The idea of combining k-NN within a region around the SVM's decision boundary in order to control false positives was first introduced in [1] to solve a problem of automatic triage. Our work extends upon and further explore those ideas. Some of the main advances in our work herein are:

- We develop a more effective technique for selecting the SVM's sensitive region;
- The k-NN classifier now works on the SVM's high-dimensional feature space (using the same kernel), instead of the original feature space of the data allowing a much better data discrimination;
- We proposed a novel technique that runs up to five times faster than the standard method and offers similar quality performance;
- We evaluated the proposed methods (and the major techniques in the literature) on several standard benchmarks, from different sources and sizes, and on different scenarios (e.g., unbalanced data).

The requirement of keeping the false positive rate bounded below a certain level, while minimizing the false negative rate is also called the *Neyman–Pearson* classification paradigm [2,3]. The requirement can also be stated as maximizing the accuracy (correct predictions), while keeping the false positive rate bounded. Thus, given a user specified threshold α, our objective is to:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \text{FN}\ (f), \\ \text{subject to} \quad & \text{FP}\ (f) \leqslant \alpha. \end{aligned} \tag{1}$$

First, we briefly discusses SVM and its features. After that, we discusses alternatives to the problem of SVM based Neyman–Pearson classification. Next we introduce our methods, with four alternatives to define the risk area and solve the false positive learning problem. We then show the evaluation methodology used to validate the proposed methods and compare them with alternative algorithms in the literature. Finally, we concludes the paper and points out some possible future research opportunities.

2. Support vector machines

In a typical classification setting, we are given a sample of training vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, each belonging to one of two classes, indicated by the respective labels $y_1, \ldots, y_n \in \{-1, +1\}$. The task is then to find a function $f : \mathbb{R}^d \to \{-1, +1\}$ that accurately predicts the label when presented with a new sample [4].

Support Vector Machines (SVM) are among the most effective methods for binary classification [4]. The idea is to find the maximum-margin hyperplane $(\mathbf{w}, b)$ in a high-dimensional space $\mathcal{H}$ that accurately separates the positive instances from the negative ones. Given a separating hyperplane $(\mathbf{w}, b)$, the support vector classifier is given by

$$f_{\mathbf{w},b}(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \Psi(\mathbf{x}) \rangle + b),$$

where $\Psi : \mathbb{R}^d \to \mathcal{H}$ is a kernel function that transforms the input data onto a high-dimensional feature space, and b is a parameter that indicates the offset of $\mathbf{w}$ with respect to the origin of $\mathcal{H}$. The transformation Ψ is implicitly defined by a *kernel* function, so that $\langle \Psi(a), \Psi(b) \rangle = \mathcal{K}(a, b)$. In this work, we used the Gaussian kernel (RBF), which is a good default kernel when we have no prior knowledge on how to represent the data under analysis. The RBF kernel is given as follows:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right),$$

where $\gamma > 0$ is the kernel parameter.

There are many different formulations for SVM. In this work, we consider the standard SVM formulation, known as C-SVM. The C-SVM uses the hinge-loss (L1) as the loss function:

$$\ell(f(\mathbf{x}_i), y_i) = \max(0, 1 - y_i f(\mathbf{x}_i)) = \xi_i. \tag{2}$$

The values ξ_i are called slack variables.

On C-SVM, the algorithm finds $\mathbf{w}$ and b by solving the following quadratic problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i. \\ \text{subject to} \quad & y_i(\langle \Psi(\mathbf{x}_i), \mathbf{w} \rangle + b) \geqslant 1 - \xi_i, \\ & \xi_i \geqslant 0, \end{aligned} \tag{3}$$

where $C \geqslant 0$ is a parameter that balances the amount misclassification and the size of the margin.

A common alternative to the hinge-loss function on SVMs is the L2-loss:

$$\ell(f(\mathbf{x}_i), y_i) = (f(\mathbf{x}_i) - y_i)^2. \tag{4}$$

SVMs that use the L2-loss are called Least-Square SVM (LS-SVM) [5].

3. Related work

There are many practical applications that require the classifier to produce a very low false positive rate. Therefore, several studies have been conducted to develop classifiers in this sense, which include techniques based on Naïve Bayes [6,7], boosting [8–10], data compression [11], neural networks [12], ensemble learning [13], partial least squares [14], and cascade of classifiers [15,16].

Support vectors machines is one of the most powerful algorithms for binary classification. However, since its standard formulation penalizes errors in both classes equally, it does not offer assurance regarding the false positive rate. This and other limitations led to the emergence of many different formulations, each of them generally focused on a particular problem.

A common method for controlling the false positive rate on SVMs is known as Bias-Shifting (BS) and was proposed by Shawe-Taylor and Karakoulas [17]. The method shifts the decision boundary toward the sensitive class by simply adjusting the threshold parameter b. This idea was motivated after Shawe-Taylor [18] showed that the distance of a data point to the decision boundary is related to its probability of misclassification. The BS technique is usually optimized by selecting the threshold t that minimizes $\text{FN}(f)$ while ensuring that $\text{FP}(f) \leqslant \alpha$ (on the training data), where α is the user-specified maximum