Image and Vision Computing 32 (2014) 550–557

Contents lists available at ScienceDirect

Image and Vision Computing

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

Exploiting Universum data in AdaBoost using gradient descent☆

Jingsong Xu [a], Qiang Wu [b], Jian Zhang [c,*], Zhenmin Tang [a]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[b] School of Computing and Communications, University of Technology, Sydney, NSW 2007, Australia
[c] Advanced Analytics Institute and School of Software, University of Technology, Sydney, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 1 August 2012
Received in revised form 15 February 2014
Accepted 25 April 2014
Available online 28 May 2014

Keywords:
AdaBoost
Gradient boost
𝔘AdaBoost
Universum
𝔘-SVM

ABSTRACT

Recently, Universum data that does not belong to any class of the training data, has been applied for training better classifiers. In this paper, we address a novel boosting algorithm called 𝔘AdaBoost that can improve the classification performance of AdaBoost with Universum data. 𝔘AdaBoost chooses a function by minimizing the loss for labeled data and Universum data. The cost function is minimized by a greedy, stagewise, functional gradient procedure. Each training stage of 𝔘AdaBoost is fast and efficient. The standard AdaBoost weights labeled samples during training iterations while 𝔘AdaBoost gives an explicit weighting scheme for Universum samples as well. In addition, this paper describes the practical conditions for the effectiveness of Universum learning. These conditions are based on the analysis of the distribution of ensemble predictions over training samples. Experiments on handwritten digits classification and gender classification problems are presented. As exhibited by our experimental results, the proposed method can obtain superior performances over the standard AdaBoost by selecting proper Universum data.

1. Introduction

Conventional machine learning algorithms take labeled data, unlabeled data or both of them for learning. Vapnik [1] proposed the third kind of data: Universum data. The Universum data contains the data that belongs to the same domain as the classification problem but it does not belong to any class of the problem. For example, in handwritten digits recognition problems, if the samples of handwritten digits '5' and '8' are prepared for learning, then other handwritten digit samples can be naturally treated as Universum data since they belong to the same domain but cannot be assigned to any of the two classes.

It is a common case that large labeled training data is included in order to obtain good quality of training. However, it is quite costly or sometime even impossible to have very large training data. To deal with such problem, semi-supervised learning is a common option when unlabeled data is available since unlabeled data helps model data distribution of the whole data. On the other hand, without unlabeled data, Universum data is still able to provide the supports to maintain the training quality with relatively small labeled data set. The reason is Universum data can be generated through a lot of ways from labeled data only [2] (mentioned later). Moreover, Universum data can carry additional valuable prior information from the domain of the problem into the training process. To the best of our knowledge, there is no comparison between semi-supervened learning and Universum based learning. But in our opinion, Universum based learning can better model the whole data set since Universum data stays in the same domain of learning problem with which we are concerned [1] while the unlabeled data may be too general and stay outside of the domain. In terms of data acquisition, Universum data can be obtained more widely.

Vapnik first discussed transductive learning with Universum since transductive learning provides prior information to estimate the upper bound of inductive inference [1]. However, the classifier trained by inductive learning is more practical to classify unknown data. Weston et al. [2] proposed an inductive algorithm, Universum Support Vector Machines (𝔘-SVM). 𝔘-SVM contains an additional regularization term for Universum data in addition to conventional SVM. The regularization is based on this assumption: the decision values on the Universum data should be close to zero. That is Universum data should fall inside the margin of the classifier since it does not belong to any class. The Universum samples which meet such assumption are called contradictions because the goal of learning is putting labeled data outside of the margin. Thus the margin should contain more Universum data to achieve better learning performance. More Universum data means more contradictions. Therefore the learning criterion for Universum based learning is called Maximal Contradiction on Universum (MCU) [1]. Two learning problems: common semi-supervised and training based on Universum are demonstrated in Fig. 1. The unlabeled data should be away from the margin like labeled data while Universum data should fall between margins.

Sinz et al. [3] analyzed the 𝔘-SVM for inference and they showed that 𝔘-SVM would give the hyperplane which had its normal lying in the

☆ This paper has been recommended for acceptance by Jan-Michael Frahm.
* Corresponding author.
E-mail addresses: xjsxujingsong@gmail.com (J. Xu), Qiang.Wu@uts.edu.au (Q. Wu), Jian.Zhang@uts.edu.au (J. Zhang), tzm.cs@mail.njust.edu.cn (Z. Tang).

http://dx.doi.org/10.1016/j.imavis.2014.04.009
0262-8856/

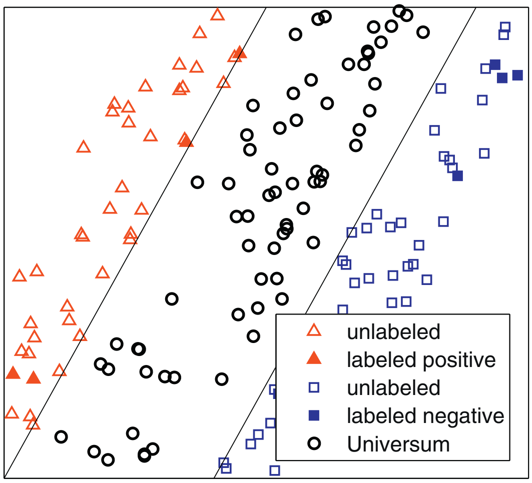

Fig. 1. The comparison of semi-supervised learning problem and Universum based learning problem. The former takes labeled positive, labeled negative and unlabeled samples for learning, and the latter learns models from labeled positive, labeled negative and Universum samples.

orthogonal to the principal directions of Universum data. They also discussed the connection of least squared version of 𝔘-SVM with Fisher discriminant analysis and oriented principal component analysis. They showed that 𝔘-SVM outperformed SVM with carefully selected Universum data. In addition to SVM, Universum data has also been extended to other learning problems, such as semi-supervised learning [4], linear discriminant analysis [5], twin support vector machine [6], cost-sensitive learning [7], linear programming [8], and domain adaptation [9]. In terms of application, besides the handwritten digits recognition problem mentioned above, it is also applied into medical imaging [10], document clustering [11], pose recognition [12,13], etc.

Universum data is always obtained from the domain of the classification problem mentioned above. Weston et al. [2] proposed four kinds of Universum data: random noise, the rest of the training data (e.g. the other digits in the handwritten digits recognition problem), artificial data from the same distribution of training data and random average of training data. Bai & Cherkassky [14] applied Universum data into gender classification and they took three kinds of Universum data: random average, empirical distribution and animal faces. In the field of Universum data selection, Sinz et al. [3] suggested that a good Universum set should contain invariant directions and be positioned "in between" the two classes of the classification problem. Chen & Zhang [15] proposed a guided formulation to pick the informative ones, i.e., in-between Universum (IBU) samples.

Boosting family contains a series of well-known algorithms with a large number of applications. Motivated by the success of 𝔘-SVM, Shen et al. [16] proposed 𝔘Boost by adding Universum data to boosting algorithms and showed that they can benefit from Universum data as SVM did. 𝔘Boost is derived from AdaBoost-CG [17] which is another view of boosting. Compared with AdaBoost-CG, AdaBoost is a stagewise method [18] which is more general and popular. The whole training procedure of AdaBoost is also much faster. Although Universum data has shown its power on 𝔘-SVM [2] and 𝔘Boost [16], to our knowledge, its importance on AdaBoost has not been evaluated.

In this paper, we propose a new boosting algorithm called 𝔘AdaBoost to improve the classification performance of AdaBoost with the help of Universum data. The learning is not straight forward since Universum data belongs to neither positive nor negative data. Stagewise AdaBoost keeps the pre-selected weak classifiers unchanged in the following training. It pays more attention on misclassified samples in the next training iteration. The weights for training samples and coefficients for the pre-selected weak classifiers are obtained according to gradient descent. Involving Universum data into AdaBoost framework needs to take these properties of AdaBoost into account. Instead of 𝔘AdaBoost, 𝔘Boost takes Universum as a conventional convex optimization problem and solves it by column generation as AdaBoost-CG does.

To tackle the above challenges, we propose explicit weighting schemes for both labeled data and Universum data which are both involved in AdaBoost training procedure. The rationale of updating weights in common AdaBoost training is to enforce the training to focus on hard samples. In this paper, this rationale is revisited and further extended on Universum data in the proposed 𝔘AdaBoost.

The major contributions of this paper are as follows:

1) Given Universum data, a new 𝔘AdaBoost learning based on the framework of AdaBoost is proposed. We propose 𝔘AdaBoost using the same functional gradient descent as AdaBoost. By taking advantage of AdaBoost, 𝔘AdaBoost is much easier and more practical to apply than 𝔘Boost [16] since 𝔘AdaBoost only needs one parameter to tune.
2) The whole training procedure of 𝔘AdaBoost is efficient. The time cost for 𝔘AdaBoost is less than 𝔘Boost. Our experimental results demonstrate such improvement.
3) 𝔘AdaBoost provides a better framework for investigating the benefits of Universum data in boosting approaches. It is known that AdaBoost is a popular algorithm in boosting algorithm family. In recent years, researches have contributed significant efforts to investigate AdaBoost in order to improve its performance. To our best knowledge, 𝔘AdaBoost is the only framework using the same approach (i.e. stagewise) as AdaBoost and integrating Universum data, so the performance evaluation on integrating Universum data into AdaBoost is more precise and convincing. In contrast, 𝔘Boost follows column generation approach [17] which is different from AdaBoost so we cannot use 𝔘Boost framework in evaluating the benefits of Universum data to AdaBoost.
4) Also, in this paper, we discuss a method for selecting effective and informative Universum data in order to better take the advantages of Universum data in AdaBoost framework. This will benefit several applications in computer vision area.

The paper is structured as follows. In Section 2, we discuss the related work about 𝔘-SVM, 𝔘Boost and the motivation to 𝔘AdaBoost. In Section 3, we propose the novel boosting formulation 𝔘AdaBoost based on the Universum data and compare it with semi-supervised boosting, AdaBoost and 𝔘Boost. In Section 4, we analyze the practical conditions for 𝔘AdaBoost. In Section 5, the performance of our model will be demonstrated with several public data sets. In Section 6, we conclude the paper.

2. Related works and motivation

2.1. Notations

In this paper, our focus is only on binary classification problems, while our method can be extended to the multi-class scenario. Let $\mathbf{X}_{\mathfrak{L}} = \{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_m,y_m)\}$ be the set of m labeled examples, where $y_i \in \{-1,1\}$ is the class label. Let $\mathbf{X}_{\mathfrak{U}} = \{\mathbf{x}_1^*,\mathbf{x}_2^*,\ldots,\mathbf{x}_n^*\}$ represent the Universum data with n samples. w_i and w_j^* represent the weights of the labeled sample $(\mathbf{x}_i, y_i)$ and $\mathbf{x}_j^*$ during boost training phase respectively.

2.2. 𝔘-SVM and 𝔘Boost

Weston et al. [2] proposed 𝔘-SVM and treated it as an inductive learning problem. The Universum examples are considered to be close to the separating hyperplane selected by SVM. The optimization objective should minimize the cumulative loss on the Universum examples. Given the Hinge loss $H_a[t] = \max\{0, a - t\}$ for the standard SVM