Information Fusion 14 (2013) 208–219

Contents lists available at SciVerse ScienceDirect

Information Fusion

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

Improving Logitboost with prior knowledge

Takafumi Kanamori [a,*], Takashi Takenouchi [b]

[a] Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan
[b] Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma City, Nara, Japan

ARTICLE INFO

Article history:
Received 26 February 2011
Received in revised form 19 August 2011
Accepted 7 November 2011
Available online 18 November 2011

Keywords:
Boosting
Prior knowledge
Mixture distribution
Multi-class

ABSTRACT

The purpose of this study is to incorporate prior knowledge into a boosting algorithm. Existing approaches require additional samples that represent the prior knowledge. Moreover, in order to adjust the balance between the information in training samples and the prior knowledge in the data domain, one needs to repeat the boosting algorithm with a different regularization parameter. These properties lead to costly computation. In this paper, we propose a boosting algorithm with prior knowledge that avoids computational issues. In our method, the mixture distribution of the estimator and prior knowledge is considered. We describe numerical experiments showing the effectiveness of our approach.

1. Introduction

Since the 1990s, statistical learning methods have been highly developed. Boosting [11] which is a learning algorithm for a classification task is one of the most significant achievements in the community of machine learning. By applying boosting to a so-called weak learner such as decision trees, one can obtain accurate classifiers or decision functions for classification tasks. In other words, boosting is regarded as a meta-learning algorithm. Adaboost [11] is one of the most efficient implementations of a boosting algorithm. Friedman et al. [12] have pointed out that the boosting algorithm is derived from a coordinate descent method [21] for loss functions. Following the approach indicated by Friedman et al. [12], one can derive a boosting algorithm from any loss function under mild assumptions. The negative log-likelihood loss function leads to the Logitboost algorithm [11], which is regarded as the maximum likelihood estimator implemented by the boosting. To improve the generalization ability of Adaboost, the Modest Adaboost has been proposed [28]. As the other approach to improve the generalization ability, FloatBoost [20] has been proposed, in which the least significant weak learner is removed. In addition, for the semi-supervised learning, SemiBoost [22] and MixtBoost [13] have been proposed. The generalization of the boosting algorithm has been studied in [5,7,23] and other works referred to in these articles. Theoretical analysis of the generalization ability of the boosting algorithm is also investigated in [4,25,29,30]. In this paper, we mainly use Logitboost algorithm.

In many learning problems, some information on the data domain may be available in addition to the observed training samples. An example is the prior distribution in Bayes inference. In the standard form, however, boosting does not allow for the direct incorporation of prior knowledge. Schapire et al. [26] have studied a boosting algorithm with prior knowledge. In their work, prior knowledge is incorporated into the loss function of the boosting algorithm. Then, the standard boosting algorithm is applied to the loss function involving prior knowledge. Boosting with prior knowledge has been applied to call classification problems in which human-crafted knowledge is available to compensate for the lack of data. Schapire et al. [26] reported that boosting with prior knowledge consistently improves the naive boosting method.

As for the other approach, recursive Bayes learning is also applicable to incorporate prior knowledge [8]. When a linear model is used under a normal distribution, the computation of recursive Bayes learning is efficiently conducted for the online setup. Otherwise, the naive application of Bayes learning entails high computational cost in terms of integrating the probability distribution over the parameter space. In Bayesian inference, some assumption on prior knowledge is required in order to reduce the computational cost. In variational Bayes method [3], the conditional independence among parameters is an important assumption. Hence, one needs to modify the provided prior knowledge in order to apply Bayes methods to large classification or regression tasks. On the other hand, in boosting with prior knowledge [26], one does not need to modify the given prior knowledge. The Bayes methods, however, can deal with many kind of statistical models, while the model used in boosting is restricted to the linear combination of basis functions. We need to select appropriate learning methods depending on the purpose of the data analysis.

[*] Corresponding author.
E-mail addresses: kanamori@is.nagoya-u.ac.jp (T. Kanamori), ttakashi@is.naist.jp (T. Takenouchi).

1566-2535/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2011.11.004

In this paper, we propose a new learning method for incorporating prior knowledge into boosting. In this framework, a significant issue is to estimate the value of the regularization parameter that determines the balance between the information in the training data and the prior knowledge of the data domain. Schapire et al. [26] empirically determine the regularization parameter because the validation method incurs high computational cost. By using our approach, the computation cost of estimation is reduced compared with [26], because in our algorithm, the learning phase by boosting and the incorporation of prior knowledge are separated. This is the main advantage of our method. Another property of our method is the application of error correcting output coding (ECOC) in multiclass classification problems. In the learning method in Schapire et al. [26], the binary reduction technique is used in order to transform a multiclass label to a binary label. The training data produced by the binary reduction technique are typically unbalanced when the number of class labels is large. The unbalanced training data often prevent the learning algorithm from reducing the training error rate. Moreover, the ECOC method does not need additional training samples, while in [26] additional training samples with weights are introduced to represent the prior knowledge. Our approach has a great advantage from the viewpoint of computational cost in comparison with existing methods. We show that the ECOC method improves the learning efficiency in the training phase.

We define some notations and fix the problem setup. Assume that we are given the training samples,

$$(x_1,y_1),\dots,(x_n,y_n)\in\mathcal{X}\times\mathcal{Y}, \tag{1}$$

where $\mathcal{X}$ is the input domain and $\mathcal{Y}:=\{1,2,\dots,K\}$ is the set of class labels. In binary classification problems, the set of labels is defined as $\mathcal{Y}=\{-1,+1\}$ instead of {1,2}. The input x_i could have some information on the features of the object i, and the label denotes the category to which the input is classified. For example, x_i denotes the ith document in the text data sets and y_i denotes the tag or the category of the text x_i. The main task of learning is to predict the label y for the given input x that may not appear in the training data. The samples in the prediction phase are called test samples or test data. We assume that all of the training and test samples are selected independently and identically from some probability distribution $p(x,y)$ on $\mathcal{X}\times\mathcal{Y}$. Let $p(x)$ be the marginal probability distribution on $\mathcal{X}$ and $p(y|x)$ be the conditional probability of the label $y\in\mathcal{Y}$ given $x\in\mathcal{X}$. For the prediction of unseen labels, the so-called decision function is often used. The decision function is a real-valued function $f(x,y)$ defined on $\mathcal{X}\times\mathcal{Y}$, and the label of x is predicted by the label $\hat{y}$ such that

$$\hat{y}=\mathrm{argmax}_{y\in\mathcal{Y}}f(x,y), \tag{2}$$

where the tie is broken in some way, e.g., by a random choice among the maximizers. Building a decision function that achieves high prediction accuracy is the main task of classification problems.

2. Boosting for classification problems

In this section, we introduce the boosting algorithm. In this paper, we focus on the boosting method for classification problems. There are many methods for predicting labels based on observations, for example, the nearest-neighbor method and the support vector machine. See [15] for details of each method. The main concept of boosting is to boost the so-called weak learner or weak classifier to a highly accurate classifier. In other words, boosting is regarded as a meta-learning algorithm. Standard learning algorithms, such as decision stumps [15] or rpart [27] are available as the weak learner to build the decision function $f(x,y)$. Note that we use the weak learner as a black-box subroutine.

Friedman et al. [12] have pointed out that the boosting algorithm is derived from the coordinate descent method [21]. Their work has also clarified that boosting has the potential to estimate the conditional probability $p(y|x)$. Indeed, there is a correspondence between decision functions and conditional probabilities. Hence, once the decision function is estimated, one can obtain an estimator of the conditional probability $p(y|x)$. This relation connects the boosting and the standard statistical tools such as the maximum likelihood estimator.

We illustrate the boosting algorithm for the estimation of the decision function $f(x,y)$. Here $f(x,y)$ is supposed to have a particular form, namely, the linear sum of the base functions $h(x,y)\in\mathcal{H}$, where $\mathcal{H}$ is a set of base functions. For example, $\mathcal{H}$ consists of all the decision functions obtained from the decision tree algorithm. In practice, $\mathcal{H}$ is implicitly defined, that is, it may be all possible outputs of a learning algorithm. Suppose that f is represented as the linear sum of base functions such that

$$f(x,y)=\sum_{t=1}^{T}\alpha_t h_t(x,y),\quad \alpha_t\in\mathbb{R},\ \ h_t\in\mathcal{H}(t=1,\dots,T). \tag{3}$$

In the following, Logitboost algorithm [5,12] is introduced. Though Adaboost [11] is the most popular boosting algorithm, we use Logitboost algorithm to deal with the noisy data. We refer Section 10.6 in [15] to explain the superiority of Logitboost in the estimation based on noisy data:

> At any point in the training process the exponential criterion concentrates much more influence on observations with large negative margins. Binomial deviance concentrates relatively less influence on such observations, more evenly spreading the influence among all of the data. It is therefore far more robust in noisy settings where the Bayes error rate is not close to zero.

Large weight on a small portion of samples will make the estimation unstable, and thus, Logitboost will be superior to Adaboost as explained in the above quotation.

Let us define $L(f)$ as the loss function

$$L(f)=\sum_{i=1}^{n}\left\{-f(x_i,y_i)+\log\sum_{z\in\mathcal{Y}}\exp(f(x_i,z))\right\}. \tag{4}$$

Eq. (4) is derived from the statistical model of conditional probability

$$q(y|x;f)=\frac{\exp\{f(x,y)\}}{\sum_{z\in\mathcal{Y}}\exp\{f(x,z)\}}. \tag{5}$$

Indeed, the loss function (4) is represented as

$$L(f)=-\sum_{i=1}^{n}\log q(y_i|x_i;f), \tag{6}$$

which is the negative log-likelihood of the statistical model (5).

We are concerned with finding the decision function that minimizes $L(f)$. For this purpose, one can use the coordinate descent method to update the decision function f to $f+\alpha h,(\alpha\in\mathbb{R},h\in\mathcal{H})$, such that $L(f+\alpha h)\leqslant L(f)$ holds. The Taylor expansion of $L(f+\alpha h)$ abound $\alpha = 0$ yields

$$L(f+\alpha h)=L(f)+\alpha\cdot\mathcal{E}(h;f)+O(\alpha^2), \tag{7}$$

in which $\mathcal{E}(h;f)$ is defined as the derivative of $L(f)$ to the direction of h,

$$\mathcal{E}(h;f)=\frac{\partial}{\partial\alpha}L(f+\alpha h)|_{\alpha=0}. \tag{8}$$

For a base function h such that $\mathcal{E}(h;f)<0$, there exists a positive coefficient $\alpha > 0$ satisfying the inequality $L(f+\alpha\ h) < L(f)$. Boosting