2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

An Adaptive Multiclass Boosting Algorithm for Classification

Shixun Wang, Peng Pan*, Yansheng Lu
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China, 430074
wsxun@hust.edu.cn, {panpeng, lys}@mail.hust.edu.cn

***Abstract*—A large number of practical domains, such as scene classification and object recognition, have involved more than two classes. Therefore, how to directly conduct multiclass classification is being an important problem. Although some multiclass boosting methods have been proposed to deal with the problem, the combinations of weak learners are confined to linear operation, namely weighted sum. In this paper, we present a novel large-margin loss function to directly design multiclass classifier. The resulting risk, which guarantees Bayes consistency and global optimization, is minimized by gradient descent or Newton method in a multidimensional functional space. At every iteration, the proposed boosting algorithm adds the best weak learner to the current ensemble according to the corresponding operation that can be sum or Hadamard product. This process grown in an adaptive manner can create the sum of Hadamard products of weak learners, leading to a sophisticated nonlinear combination. Extensive experiments on a number of UCI datasets show that the performance of our method consistently outperforms those of previous multiclass boosting approaches for classification.**

Keywords—multiclass boosting; classification; loss function; nonlinear combination; probabilistic outputs

I. Introduction

In recent years, the solutions to various problems in machine learning and computer vision have involved the design of a classifier. One reliable tool for this design is boosting, which is a powerful and effective technique for combining multiple weak learners to produce a strong ensemble. The weak learners are trained in sequence, and each one is learnt using a weighted data set where the data points are reweighed in the light of the performance of the previous learners. Thus, the next weak learner may focus more attentions on the data points that are misclassified. In the literature, a lot of boosting algorithms have been proposed to learn the classifier, including AdaBoost [1], LogitBoost [2], GradBoost [3], TaylorBoost [4] and TangentBoost [5]. Most of them are primarily designed for binary problems.

However, more and more practical applications about multiclass classification have appeared in the daily life of people. Such applications contain handwriting recognition, medical diagnosis, 3D object recognition and wine quality grading. In these cases, each data point belongs to one of multiple different categories. For instance, on the basis of physicochemical tests, the wine quality can be classified into “poor”, “middle” and “excellent”. Unlike the traditional binary boosting methods, the multiclass boosting approaches can effectively deal with this case.

In general, there are two main strategies for designing multiclass boosting algorithms, namely indirect and direct strategies. The first is to decompose the multiclass problem into several independent binary sub-problems which can be efficiently solved using binary classifiers. Popular methods in this type include one-versus-all and one-versus-one. In spite of their gains in some cases, these schemes can have several disadvantages, such as imbalanced data distribute, increased complexity, getting an optimal joint classifier cannot be guaranteed due to ambiguous classification, and the scores generated by different classification problems are on different scales. As for the second strategy, a set of codewords usually plays an important role when multiclass weak learners such as trees are directly boosted. Some methods in this class require specific weak learners which have high probability to cause the phenomenon of over-fitting, or cannot guarantee a large multiclass margin. As a result, multiclass boosting is still worth researching.

If the classifier which minimizes the risk associated with a loss function converges asymptotically to the Bayes decision rule, then this loss function has a property of Bayes consistency. Margin enforcing means penalizing data points which are correctly classified but near the boundary. The utility of such loss function can perform well on both the small and large datasets. Nevertheless, the weakness of weak learners can limit the classification performance in some complex cases. An effective solution is to combine weak learners in a nonlinear rather than linear way, using at least two predefined operations such as sum and Hadamard product. To the best of our knowledge, such research has not been found for multiclass boosting classification in the previous papers.

In this paper, the classification problem on which we focus is directly addressed by multiclass boosting. We present a new loss function which has two main properties, namely Bayes consistency and margin enforcing. A concept framework for the design of multiclass boosting algorithm, which learns more sophisticated combinations of multiclass weak learners, is

*Corresponding Author

introduced. This concept framework is generalized from binary to multiclass boosting by Hadamard product operation. When the risk is minimized in a multidimensional functional space, an optimization strategy that may be gradient descent or Newton method is considered to choose the best weak learner. The resulting boosting algorithm can create a final classifier which learns the sum of Hadamard products of weak learners. In particular, the two operations, sum and Hadamard product, can be selected in an adaptive way. Experimental results on several UCI datasets show that the proposed algorithm performs better than the previous boosting algorithms for multiclass classification.

As will be detailed in Section II, the existing boosting approaches for multiclass classification usually focus on a linear combination of weak learners, which may be insufficient to produce an accurate classifier. However, our method considers more complex classifier structure. In summary, the major contribution of this paper is divided into the following two parts.

- A new loss function for multiclass boosting is proposed, which gives rise to a convex risk. As long as the functional space is a convex set, then the discussed problem becomes a convex optimization which may be convergent to the global optimum.
- A corresponding algorithm based on the proposed loss is implemented, which also has probabilistic outputs so that some further applications can use them.

The rest of this paper is organized as follows. After a review of related works in Section II, we present several preliminaries in Section III. Section IV describes our adaptive multiclass boosting algorithm, which can build a fairly complex classifier. The experimental results and analyses are listed in Section V. Finally, the concluding discussions are provided in Section VI.

II. Related Works

The beginning of boosting algorithm is AdaBoost [1], which has a gradient descent procedure for minimizing the risk associated with an exponential loss. In [2], LogitBoost which carries out Newton method is proposed to minimize the logistic risk. Based on Taylor series expansion of the risk, TaylorBoost [4] may be applied to any loss function, which results in a family of boosting algorithms of either first or second order. It is shown that GradBoost [3] and LogitBoost are the special cases of TaylorBoost. However, these methods are primarily presented for two classes, and limited because of the linear combination of weak learners.

Recently, many efforts [6]–[9] have been devoted to create new combinations of weak learners. The goal of [6] is to construct products of several decision stumps, which only depends on a single operation. Gated classifiers [7] having the ability to handle intra-class variation, are built by combining sets of previously selected weak learners utilizing the network of logical gates (and, or). The technique in [8] combines the binary quantized feature responses with sequential forward selection to design new weak learners. In [9], the boosting algorithms for feature extraction are derived, and grow a final predictor by selecting the most likely operation from a pair of predefined operations which could be sum and product. However, these approaches mainly aim at solving a binary classification problem.

In parallel, a lot of efforts [10]–[16] have appeared in multiclass boosting. The ECOC method in [10] designs a coding matrix in which each row corresponds to a certain class and each column is used to train a binary classifier. The idea of ECOC is also generalized in [11], where the elements of coding matrix are taken from $\{-1, 0, +1\}$. The weight distribution on instances and the label weighting function are simultaneously manipulated in AdaBoost-M2 [1], where the goal of weak learner is to minimize a pseudo-loss. RUSBoost [12] introducing random undersampling into AdaBoost algorithm is a variant of the AdaBoost-M2 procedure. In AdaBoost-MH [13], each example is augmented with a feature that identifies a class, and receives a binary label depending on whether the example belongs to this class. However, these approaches do not directly boost multiclass classifier.

SAMME [14] directly extends the AdaBoost algorithm to the multiclass case without reducing it to multiple binary problems, and is equivalent to a forward stagewise additive modeling algorithm that minimizes a multiclass exponential loss function. In [15], a general framework is created, where the optimal requirements on the weak learner are identified. However, the corresponding boosting method is not shown to be Bayes consistency. An optimal set of codewords and a margin enforcing loss for multiclass boosting are introduced in [16], where the related risk is only minimized by gradient descent. Two boosting algorithms are proposed, which are different in terms of the strategy used to update the predictor. Although these methods are directly designed for multiclass problem, it should be noted that the combination of multiclass weak learners is linear.

III. Multiclass Setting and Loss Function

In this section, we present the multiclass setting and a new loss function for multiclass boosting. The boldface letters are used to denote vectors.

A. Multiclass Setting

Let the labeled dataset be denoted as $(X, C) = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_i, c_i), \ldots, (\mathbf{x}_N, c_N)\}$, in which $\mathbf{x}_i \in \mathbb{R}^d$, $c_i \in \{1, \ldots, K\}$ denotes a class label, and the data are independently and identically distributed samples from an unknown probability distribution. Usually, the objective is to learn an optimal classification rule $F(\mathbf{x})$: $X \rightarrow \{1, \ldots, K\}$ from the training data, so that a class label c can be assigned to a new input $\mathbf{x}$. In the sense of minimum error rate, the optimal classifier should be Bayes decision rule

$$F(\mathbf{x}) = \arg\max_k P_{C|X}(c = k \mid \mathbf{x}). \tag{1}$$

Due to the intractability of estimating $P_{C|X}(c \mid \mathbf{x})$, there is a difficulty to implement Bayes decision rule. One possible solution is to resort to boosting approach, which combines some weak classifiers to estimate the Bayes decision rule.

Note that the class labels ±1 play a significant role in a binary classification. Therefore, there is a need to recode the class label c with a multidimensional vector $\mathbf{y}$. For this, a set of K distinct unit codewords $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^K\}$, which are the vertices of a $K-1$ dimensional regular simplex centered at the origin [17], has been used in [16]. So each class label k can be mapped into a codeword $\mathbf{y}^k \in \mathbb{R}^{K-1}$ that identifies the class label.

Let $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^{K-1}$ be a predictor, the margin of $\mathbf{f}(\mathbf{x})$ with respect to class k can be defined as in [16]

$$M\left(\mathbf{f}(\mathbf{x}), \mathbf{y}^k\right) = \left[< \mathbf{f}(\mathbf{x}), \mathbf{y}^k > - \max_{l \neq k} < \mathbf{f}(\mathbf{x}), \mathbf{y}^l >\right] / 2, \quad (2)$$

where $< \cdot, \cdot >$ denotes the standard inner product. To attain the probabilistic results, we may naturally implement the following classifier

$$F(\mathbf{x}) = \arg\max_k \sigma\left(< \mathbf{f}(\mathbf{x}), \mathbf{y}^k >\right) / Z, \quad (3)$$

where $Z = \sum_k \sigma(< \mathbf{f}(\mathbf{x}), \mathbf{y}^k >)$ is a normalization constant, and $\sigma(\cdot)$ a sigmoid function. Since the sigmoid function is monotonically increasing with regard to its argument, maximizing the function itself is equivalent to maximizing its argument. Equivalently, (3) can satisfy the following expression

$$F(\mathbf{x}) = \arg\max_k M\left(\mathbf{f}(\mathbf{x}), \mathbf{y}^k\right), \quad (4)$$

as is easily verified. Therefore, $F(\mathbf{x})$ can find a class which has the largest margin for the predictor $\mathbf{f}(\mathbf{x})$. Then we should search for an optimal predictor which minimizes the classification risk

$$R_M(\mathbf{f}) = E_{X,Y}\left\{L_M[\mathbf{y}, \mathbf{f}(\mathbf{x})]\right\} \approx \sum_{i=1}^{N} L_M[\mathbf{y}_i, \mathbf{f}(\mathbf{x}_i)], \quad (5)$$

where $L_M[\cdot,\cdot]$ is a multiclass loss function. In general, the optimal predictor is approximated as a linear combination of multiclass weak learners. In this case, the risk can be minimized by solving the optimization problem

$$\begin{cases} \min_{\mathbf{f}(\mathbf{x})} & R_M[\mathbf{f}(\mathbf{x})] \\ s.t & \mathbf{f}(\mathbf{x}) \in span(G), \end{cases} \quad (6)$$

where $G = \{\mathbf{g}_1(\mathbf{x}), \ldots, \mathbf{g}_m(\mathbf{x})\}$ is the set of all multiclass weak learners $\mathbf{g}_i(\mathbf{x}) : X \to \mathbb{R}^{K-1}$, and $span(G)$ the functional space of linear combinations of $\mathbf{g}_i(\mathbf{x})$.

B. Loss Function

To derive some useful properties, we rely on the following multiclass loss function

$$L_M[\mathbf{y}, \mathbf{f}(\mathbf{x})] = \sum_{i=1}^{K} \log\left[1 + \exp\left(- < \mathbf{f}(\mathbf{x}), \mathbf{y} - \mathbf{y}^k >\right)\right]. \quad (7)$$

Firstly, as can be seen from Appendix A, the lower bound of (7) is $\log\{2 + 2 \bullet \exp[-2M(\mathbf{f}(\mathbf{x}), \mathbf{y})]\}$. Owing to the monotonic properties of the logarithm and exponential functions, the minimization of empirical risk encourages the predictors which have large margin for every example. Note that when $K = 2$, the loss (7) reduces to

$$L_2[y, f(\mathbf{x})] = \log 2 + \log\left[1 + \exp\left(-2y \bullet f(\mathbf{x})\right)\right], \quad (8)$$

which is equivalent to the logistic loss [2]. Secondly, it is shown in Appendix B that the objective function in (6) is convex. If the functional space is a convex set, then the optimization problem is convex, which can be effectively solved by numerical methods and guarantees convergence to the global optimum. Finally, it can be clearly seen from Appendix C that

$$\arg\max_k \sigma\left(< \mathbf{f}^*(\mathbf{x}), \mathbf{y}^k >\right) = \arg\max_k P_{Y|X}(\mathbf{y}^k \mid \mathbf{x}), \quad (9)$$

which implements Bayes decision rule (1). In addition, once the optimal predictor $\mathbf{f}^*(\mathbf{x})$ is estimated, there has a way to attain the class probability, namely

$$P_{Y|X}(\mathbf{y}^k \mid \mathbf{x}) = \sigma\left(< \mathbf{f}^*(\mathbf{x}), \mathbf{y}^k >\right) / Z. \quad (10)$$

After t boosting iterations, we set the estimate of the optimal predictor to $\mathbf{f}^t(\mathbf{x})$. Around point $\mathbf{f}^t(\mathbf{x})$, the first and second order functional derivatives of the risk $R_M(\mathbf{f}^t + \mathbf{g})$, along the direction of multiclass weak learner $\mathbf{g}(\mathbf{x})$, are

$$\begin{aligned} \delta R_M[\mathbf{f}^t; \mathbf{g}] &= \left.\frac{\partial R_M[\mathbf{f}^t + \varepsilon \mathbf{g}]}{\partial \varepsilon}\right|_{\varepsilon=0} \\ &= \left.\frac{\partial}{\partial \varepsilon} \sum_{i=1}^{N} L_M[\mathbf{y}^i, \mathbf{f}^t(\mathbf{x}_i) + \varepsilon \mathbf{g}(\mathbf{x}_i)]\right|_{\varepsilon=0} \\ &= -\sum_{i=1}^{N} < \mathbf{g}(\mathbf{x}_i), \mathbf{w}_i > \end{aligned} \quad (11)$$

$$\begin{aligned} \delta^2 R_M[\mathbf{f}^t; \mathbf{g}] &= \left.\frac{\partial^2 R_M[\mathbf{f}^t + \varepsilon \mathbf{g}]}{\partial \varepsilon^2}\right|_{\varepsilon=0} \\ &= \left.\frac{\partial^2}{\partial \varepsilon^2} \sum_{i=1}^{N} L_M[\mathbf{y}^i, \mathbf{f}^t(\mathbf{x}_i) + \varepsilon \mathbf{g}(\mathbf{x}_i)]\right|_{\varepsilon=0} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} < \mathbf{g}(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k >^2 \lambda_{i,k} \end{aligned} \quad (12)$$

where

$$\mathbf{w}_i = \sum_{k=1}^{K} (\mathbf{y}_i - \mathbf{y}^k) \frac{\exp\left(-<\mathbf{f}^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)}{1+\exp\left(-<\mathbf{f}^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)} \tag{13}$$

$$\lambda_{i,k} = \frac{\exp\left(-<\mathbf{f}^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)}{\left[1+\exp\left(-<\mathbf{f}^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)\right]^2}. \tag{14}$$

At t + 1 iteration, the direction which brings the greatest decrease of the risk is

$$\mathbf{g}^* = \arg\min_{\mathbf{g}\in G} \delta R_M[\mathbf{f}^t;\mathbf{g}] \tag{15}$$

when gradient descent is used [1][4][16], or

$$\mathbf{g}^* = \arg\max_{\mathbf{g}\in G} \frac{\left\{\delta R_M[\mathbf{f}^t;\mathbf{g}]\right\}^2}{\delta^2 R_M[\mathbf{f}^t;\mathbf{g}]} \tag{16}$$

when Newton method is used [2][4].

IV. Adaptive Multiclass Boosting Algorithm

In this section, a multiclass boosting algorithm that can learn more sophisticated combination of multiclass weak learners is presented in detail.

Let the functional space, which is a nonlinear combination of multiclass weak learners, such as the sum of Hadamard products, be Ω_G

$$\Omega_G = \left\{ \mathbf{h}(\mathbf{x}) \mid \mathbf{h}(\mathbf{x}) = \sum_j \mathbf{g}_{j,1}(\mathbf{x}) \odot \ldots \odot \mathbf{g}_{j,m}(\mathbf{x}),\ \mathbf{g}\in G \right\} \tag{17}$$

where $\odot$ denotes Hadamard product. For any $\mathbf{h}_1, \mathbf{h}_2 \in \Omega_G$, it is easy to discern that $\mathbf{h}_1 + \mathbf{h}_2$ still belongs to Ω_G. Therefore, the functional space Ω_G is a convex set. It follows that the optimization problem

$$\begin{cases} \min_{\mathbf{f}(\mathbf{x})} & R_M[\mathbf{f}(\mathbf{x})] \\ s.t & \mathbf{f}(\mathbf{x}) \in \Omega_G \end{cases} \tag{18}$$

produces a global minimum. Specifically, the predictor after t iterations is assumed to be of the form

$$\mathbf{f}^t(\mathbf{x}) = \sum_{j=1}^{S} \mathbf{p}_j^t(\mathbf{x}), \tag{19}$$

where each term is defined as

$$\mathbf{p}_j^t(\mathbf{x}) = \mathbf{g}_{j,1}(\mathbf{x}) \odot \ldots \odot \mathbf{g}_{j,m^j}(\mathbf{x}),\ m^j \in \mathbb{N}. \tag{20}$$

Similar to [9], there are two types of updates at t+1 iteration, namely additive and Hadamard product. In the additive case, a multiclass weak learner is added to the current predictor $\mathbf{f}^t$, which may be regarded as standard multiclass boosting. From (11) and (12), the optimal weak learner $\mathbf{g}_0^*$ is given by (15) or (16), depending on the choice of optimization strategy. Then the optimal step size can be obtained by

$$\alpha_0^* = \arg\min_{\alpha\in\mathbb{R}} R_M(\mathbf{f}^t + \alpha \mathbf{g}_0^*). \tag{21}$$

Hence, the updated predictor has the following risk

$$R_M^0(\mathbf{f}^{t+1}) = R_M(\mathbf{f}^t + \alpha_0^* \mathbf{g}_0^*). \tag{22}$$

As for the second case, each term in (19) can be updated by a new weak learner, such as $\mathbf{p}_j^{t+1}(\mathbf{x}) = \mathbf{p}_j^t(\mathbf{x}) \odot \mathbf{g}(\mathbf{x})$. Therefore, the updated predictor can be achieved

$$\begin{aligned} \mathbf{f}^{t+1}(\mathbf{x}) &= \sum_{i\neq j} \mathbf{p}_i^t(\mathbf{x}) + \mathbf{p}_j^t(\mathbf{x}) \odot \mathbf{g}(\mathbf{x}) \\ &= \mathbf{Q}_j^t(\mathbf{x}) + \mathbf{p}_j^t(\mathbf{x}) \odot \mathbf{g}(\mathbf{x}). \end{aligned} \tag{23}$$

where $\mathbf{Q}_j^t(\mathbf{x}) = \mathbf{f}^t(\mathbf{x}) - \mathbf{p}_j^t(\mathbf{x})$.

Around point $\mathbf{Q}_j^t(\mathbf{x})$, the first and second order functional derivatives of the risk $R_M(\mathbf{f}^{t+1})$ with respect to the above update in $\mathbf{f}^t(\mathbf{x})$ are

$$\begin{aligned} \delta R_M[\mathbf{f}^t;\mathbf{g},j] &= \left.\frac{\partial R_M[\mathbf{Q}_j^t + \varepsilon \mathbf{p}_j^t \odot \mathbf{g}]}{\partial \varepsilon}\right|_{\varepsilon=0} \\ &= \left.\frac{\partial}{\partial \varepsilon} \sum_{i=1}^{N} L_M[\mathbf{y}^i, \mathbf{Q}_j^t(\mathbf{x}_i) + \varepsilon \mathbf{p}_j^t(\mathbf{x}_i) \odot \mathbf{g}(\mathbf{x}_i)]\right|_{\varepsilon=0} \\ &= -\sum_{i=1}^{N} < \mathbf{g}(\mathbf{x}_i), \boldsymbol{\varphi}_i > \end{aligned} \tag{24}$$

$$\begin{aligned} \delta^2 R_M[\mathbf{f}^t;\mathbf{g},j] &= \left.\frac{\partial^2 R_M[\mathbf{Q}_j^t + \varepsilon \mathbf{p}_j^t \odot \mathbf{g}]}{\partial \varepsilon^2}\right|_{\varepsilon=0} \\ &= \left.\frac{\partial^2}{\partial \varepsilon^2} \sum_{i=1}^{N} L_M[\mathbf{y}^i, \mathbf{Q}_j^t + \varepsilon \mathbf{p}_j^t \odot \mathbf{g}]\right|_{\varepsilon=0} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} < \mathbf{g}(\mathbf{x}_i), \mathbf{p}_j^t(\mathbf{x}_i) \odot (\mathbf{y}_i - \mathbf{y}^k) >^2 \theta_{i,k} \end{aligned} \tag{25}$$

```
**Algorithm 1** Adaptive multiclass boost (SOHP-MCBoost)
Input: Dataset ($X$, $C$), the number of classes $K$, a set of codewords $Y$, multiclass loss function $L_M[\cdot,\cdot]$ and the number of iterations $T$.
Initialization: Set t = 0, S = 0 and $\mathbf{f}^t(\mathbf{x}) = \mathbf{0} \in \mathbb{R}^{K-1}$.
while $t < T$ do
    For gradient descent, find the best additive update $\alpha_0^* \mathbf{g}_0^*$ with (11), (15) and (21); For Newton method, find this best update with (11), (12), (16) and (21).
    Compute the updated risk $R_M^0$ with (22).
    for $j$=1 to $S$ do
        For $j$th term in (19), find the best direction $\mathbf{g}_j^*$ by using (24) and (25) in (15) or (16), and the optimal step size $\alpha_j^*$ by (28).
        Compute the update risk $R_M^j$ via (29).
    end for
    Set $j_* = \arg\min_j R_M^j, \quad j \in \{0, \ldots, S\}$.
    If $j_*$ = 0, calculate $\mathbf{p}_{S+1}^{t+1} = \alpha_0^* \mathbf{g}_0^*$ and $S$ = $S$ + 1; Otherwise, calculate $\mathbf{p}_{j_*}^{t+1} = \alpha_{j_*}^* \mathbf{p}_{j_*}^t \odot \mathbf{g}_{j_*}^*$.
    For $j \neq j_*$, update $\mathbf{p}_j^{t+1} \leftarrow \mathbf{p}_j^t$.
    Update $\mathbf{f}^{t+1}(\mathbf{x}) \leftarrow \sum_{j=1}^{S} \mathbf{p}_j^{t+1}(\mathbf{x})$.
    Set $t = t + 1$.
end while
Output: decision rule
    $F(\mathbf{x}) = \arg\max_k \frac{\sigma\left(<\mathbf{f}^T(\mathbf{x}), \mathbf{y}^k>\right)}{\sum_{k=1}^{K} \sigma\left(<\mathbf{f}^T(\mathbf{x}), \mathbf{y}^k>\right)}$
```

where

$$\boldsymbol{\varphi}_i = \sum_{k=1}^{K} \mathbf{p}_j^t(\mathbf{x}_i) \odot (\mathbf{y}_i - \mathbf{y}^k) \frac{\exp\left(-<\mathbf{Q}_j^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)}{1 + \exp\left(-<\mathbf{Q}_j^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)} \tag{26}$$

$$\theta_{i,k} = \frac{\exp\left(-<\mathbf{Q}_j^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)}{\left[1 + \exp\left(-<\mathbf{Q}_j^t(\mathbf{x}_i), \mathbf{y}_i - \mathbf{y}^k>\right)\right]^2}. \tag{27}$$

Using the combination of (15) or (16) with (24) and (25), we can obtain the best weak learner $\mathbf{g}_j^*$. Moreover, the optimal step size along this direction is

TABLE I. SUMMARY OF TEN SELECTED UCI DATASETS

Dataset	#Instances	#Test	#Classes	#Attributes
Glass	214	5-fold	6	9
Seeds	210	5-fold	3	7
Vehicle	846	5-fold	4	18
Redwine	1599	5-fold	6	11
Yeast	1484	5-fold	10	8
Isolet	7797	1559	26	617
Landsat	6435	2000	6	36
Letter	20000	4000	26	16
Pendigit	10992	3498	10	16
Shuttle	58000	14500	7	9

$$\alpha_j^* = \arg\min_{\alpha \in \mathbb{R}} R_M(\mathbf{Q}_j^t + \alpha \mathbf{p}_j^t \odot \mathbf{g}_j^*). \tag{28}$$

As a result, the updated predictor has the following risk

$$R_M^j(\mathbf{f}^{t+1}) = R_M(\mathbf{Q}_j^t + \alpha_j^* \mathbf{p}_j^t \odot \mathbf{g}_j^*). \tag{29}$$

For each update strategy, this method computes the optimal multiclass weak learners and the corresponding risks. Then it chooses the update which leads to the largest decrease of the classification risk. This procedure is summarized in Algorithm 1, and named as SOHP-MCBoost for brevity.

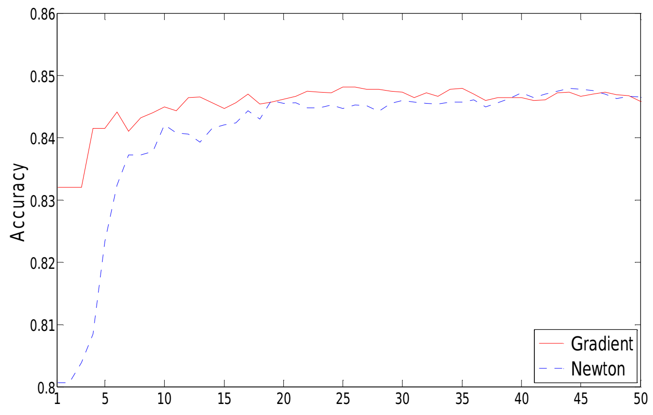
V. EXPERIMENT

In this section, some extensive experimental evaluations of the proposed multiclass boosting algorithm are described, which are compared with those of the existing methods for classification. All the experiments are fully implemented in MTALAB (R2012b) which is installed on an Intel Core 2 CPU E7500, 2.93GHz, 2GB RAM PC.

A. Dataset

Two types of datasets, namely synthetic data and real data, are used to carry out our experiments. To get the synthetic data, we consider the three-class and four-class problems in a two-dimensional space. The former problem has Gaussian classes of means [1, 2], [0, -1], [-2, 1] and covariances [1, 0.5; 0.5, 2], [1, 0.4; 0.4, 1], [2, 0.3; 0.3, 1], respectively. Similarly, for the latter problem, the class means are [2, 2], [-2, 2], [-2, -2], [2, -2], and the corresponding covariances are [1, 0.8; 0.8, 2], [2, 0.6; 0.6, 1], [1, 0.4; 0.4, 2], [2, 0.2; 0.2, 1]. In each class, training and test sets of 1000 examples are randomly sampled. Averaged results over five such independently drawn training-test set combinations are utilized.

The real datasets summarized in Table 1 are taken from the UCI machine learning repository [18]. Since no test sets are provided in the Glass, Seeds, Vehicle, Redwine and Yeast datasets, 5-fold cross-validation is used to evaluate the accuracy. That is to say, each dataset is randomly split into five folds, each of which is tested with the remaining four as training data. For the other datasets, the predefined training and test splits are used. In all case, the weak learners are decision trees which are built with a greedy and top-down recursive procedure. Since the extension of ROC curves for multiple



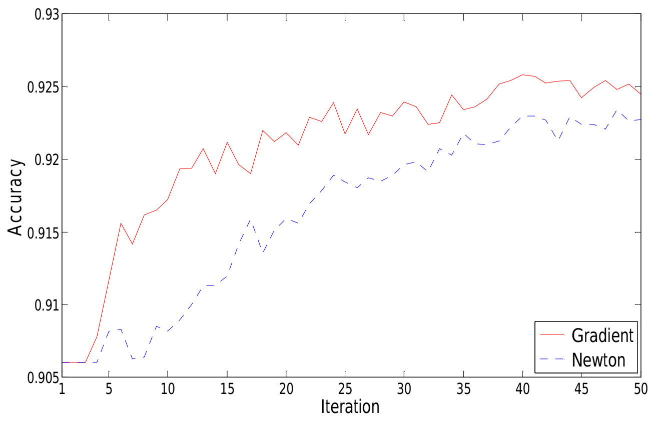


Fig. 1. Average accuracy curves of our method with gradient descent and Newton method, for three classes on the top and four classes on the bottom

class classification problems is intractable, the classification accuracy is taken as performance measure.

B. *Experimental results*

To begin with, we evaluate different optimization strategies used for weak learner selection in our multiclass boosting. Fig. 1 shows the average accuracy, which are achieved on the two synthetic datasets. As can be seen from this figure, the classification performance with Newton method is not superior to that with gradient descent. In particular, the accuracy gap is obvious for four-class synthetic dataset. A viewpoint in [4] is that Newton method is more accurate than gradient descent for binary problem, but it is unclear whether such viewpoint still holds promise for multiclass problem. Here we provide an experimental supplement. Thereafter, gradient descent is used as the optimization strategy for selecting weak learners in the remaining experiments.

The next experiment is designed to examine that the performance of multiclass boosting can be affected by tuning the depth of decision tree. In Table 2, the average results (with 50 iterations) on five UCI datasets are shown, and Tree(*num*) means that the depth of this decision tree is *num*. It can be clearly seen from this table that the moderate tree leads to the best performance in 4 of the 5 datasets. The simple tree cannot capture the interesting and important trends in the dataset, and the more complex tree can cause the severe over-fitting. Particularly, the phenomenon of over-fitting is visible in the

TABLE II. THE AVERAGE CLASSIFICATION ACCURACY (%) ON VARIOUS DATASETS USING DIFFERENT WEAK LEARNERS FOR SOHP-MCBOOST. BOLD NUMBER MEANS THE BEST ACCURACY

Dataset	Tree(1)	Tree(2)	Tree(3)
Glass	71.51	**77.08**	76.63
Seeds	90.95	**92.86**	89.52
Vehicle	72.34	**76.60**	75.89
Redwine	**57.41**	54.35	50.22
Yeast	54.92	**55.12**	52.92

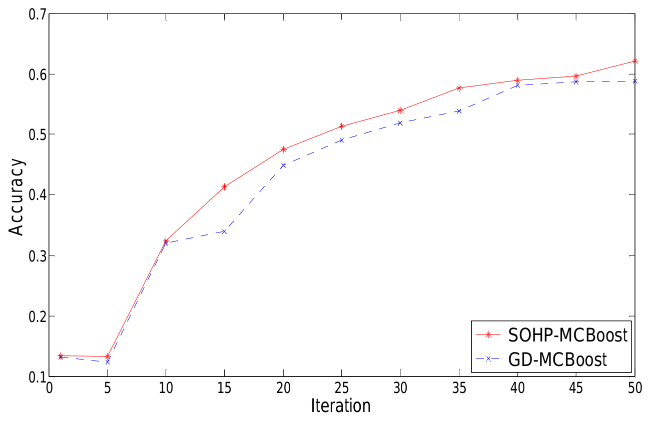


Fig. 2. Plot of the accuracy curves on the Letter dataset, for SOHP-MCBoost and GD-MCBoost

Redwine dataset, where increasing the depth of tree results in lower accuracy. Henceforth, the tree whose depth is two is utilized as the weak learners.

Finally, the performance of our method is compared with those of the existing methods in Table 3. In this table, all classifiers are trained with 50 iterations. The LPBoost and TotalBoost [19], which perform multiclass classification by attempting to maximize the minimal margin in the training set, make use of linear and quadratic programming, respectively. As can be seen, our proposed approach has the best accuracy on all datasets. For example, the gain can be improved from a previous best of 58.8% to 62.08% in the Letter dataset. Unlike GD-MCBoost, our method searches a larger space of weak learners, which can exploit some underlying structures from a dataset. Moreover, the weak learners are continually combined into complex combinations if and only if such combinations can lead to better performance. These may be the reasons behind the improvement. Compared with the other methods, our method has a Bayes consistent and large margin solution, which partly explains the superiority. In addition, Fig. 2 shows the accuracy curves for our method and GD-MCBoost on the Letter dataset.

VI. CONCLUSIONS

So far, we have acquired the optimization methods in the functional space for learning the final classifier, and well demonstrated the effectiveness of the proposed concept framework. Although our algorithm is a variant of some existing boosting algorithms, it is different in aspects of 1) convex optimization using a new multiclass loss function, 2)

TABLE III. ACCURACY (%) OF MULTICLASS BOOSTING ALGORITHMS ON FIVE UCI DATASETS. BOLD NUMBER MEANS THE HIGHEST ACCURACY. THE ALGORITHMS ON THE FIRST FOUR ROWS ARE TRAINED USING TREES OF 4 LEAVES, AND THE REST USE DEPTH-2 DECISION TREES AS WEAK LEARNERS

Algorithm	Isolet	Landsat	Letter	Pendigit	Shuttle	Average
AdaBoost-M2 [1]	19.24	78.90	16.33	64.92	93.91	54.66
LPBoost [19]	11.55	71.60	9.83	45.68	32.59	34.25
TotalBoost [19]	30.21	73.65	22.93	76.13	74.02	55.39
RUSBoost [12]	11.55	44.95	9.83	29.27	79.16	34.95
AdaBoost-SAMME [14]	61.00	79.80	45.65	83.82	99.70	73.99
AdaBoost-Cost [15]	63.69	83.95	42.00	80.53	99.55	73.94
GD-MCBoost [16]	84.28	86.35	58.80	92.94	99.73	84.82
SOHP-MCBoost	**85.82**	**87.15**	**62.08**	**94.03**	**99.97**	**85.81**

more sophisticated combinations of multiclass weak learners, and 3) probability distribution over classes. Experimental results on UCI datasets have shown that our method produces better classification performance. Theoretically, there are two special combinations in the functional space. One is only based on Hadamard product, which may be regarded as a generalization of [6] to multiclass. Suppose the computational cost for building each decision tree is $O(\mu)$, then the best computational cost of our algorithm is $O(\mu T)$ when this extreme case occurs. The other is the combinations of pure addition operators, which is similar to GD-MCBoost. This case can result in the worst time complexity for our method, namely $O(\mu T^2)$.

Additionally, for a specific dataset, the two operations can be combined in an adaptive manner, which need not predefine the number of operators and reduces the risk of over-fitting. Since the work in this paper can be regarded as a proof of concept to some extent, the research on real applications which are characterized by noise and high dimensionality is our ongoing work. The codes used here are completely independent of dataset, therefore an interesting benefit may be obtained by learning data-dependent codes for multiclass. Furthermore, we would like to explore multiclass semi-supervised boosting learning in the future.

APPENDIX

A. The lower bound of multiclass loss

From (7), one can express the multiclass loss as

$$\begin{aligned} L_M[\mathbf{y},\mathbf{f}(\mathbf{x})] &= \log 2 + \sum_{\mathbf{y}^k \neq \mathbf{y}}^{K} \log\left[1+\exp\left(-<\mathbf{f}(\mathbf{x}),\mathbf{y}-\mathbf{y}^k>\right)\right] \\ &= \log 2 + \log \prod_{\mathbf{y}^k \neq \mathbf{y}}^{K}\left[1+\exp\left(-<\mathbf{f}(\mathbf{x}),\mathbf{y}-\mathbf{y}^k>\right)\right]. \end{aligned} \tag{30}$$

For convenience, we omit **x** in **f**(**x**). In (30), if we define the product term as P, then

$$\begin{aligned} P = \exp[-2M(\mathbf{f},\mathbf{y})] \bullet \prod_{\mathbf{y}^k \neq \mathbf{y}}^{K}\left[1+\exp\left(-<\mathbf{f},\mathbf{y}-\mathbf{y}^k>\right)\right] \\ \bullet \exp[2M(\mathbf{f},\mathbf{y})] \end{aligned} \tag{31}$$

Substituting (2) in (31)

$$\begin{aligned} P &= \exp[-2M(\mathbf{f},\mathbf{y})] \bullet \prod_{\mathbf{y}^k \neq \mathbf{y}}^{K}\left[1+\exp\left(-<\mathbf{f},\mathbf{y}-\mathbf{y}^k>\right)\right] \\ &\quad \bullet \exp\left[<\mathbf{f},\mathbf{y}> - \max_{\mathbf{y}^l \neq \mathbf{y}} <\mathbf{f},\mathbf{y}^l>\right] \\ &= \exp[-2M(\mathbf{f},\mathbf{y})] \bullet \left(\exp\left[<\mathbf{f},\mathbf{y}-\mathbf{y}^{ll}>\right]+1\right) \\ &\quad \bullet \prod_{\mathbf{y}^k \neq \mathbf{y},\mathbf{y}^{ll}}^{K}\left[1+\exp\left(-<\mathbf{f},\mathbf{y}-\mathbf{y}^k>\right)\right] \\ &\geq 1+\exp[-2M(\mathbf{f},\mathbf{y})], \end{aligned} \tag{32}$$

where $ll = \arg\max_{\mathbf{y}^l \neq \mathbf{y}} <\mathbf{f},\mathbf{y}^l>$. Hence, we have

$$L_M[\mathbf{y},\mathbf{f}] \geq \log 2 + \log\left[1+\exp\left(-2M(\mathbf{f},\mathbf{y})\right)\right]. \tag{33}$$

B. The convexity of objective funtion

Given **x**, let the probability of class $\mathbf{y}^k$ be $\beta_k = P_{Y|X}(\mathbf{y}^k \mid \mathbf{x})$. We have

$$\begin{aligned} R_M(\mathbf{f} \mid \mathbf{x}) &= E_{Y|X}\{L_M[\mathbf{y},\mathbf{f}(\mathbf{x})] \mid \mathbf{x}\} \\ &= \sum_{k=1}^{K} \beta_k L_M[\mathbf{y}^k,\mathbf{f}(\mathbf{x})] \\ &= \sum_{k=1}^{K}\sum_{j=1}^{K} \beta_k \log\left[1+\exp\left(-<\mathbf{f}(\mathbf{x}),\mathbf{y}^k-\mathbf{y}^j>\right)\right], \end{aligned} \tag{34}$$

$$\frac{\partial R_M(\mathbf{f} \mid \mathbf{x})}{\partial \mathbf{f}(\mathbf{x})} = -\sum_{k=1}^{K}\sum_{j=1}^{K} \beta_k \boldsymbol{\eta}_{k,j} \frac{\exp\left(-<\mathbf{f}(\mathbf{x}),\boldsymbol{\eta}_{k,j}>\right)}{1+\exp\left(-<\mathbf{f}(\mathbf{x}),\boldsymbol{\eta}_{k,j}>\right)}, \tag{35}$$

$$\frac{\partial^2 R_M(\mathbf{f} \mid \mathbf{x})}{\partial \mathbf{f}(\mathbf{x})^2} = \sum_{k=1}^{K}\sum_{j=1}^{K} \beta_k [\boldsymbol{\eta}_{k,j}\boldsymbol{\eta}_{k,j}^T] \frac{\exp\left(-<\mathbf{f}(\mathbf{x}),\boldsymbol{\eta}_{k,j}>\right)}{\left[1+\exp\left(-<\mathbf{f}(\mathbf{x}),\boldsymbol{\eta}_{k,j}>\right)\right]^2}. \tag{36}$$

where $\boldsymbol{\eta}_{k,j} = \mathbf{y}^k - \mathbf{y}^j$. In [16], all the codewords are different. That is to say, for any k and j, $\boldsymbol{\eta}_{k,j} \neq 0$. As a result, the matrices $[\boldsymbol{\eta}_{k,j}\boldsymbol{\eta}_{k,j}^T]$ are positive definite. It is easily verified that (36) is strictly positive definite. Therefore, the objective function $R_M(\mathbf{f} \mid \mathbf{x})$ is strictly convex.

C. Bayes consistency

Using the following form

$$1 + \exp\left(< \mathbf{f}(\mathbf{x}), \boldsymbol{\eta}_{k,j} >\right) = \sqrt{\frac{\beta_k}{\beta_j}}, \tag{37}$$

from (35), we obtain

$$\begin{aligned} \frac{\partial R_M(\mathbf{f} \mid \mathbf{x})}{\partial \mathbf{f}(\mathbf{x})} &= -\sum_{k=1}^{K}\sum_{j=1}^{K} \beta_k \boldsymbol{\eta}_{k,j} \frac{1}{1 + \exp\left(< \mathbf{f}(\mathbf{x}), \boldsymbol{\eta}_{k,j} >\right)} \\ &= -\sum_{k=1}^{K}\sum_{j=1}^{K} \beta_k \boldsymbol{\eta}_{k,j} \sqrt{\frac{\beta_j}{\beta_k}} \\ &= -\sum_{k=1}^{K} \mathbf{y}^k \sqrt{\beta_k} \sum_{j=1}^{K} \sqrt{\beta_j} + \sum_{k=1}^{K} \sqrt{\beta_k} \sum_{j=1}^{K} \mathbf{y}^j \sqrt{\beta_j} = 0. \end{aligned} \tag{38}$$

Therefore, if the optimum of $R_M(\mathbf{f} \mid \mathbf{x})$ is $\mathbf{f}^*(\mathbf{x})$, then (37) holds. It follows that

$$\begin{aligned} < \mathbf{f}^*(\mathbf{x}), \boldsymbol{\eta}_{k,j} > &= < \mathbf{f}^*(\mathbf{x}), \mathbf{y}^k > - < \mathbf{f}^*(\mathbf{x}), \mathbf{y}^j > \\ &= \log\left(\sqrt{\beta_k} - \sqrt{\beta_j}\right) - \log\left(\sqrt{\beta_j}\right), \end{aligned} \tag{39}$$

and

$$< \mathbf{f}^*(\mathbf{x}), \mathbf{y}^k > = \log\left(\sqrt{P_{Y|X}(\mathbf{y}^k \mid \mathbf{x})} - c_1\right) + c_2, \tag{40}$$

for constants c_1 and c_2. It shows that implementing (3) is identical to executing Bayes decision rule (1).

REFERENCES

[1] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, 55(1): 119-139, 1997.

[2] J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," Annals of Statistics, 28(2): 337-407, 2000.

[3] L. Mason, J. Baxter, P. Bartlett and M. Frean, "Boosting algorithms as gradient descent in function space," Advances in Neural Information Processing Systems, 2000.

[4] M. J. Saberian, H. Masnadi-Shirazi and N. Vasconcelos, "Taylorboost: first and second order boosting algorithms with explicit margin control," IEEE International Conference on Computer Vision and Pattern Recognition, 2011.

[5] H. Masnadi-Shirazi, N. Vasconcelos and V. Mahadevan, "On the design of robust classifiers for computer vision," IEEE International Conference on Computer Vision and Pattern Recognition, 2010.

[6] B. Kégl and R. Busa-Fekete, "Boosting products of base classifiers," ACM International Conference on Machine Learning, pp. 497-504, 2009.

[7] O. Danielsson, B. Rasolzadeh and S. Carlsson, "Gated classifiers: Boosting under high intra-class variation," IEEE International Conference on Computer Vision and Pattern Recognition, pp. 2673-2680, 2011.

[8] T. Mita, T. Kaneko B. Stenger and O. Hori, "Discriminative feature co-occurrence selection for object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(7): pp. 1257-1269, 2008.

[9] M. J. Saberian and N. Vasconcelos, "Boosting algorithms for simultaneous feature extraction and selection," IEEE International Conference on Computer Vision and Pattern Recognition, 2012.

[10] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," Journal of Artificial Intelligence Research, 2: 263-286, 1995.

[11] E. L. Allwein, R. E. Schapire and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," Journal of Machine Learning Research, 1: 113-141, 2001.

[12] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse and A. Napolitano, "RUSBoost: Improving classification performance when training data is skewed," IEEE International Conference on Pattern Recognition, 2008.

[13] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," Machine Learning, 37(3): 297-336, 1999.

[14] J. Zhu, H. Zou, S. Rosset and T. Hastie, "Multi-class adaboost," Statistics and Its Interface, 2: 349-360, 2009.

[15] I. Mukherjee and R. E. Schapire, "A theory of multiclass boosting," Advances in Neural Information Processing Systems, 2010.

[16] M. J. Saberian and N. Vasconcelos, "Multiclass boosting: theory and algorithms," Advances in Neural Information Processing Systems, 2011.

[17] H. S. M. Coxeter, "Regular polytopes," DoverPublications. 1973.

[18] http://archive.ics.uci.edu/ml/datasets.html

[19] M. K. Warmuth, J. Liao and G. Rätsch, "Totally corrective boosting algorithms that maximize the margin," ACM International Conference on Machine Learning, 2006.