Multi-class HingeBoost*  
Method and Application to the Classification of Cancer Types Using Gene Expression Data

Z. Wang
Department of Research, Connecticut Children’s Medical Center, Department of Pediatrics, University of Connecticut School of Medicine, Hartford, Connecticut, USA

Keywords  
Classification, boosting, variable selection, smoothing splines, regression trees

Summary  
Background: Multi-class molecular cancer classification has great potential clinical implications. Such applications require statistical methods to accurately classify cancer types with a small subset of genes from thousands of genes in the data.

Objectives: This paper presents a new functional gradient descent boosting algorithm that directly extends the HingeBoost algorithm from the binary case to the multi-class case without reducing the original problem to multiple binary problems.

Methods: Minimizing a multi-class hinge loss with boosting technique, the proposed HingeBoost has good theoretical properties by implementing the Bayes decision rule and providing a unifying framework with either equal or unequal misclassification costs. Furthermore, we propose Twin HingeBoost which has better feature selection behavior than HingeBoost by reducing the number of ineffective covariates. Simulated data, benchmark data and two cancer gene expression data sets are utilized to evaluate the performance of the proposed approach.

Results: Simulations and the benchmark data showed that the multi-class HingeBoost generated accurate predictions when compared with the alternative methods, especially with high-dimensional covariates. The multi-class HingeBoost also produced more accurate prediction or comparable prediction in two cancer classification problems using gene expression data.

Conclusions: This work has shown that the HingeBoost provides a powerful tool for multi-classification problems. In many applications, the classification accuracy and feature selection behavior can be further improved when using Twin HingeBoost.

Correspondence to:
Zhu Wang
Department of Research, Connecticut Children’s Medical Center, Department of Pediatrics, University of Connecticut School of Medicine
282 Washington Street
Hartford, CT 06106
USA
E-mail: zwang@ccmckids.org

1. Introduction

The traditional cancer diagnostic methods are based primarily on subjective judgment of clinical and histopathological information. With emerging microarray gene expression data, objective and systematic approach for cancer diagnostic has great potential clinical implications [1–4]. In this context, it is important that the statistical methods not only can accurately classify cancer types, but also can select a small subset from thousands of genes. While there are many successful binary classification algorithms including the support vector machine (SVM) [5] and boosting [6], methodologies for the multi-class problems are less developed. Many of the popular multi-classification algorithms expand a binary classification algorithm to a multi-class problem based on the so-called “one-vs-all” or “one-vs-one” approach [7]. For a K-class problem (K>2) with one-vs-all approach, each class is separately compared against all other classes, and K functions are estimated to represent confidence for each class. The classification rule is to assign the class with the largest estimate. The one-vs-all approach has been applied to machine learning binary classifiers including the SVM and boosting [8,9]. However, this approach may not necessarily induce the optimal results for the original K-class problem. Indeed, when there is a lack of a dominant class (the conditional probability for the class is greater than 0.5), the final prediction of class labels can be unclear [8]. The one-vs-one approach constructs all possible pairs of classification rules and then combines them by means of a simple voting scheme. Therefore, this method requires the large number K(K – 1)/2 of binary classifiers and the size of training sample could become small for a given pair.

Recent methodological work has focused on developing new methods directly targeting on the original K-class problem, such as multi-class SVM [8] and multi-class boosting [10]. Minimizing a hinge loss function, the solution of the binary SVM directly targets the Bayes decision rule without estimating the conditional probability [11]. Using a multi-class hinge loss, the multi-class SVM (MSVM) [8] extends the binary SVM to the multi-class case and implements the Bayes decision rule as well. Boosting is another machine learning algorithm which has attracted much attention

* Supplementary material published on our website www.methods-online.com.
in the statistic community. Boosting has been widely applied with microarray data due to its accurate prediction and computational advantages with high-dimensional data. A recent review can be found in [12]. Also see [13] for a comparison between the traditional logistic regression and boosting. Taking full advantages of the SVM and boosting, the HingeBoost algorithm was developed to minimize the hinge loss with the functional gradient descent boosting algorithm [14]. The HingeBoost has shown its excellent performance and can incorporate unequal classification costs. Different from the SVM, the HingeBoost can conduct variable selection and function estimation simultaneously. Following this line of research, this paper proposes a new multi-class functional gradient descent boosting algorithm which minimizes a multi-class hinge loss function. The new algorithm extends the binary HingeBoost to the multi-class case and can be thought of as a unique combination of multi-class SVM and boosting. As a result, the developed multi-class HingeBoost has good theoretical properties by implementing the Bayes decision rule and providing a unifying framework with either equal or unequal classification costs. In the spirit of twin boosting [15], we further extend multi-class HingeBoost to Twin HingeBoost which has better feature selection behavior than HingeBoost by reducing the number of ineffective variables. The new algorithm is highly competitive as demonstrated through simulated data, benchmark data and two applications of cancer classification using microarray data.

2. Methods

2.1 Binary HingeBoost

In a binary classification problem, we have training data \((\vec{x}_i, y_i)\) for \(i = 1, \ldots, n\), where \(\vec{x}_i\) is a vector of length \(p\) and \(y_i = 1\) or \(-1\). The HingeBoost [14] is a functional gradient descent boosting algorithm which minimizes a popular hinge loss with modifications to accommodate unequal classification costs.

\[
L^{(b)} = \frac{1}{n} \sum_{i=1}^{n} \left\{ \sum_{j=1, j \neq b}^{K} C_q(f_j^{(b)}(\vec{x}_i) - y_i) + C_b \left( \frac{1}{n} \sum_{j=1, j \neq b}^{K} (f_j^{(b)}(\vec{x}_i) - y_i) \right) \right\}
\]  

(4)

\fig{1}{Equation 4}

where \(f(\vec{x})\) is a classifier and \(z_i = (\max(z, 0), 0)\). The classification rule is to predict a new response as 1 if \(f(\vec{x}) > 0\) and \(-1\) otherwise. Although HingeBoost implements the Bayes rule that predicts the dominant class, using HingeBoost for multi-class problems in the one-vs-all approach may not be optimal for some data.

2.2 Multi-class HingeBoost

For a \(K\)-class problem with a response belonging to \(1, \ldots, K\), we can recode the response to a vector \(\vec{y}_i = (y_{i1}, \ldots, y_{iK})\) whose \(r\)th element is 1 and all other elements are \(-1/(K-1)\) if the response falls into class \(r\). Denote \(C_b\) the cost of misclassifying a subject \(i\) from class \(r\) to class \(j\). For equal costs, \(C_b\) is 1 if \(j \neq r\), and 0 if \(j = r\). The following multi-class hinge loss was studied in [8]:

\[
L_{\text{lasso}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{K} C_q(f_j(\vec{x}_i) - y_{ij}) + C_b \left( \frac{1}{n} \sum_{j=1}^{K} (f_j(\vec{x}_i) - y_{ij}) \right),
\]  

subject to

\[
f_1(\vec{x}_i) + \ldots + f_K(\vec{x}_i) = 0.
\]  

(3)

The classification rule is \(\arg \max_i f_j(\vec{x}_i)\), which shares the Bayes decision rule. The sum-to-zero constraint assures identifiability of functions. The MSVM was proposed to minimize a penalized version of loss (2), which can be extended to take into account sampling bias together with unequal costs [8]. In this paper, we propose a functional gradient descent boosting algorithm to minimize loss (2). Since the binary hinge loss (1) is a special case of multi-class hinge loss (2), the proposed multi-class boosting algorithm naturally extends the binary HingeBoost to the multi-class case. The constraint (3) introduces a redundant function component, referred to as base class function. The base class function can be eliminated from a boosting algorithm initially, and can be computed after obtaining functions for other classes [16]. Suppose class \(b\) is considered as base class, then the above constrained objective function can be reduced (Fig. 1).

Estimation of \(f_j^{(b)}, j = 1, \ldots, K, j \neq b\) can be achieved by minimizing \(L^{(b)}\) loss through the following boosting algorithm. At each boosting iteration, base class \(b\) can be adaptively chosen through exhaustive search to minimize the hinge loss (2) so that the best performance can be achieved. See [16] for a similar strategy. The multi-class HingeBoost (hereafter referred to as HingeBoost for short) algorithm is described below:

1. Initialize \(f_j(\vec{x}) = 0\) for class \(j = 1, \ldots, K\).
2. For iteration \(m = 1, M\), repeat:
   a) For class \(b = 1\) to \(K\), repeat:
      i. For class \(j = 1\) to \(K\) and \(j \neq b\), repeat:
   A. Compute the residual vector \(V_j^{(b)} = (V_{j1}^{(b)}, \ldots, V_{jK}^{(b)})\), defined as negative gradient of loss function (4)

\[
V_{j}^{(b)} = \frac{\partial L^{(b)}}{\partial f_j^{(b)}} = f_j^{(b)} - \bar{f}_j^{(b)}(\vec{x}_i)
\]  

\[
= -C_q[\text{sign}(f_j^{(b)}(\vec{x}_i) - y_{ij}) + 1/2] + C_b[\text{sign}(\sum_{j=1, j \neq b}^{K} f_j^{(b)}(\vec{x}_i) - y_{ij}) + 1/2].
\]

B. Construct a regression model \(h_j(\vec{x})\) for predictor \(\vec{x}\) and working response vector \(V_j^{(b)}\) with a weak base learner. For instance, with componentwise linear least squares as base learner, \(h_j(\vec{x}^{(b)})\) can approximate the residuals \(V_j^{(b)}\) and \(q\) is determined by

\[
q = \arg \min_{1 \leq j \leq p} \left( \sum_{i=1}^{n} (V_{ij}^{(b)} - h_j(x_i^{(b)}))^2 \right).
\]
Table 1: Misclassification error for multi-class HingeBoost (HingeBoost.M), one-vs-one binary HingeBoost (HingeBoost.OVA), one-vs-one SVM, one-vs-all AdaBoost and regularized multinomial regression (glmnet) with Examples 1–4. The numbers in parentheses are the standard deviations of the estimated misclassification error.

<table>
<thead>
<tr>
<th>Method</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
<th>Example 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayes</td>
<td>0.3841</td>
<td>0.3763</td>
<td>0.5408</td>
<td>0.5387</td>
</tr>
<tr>
<td>HingeBoost.M</td>
<td>0.4024 (0.0116)</td>
<td>0.3829 (0.0124)</td>
<td>0.5875 (0.0346)</td>
<td>0.5846 (0.0213)</td>
</tr>
<tr>
<td>HingeBoost.OVA</td>
<td>0.4033 (0.0146)</td>
<td>0.3814 (0.0113)</td>
<td>0.5927 (0.0379)</td>
<td>0.5889 (0.0240)</td>
</tr>
<tr>
<td>SVM</td>
<td>0.5130 (0.0231)</td>
<td>0.4220 (0.0124)</td>
<td>0.6208 (0.0161)</td>
<td>0.5963 (0.0132)</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>0.5678 (0.0737)</td>
<td>0.4193 (0.0047)</td>
<td>0.6152 (0.0222)</td>
<td>0.5961 (0.0116)</td>
</tr>
<tr>
<td>glmnet</td>
<td>0.4322 (0.0342)</td>
<td>0.4193 (0.0052)</td>
<td>0.5717 (0.0116)</td>
<td>0.5889 (0.0114)</td>
</tr>
</tbody>
</table>

3.1 Simulation Study

To compare multi-class HingeBoost and one-vs-all binary HingeBoost, we generate three-class examples with the conditional probabilities as follows:

Example 1

\[
\begin{align*}
q_1(x^{(1)}) &= 0.97 \exp(-3x^{(1)}), \\
q_2(x^{(1)}) &= \exp(-2.5(x^{(1)} - 1.2)^2), \\
q_3(x^{(1)}) &= 0.1 - q_1(x^{(1)}) - q_2(x^{(1)}).
\end{align*}
\]

Example 2

\[
\begin{align*}
q_1(x^{(1)}) &= 0.7 - 0.6(x^{(1)})^4, \\
q_2(x^{(1)}) &= 1 + 0.6(x^{(1)})^4, \\
q_3(x^{(1)}) &= 0.2.
\end{align*}
\]

Example 3

\[
\begin{align*}
q_1(x^{(1)}) &= 0.45 - 0.4(x^{(1)})^4, \\
q_2(x^{(1)}) &= 0.3 + 0.4(x^{(1)})^4, \\
q_3(x^{(1)}) &= 0.2.
\end{align*}
\]

Example 4

\[
\begin{align*}
q_1(x^{(1)}) &= 0.45 - 0.3(x^{(1)})^4, \\
q_2(x^{(1)}) &= 0.35 + 0.3(x^{(1)})^4, \\
q_3(x^{(1)}) &= 0.2.
\end{align*}
\]

We generated \(x^{(i)}\) from the uniform distribution on [0,1] in Example 1, and on [−1, 1] in the other examples, for \(s = 1, \ldots, 50\). Similar simulations were used to compare MSVM and one-vs-all SVM, although only one effective predictor was considered [8]. Apparently, with the addi-

3. Results

We report the results from some empirical studies on HingeBoost for multi-class problems. We demonstrate that HingeBoost can fit a wide variety of data structures with different base learners. The proposed multi-class HingeBoost was compared with the alternative methods including the one-vs-all HingeBoost, one-vs-all AdaBoost, one-vs-one SVM [17] and regularized multinomial regression [18]. With simulated data, the multi-class HingeBoost has a better performance compared with one-vs-all HingeBoost if there is no dominant class. The HingeBoost algorithm can outperform the popular AdaBoost boosting [19] for some benchmark data. Finally, we illustrate that the HingeBoost and Twin HingeBoost are powerful tools for cancer type classification using microarray gene expression data.
ional 49 ineffective predictors, the examples studied in this paper are more complicated than those in [8]. These examples represent different patterns of class conditional probabilities. Except for Example 2, the examples have no dominant class in a large subset of \( x^{(1)} \). In Example 1, prediction of class 2 based on \( f_2(x^{(1)}) \) can be difficult for the one-vs-all scheme since \( p_2(x^{(1)}) \) is only close to 0.5 at most. Example 3 has no dominant class for most of \( x^{(1)} \) although one class is more likely than other two classes. Example 4 has two competitive classes. In each example, we randomly generated 200 training samples and the error rates were from 10,000 test data. This process was repeated for 100 times. We applied the multi-class HingeBoost and one-vs-all binary HingeBoost as base learners to the above examples. Table 1 and Figures 1–4 in the Online Supplementary Material of the paper summarize the results. Also enclosed are the Bayes misclassification error rates. In Table 1, the tuning parameters were selected by 5-fold cross-validation except for the SVM and regularized multinomial regression in which the default 10-fold cross-validation was used in the implementations [17, 18]. The results demonstrate that when there is no dominant class (Example 1, 3 and 4), multi-class HingeBoost typically yields smaller test error rates than the one-vs-all HingeBoost, which is consistent with findings in [8]. When no dominant class appears in the data, the multi-class HingeBoost has less variable classification error rates across boosting iterations than the one-vs-all HingeBoost. Supplementary Figures 1–4 suggest that the two algorithms can produce their optimal or close to the optimal prediction accuracy with very small number of predictor variables. Except for Example 2, the multi-class HingeBoost shows great resistance to overfitting even with additionally selected noise predictors, while the one-vs-all HingeBoost degrades the performance when selecting more noise variables. In summary, for data without dominant class, the multi-class HingeBoost appears to be more robust than the one-vs-all HingeBoost. Compared to the SVM, HingeBoost can conduct variable selection, which is a contributing factor with smaller error rates. Trees with two terminal nodes were used as base learner for the AdaBoost since the generative functions are additive. It is not surprising that HingeBoost with smoothing splines has better prediction accuracy than the AdaBoost due to the underlying smooth conditional probabilities. With the linear functions, the regular-
ized multinomial regression may not fit the underlying nonlinear conditional probabilities well, as demonstrated in Table 1.

The simulation study was conducted with $p = 50$ and $n = 200$, which should have provided a challenging task in the context of nonparametric modeling. As a reviewer suggested, to investigate the performance of the proposed approach in the high-dimensional setting, additional simulations were conducted. Specifically, every simulated data set now contains 499 ineffective predictors, in addition to one effective predictor. Other simulation configurations remain the same. The results from running both multi-class and one-vs-all HingeBoost (not shown) were compared with those in Supplementary Figures 1–4 with $p = 50$. The multi-class HingeBoost still performs better than the one-vs-all counterpart when there is no dominant class (and comparable otherwise). Compared with $p = 50$, the prediction accuracy with $p = 500$ only slightly degrades although more predictors are selected. For instance, in Example 1, the mean test errors with the multi-class HingeBoost are between 0.4025 to 0.4085 across the boosting iterations, almost unnoticeably increased from a range of 0.4006 to 0.4048 with $p = 50$. In summary, the simulation results show that the HingeBoost can produce accurate classification even with high-dimensional covariates.

### 3.2 Benchmark Data

We compare the HingeBoost and the one-vs-all AdaBoost algorithm, along with other methods, for four benchmark data sets available from the UCI repository of machine learning databases. The characteristics of these data are summarized in Table 1. We utilized regression trees as base learners in the HingeBoost and AdaBoost. The number of terminal nodes is related to the depth of the tree, and the degree of interactions. For instance, a stump has a depth of 1 with two terminal nodes and can fit main effects; while a tree with a depth of 2 can have at most four terminal nodes and fit main effects plus two-way interactions. Thus, to fit the data adequately, the depth should be at least as large as the number of dominant variable interactions. Although the degree of interaction is typically unknown in practice, one can calibrate the value by trying several different degrees and choosing the one that produces the best prediction on a test data [20, page 363]. For the benchmark data sets considered, boosted trees with main effects or two-way interactions are generally inferior to those with higher interactions. To illustrate, we present the results for maximum tree depth 6 in Table 2. The tuning parameters were selected as in Section 3.1. To visually compare the performances between HingeBoost and AdaBoost, we also present the predictions for maximum tree depth 3 and 6 in Supplementary Figures 5 and 6, respectively. Prediction accuracy by boosted trees with maximum depth 4 or 5 roughly falls between the results with depth 3 and depth 6. These figures show that the multi-class HingeBoost outperforms or nearly outperforms AdaBoost with smaller test error rates across almost all the boosting iterations, with an exception for the thyroid data for which the difference is indistinguishable with near-perfect predictions. In addition, the multi-class HingeBoost typically outperforms one-vs-all HingeBoost when the tree depth is chosen appropriately. In other words, except for glass with tree depth 3, the multi-class HingeBoost has smaller misclassification errors than the one-vs-all HingeBoost in most boosting iterations. For glass, the multi-class HingeBoost with tree depth 6 appears to be better than one-vs-all HingeBoost with tree depth 3. Furthermore, Supplementary Figures 5 and 6 suggest that the maximum depth 6 is a better choice than depth 3 for both the multi-class HingeBoost and AdaBoost, except for thyroid. Within the one-vs-all scheme, these figures also demonstrate that the HingeBoost can generate smaller misclassification errors than AdaBoost or at least comparable ones. Table 2 shows that the multi-class HingeBoost is much better than the one-vs-one SVM and regularized multinomial regression in these applications, partly because the HingeBoost can conveniently model interactions among the predictor variables.

### 3.3 Gene Expression Data

Classifying the small round blue cell tumors (SRBCTs) of childhood into four categories was studied using gene expression profiles [3]. With 2,308 gene profiles in 63 training samples and 20 test samples, perfect classification was reported. To apply the HingeBoost, we first took the logarithm base 10 of the gene levels, then standardized the results. We then selected top genes based on a marginal relevance measure as in [8]. We applied the HingeBoost with smoothing splines and summarized the results in Figure 2. The results are based on top 300 genes and similar results have been achieved with top 500 genes. It is clear that the HingeBoost can yield perfect classification in a wide scope of boosting iterations. In addition, Twin HingeBoost can further reduce the number of genes used for classification, and still maintain the perfect classification. In Figure 2, Twin HingeBoost utilized boost iteration 20 in the first round of HingeBoost and the results were robust to other choices. The choice of 20 iterations can be justified by the performance of 5-fold cross-validation (Supplementary Figure 7).

---

**Table 2** Misclassification error for multi-class HingeBoost (HingeBoost.M), one-vs-all binary HingeBoost (HingeBoost.OVA), one-vs-one SVM, one-vs-all AdaBoost and regularized multinomial regression (glmnet) with the benchmark data sets. The maximum depth of tree was fixed at 6.

<table>
<thead>
<tr>
<th>Method</th>
<th>Segmentation</th>
<th>Thyroid</th>
<th>Satimage</th>
<th>Glass</th>
</tr>
</thead>
<tbody>
<tr>
<td>HingeBoost.M</td>
<td>0.0505</td>
<td>0.0055</td>
<td>0.1000</td>
<td>0.2111</td>
</tr>
<tr>
<td>HingeBoost.OVA</td>
<td>0.0757</td>
<td>0.0073</td>
<td>0.1185</td>
<td>0.3097</td>
</tr>
<tr>
<td>SVM</td>
<td>0.0957</td>
<td>0.0452</td>
<td>0.1230</td>
<td>0.3367</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>0.0610</td>
<td>0.0073</td>
<td>0.1010</td>
<td>0.2061</td>
</tr>
<tr>
<td>glmnet</td>
<td>0.0924</td>
<td>0.0455</td>
<td>0.1635</td>
<td>0.3926</td>
</tr>
</tbody>
</table>
Classifying cancer into 14 types using gene expression levels was studied in [1, 2]. The GCM data has 16,063 genes for 144 training and 46 test samples, and the best prediction accuracy was reported to be 0.78 using the one-vs-all SVM. To proceed the analysis, pre-processing steps were taken before standardization [21]: a) thresholding (truncated below 20 and above 16,000), b) filtering (removal of genes with max/min ≤ 5 or max – min ≤500), c) logarithmic base 10 transformation. With 10,884 genes after filtering, the HingeBoost and Twin HingeBoost with componentwise linear least squares were applied to the data. For Twin HingeBoost, we ran the first round of HingeBoost 150 iterations, which is a reasonable stopping point according to the performance of 5-fold cross-validation (Supplementary Figure 7). Figure 3 demonstrated the results from HingeBoost and Twin HingeBoost. It is clear that misclassification error rate from Twin HingeBoost is smaller than 0.22 after about 20 boosting iterations and the best accuracy is 0.85 increasing by 9% than the previously documented best results. The unsmooth error curve seems to associate with the large predictor space (10,884 genes after filtering steps) and large class sizes (14 cancer types). Perhaps equally interesting is the fact that only 500 genes are needed to reach the maximal accuracy. This is somewhat dramatically distinct to the results with one-vs-all SVM in which the best classification accuracy is achieved when all genes are utilized for each one-vs-all classification and smaller number of genes only led to decreased accuracy [1]. Here, we demonstrate that with smaller number of genes, better prediction accuracy can be achieved with Twin HingeBoost. In conclusion, Twin HingeBoost outperforms HingeBoost and one-vs-all SVM in this application.

4. Conclusions

This paper presented a new boosting algorithm tailored to optimize a well defined loss function for multicategory classification problems. Although the proposed multi-class HingeBoost shares the same loss function with the original MSVM approach, they are different in two aspects at least. First, HingeBoost can model a variety of data structures when appropriate base learners are used, as illustrated with regression trees containing high order interactions. Second, the HingeBoost can conduct variable selection. With the addition of adaptive variable selection in Twin HingeBoost, we have demonstrated useful applications in cancer classification with microarray gene expression profiles. Different from other existing multi-class boosting algorithms (for instance, AdaBoost), HingeBoost has been extended to contain the adaptive variable selection feature, and can incorporate equal or unequal misclassification costs, and possibly non-representative training sets in a unifying framework.

The proposed HingeBoost has been implemented in the open source R package bst [22], freely available from the Comprehensive R Archive Network http://CRAN.R-project.org.

Acknowledgments

The author thanks three reviewers for their constructive remarks and suggestions, which have greatly improved the article.

References