BIOINFORMATICS

Vol. 19 no. 9 2003, pages 1061–1069
DOI: 10.1093/bioinformatics/btf867

Boosting for tumor classification with gene expression data

Marcel Dettling and Peter Bühlmann*

Seminar für Statistik, ETH Zürich, CH-8092, Switzerland

Received on February 28, 2002; revised on April 19, 2002; accepted on September 5, 2002

ABSTRACT

Motivation: Microarray experiments generate large datasets with expression values for thousands of genes but not more than a few dozens of samples. Accurate supervised classification of tissue samples in such high-dimensional problems is difficult but often crucial for successful diagnosis and treatment. A promising way to meet this challenge is by using boosting in conjunction with decision trees.

Results: We demonstrate that the generic boosting algorithm needs some modification to become an accurate classifier in the context of gene expression data. In particular, we present a feature preselection method, a more robust boosting procedure and a new approach for multicategorical problems. This allows for slight to drastic increase in performance and yields competitive results on several publicly available datasets.

Availability: Software for the modified boosting algorithms as well as for decision trees is available for free in *R* at http://stat.ethz.ch/~dettling/boosting.html.

Contact: dettling@stat.math.ethz.ch

1 INTRODUCTION

The recently developed microarray technology allows for measuring expression levels of thousands of genes simultaneously. We focus on the case where the experiments monitor gene expression values of different individuals or tissue samples, and where each experiment is equipped with an additional categorical outcome variable describing a cancer (pheno)type. In such a supervised setting, our goal is to predict the unknown class label of a new individual on the basis of its gene expression profile, since precise diagnosis of cancer type is often crucial for successful treatment. Given the availability of efficient classification techniques, bio-molecular information could become as, or even more important than traditional clinical factors.

Classification of different phenotypes, predominantly cancer types, using microarray gene expression data has been considered by Golub *et al.* (1999), Alon *et al.* (1999), Ben-Dor *et al.* (2000), Furey *et al.* (2000), Slonim *et al.* (2000), Dudoit *et al.* (2002), West *et al.* (2001) and Zhang *et al.* (2001), among others. The methods used in these studies range from classical discriminant analysis over Bayesian approaches and clustering methods to flexible tools from machine learning such as bagging, boosting and support vector machines. Explicitly, boosting decision trees has been applied for the classification of gene expression data in Ben-Dor *et al.* (2000) and Dudoit *et al.* (2002). Both studies compare the original AdaBoost algorithm that was proposed by Freund and Schapire (1997) against other classifiers, and both recognize that boosting does not yield very impressive results.

In this paper we demonstrate that the performance of boosting for classification of gene expression data can often be drastically improved by modifying the algorithm as follows: First, we perform feature preselection with the nonparametric scoring method of Park *et al.* (2001). Then, we apply the LogitBoost procedure introduced by Friedman *et al.* (2000) instead of the AdaBoost procedure. The former was found to have a slight edge over AdaBoost in a variety of more traditional classification problems (Friedman *et al.*, 2000),and it usually performs better on noisy data or when there are misspecifications or inhomogeneities of the class labels in the training data, which is frequently the case with microarray gene expression data. Finally, if discrimination has to be done for more than two tumor types, we reduce multiclass to multiple binary problems so that different gene subsets and different model complexity for discriminating different tumor types are allowed. This multiclass approach turns out to be much more accurate than the direct multiclass LogitBoost algorithm of Friedman *et al.* (2000). On six publicly available datasets and with a simulation study we show that the sum of these modifications leads to a classification procedure which performs very competitively, does not require sophisticated fine tuning and is fairly easy to implement.

2 METHODS

2.1 The stochastic framework

We assume that we are given n training data pairs

$$(x_1, y_1), \ldots, (x_n, y_n),$$

*To whom correspondence should be addressed.

with $x_i \in \mathbb{R}^p$ and $y_i \in \{0, \ldots, J-1\}$, which are independent and identically distributed realizations of a random vector (X, Y). The interpretation is that the feature or input vector X models the p-dimensional gene expression profile and the response or output variable Y denotes the class label. Today, the sample size n is typically in the range of 20 to 80 and the number of monitored genes p varies between 2000 and 20 000.

In the standard classification problem, the goal is to predict the class label Y, based on the expression vector X. This amounts to construct a classifier function

$$\mathcal{C} : X \mapsto \mathcal{C}(X) \in \{0, \ldots, J-1\},$$

which can subsequently be used to predict the unknown class label of a *new* tissue sample based on its expression vector. The optimal classifier is such that the misclassification risk

$$\mathbb{P}\left[\mathcal{C}(X) \neq Y\right] \text{ is minimal.} \qquad (1)$$

Note that this quantity is most often different from zero. The solution of Equation (1) requires knowledge of the true, but generally inaccessible conditional probability distribution $\mathbb{P}[Y = j|X]$ and is called *Bayes classifier*,

$$\mathcal{C}_{Bayes}(X) = \text{argmax}_{j\in\{0,\ldots,J-1\}}\mathbb{P}[Y = j|X]. \qquad (2)$$

In practice, it can be constructed via estimated conditional probabilities $\widehat{\mathbb{P}}[Y = j|X]$. This is a classical task for $p \ll n$, but expression data with many more features than samples ($p \gg n$) create a new challenge. A promising way to find a good discriminative model is by using boosting in conjunction with decision trees.

2.2 Binary classification of gene expression data

We focus first on binary problems with response $Y \in \{0, 1\}$. The best way to handle multi-categorical problems is explained later in Section 2.3.

2.2.1 Feature preselection The intrinsic problem with classification from microarray data is that sample size n is much smaller than the dimensionality of the feature space, i.e. the number of genes p. Many genes are non-differentially expressed across the samples and irrelevant for phenotype discrimination. Dimensionality reduction of the feature space has been performed by many authors, see for example Golub *et al.* (1999), Ben-Dor *et al.* (2000) and Dudoit *et al.* (2002), among others. It drastically eases the computational burden and for many problems improves class prediction due to the reduced amount of noise. Our feature selection is based on scoring each individual gene g, with $g \in \{1, \ldots, p\}$, according to its strength for phenotype discrimination. We use a nonparametric method that is based on ranks and was presented by Park *et al.* (2001). It is in fact equivalent to the test statistic of Wilcoxon's two sample test,

$$\text{Score}(g) = s(g) = \sum_{i\in\mathcal{N}_0}\sum_{j\in\mathcal{N}_1} 1_{[x_j^{(g)} - x_i^{(g)} \leq 0]},$$

where $x_i^{(g)}$ is the expression value of gene g for individual i and $\mathcal{N}_m$ represents the set of the n_m indices $\in \{1, \ldots, n\}$ having response in $m \in \{0, 1\}$. The score function can be interpreted as counting for each individual having response value zero, the number of instances with response one that have smaller expression values, and summing up these quantities. Viewing it as Wilcoxon's test statistic, it allows ordering of the genes according to their potential significance. It captures to what extent a gene g discriminates the response categories and it is easy to notice that both values near the minimum score zero and the maximum score $n_0 n_1$ indicate a differentially expressed, informative gene. The quality measure

$$q(g) = \max(s(g), n_0 n_1 - s(g))$$

thus gives the highest values to those genes whose expression levels have the best strength for phenotype discrimination. We then simply take the $\widetilde{p} \leq p$ genes with the highest values of $q(g)$ as our top features and restrict the boosting classifier to work with this subset. The number of predictor variables is a tuning parameter whose optimal value varied across different datasets. A formal choice of $\widetilde{p}$ is possible via cross validation on the training data or by determining the correct null distribution by bootstrap methods and a decision on significance levels as in Park *et al.* (2001). Many more variable selection criteria for gene expression data have been proposed in the literature. We think that our approach based on Wilcoxon's test statistic is more suitable in the context of gene expression data than the *t*-statistic used in Dudoit *et al.* (2002), or the TNoM score (Ben-Dor *et al.*, 2000) which corresponds to counting the number of errors made by the best stump, a decision tree with two terminal nodes. The situation is similar to the trade-off between *t*-, Wilcoxon- and sign-test. It is known from robustness theory that the *t*-test is highly sensitive to outliers and (even small) deviations from the normal distribution, whereas the sign-test (TNoM score) wastes useful information about the magnitude of gene expression levels. A good compromise is the Wilcoxon-test which has nearly optimal power properties over a large class of data-generating distributions, see Hampel *et al.* (1986).

2.2.2 Binary LogitBoost with decision trees Boosting, first introduced by Freund and Schapire (1996) has been found to be a powerful classification technique with remarkable success on a wide variety of problems,

especially in higher dimensions. It aims at producing an accurate combined classifier from a sequence of *weak* (or *base*) classifiers, which are fitted to iteratively reweighted versions of the data. In each boosting iteration m, with $m \in \{1, 2, \ldots, M\}$, the observations that have been misclassified at the previous step have their weights increased, whereas the weights are decreased for those that were classified correctly. The mth weak classifier $f^{(m)}$ is thus forced to focus more on individuals that have been difficult to classify correctly at earlier iterations. The combined classifier is equivalent to a weighted majority vote of the weak classifiers for shifted labels $\in \{-1, 1\}$,

$$\mathcal{C}^{(M)}(X) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m f^{(m)}(X)\right).$$

Three elements need to be chosen: (i) the type of weak learners $f^{(m)}$; (ii) the reweighting of the data and the aggregation weights α_m; and (iii) the number of boosting iterations M. Regarding issue (i), we exclusively focus on decision trees, see Breiman *et al.* (1984). These are the most popular learners in conjunction with boosting. In fact, we even further restrict here to *stumps*, which are trees with two terminal nodes only, since in the context of gene expression data, this always yielded better or equal performance than boosting larger trees. Concerning issue (ii), the reweighting of the data and the choice of aggregation weights can be coherently motivated by the principle of functional gradient descent (Breiman, 1999; Friedman *et al.*, 2000), from which several versions of boosting for classification emerge. We build here on the LogitBoost introduced by Friedman *et al.* (2000): it relies on the binomial log-likelihood as a loss function, which is a more natural criterion in classification than the exponential criterion underlying the AdaBoost algorithm. Since the former increases linearly instead of exponentially for strongly negative margins (see Hastie *et al.*, 2001), it is more robust in noisy problems where the misclassification risk of Equation (1) is substantial, and also in situations where mislabeled training data points or inhomogeneities in the training samples are present, all of which can be the case with gene expression data. Finally regarding (iii), the choice of the stopping parameter is often neglected and the boosting process is stopped at a usually large, but arbitrarily fixed number of iterations. Alternatively, we consider an empirical approach for the choice of M in the next section. The binary LogitBoost with decision stumps as weak learner works then as follows:

`Step 1: Initialization`
Start with an initial committee function $F^{(0)}(x) \equiv 0$ and initial probabilities $p^{(0)}(x) \equiv 1/2$; $p(x)$ is an abbreviation for $\widehat{\mathbb{P}}[Y = 1|X = x]$.

`Step 2: LogitBoost iterations`
For $m = 1, 2, \ldots, M$ repeat:

`A. Fitting the weak learner`

`(i)` Compute working response and weights for all $i = 1, \ldots, n$

$$w_i^{(m)} = p^{(m-1)}(x_i) \cdot \left(1 - p^{(m-1)}(x_i)\right),$$
$$z_i^{(m)} = \frac{y_i - p^{(m-1)}(x_i)}{w_i^{(m)}}.$$

`(ii)` Fit a regression stump $f^{(m)}$ by weighted least squares

$$f^{(m)} = \text{argmin}_f \sum_{i=1}^{n} w_i^{(m)}(z_i^{(m)} - f(x_i))^2.$$

`B. Updating and classifier output`

$$F^{(m)}(x_i) = F^{(m-1)}(x_i) + \frac{1}{2} f^{(m)}(x_i).$$
$$\mathcal{C}^{(m)}(x_i) = \text{sign}\left(F^{(m)}(x_i)\right),$$
$$p^{(m)}(x_i) = \left(1 + \exp\left(-2 \cdot F^{(m)}(x_i)\right)\right)^{-1}.$$

To increase understanding of the LogitBoost algorithm, we point out that each committee function $F^{(m)}(x)$ is an estimate of half of the log-odds ratio

$$F(x) = \frac{1}{2} \log\left(\frac{p(x)}{1 - p(x)}\right).$$

LogitBoost thus fits an additive logistic regression model by stagewise optimization of the binomial log-likelihood. More details can be found in Friedman *et al.* (2000),

A very useful property of our classification method is that it directly yields probability estimates $\widehat{\mathbb{P}}[Y = j|X = x]$. This is crucial for constructing classifiers respecting non-equal misclassification costs. Moreover, it allows to build classifiers which have the option to assign the label 'no class' (or 'doubt') for certain regions in the space of gene expression vectors x, see for example Ripley (1996).

An important advantage of LogitBoost compared to methods like neural nets or support vector machines is that it works well without fine tuning and no sophisticated nonlinear optimization is necessary. Provided that a decision tree algorithm is available, e.g. versions of CART (Breiman *et al.*, 1984) or C4.5 (Quinlan, 1993), LogitBoost with trees can be implemented very easily. Software for decision trees is widely available: for example for free as an *R*-Package called `rpart`, at http://www.stat.math.ethz.ch/CRAN.

2.2.3 Choice of the stopping parameter The stopping parameter M is often simply fixed at a large number in the range of dozens or hundreds. This, because boosting is generally quite resistant against overfitting so that the choice of M is typically not very critical, see also Figure 1. An alternative is to use an empirical approach for estimation of M by leave-one-out cross validation on the training data. The idea is to compute the binomial log-likelihood

$$\ell(m) = \sum_{i=1}^{n} \log\left(\widehat{p}^{(m)}(x_i)\right) \cdot 1_{[Y_i=1]} + \\ + \log\left(1 - \widehat{p}^{(m)}(x_i)\right) \cdot 1_{[Y_i=0]}, \qquad (3)$$

for each boosting iteration m across the samples and to choose the stopping parameter as the m for which $\ell(m)$ is maximal. We observed that $\ell(m)$ usually peaks somewhere between 10 and 100 boosting iterations. However empirically, we could not exploit significant advantages of estimated stopping parameters against a choice of $M = 100$ in the gene expression data we considered.

2.3 Reducing multiclass to binary

Here we explain how multi-response problems ($J > 2$) can be handled in conjunction with boosting. We recommend to compare each response class separately against all other classes. This *one-against-all* approach for reduction to J binary problems is very popular in the machine learning community, since many algorithms are solely designed for binary problems. It works by defining the response in the jth problem as

$$Y^{(j)} = \begin{cases} 1, & \text{if } Y = j, \\ 0, & \text{else} \end{cases}$$

and running j times the entire procedure including feature preselection, binary LogitBoost and stopping parameter estimation on the data $(x_1, y_1^{(j)}), \ldots, (x_n, y_n^{(j)})$. This yields estimates $\widehat{\mathbb{P}}[Y^{(j)} = 1|X]$ for $j \in \{0, \ldots, J-1\}$, which can be converted into probability estimates for $Y = j$ via normalization,

$$\widehat{\mathbb{P}}[Y = j|X] = \frac{\widehat{\mathbb{P}}\left[Y^{(j)} = 1|X\right]}{\sum_{k=1}^{J} \widehat{\mathbb{P}}\left[Y^{(k)} = 1|X\right]}.$$

This expression can be plugged into the Bayes classifier of Equation (2) and it is easy to see that this yields

$$\mathcal{C}(X) = \text{argmax}_{j \in \{0,\ldots,J-1\}} \widehat{\mathbb{P}}\left[Y^{(j)} = 1|X\right]$$

as our final classifier in multiclass problems. More sophisticated and computationally more expensive approaches for reducing multiclass to binary problems also exist, see Hastie and Tibshirani (1998) or Allwein *et al.* (2000) for a thorough discussion.

The one-against-all approach allows for different preselected features, different chosen variables for the decision trees in the LogitBoost algorithm, and for different model complexity via different stopping parameters for every class discrimination. This adaption seems to be very important with gene expression data. We observed, that the multiclass LogitBoost of Friedman *et al.* (2000), which treats the multiclass problem more simultaneously, performed much worse in our study. In the NCI dataset, comprising $J = 8$ different tumor types, it yielded an error rate of 36.1%, whereas with the one-against-all method, the error-rate was only 22.9%. For the Lymphoma dataset with $J = 3$ response classes, the one-against-all approach is also superior with 1.61% versus 8.06%.

3 RESULTS

3.1 Real data

We explored the performance of our classification technique on six publicly available datasets.

Leukemia
This dataset contains gene expression levels of $n = 72$ patients either suffering from acute lymphoblastic leukemia (ALL, 47 cases) or acute myeloid leukemia (AML, 25 cases) and was obtained from Affymetrix oligonucleotide microarrays. More information can be found in Golub *et al.* (1999); the raw data are available at http://www-genome.wi.mit.edu/cancer/. Following the protocol in Dudoit *et al.* (2002), we preprocess them by thresholding, filtering, a logarithmic transformation and standardization, so that the data finally comprise the expression values of $p = 3571$ genes.

Colon
In this dataset, expression levels of 40 tumor and 22 normal colon tissues for 6500 human genes are measured using the Affymetrix technology. A selection of 2000 genes with highest minimal intensity across the samples has been made by Alon *et al.* (1999), and these data are publicly available at http://microarray.princeton.edu/oncology/. As for the leukemia dataset, we process these data further by carrying out a base 10 logarithmic transformation and standardizing each tissue sample to zero mean and unit variance across the genes.

Estrogen and Nodal
These datasets were first presented in recent papers of West *et al.* (2001) and Spang *et al.* (2001). Their common expression matrix monitors 7129 genes in 49 breast tumor samples. The data are available at http://mgm.duke.edu/genome/dna_micro/work/ and were obtained by applying the Affymetrix gene chip technology. We thresholded the

raw data with a floor of 100 and a ceiling of 16 000 and then applied a base 10 logarithmic transformation. Finally, each experiment was standardized to zero mean and unit variance across the genes. Two different response variables are available: The first one describes the status of the estrogen receptor (ER). 25 samples are ER+, whereas the remaining 24 samples are ER−. The second response variable describes the lymph nodal (LN) status, which is an indicator for the metastatic spread of the tumor, a very important risk factor for disease outcome. Also here, 25 samples are positive (LN+) and 24 samples are negative (LN−).

Lymphoma

This dataset is publicly available at http://llmpp.nih.gov/lymphoma/data/figure1 and contains gene expression levels of the $J = 3$ most prevalent adult lymphoid malignancies: 42 samples of diffuse large B-cell lymphoma, 9 observations of follicular lymphoma and 11 cases of chronic lymphocytic leukemia. The total sample size is $n = 62$, and the expression of $p = 4026$ well-measured genes, preferentially expressed in lymphoid cells or with known immunological or oncological importance is documented. More information on these data can be found in Alizadeh *et al.* (2000). We imputed missing values and standardized the data as described in Dudoit *et al.* (2002).

NCI

This dataset comprises gene expression levels of $p = 5244$ genes for $n = 61$ human tumor cell lines from cDNA microarrays, which can be divided in $J = 8$ classes: 7 breast, 5 central nervous system, 7 colon, 6 leukemia, 8 melanoma, 9 non-small cell lung carcinoma, 6 ovarian and 9 renal tumors. A more detailed description of the data can be found on the website http://genome-www.stanford.edu/nci60 and in Ross *et al.* (2000). We work with preprocessed data as described in Dudoit *et al.* (2002).

3.1.1 Empirical study We performed leave-one-out cross validation to explore the classification potential of our method. This means that we set aside the ith observation and carry out feature selection, stopping parameter estimation and classifier fitting by considering only the remaining $(n - 1)$ data points. We then predict $\widehat{Y}_i$, the class label of the ith observation and repeat this process for all observations in the training sample. Each observation is held out and predicted exactly once. We determine the test set error using symmetrically equal misclassification costs

$$Error = \frac{1}{n}\sum_{i=1}^{n} 1_{[Y_i \neq \widehat{Y}_i]}.$$

Table 1 reports test set errors with different gene subset size from feature selection for several classifiers. LogitBoost is reported with the optimal stopping time, yielding the minimal cross-validated error across the boosting iterations. This stopping time is not known in real life problems and results in an over-optimistic misclassification rate. Thus, also the error after a fixed number of 100 iterations as well as the error using our stopping parameter estimate from Equation (3) are given. A close competitor to LogitBoost is the discrete AdaBoost algorithm of Freund and Schapire (1996). We report its error rate after 100 iterations and observe that its accuracy is inferior to LogitBoost in 19 cases, equal in 11 cases and superior in 12 cases. LogitBoost thus seems to have an edge over AdaBoost, but this is far from being significant. To illustrate the benefit of boosting, we also ran the (optimally tuned) CART algorithm (Breiman *et al.*, 1984) to produce single classification trees. Boosting uses them as weak learners and leads to massive improvements in all except the estrogen and nodal datasets. As a benchmark method we applied the 1-nearest-neighbor classifier (Fix and Hodges, 1951) with simultaneous classification in multiclass problems, using all the genes from the one-against-all approach in conjunction with boosting. This simple rule is known to perform reasonably well on gene expression data in connection with precedent feature selection. For the smaller gene subsets, it is better than boosting for the leukemia and lymphoma data, at about the same level for the colon and NCI data and worse than boosting for the estrogen and nodal data. With larger gene subsets, if many noise variables are present, its accuracy often deteriorates severely.

It is known that repeated random splitting of the data into training and larger test sets may yield more accurate estimates of the test set error than leave one out cross validation, but the former has the disadvantage of being difficult to reproduce. In our setting, the error rates from random splitting (data not shown) were often at a somewhat higher level, but the relationship between the classifiers remained unchanged.

The choice of the stopping parameter for boosting is not very critical in all six datasets. Our classifier did not overfit much and Figure 1 shows that the error-rates are at, or close to the minimal error rate for many boosting iterations. We conjecture that stopping after a large, but arbitrary number of 100 iterations is a reasonable strategy in the context of gene expression data. Our data-driven approach for estimating the stopping parameters by cross validation on the training data does not improve and most often yields slightly worse results, probably due to additional random variation.

3.1.2 ROC curves In our evaluation, we determined the test set error using symmetrically equal misclassification costs. In a clinical setting, one often prefers to punish misclassifications asymmetrically, since false negative errors, i.e. classifying a tumorous tissue as normal can be

Table 1. Test set error rates based on leave one out cross validation for leukemia, colon, estrogen, nodal, lymphoma and NCI data with gene subsets from feature selection ranging between 10 to all genes for several classifiers. LogitBoost error rates are reported with optimal stopping (minimum cross-validated error across iterations), after a fixed number of 100 iterations as well as with the estimated stopping parameter. The cross validation with estimated stopping parameters for the lymphoma and NCI data with all genes was not feasible

Leukemia	10	25	50	75	100	200	3571
LogitBoost, optimal	4.17%	2.78%	4.17%	2.78%	2.78%	2.78%	2.78%
LogitBoost, estimated	6.94%	5.56%	5.56%	4.17%	4.17%	5.56%	5.56%
LogitBoost, 100 iterations	5.56%	2.78%	4.17%	2.78%	2.78%	2.78%	2.78%
AdaBoost, 100 iterations	4.17%	4.17%	4.17%	4.17%	4.17%	2.78%	4.17%
1-nearest-neighbor	4.17%	1.39%	4.17%	5.56%	4.17%	2.78%	1.39%
Classification tree	22.22%	22.22%	22.22%	22.22%	22.22%	22.22%	23.61%
Colon	10	25	50	75	100	200	2000
LogitBoost, optimal	14.52%	16.13%	16.13%	16.13%	16.13%	14.52%	12.90%
LogitBoost, estimated	22.58%	19.35%	22.58%	20.97%	22.58%	19.35%	19.35%
LogitBoost, 100 iterations	14.52%	22.58%	22.58%	19.35%	17.74%	16.13%	16.13%
AdaBoost, 100 iterations	16.13%	24.19%	24.19%	17.74%	20.97%	17.74%	17.74%
1-nearest-neighbor	17.74%	14.52%	14.52%	20.97%	19.35%	17.74%	25.81%
Classification tree	19.35%	22.58%	29.03%	32.26%	27.42%	14.52%	16.13%
Estrogen	10	25	50	75	100	200	7129
LogitBoost, optimal	4.08%	4.08%	2.04%	2.04%	2.04%	4.08%	2.04%
LogitBoost, estimated	6.12%	6.12%	6.12%	6.12%	6.12%	6.12%	6.12%
LogitBoost, 100 iterations	8.16%	6.12%	6.12%	4.08%	4.08%	8.16%	6.12%
AdaBoost, 100 iterations	8.16%	8.16%	2.04%	2.04%	6.12%	4.08%	4.08%
1-nearest-neighbor	4.08%	8.16%	18.37%	12.24%	14.29%	14.29%	16.33%
Classification tree	4.08%	4.08%	4.08%	4.08%	4.08%	4.08%	4.08%
Nodal	10	25	50	75	100	200	7129
LogitBoost, optimal	16.33%	18.37%	22.45%	22.45%	22.45%	18.37%	20.41%
LogitBoost, estimated	22.45%	30.61%	30.61%	34.69%	28.57%	26.53%	24.49%
LogitBoost, 100 iterations	18.37%	20.41%	26.53%	42.86%	42.86%	18.37%	22.45%
AdaBoost, 100 iterations	18.37%	16.33%	28.57%	40.82%	36.73%	22.45%	28.57%
1-nearest-neighbor	18.37%	30.61%	30.61%	42.86%	36.73%	36.73%	48.98%
Classification tree	22.45%	20.41%	20.41%	20.41%	20.41%	20.41%	20.41%
Lymphoma	10	25	50	75	100	200	4026
LogitBoost, optimal	1.61%	3.23%	1.61%	1.61%	1.61%	3.23%	8.06%
LogitBoost, estimated	3.23%	3.23%	3.23%	1.61%	3.23%	3.23%	-%
LogitBoost, 100 iterations	1.61%	3.23%	1.61%	1.61%	1.61%	3.23%	8.06%
AdaBoost, 100 iterations	4.84%	3.23%	1.61%	1.61%	1.61%	1.61%	3.23%
Nearest neighbor	1.61%	0.00%	0.00%	0.00%	0.00%	1.61%	1.61%
Classification tree	22.58%	22.58%	22.58%	22.58%	22.58%	22.58%	25.81%
NCI	10	25	50	75	100	200	5244
LogitBoost, optimal	32.79%	31.15%	27.87%	22.95%	26.23%	24.59%	31.15%
LogitBoost, estimated	36.07%	44.26%	36.07%	39.34%	44.26%	47.54%	-%
LogitBoost, 100 iterations	37.70%	44.26%	34.43%	29.51%	26.23%	24.59%	36.07%
AdaBoost, 100 iterations	50.82%	37.70%	34.43%	29.51%	32.79%	29.51%	36.07%
Nearest neighbor	36.07%	29.51%	27.87%	24.59%	22.95%	22.95%	27.87%
Classification tree	70.49%	68.85%	65.57%	65.57%	60.66%	62.30%	62.30%

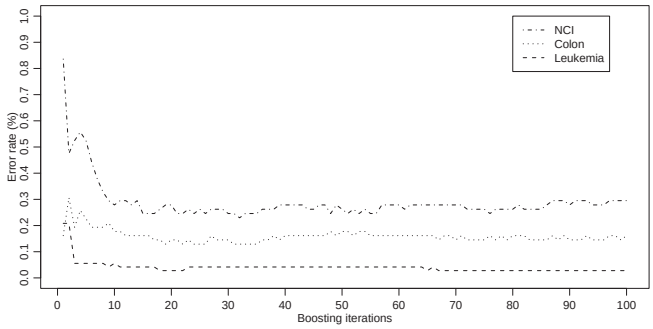

Fig. 1. Test set error curves for leukemia, colon and NCI data. The number of genes was chosen such that the performance was optimal: 75 for leukemia and NCI data, and 2000 for the colon data. The error curves for estrogen, nodal and lymphoma data look similar and are not displayed for reasons of clarity.

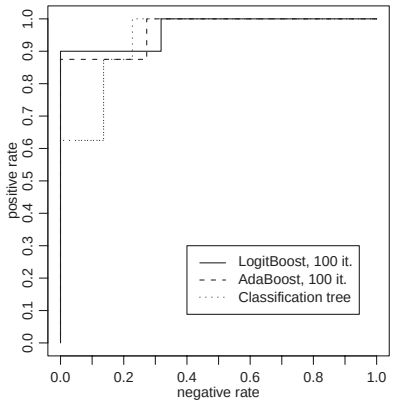

Fig. 2. ROC curves for LogitBoost, AdaBoost and classification trees applied on colon data without feature preselection. On the x-axis is the fraction of negative examples classified as positives (tumorous), the y-axis show the fraction of positive examples classified as positives. Each point on the curves represents the fractions achieved with a particular probability $\beta \in [0, 1]$ as threshold for positive classification. The probabilities for class membership were estimated by leave-one-out cross validation.

fatal, whereas false positive errors, i.e. predicting a normal tissue as a tumor may be less serious since in this case additional tests will be carried out.

ROC curves illustrate how accurate classifiers are under asymmetric losses, by plotting the tradeoff between false positives and false negatives. Each point on the two-dimensional ROC curve corresponds to a particular probability $\beta \in [0, 1]$ that was used as a threshold for positive (tumorous) classification. The (x, y) coordinates of each point are then the fractions of negative and positive samples that are classified as positive with this particular threshold β. In the ideal case, the ROC curve goes through $(0, 1)$, the upper left corner of the plot.

Figure 2 shows the ROC curves for LogitBoost after 100 iterations, AdaBoost after 100 iterations and classification trees applied to the colon data with $\widetilde{p} = 2000$ predictor variables. The class membership probabilities for each sample were determined by leave one out cross validation. We can see that both boosting classifiers yield a similar curve which comes closer to the ideal ROC curve than the one from classification trees. Note that this is a case where the test set errors under equal misclassification losses are very similar. However for this example, Boosting has an advantage for small negative rates.

3.1.3 Validation of the results The leukemia dataset has been considered by many authors. On the original test set comprising 34 observations, LogitBoost assigns the correct label to 33 of the 34 patients. This can be directly compared to the study in Golub *et al.* (1999), where 29 observations were classified correctly by their weighted voting scheme. Furey *et al.* (2000), working with support vector machines, report results ranging between 30 to 32 correct classifications. Ben-Dor *et al.* (2000) applied AdaBoost and carried out cross validation. After 10 000 boosting iterations, they obtained 2.78% misclassified and 1.39% unclassified instances without feature preselection, and 1.39% either mis- or unclassified instances with several gene subsets.

The colon dataset has been cross validated by Ben-Dor *et al.* (2000) with various classifiers, with and without precedent feature selection. AdaBoost performed comparably bad and the best result they report are 17.74% of misclassified, plus another 9.68% of unclassified instances. Our best results here are in the range between 12.90% and 14.52% wrongly classified observations. We gain evidence that LogitBoost can be superior over AdaBoost in some cases. The best support vector machine of Furey *et al.* (2000) misclassified only 6 tissue samples in the full cross validation cycle, being equivalent to an error-rate of 9.68%, whereas our error-rate of 12.9% corresponds to 8 misclassifications.

The NCI dataset has been extensively analyzed by Dudoit *et al.* (2002). They tried several classification methods including AdaBoost on a precedently reduced feature space. Also in their study, AdaBoosting was not among the best classifiers with a median error of about 48% in 150 random divisions in training and test set. Our method with reduction to binary problems and LogitBoost shows a considerable improvement to an error of only 22.9%, but a part of this reduction could be caused by the two different setups, i.e. random divisions versus cross validation for estimating the test set error.

For the estrogen and nodal datasets, we obtain better predictions than West *et al.* (2001) with their Bayesian approach, even without omitting the most difficult cases

as they do. A validation of the results for the lymphoma dataset in comparison to other studies is not possible. Since our classifier does well with respect to the benchmarks, we expect that it yields competitive results here too.

3.2 Simulation

Due to the scarcity of samples in real datasets, relevant differences between classification methods may be difficult to detect. We consider here simulated gene expression data: by generating large test sets, the performance of our modified LogitBoost classifier can be much more accurately compared against the benchmark classifiers and assessing significant differences becomes possible. We start by producing gene expression profiles from a multivariate normal distribution, $X \sim \mathcal{N}_p(0, \Sigma)$, where the covariance structure Σ is from the colon dataset. This reflects the real situation with microarray data, yielding gene expression profiles with $p = 2000$ genes. We continue by assigning one out of two response classes to the simulated expression profiles according to $Y \mid X = x \sim Bernoulli(p(x))$, where the conditional probabilities are from the model

$$\log \frac{p(x)}{1-p(x)} = \sum_{j=1}^{10} \beta_j \bar{x}^{(\mathcal{C}_j)} \left(1 + \gamma_j \bar{x}^{(\mathcal{C}_j)}\right) \left(1 + \delta_j \bar{x}^{(\mathcal{C}_j)}\right)$$

The $\bar{x}^{(\mathcal{C}_j)} = \sum_{g \in \mathcal{C}_j} x^{(g)}/|\mathcal{C}_j|$ are mean values across random gene clusters $\mathcal{C}_j \subseteq \{1, \ldots, p\}$ of uniformly random size between 1 and 10 genes, the expected number of relevant genes is thus $10 \cdot 5.5 = 55$. The model coefficients β_j, γ_j and δ_j were randomly drawn from normal distributions with zero mean and standard deviation $\sigma = 2, 1$ and 1/2, respectively. This leads to a complex non-additive decision boundary, where LogitBoost with stumps, which fits an additive model, is not in favor of the benchmark classifiers[†].

The training sample size was chosen to be $n = 200$ and we considered the performance of the classifiers on single but large test sets comprising 1000 new observations. The process was independently repeated 20 times, which enables to explore whether LogitBoost yields significantly better test set error-rates than the benchmark classifiers by performing paired Wilcoxon signed rank tests for the hypothesis of equal misclassifications against the two-sided alternative. The test always points towards better accuracy of LogitBoost, results are given in Table 2.

Not only when the LogitBoost algorithm was optimally stopped, but also after a fixed number of 150 iterations (which was found to be a reasonable ad-hoc choice for this problem) it significantly outperformed the benchmark methods. This confirms our findings from real data that our classifier is more accurate than the benchmarks.

[†] LogitBoost with larger trees would allow to pick up nonadditive decision boundaries.

Table 2. Percentual improvement and p-values of LogitBoost (stopped optimally and after a fixed number of 150 iterations) against the generic 1-nearest-neighbor method and classification trees in 20 independent realizations from our simulation model. The p-values are from paired two-sided Wilcoxon signed rank tests for equal test set error and are always in favor of LogitBoost

	1-Nearest-Neighbor	
LogitBoost, optimal	12.37%,	$p = 1.7 \cdot 10^{-4}$
LogitBoost, 150 iterations	7.54%,	$p = 1.4 \cdot 10^{-3}$
	Classification Tree	
LogitBoost, optimal	10.21%,	$p = 1.1 \cdot 10^{-3}$
LogitBoost, 150 iterations	5.27%,	$p = 1.7 \cdot 10^{-2}$

4 CONCLUSIONS

We propose modifications and extensions of boosting classifiers for microarray gene expression data from several tissue or cancer types. We applied precedent feature selection and used the more robust LogitBoost combined with an alternative approach for binary problems. The results on six real and a simulated datasets indicate that these modifications are successful and make boosting a competitive player for predicting expression data. Our feature preselection generally improved the predictive power of a classifier. Moreover, we observed slightly better performance of LogitBoost over AdaBoost, and our whole procedure (feature selection plus LogitBoost) compares favorably with previously published results using AdaBoost. Finally, we propose to reduce multiclass problems to multiple binary problems which are solved separately. This was found to have a great potential for more accurate results on gene expression data, where the choice of predictor variables is crucial.

Our LogitBoost classifier is very suitable for application in a clinical setting. In comparison to other methods, it yields good results, is easy to implement and does not require sophisticated tuning and model or kernel selection as with neural networks or support vector machines. Unlike several other classifiers, it directly provides class membership probabilities. They are essential to quantify the uncertainty of a class label assignment and allow decisions under unequal misclassification costs which are often encountered in practice.

ACKNOWLEDGEMENTS

Many thanks are due to Rainer Spang, Mike West and Joe Nevins for providing the estrogen and nodal datasets, and to Jane Fridlyand for providing the preprocessed NCI data.

REFERENCES

Alizadeh,A., Eisen,M., Davis,R., Ma,C., Lossos,I., Rosenwald,A., Boldrick,J., Sabet,H., Tran,T., Yu,X. *et al.*, (2000) Distinct types of diffuse large B-cell-lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Alon,U., Barkai,N., Notterman,D., Gish,K., Mack,S. and Levine,J. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, **96**, 6745–6750.

Allwein,E., Schapire,R. and Singer,Y. (2000) Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Machine Learning Res.*, **1**, 113–141.

Ben-Dor,A., Bruhn,L., Friedman,N., Nachman,I., Schummer,M. and Yakhini,Z. (2000) Tissue classification with gene expression profiles. *J. Comput. Biol.*, **7**, 559–583.

Breiman,L., Friedman,J., Olshen,R. and Stone,C. (1984) *Classification and Regression Trees*. Wadsworth, Belmont, CA.

Breiman,L. (1999) Prediction games & arcing algorithms. *Neural Computation*, **11**, 1493–1517.

Dudoit,S., Fridlyand,J. and Speed,T. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *JASA*, **97**, 77–87.

Fix,E. and Hodges,J. (1951) Discriminatory Analysis—Nonparametric Discrimination: Consistency Properties. *US Air Force School of Aviation Medicine*, Random Field, Texas. Published in Agrawala, A., (1977), Machine Recognition of Patterns, Report No. 4, IEEE Press, New York.

Freund,Y. and Schapire,R. (1996) Experiments with a New boosting algorithm. *Machine Learning: Proceedings to the Thirteenth International Conference*. Morgan Kaufmann, San Francisco, pp. 148–156.

Freund,Y. and Schapire,R. (1997) A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. System Sci.*, **55**, 119–139.

Friedman,J., Hastie,T. and Tibshirani,R. (2000) Additive logistic regression: a statistical view of boosting. *Ann. Stat.*, **28**, 337–407 (with discussion).

Furey,T., Cristianini,N., Duffy,N., Bednarski,D., Schummer,M. and Haussler,D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

Golub,T., Slonim,D., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J., Coller,H., Loh,M., Downing,J.*et al.*, (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Hampel,F., Ronchetti,E., Rousseeuw,P. and Stahel,W. (1986) *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York.

Hastie,T. and Tibshirani,R. (1998) Classification by pairwise coupling. *Ann. Stat.*, **26**, 451–471.

Hastie,T., Tibshirani,R. and Friedman,J. (2001) *The Elements of Statistical Learning*. Springer, New York.

Quinlan, (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco.

Park,P., Pagano,M. and Bonetti,M. (2001) A nonparametric scoring algorithm for identifying informative genes from microarray data. *Pac. Symp. Biocomput.*, **6**, 52–63.

Ripley,B. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.

Ross,D., Scherf,U., Eisen,M., Perou,C., Rees,C., Spellman,P., Iyer,V., Jeffrey,S., Van de Rijn,M.*et al.*, (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Genet.*, **24**, 227–235.

Slonim,D., Tamayo,P., Mesirov,J., Golub,T. and Lander,E. (2000) Class prediction and discovery using gene expression data. In *Proceedings of the 4th International Confererence on Computational Molecular Biology*. Universal Academy Press, Tokyo, Japan, pp. 263–272.

Spang,R., Blanchette,C., Zuzan,H., Marks,J., Nevins,J. and West,M. (2001) Prediction and uncertainty in the analysis of gene expression profiles. In Wingender,E., Hofestädt,R. and Liebich,I. (eds), *Proceedings of the German Conference on Bioinformatics GCB 2001*. Braunschweig.

West,M., Blanchette,C., Dressman,H., Huang,E., Ishida,S., Spang,R., Zuzan,H., Olson,J., Marks,J. and Nevins,J. (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. *PNAS*, **98**, 11462–11467.

Zhang,H., Yu,C., Singer,B. and Xiong,M. (2001) Recursive partitioning for tumor classification with gene expression microarray data. *PNAS*, **98**, 6730–6735.