Original Paper

GroupAdaBoost: Accurate Prediction and Selection of Important Genes

Takashi Takenouchi,† Masaru Ushijima†† and Shinto Eguchi†††

In this paper, we propose GroupAdaBoost which is a variant of AdaBoost for statistical pattern recognition. The objective of the proposed algorithm is to solve the "$p \gg n$" problem arisen in bioinformatics. In a microarray experiment, gene expressions are observed to extract any specific pattern of gene expressions related to a disease status. Typically, p is the number of investigated genes and n is number of individuals. The ordinary method for predicting the genetic causes of diseases is apt to over-learn from any particular training dataset because of the "$p \gg n$" problem. We observed that GroupAdaBoost gave a robust performance for cases of the excess number p of genes. In several real datasets which are publicly available from web-pages, we compared the analysis of results among the proposed method and others, and a small scale of simulation study to confirm the validity of the proposed method. Additionally the proposed method effectively worked for the identification of important genes.

1. Introduction

Microarray technology allows us to monitor several thousand gene expressions related to a disease status in a single experiment. This technology leads us beyond the usual assumptions of conventional statistical analysis and this poses a serious problem. There are two main objectives with microarray analysis. One is discriminant analysis (supervised learning), which aims to predict the unknown class label of a new individual from a monitored gene expression profile. The other is the identification of those marker genes (variable selection) that characterize the different disease classes. For the discriminant analysis, there are many proposed methods such as Fisher discriminant analysis, support vector machine and machine learning methods as bagging or Boosting.

Boosting method constructs a classification machine by combining a lot of weak classification machines and its learning process is implemented by sequentially reweighting all the examples according to classification results. The typical Boosting algorithm AdaBoost was compared with other methods and reported that AdaBoost does not yield many impressive results [2),4)]. LogitBoost (see Friedman, et al. [5)]), which is a variation of AdaBoost, is applied and some results is obtained for publicly available datasets by Dettling and Bühlmann [3)].

The serious problem with classification from gene expression profiles is that the sample size n is generally too small relative to the number p of monitored genes. Many gene expressions are considered to be non-differentially expressed across the sample and do not give any important information. This superabundance of information on gene expression makes it difficult to get any useful predictive results. Moreover, there are several solutions for prediction, which situation inhibits the building of medical knowledge from the analysis. That p is extremely huge makes conventional classification algorithms in-executable: the training dataset often can be completely learned with the training error 0 even when any gene expressions do not have information. To avoid those problem, conventional methods apply a variable selection method to expression data for the reduction of gene expression and then only those selected gene expressions are used to construct a classification machine for producing the diagnosis. However, this can be problematic because the set of all the genes involves a considerable number of non-informative genes and the preprocedure often falls into a difficult situation in which there are no evident separations between important genes and unimportant genes. Once a gene expression has been truncated by the pre-procedure, the information is never utilized in the prediction procedure. AdaBoost can be applied to microarray data without any preselection but does not sufficiently catch important expressions. Because the usual Boosting

† Graduate School of Information Science, Nara Institute of Science and Technology
†† Genome Center, Japanese Foundation for Cancer Research
††† Institute of Statistical Mathematics, Japan and Department of Statistical Science, Graduate University of Advanced Studies

method selects the best representative expression as a classification machine in each learning step, the algorithm tends to look over important genes having similar expressions over individuals. For example, in our data analysis, AdaBoost selects only 15 genes for classification of the well-known publicly dataset, Leukemia and this result by AdaBoost gives less information than that of Golub, et al. [6]. However in their naive analysis, the set of 50 genes was suggested to have an association with leukemic diseases.

To overcome the above difficulty, we propose a new Boosting algorithm called GroupAdaBoost where the similarity between gene expressions is positively incorporated into AdaBoost. GroupAdaBoost deals with genes having similar performance as a group and enables us to jointly execute a selection of important genes and design of a classification machine without pre-selection. Consequently, GroupAdaBoost can select important genes that are highly influential for the classification of diseases.

2. GroupAdaBoost

2.1 Framework of Boosting

We focus on the binary classification problem. See Mclachlan [7] for extensive discussion for classification methods. Let us assume a set of training dataset $D = \{(\boldsymbol{x}_i, y_i) : i = 1, \cdots, n\}$, where $\boldsymbol{x} \in \mathbf{R}^p$ is an input vector and $y \in \{1, -1\}$ is a class label. In our context the input vector $\boldsymbol{x}$ has expression profiles of p monitored genes as the components, y represents a disease status of an individual having expression $\boldsymbol{x}$. Typically, the sample size n is about several tens, and there are several thousands or larger of monitored genes, p. The aim of the classification problem is to construct a classification machine $F : \boldsymbol{x} \to \mathbf{R}$ that minimizes a misclassification error $\Pr(\mathrm{sgn}(F(\boldsymbol{x})) \neq y)$.

In the context of Boosting, we intentionally use only weak classification machines, $f(\boldsymbol{x}) \in \{1, -1\}$. We employ a decision stump as a weak classification machine:

$$f(\boldsymbol{x}; j, b) = \mathrm{sgn}(x_j - b),$$

where $j = 1, \cdots, p$ and b is a threshold value in a range of the j-th gene expression profile x_j. This implies that the classification machine $f(\boldsymbol{x}; j, b)$ determines the label for an input $\boldsymbol{x}$ whether the quantity of the expression level of j-th gene, x_j is larger than the threshold b or not. The decision stump has a preferable property for analyzing gene expression data: the decision result is dependent on only ranking of the expression levels x_j and thus the stump is invariant for any kind of monotone transformation in pre-processing such as centralization or log transformation. Let $\mathcal{F}$ be a set of weak classification machines.

$$\mathcal{F} = \{f(\boldsymbol{x}; j, b) | j = 1, \cdots, p, b \in \mathbf{R}\}.$$

The Boosting method aims to construct a strong classification machine $\mathrm{sgn}(F_T(\boldsymbol{x}))$ by linearly combining weak classification machines as

$$F_T(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t f(\boldsymbol{x}; j_t, b_t), \tag{1}$$

where j_t is the gene number related in the t-step and b_t is the optimized threshold. The derivation of (α_t, j_t, b_t) will be given in a subsequent discussion. This discriminant function is viewed as a weighted majority vote of stumps associated with T gene expressions and an absolute value of $F_T(\boldsymbol{x})$ represents a degree of confidence concerned with the classification of $\boldsymbol{x}$. The prediction of label is decided by the sign of $F_T(\boldsymbol{x})$. A typical Boosting algorithm is AdaBoost, which is derived from a sequential minimization of the exponential loss function

$$L_{\exp}(F) = \frac{1}{n} \sum_{i=1}^{n} \exp(-F(\boldsymbol{x}_i) y_i). \tag{2}$$

We note that, if in the i-th example $F(\boldsymbol{x}_i)$ has the same sign as y_i, it gives less influence for the exponential loss than that with distinct sign. Thus the minimization of $L_{\exp}(F)$ with respect to F qualitatively aims at matching the signs of y_i and $F(\boldsymbol{x}_i)$ over the training dataset. AdaBoost works efficiently for usual classification problems but is not appropriate for gene expression datasets. In the present dataset, there are too many feature variables comparing with n and some of these have similar information. Because AdaBoost selects only one variable in one step, other variables with similar information are abandoned. To overcome this problem, we propose GroupAdaBoost, as a simple modification of AdaBoost in a learning step. We will demonstrate its performance using real datasets, and discuss the theoretical considerations.

2.2 GroupAdaBoost Algorithm

In this section, we introduce the algorithm GroupAdaBoost. For this, we overview the learning step of algorithm of AdaBoost, which consists of the following three procedures and

is proceeded by a sequentially updated weight distribution for examples. At first, the algorithm selects the best weak classification machine having minimum weighted error rate. Secondly, a coefficient for the selected classification machine is calculated according to the performance of the selected machine. Thirdly, the weight distribution is updated as putting a high weight into only misclassified examples. These three procedures are updated and finally provide the discriminant function (1). Let us define GroupAdaBoost we propose. It follows almost the same procedures except for the first procedure. AdaBoost selects the best machine in terms of weighted error rates, whereas GroupAdaBoost selects a group of G classification machines as follows.

GroupAdaBoost(G)

(1) Set the initial condition as $w_1(i) = \frac{1}{n}$ and $F_0(\boldsymbol{x}) = 0$.

(2) For $t = 1, \cdots, T$.

(a) Select a weak machine for the j-th gene as

$$f_t(x_j; b_j) = \underset{f \in \mathcal{F}_j}{\operatorname{argmin}}\, \varepsilon_t(f),$$

where $\mathcal{F}_j = \{f(x_j; b); b \in \mathbf{R}\}$ is a set of weak classification machines associated with j-th gene and $\varepsilon_t(f)$ is a weighted error rate,

$$\varepsilon_t(f) = \sum_{i=1}^{n} w_t(i) \mathrm{I}(f(\boldsymbol{x}_i) \neq y_i).$$

Note that the threshold value b_j depends on the step number t. From a sequence of weak machines $\{f_t(x_1; b_1), \cdots, f_t(x_p; b_p)\}$, extract G weak classification machines in the order of their weighted error rate, with the smallest first,

$$\{f_t(x_{(1)}; b_{(1)}), \cdots, f_t(x_{(G)}; b_{(G)})\},$$

where the subscript (g) denotes the gene number of the g-th smaller weighted error rate. Thus, this is a group of the G best-weighted error rates.

(b) For the extracted weak classification machine, $f_t(x_{(g)}; b_{(g)})(g = 1, \cdots, G)$, calculate the coefficient

$$\alpha_{t,(g)} = \frac{1}{2} \log \frac{1 - \varepsilon_t(f_t(\cdot; b_{(g)}))}{\varepsilon_t(f_t(\cdot; b_{(g)}))},$$

and construct the t-th machine as

$$\bar{f}_t(\boldsymbol{x}) = \frac{1}{G} \sum_{g=1}^{G} \alpha_{t,(g)} f_t(x_{(g)}; b_{(g)}).$$

(c) Update a weight distribution as

$$w_{t+1}(i) = \frac{w_t(i) \exp(-\bar{f}_t(\boldsymbol{x}_i) y_i)}{Z_t},$$

where

$$Z_t = \sum_{i=1}^{n} w_t(i) \exp(-\bar{f}_t(\boldsymbol{x}_i) y_i),$$

and update the discriminant function as

$$F_t(\boldsymbol{x}) = F_{t-1}(\boldsymbol{x}) + \bar{f}_t(\boldsymbol{x}).$$

(3) Output the function

$$F_T(\boldsymbol{x}) = \sum_{t=1}^{T} \bar{f}_t(\boldsymbol{x}).$$

If we set $G = 1$, then GroupAdaBoost reduces to the usual AdaBoost analysis. In the first procedure, GroupAdaBoost can be expected to catch a group of classification machines having similar properties or equivalent genes comparable with the best machine. The constant, G, is typically determined to be a number required by medical information and the number within several tens is appropriate in our analysis. Alternatively, we can choose classification machines by other measures, for example, from a range of weighted error rates. The coefficient $\alpha_{t,(g)}$ calculated in the procedure (b) is the same with AdaBoost and becomes higher as the weighted error rate gets lower. The discriminant function, $\bar{f}_t(\boldsymbol{x})$ of step t is constructed for the weighted majority vote. The sign of $\bar{f}_t(\boldsymbol{x})$ means a predicted label for the input $\boldsymbol{x}$, and the absolute value of $\bar{f}_t(\boldsymbol{x})$ represents a confidence of classification. In the procedure (c), the weight distribution is updated according to classification results and its confidences. A weight of an example with a high confidence is exponentially updated but its update is moderated if $\bar{f}_t(\boldsymbol{x})$ has a low confidence. By this update rule, GroupAdaBoost sequentially reinforces the discriminant function $F_t(\boldsymbol{x})$. As a result, GroupAdaBoost jointly executes accurate classification and efficient correction of important gene expressions.

2.3 Loss Function Related with GroupAdaBoost

We now consider a relation between the GroupAdaBoost algorithm and the exponential loss function (2). Procedures of GroupAdaBoost are derived from a approximate minimization of the exponential loss function while procedures of AdaBoost are derived from exactly sequential minimization. Let us assume that we obtain the discriminant function $F_{t-1}(\boldsymbol{x})$ and consider an update from $F_{t-1}(\boldsymbol{x})$ to $F_t(\boldsymbol{x})$, where

$$F_t(\boldsymbol{x}) = F_{t-1}(\boldsymbol{x}) + \frac{1}{G}\sum_{g=1}^{G}\alpha_{t,(g)} f_t(x_{(g)}; b_{(g)}). \quad (3)$$

From the convexity of the exponential function, we obtain the following inequality:

$$\begin{aligned} & L_{exp}(F_t) \\ & \leq \frac{1}{G}\sum_{g=1}^{G} L_{exp}(F_{t-1} + \alpha_{t,(g)} f_t(x_{(g)}; b_{(g)})) \\ & \leq L_{exp}(F_{t-1}). \end{aligned} \quad (4)$$

This shows that the loss function monotonically decreases by the update (3). A minimizer of the exponential function is equivalent to the Bayes rule which is the optimal discriminant function and minimizes the naive error rate. See Murata, et al. [8] for detailed discussion. GroupAdaBoost approximately minimizes the exponential loss function that is updated by a set of weak classification machines. Note that GroupAdaBoost does not directly minimize $L_{exp}(F)$.

2.4 Score of a Gene

When the discriminant function $F_T(\boldsymbol{x})$ is obtained, we define a score for x_j associated with the j-th gene as

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{g=1}^{G} \mathrm{I}((g) = j)\alpha_{t,(g)}. \quad (5)$$

Note that the number (g) defined in step (a) of the algorithm in Section 2.2 implicitly depends on t. The score value implies the contribution of classification machines associated with the j-th gene expression per one step or total confidence of j-th gene expression. In Section 3.5, we will discuss a selection of important genes based on the score.

2.5 Choice of the Learning Step Number

GroupAdaBoost and any other Boosting method, including LogitBoost, are apt to overfit the training dataset unless the algorithm is stopped at an appropriate step. See Takenouchi and Eguchi [9]. Thus, the number of learning steps T should be estimated. If we had sufficient examples, we could set aside a test dataset and use it to assess the performance of a method. Now the sample size n of gene expression dataset is typically small compared with p, we employ a K-fold cross validation technique. If we set $K = n$, this reduces to the leave one out cross validation employed in many data analysis. But the leave one out cross validation does not work well as an estimator of the generalization error, we use another value of K, typically 10 as in Ambroise and McLachlan [1].

First, we divide the training dataset into K roughly equal-sized sets $D_1, \cdots, D_K$ in which each D_k is a subset of D and satisfies $D = D_1 \cup \cdots \cup D_K$ and $D_j \cap D_k = \emptyset$ for any different $j, k \in \{1, \cdots, K\}$. Second, we run GroupAdaBoost on the dataset without D_k and construct the classification machines $F_t^{(-k)}(\boldsymbol{x})$ $(t = 1, \cdots, T)$. We calculate the misclassification rate $\epsilon(F_t^{(-k)}; D_k)$ on D_k and do this for $k = 1, \cdots, K$. Finally, we compute the cross validation error at step t by averaging the K estimates of misclassification rate as

$$\epsilon(t) = \frac{1}{K}\sum_{k=1}^{K}\epsilon(F_t^{(-k)}; D_k).$$

Note that, if we set $K = n$, the above method means the leave one out cross-validation. An optimal learning step is determined as a point that minimizes $\epsilon(t)$.

2.6 Experiment with a Synthetic Dataset

In this subsection, we investigate the performance of GroupAdaBoost with a synthetic dataset. In particular, we want to investigate the relationship between the number of groups, G, and a number of feature vectors giving important information. Assume that the feature vector $\boldsymbol{x}$ is uniformly distributed on $[-1, 1]^p$ and the conditional probability of y is in the logistic model,

$$p(y|\boldsymbol{x}) = \frac{1}{1 + \exp(-2yF^*(\boldsymbol{x}))}, \quad (6)$$

where $F^*(\boldsymbol{x}) = F^1(\boldsymbol{x}) + F^0(\boldsymbol{x})$ and

$$F^1(\boldsymbol{x}) = \sum_{s=1}^{u} 10x_s,$$

$$F^0(\boldsymbol{x}) = \sum_{s=u+1}^{p} \frac{0.1}{p}x_s.$$

Under the above setting, the Bayes rule of (6) is $F^*(\boldsymbol{x})$ and is mainly influenced by $F^1(\boldsymbol{x})$ because the rule is determined by the sign of $F^*(\boldsymbol{x})$. See McLachlan [7] for detail discussion for Bayes rule. Now, let us consider the relationship between u and G. We generated 20 sets of training datasets containing 50 examples and a test dataset containing 2000 examples for $p = 1000$, $u = 10, 20, 50$. The number of learning steps T is estimated by the 10-fold cross-validation with each training dataset for

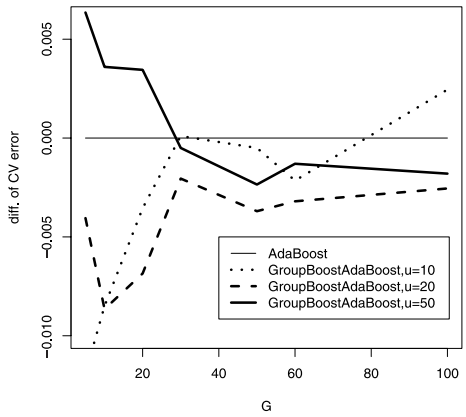

Fig. 1 The difference in the mean cross-validation error rate against G. The level 0 corresponds to AdaBoost. The dataset was generated by (6) and the dimension p of feature vector was 1000.

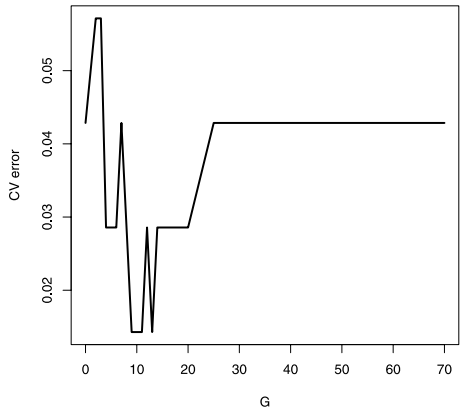

Fig. 2 The mean of 10-fold cross-validation error rate against G for the leukemia dataset. The point $G = 1$ indicates AdaBoost.

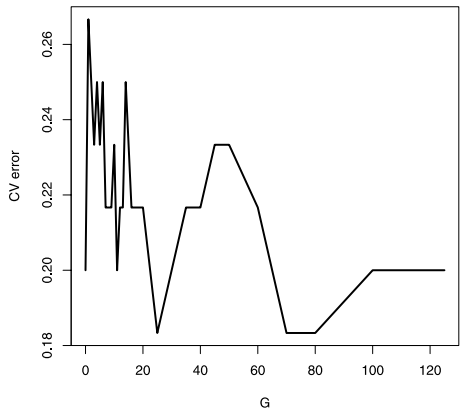

Fig. 3 The mean of 10-fold cross-validation error rate against G for colon datasets. The point $G = 1$ indicates AdaBoost.

the fixed G. **Figure 1** shows the result. Differences in the cross-validation error rate between AdaBoost and GroupAdaBoost for fixed G are shown in the figure. The level 0 indicates AdaBoost; if lines are under the level 0, then GroupAdaBoost is superior to AdaBoost. If we appropriately choose G, GroupAdaBoost is superior to AdaBoost. We could observe that the number G that minimizes the cross-validation error is near to the number u of influential feature variables.

3. Results

We applied GroupAdaBoost to three publicly available real datasets. A preliminary experimental analysis was performed in Takenouchi, et al. [10]. The test error rate was estimated from the 10-fold cross-validation. Note that the stopping parameter T was also determined by the 10-fold cross-validation for a dataset without validation examples. Consequently, we performed two sequences of cross-validations: one was to estimate the generalization performance of GroupAdaBoost, and the other was to estimate the optimal stopping point T.

3.1 Leukemia

We explored the performance of our method on a leukemia dataset. This dataset contains gene expression data from patients suffering from acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML). The dataset consisted of 72 observations and a feature vector $\boldsymbol{x}$ containing 7129 variables. More information about this dataset can be found in Golub, et al. [6].

3.2 Colon

This dataset contains 2000 gene expressions of 62 patients, with 40 tumor and 22 normal colon tissues, measured using Affymetrix gene chip technology. This dataset is available at http://microarray.princeton.edu/oncology/.

3.3 Estrogen

This dataset monitors 7129 genes in 49 breast tumor samples. The dataset is available at http://mgm.duke.edu/genome/dna_micro/work/ and was obtained using Affymetrix gene chip technology. The label describes the status of the estrogen receptor (ER), in which 25 samples are positive (ER+) and 24 are negative (ER−).

3.4 Discussion

The 10-fold cross-validation error rate against G for each dataset, is shown in **Fig. 2**, **Fig. 3** and **Fig. 4**, respectively. We observed that the validation error rate was minimized at a rela-

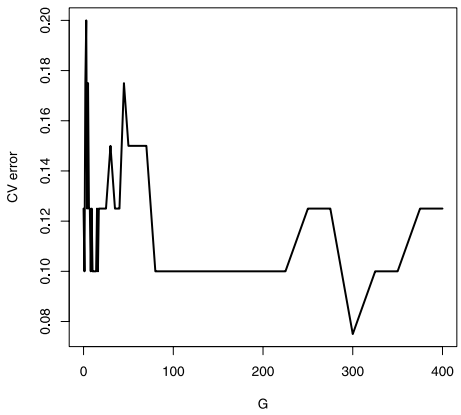

Fig. 4 The mean of 10-fold cross-validation error rate against G for estrogen datasets. The point $G = 1$ indicates AdaBoost.

tively large G for all datasets. We also observed that the mean training error is minimized near $G = 1$, which supports AdaBoost or GroupAdaBoost with a small G. We observed that AdaBoost apt to fall into over-learning; GroupAdaBoost escapes from the over-learning thanks to the improved step in the algorithm. Note that an extremely large G also loads to over-learning and G of several tens seems to be appropriate. GroupAdaBoost worked much better than AdaBoost in the sense of the estimated test error and this result is comparable to that of Dettling and Bühlmann [3], using the intensive preselection of genes. For the leukemia dataset, GroupAdaBoost wrongly classified only one example through the validation process.

3.5 Selection of Important Genes

Another objective of this paper was to select important genes as the weak learners. To verify the importance of selected genes, we refer to the original papers. Here, we discuss only the leukemia and estrogen datasets because there was no detailed information for important genes for the colon dataset in the original paper.

Table 1 and **Table 2** show the list of the top 30 genes for the leukemia and estrogen datasets identified by GroupAdaBoost with the highest score. The score for the classification machine is defined in Eq. (5) and we averaged over 10 classification machines, obtained by 10-fold cross validation.

For the leukemia dataset, we refer to Golub, et al. [6] who showed 50 genes as informative genes which were most closely correlated with AML-ALL class distinction. Twelve genes in Table 1 are listed in Golub, et al. [6]. In particular, the genes CD33 and MB-1 are known to be useful in distinguishing lymphoid from myeloid lineage cells, and so the genes likely to be associated with ALL-AML can be distinguished.

Table 1 Top 30 genes associated with disease from the leukemia dataset. The "∗" marked genes are also listed in 50 informative genes in Golub, et al. [6]. Gene names or symbols used in Golub, et al. [6] are written in parentheses.

GenBank ID	Gene Symbol
D88422	CSTA
J05243	SPTAN1
M11722	DNTTT
M23197*	CD33 (CD33)
M27891*	CST3 (Cystatin C)
M31166	TSG-14
M31303*	STMN1 (Op18)
M63379	TRPM-2
M68891	GATA2
M84526*	ADN (Adipsin)
M92287*	CCND3 (CCND3)
M96326*	AZU1 (Azurocidin)
U05259*	CD79A (MB-1)
U46499	MGST1
U88047	DRIL1
X62320	GRN
X62654	CD63
X90858	UPP1
X95735*	ZYX (Zyxin)
Y07604	NME4
J02783	P4HB
U22376*	MYB (C-myb)
L07807	DNM1
L09209	APLP2
M83652*	PFC (Properdin)
M83667	CEBPD
X85116*	STOM (Epb72)
U49020	MEF2A
L11672	ZNF91
M31523*	TCF3 (E2A)

For the estrogen dataset, we refer to West, et al. [11] who showed the list of the 40 genes most highly correlated with ER status. Fifteen genes in Table 2 are also listed in Table 1 of West, et al. [11], and 8 genes are in the protein synthetic pathway of ER or are involved in ER itself. For example, pS2, LIV-1, and GATA3 have already been reported to have a relationship with ER status in several articles.

Therefore, we can confirm that many important genes are included in our lists through the use of GroupAdaBoost. Thus, important genes can be selected accurately as those effective for sample classification.

4. Conclusions

We have proposed the new algorithm GroupAdaBoost for analyzing microarray problems under the difficult situation "$p \gg n$". The performance of the algorithm has been explored

Table 2 Top 30 genes associated with disease from the estrogen dataset. The marked genes are in the 40 genes list which are correlated with ER status, and the "**" marked genes are in the protein synthetic pathway of ER or are involved in ER itself. Gene symbols used in West, et al. [11] are written in parentheses.

GenBank ID	Gene Symbol
D38437*	PMS2L3 (PMS2L3)
J03778**	MAPT (MAPT)
J03827	NSEP1
L08044**	TFF3 (TFF3)
L12535	RSU-1
L17131	HMGA1
L26336	HSPA2
L40401	ZAP128
M29874*	CYP2B6 (CYP2B)
M33493	TPSB2
M99701	TCEAL1/TCEAL3
U05340	CDC20
U07919	ALDH1A3
U09770*	CRIP1 (CRIP1)
U22376**	MYB (MYB)
U41060**	SLC39A6 (LIV-1)
U42408	LAD1
U79293*	Clone 23948 (N/A)
X03635**	ESR1 (ESR1)
X13238*	COX6C (COX6C)
X17059*	NAT1 (NAT1)
X52003**	TFF1 (pS2)
X52947	GJA1
X55037**	GATA3 (GATA3)
X58072**	GATA3 (GATA3)
X76180	SCNN1A
X83425	LU
X87176	HSD17B4
X87212*	CTSC (CTSC)
Z48633	N/A

with publicly available real datasets and synthetic datasets. We observed that GroupAdaBoost overcomes the sensitivity of the preselection of genes and has a high generalization ability by applying the adaptive selection of tuning parameters. Additionally, the grouping and selection of weak classification machines by GroupAdaBoost effectively worked for the identification of important genes.

Acknowledgments A part of work was supported by Transdisciplinary Research Integration Center, Research Organization of Information and Systems.

References

1) Ambroise, C. and McLachlan, G.J.: Selection bias in gene extraction on the basis of microarray gene-expression data, *PNAS*, Vol.99, pp.6562–6566 (2002).
2) Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, J., Schummer, M. and Yakhini, Z.: Tissue classification with gene expression profiles, *Journal of Computational Biology*, Vol.7, pp.559–583 (2000).
3) Dettling, M. and Bühlmann, P.: Boosting for tumor classification with gene expression data, *Bioinformatics*, Vol.19, pp.1061–1069 (2003).
4) Dudoit, S., Fridlyand, J. and Speed, T.: Comparison of discrimination methods for the classification of tumors using gene expression data, *Journal of the American Statistical Association*, Vol.97, pp.77–87 (2002).
5) Friedman, J.H., Hastie, T. and Tibshirani, R.: Additive logistic regression: A statistical view of boosting, *Annals of Statistics*, Vol.28, pp.337–407 (2000).
6) Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, Vol.286, pp.531–537 (1999).
7) McLachlan, G.J.: *Discriminant analysis and statistical pattern recognition*, Wiley, New York (1992).
8) Murata, N., Takenouchi, T., Kanamori, T. and Eguchi, S.: Information geometry of *U*-boost and bregman divergence, *Neural Computation*, Vol.16, pp.1437–1481 (2004).
9) Takenouchi, T. and Eguchi, S.: Robustifying AdaBoost by adding the naive error rate, *Neural Computation*, Vol.16, pp.767–787 (2004).
10) Takenouchi, T., Ushijima, M, and Eguchi, S.: GroupAdaBoost for selecting important genes, In *BIBE 2005*, IEEE Computer Society, pp.218–221 (2005).
11) West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J.A., Marks, J.R. and Nevins, J.R.: Predicting the clinical status of human breast cancer by using gene expression profiles, *PNAS*, Vol.98, pp.11462–11467 (2001).

(Received July 31, 2006)
(Accepted September 27, 2006)

(Communicated by *Tetsuo Shibuya*)

Takashi Takenouchi received his Ph.D. degree in 2004 from the Department of statistical Science, Graduate University of Advanced Studies after having obtained the master degree in Engineering from University of Tokyo in 2001. His current position is postdoctoral fellow of Nara Institute of Science and Technology, Japan. His major interests are the learning theory and information geometry.

Masaru Ushijima received his M.Sc. degree in Engineering from the University of Tokyo, Japan in 1997. From April 2000 to March 2002, he was at Tokyo University of Science as an assistant professor. He is currently in the Bioinformatics group, Genome Center, Japanese Foundation for Cancer Research as a researcher. His major interest is bioinformatics, especially the methodology of genome-related data analysis.

Shinto Eguchi is graduated from Graduate School of Osaka University, and received from his Ph.D. from Hiroshima University in 1985. Now he is a professor of Institute of Statistical Mathematics and department of statistical Science, Graduate University of Advanced Studies. His major interests are the principle of statistical inference, machine learning theory and bioinformatics.