Biomedical Informatics Insights

ORIGINAL RESEARCH

Comparison of LDA and SPRT on Clinical Dataset Classifications

Chih Lee, Brittany Nkounkou and Chun-Hsi Huang

Computer Science and Engineering Department, University of Connecticut, Storrs, CT 06269, USA.
Corresponding author email: huang@engr.uconn.edu

Abstract: In this work, we investigate the well-known classification algorithm LDA as well as its close relative SPRT. SPRT affords many theoretical advantages over LDA. It allows specification of desired classification error rates α and β and is expected to be faster in predicting the class label of a new instance. However, SPRT is not as widely used as LDA in the pattern recognition and machine learning community. For this reason, we investigate LDA, SPRT and a modified SPRT (MSPRT) empirically using clinical datasets from Parkinson's disease, colon cancer, and breast cancer. We assume the same normality assumption as LDA and propose variants of the two SPRT algorithms based on the order in which the components of an instance are sampled. Leave-one-out cross-validation is used to assess and compare the performance of the methods. The results indicate that two variants, SPRT-ordered and MSPRT-ordered, are superior to LDA in terms of prediction accuracy. Moreover, on average SPRT-ordered and MSPRT-ordered examine less components than LDA before arriving at a decision. These advantages imply that SPRT-ordered and MSPRT-ordered are the preferred algorithms over LDA when the normality assumption can be justified for a dataset.

Keywords: clinical data classification, linear discriminant analysis, sequential probability ratio test, supervised learning

Biomedical Informatics Insights 2011:4 1–7

doi: 10.4137/BII.S6935

This article is available from http://www.la-press.com.

Introduction

Classification algorithms[1] find many applications in analysis of biological and clinical data. With a reference dataset of tumor and normal expression profiles, a classification algorithm can be utilized to "interpret" the expression profile of a new tissue sample, tagging it as tumor or normal. This is a typical example of a binary classification problem, where we distinguish positive instances from negative ones. Formally speaking, a training dataset consists of n instances $\{(\boldsymbol{x}_i, y_i)|i = 1, \ldots, n\}$, where $y_i \in \{1, 2\}$ and $\boldsymbol{x}_i \in \mathbb{R}^p$, that is, y_i is a nominal variable representing the label and $\boldsymbol{x}_i$ is known as the p-dimensional feature vector of instance i. A classification algorithm "learns" from the training dataset and predicts the label of a new instance based on its feature vector.

While there are classification algorithms designed to handle multi-class classification problems, a binary classification algorithm can be readily extended to addressing multi-class problems. Two techniques are widely used in a k-class classification problem. One transforms the problem into k binary classification problems by treating one class as class 1 and all the other classes as class 2. Alternatively, one converts the problem into $k(k-1)/2$ binary classification problems by considering all the pairs of k classes. In both cases, majority vote is used to determine the class label of a new instance. Therefore, binary classification algorithms can be viewed as building blocks of more sophisticated classification algorithms. For this reason, we focus on binary classification algorithms in this work.

Linear discriminant analysis (LDA)[1] is a simple, well-studied and widely used statistical classification algorithm. LDA assumes that, conditioned on the class label of an instance, the feature vector $\boldsymbol{x}$ follows a p-dimensional multivariate normal distribution. Moreover, all the classes have a common dispersion matrix, which makes the decision function linear in $\boldsymbol{x}$. In the binary case, LDA is equivalent to computing the likelihood ratio of posterior probabilities. Consequently, binary LDA is closely related to likelihood ratio (LR) tests.[10]

In the binary classification setting, predicting the class label of a new instance can be viewed as a statistical test of hypothesis[10] with the null hypothesis being "the new instance belongs to class 1" and the alternative hypothesis being "the new instance belongs to class 2". There are naturally two types of errors. One, known as the type I error, arises when we wrongly assign a class-1 instance to class 2. The other, the type II error, occurs when we wrongly assign a class-2 instance to class 1. In a LR test, a desired type I error rate α can be specified, while the type II error rate β depends on α. In other words, once we set the type I error rate, the type II error rate is fixed but can be undesirably large.

To allow setting α and β freely, Wald's sequential probability ratio test (SPRT)[3] was proposed to meet the need under assumptions that feature components of an instance can be observed sequentially and $\alpha + \beta < 1$. Shortly after the appearance of SPRT, Fu[2] applied the procedure to classification problems and proposed variants of SPRT assuming the feature components are independent. SPRT offers (at least) a couple of attractive theoretical advantages over binary LDA. First, freedom of specifying desired error rates permits us to control the classification accuracy of SPRT. Second, SPRT may not need all the p feature components to assign a new instance, requiring less computation.

While LDA is well-known, it is surprising that articles and books citing[2] barely mention the sequential classification procedures proposed by Fu[2] since the assumptions can be easily satisfied. In this work, we empirically compare binary LDA to SPRT and its variants using clinical/biological datasets. Leave-one-out cross-validation is used to assess the performance of the compared classification algorithms. We seek to reveal the effect of the theoretical advantages of SPRT on real datasets.

Methodology

LDA

Linear discriminant analysis (LDA)[1] is a well-studied classification algorithm. It assigns an instance described by $\boldsymbol{x} = (x_1 x_2 \ldots x_p)^T$, a p-dimensional feature vector, to $\arg\max_g \Pr(G = g \,|\boldsymbol{X} = \boldsymbol{x})$, the class with the highest posterior probability. By Bayes' theorem, we know that

$$\Pr(G = g|\boldsymbol{X} = \boldsymbol{x}) \propto \Pr(\boldsymbol{X} = \boldsymbol{x}|G = g)\Pr(G = g)$$

and if we further assume that $\Pr(G = g)$'s are the same for all g,

$$\Pr(G = g|\boldsymbol{X} = \boldsymbol{x}) \propto \Pr(\boldsymbol{X} = \boldsymbol{x}|G = g).$$

Therefore, class label assignment amounts to finding $\arg\max_g \Pr(\boldsymbol{X}=\boldsymbol{x}|G=g)$.

LDA assumes that, conditioned on the class label, the feature vector of an instance is distributed as a multivariate normal distribution. That is,

$$\Pr(G=g|\boldsymbol{X}=\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \times \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_g\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_g\right)\right),$$

where $\boldsymbol{x}$ is the component vector of an instance belonging to class g with mean vector $\boldsymbol{\mu}_g$ and dispersion matrix $\boldsymbol{\Sigma}$ common to all classes. The parameters $\boldsymbol{\mu}_g$'s and $\boldsymbol{\Sigma}$ can be estimated with a training dataset. In the binary case, finding $\arg\Pr(\boldsymbol{X}=\boldsymbol{x}|G=g)$ is equivalent to computing the likelihood ratio

$$\Lambda = \frac{\Pr(\boldsymbol{X}=\boldsymbol{x}\,|\,G=1)}{\Pr(\boldsymbol{X}=\boldsymbol{x}\,|\,G=2)}.$$

The instance $\boldsymbol{x}$ is assigned to class 1 if $\Lambda > 1$ or class 2 if $\Lambda < 1$. Binary LDA is therefore closely related to the probability ratio test.

SPRT

Fu[2] assumes that the components of $\boldsymbol{x}$ are independent and can be observed sequentially. Consequently,

$$\Lambda = \frac{\Pr(\boldsymbol{X}=\boldsymbol{x}\,|\,G=1)}{\Pr(\boldsymbol{X}=\boldsymbol{x}\,|\,G=2)} = \prod_{i=1}^{p} \frac{\phi\left(\frac{x_i-\mu_{1i}}{\sigma_i}\right)}{\phi\left(\frac{x_i-\mu_{2i}}{\sigma_i}\right)},$$

where $\phi(\cdot)$ is the pdf of the standard normal distribution, μ_{1i}'s and μ_{2i}'s are components of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, respectively, and σ_i's are the standard deviations of the x_i's. Equivalently, we have

$$\log\Lambda = \sum_{i=1}^{p} \log\left(\frac{\phi\left(\frac{x_i-\mu_{1i}}{\sigma_i}\right)}{\phi\left(\frac{x_i-\mu_{2i}}{\sigma_i}\right)}\right) = \sum_{i=1}^{p} Z_i,$$

where $$Z_i = \log\left(\frac{\phi\left(\frac{x_i-\mu_{1i}}{\sigma_i}\right)}{\phi\left(\frac{x_i-\mu_{2i}}{\sigma_i}\right)}\right).$$

Wald's sequential probability ratio test (SPRT)[3] is then readily applicable to binary classification problems. Setting the desired classification error rates, we obtain the decision boundaries b and a ($>b$). Given a new instance $\boldsymbol{x}$, we sample (without replacement) its components one at a time and compute $\Sigma_{i=1}^{j} Z_i$ until $\Sigma_{i=1}^{j} Z_i$ falls out of range (b, a), where j is the number of components sampled. Upon termination, we assign $\boldsymbol{x}$ to class 1 if $\Sigma_{i=1}^{j} Z_i > a$ or to class 2 if $\Sigma_{i=1}^{j} Z_i < b$. Unlike many other classification algorithms, the number of components examined before making a decision is not a constant but depends on the given new instance.

The decision boundaries a and b are computed from inputs α and β, which are the desired error rates for class 1 and class 2, respectively, such that $0 < \alpha, \beta < 1$. The decision boundaries are then given by

$$a = \log\left(\frac{1-\beta}{\alpha}\right) \quad \text{and}$$
$$b = \log\left(\frac{\beta}{1-\alpha}\right),$$

where we take the logarithms of the fractions because a and b are the boundaries for $\log\Lambda$ as opposed to simply Λ.

In practice, we have limited number of components for each instance, that is, we can never sample more than p components. Mukhopadhyay[3] addressed this issue by truncation, ie, setting a maximum number of components desired to be examined. Even if a constant less than p is not specified, truncation must be utilized if the algorithm has not made a decision after examining the last component. That is,

$$b < \sum_{i=1}^{p} Z_i < a.$$

In this case, the decision boundary is truncated to $1/2(a + b)$, such that the given instance $\boldsymbol{x}$ is assigned to

$$\text{class 1,} \quad \text{if} \quad \sum_{i=1}^{j} Z_i > \frac{1}{2}(a+b),$$
$$\text{class 2,} \quad \text{if} \quad \sum_{i=1}^{j} Z_i < \frac{1}{2}(a+b).$$

The same truncation can be applied at any specified constant k such that $1 < k \leq p$ and $b < \sum_{i=1}^{k} Z_i < a$. However, parameters a, b and k are somewhat interdependent. It was acknowledged in[3] that, if k is specified, the class error rates, which determine a and b, may not be closely met when truncation happens. Since we are not particularly concerned with minimizing the running time of our algorithms, we simply truncate at the total number of components p each time.

Modified SPRT (MSPRT)

Using Fu's[2] modified SPRT method, the computation of $\log \Lambda$ is the same as that of SPRT. However, the decision boundaries are not constant as we sample components of instance $\boldsymbol{x}$. The decision boundaries at the jth iteration are b_j and a_j $(>b_j)$, given by

$$a_j = a\left(1-\frac{j}{k}\right)^{r_1},$$
$$b_j = b\left(1-\frac{j}{k}\right)^{r_2},$$

where $0 < r_1, r_2 \leq 1$, $a > 0$, $b > 0$, and k is the truncation parameter. In this work, we set $r_1 = r_2 = 1$ for all the MSPRT experiments. As $j \to k$, $a_j, b_j \to 0$, thus ensuring that a decision will be made by the kth component. MSPRT is generally distinct from SPRT in that its decision boundaries gradually decrease to 0 at iteration k, while the decision boundaries of SPRT remain constant, with an abrupt decision made at iteration k.

Component sampling

As introduced in Section 2.1, the main assumption of LDA is that, conditioned on its class label, the p components of an instance are jointly normal with its class mean vector and a common dispersion matrix common to all the classes. We make the same assumption in our implementations of SPRT and MSPRT. To obtain independent components for an instance, we apply a linear transformation to the original p components in $\boldsymbol{X}$, resulting in p new independent components in $\boldsymbol{Y}$. Specifically, $\boldsymbol{Y} = \mathbf{P}^T\boldsymbol{X}$ such that

$$\text{Cov}(\boldsymbol{Y},\boldsymbol{Y}) = \mathbf{P}^T\boldsymbol{\Sigma}\mathbf{P} = \mathbf{D},$$

where $\text{Cov}(\boldsymbol{Y},\boldsymbol{Y})$ denotes the dispersion matrix of $\boldsymbol{Y}$, $\mathbf{P}$ is a $p \times p$ orthogonal matrix, $\boldsymbol{\Sigma}$ is the dispersion matrix of $\mathbf{X}$ and $\mathbf{D}$ is a $p \times p$ diagonal matrix containing the variances of components in $\boldsymbol{Y}$.[10] Independence among the new components follows immediately from the normality assumption. Once we estimate the dispersion matrix $\boldsymbol{\Sigma}$ using a training dataset, matrices $\mathbf{P}$ and $\mathbf{D}$ can be computed through eigen decomposition. In this work, we do not keep all the components in $\boldsymbol{Y}$. Components with small variances are discarded, ie, Y_i is kept if $D_{ii} > 10^{-3}\sum_{j=1}^{p} D_{jj}$.

The linear transformation does not affect LDA in any way. It is mainly for the ease of implementing SPRT and MSPRT. All the (new) components are considered when predicting the label of a new instance with LDA, while SPRT and MSPRT may not use all of the components. This raises the following question: In what order should we sample the components when using SPRT and MSPRT? We investigate two ways of sampling components: randomly and in the order of decreasing variance. We denote the former by SPRT/MSPRT-random and the latter by SPRT/MSPRT-ordered. Since sampling components in random order does not yield deterministic results, for a new instance, 100 runs of SPRT/MSPRT-random are performed and the majority prediction is assigned to the instance.

Results and Discussion

We conducted leave-one-out (LOO) cross-validation (CV) experiments on three binary biological/clinical datasets. The first is a Parkinson's disease dataset,[4,5] including 195 instances with 22 components. The second is a colon cancer microarray dataset,[6] preprocessed by,[7] including 63 instances with 2,000 components. We ranked the genes of the colon cancer dataset by a simple index (BSS/WSS) as described in,[8] narrowing the dataset down to only 500 components. The third one is a breast cancer dataset,[9] including 683 instances with 10 components. The results of our algorithms on the three clinical dataset are shown in Figures 1, 2 and 3, respectively.

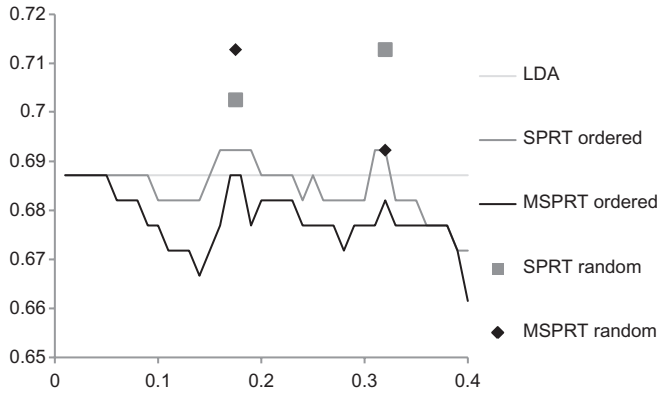

Figure 1. Accuracy rates for Parkinson's disease dataset. SPRT-ordered and MSPRT-ordered were run with $\alpha = \beta$ from 0.01 to 0.40 with a step size of 0.01. The accuracy rates for SPRT-random and MSPRT-random were calculated by the majority predictions over 100 runs each at $\alpha = \beta =$ **0.175** and $\alpha = \beta =$ **0.32**. The highest accuracy rate for the Parkinson's disease dataset was 0.7128, generated by MSPRT-random at $\alpha = \beta =$ **0.175** and by SPRT-random at $\alpha = \beta =$ **0.32**.

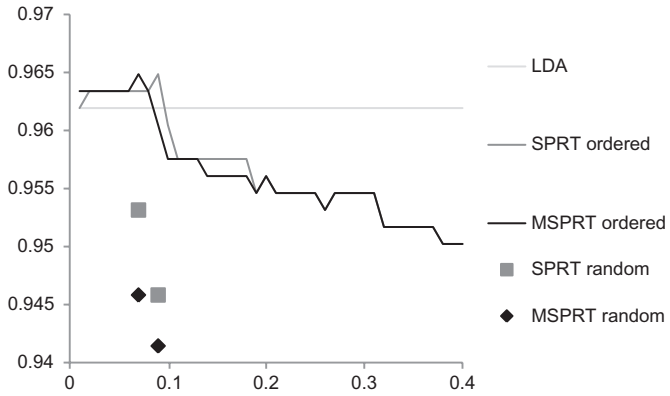

Figure 3. Accuracy rates for breast cancer dataset. SPRT-ordered and MSPRT-ordered were run with $\alpha = \beta$ from 0.01 to 0.40 with a step size of 0.01. The accuracy rates for SPRT-random and MSPRT-random were calculated by the majority predictions over 100 runs each at $\alpha = \beta =$ **0.07** and $\alpha = \beta =$ **0.09**. The highest accuracy rate for the breast cancer dataset was 0.9648, generated by MSPRT-ordered at $\alpha = \beta =$ **0.07** and by SPRT-ordered at $\alpha = \beta =$ **0.09**.

For the Parkinson's disease dataset, both SPRT-random and MSPRT-random outperform LDA at the chosen α and β values. SPRT-ordered outperforms LDA at some α and β values, while MSPRT-ordered is comparable to LDA at some α and β values. The highest accuracy rate is achieved by SPRT-random and MSPRT-random at $\alpha = \beta = 0.32$ and $\alpha = \beta = 0.175$, respectively. For the colon cancer dataset, the MSPRT-ordered reaches the maximum accuracy rate among all the methods at $\alpha = \beta = 0.22$. Most accuracy rates by SPRT-ordered are right under those by the MSPRT-ordered. Also note that the MSPRT-random and SPRT-random reaches the LDA accuracy rate, but the MSPRT-random falls below the accuracy rate achieved by LDA. Similarly, SPRT-ordered and MSPRT-ordered outperform LDA at some α and β values. Finally, for the breast cancer dataset, both SPRT-ordered and MSPRT-ordered reach peaks of accuracy that are above the LDA accuracy rate. Note the two peak values are the same but occur at different α and β values. SPRT-random and MSPRT-random both fall short of the LDA accuracy rate.

As described in Section 2.2 and 2.3, SPRT and MSPRT may not use all the components before arriving at a decision. Hence, we investigated the relationship between prediction accuracy and the average number of components examined by SPRT-ordered and MSPRT-ordered using the colon cancer

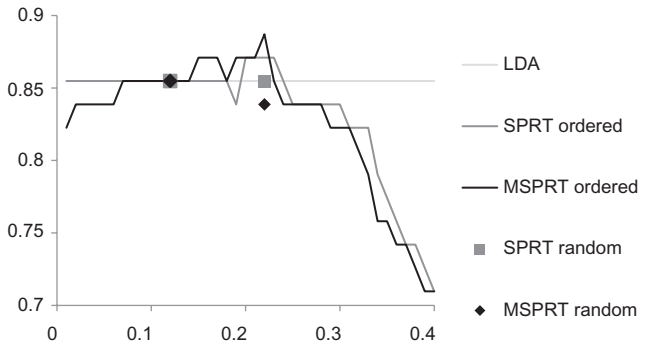

Figure 2. Accuracy rates for the colon cancer dataset. SPRT-ordered and MSPRT-ordered were run with $\alpha = \beta$ from 0.01 to 0.40 with a step size of 0.01. The accuracy rates for SPRT-random and MSPRT-random were calculated by the majority predictions over 100 runs each at $\alpha = \beta =$ **0.12** and $\alpha = \beta =$ **0.22**. The highest accuracy rate for the colon cancer dataset was 0.8871, generated by MSPRT-ordered at $\alpha = \beta =$ **0.22**.

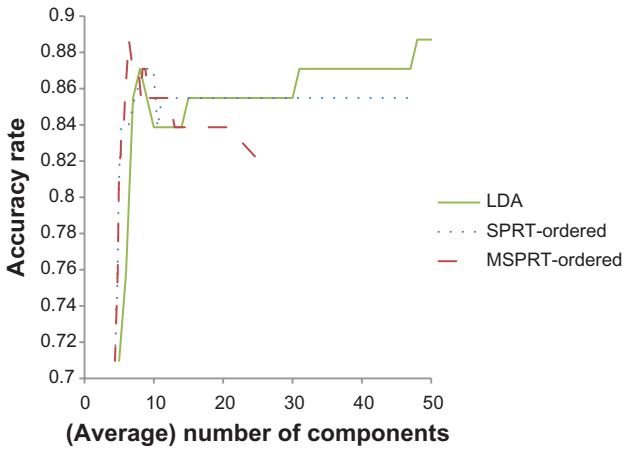

Figure 4. Scatter plots of accuracy rate against number of components. SPRT-ordered requires about 9 components on average to attain an accuracy rate of 0.8710. MSPRT-ordered uses about 6.4 components on average to achieve the maximal accuracy rate of 0.8871, while LDA needs at least 48 components to obtain the same accuracy rate.

dataset since it has the most number of components. To compare to LDA, we ordered the components by their variances and used only K components with the most variances to conduct LOO CV experiments, where K ranges from 5 to 50. Unlike SPRT-random and MSPRT-random, LDA always uses all the K components to infer the class label of a new instance. Figure 4 shows scatter plots of accuracy rate versus number of components for the three methods. It is evident that MSPRT-ordered requires significantly less components than LDA to attain the maximal accuracy rate of 0.8871. SPRT-ordered also requires less than 10 components on average to achieve its performance peak. It may appear that LDA reaches its performance peak at $K = 48$ and remains at the peak as K increases. This is not true since K can go up to 500, the total number of components, for this data set. We know that, at $K = 500$, this classifier is simply the LDA without component selection and the accuracy rate is 0.8548 (see Fig. 2), which is less than the maximal accuracy rate.

We note that the desired error rates, α and β, in SPRT/MSPRT are considered model parameters, which can be tuned by CV experiments on a training dataset. From the LOO CV results presented above, we can see that SPRT-ordered and MSPRT-ordered are superior to LDA in terms of prediction accuracy. Moreover, on average SPRT-ordered and MSPRT-ordered require less components than LDA to achieve the same accuracy. In some sense, SPRT-ordered and MSPRT-ordered perform implicit feature selection when labeling a new instance. Consequently, we do not need to find the optimal number of components as was done for LDA.

Inspired by the random forest algorithm,[11] SPRT-random and MSPRT-random can be viewed as ensemble classification algorithms, where a run of SPRT-random or MSPRT-random is analogous to a decision tree. What is different is that we did not perform bootstrapping on the training dataset for each run as in random forest. It is difficult, however, to compare SPRT-random and MSPRT-random to the other methods investigated in this work since the accuracy rates are available only at a few α and β values. More experiments need to be done at a range of α and β values to better understand the behavior of SPRT-random and MSPRT-random. We will also investigate the effect of introducing bootstrapping into our SPRT-random and MSPRT-random algorithms.

Finally, although Fu[2] assumed independence among components, we note that this assumption is not necessary. This follows from the fact that the proof of Theorem 3.2.1 in[10] does not assume independence among components. As long as the joint distribution of the components is known, the SPRT and MSPRT algorithms will work correctly. Because of the normality assumption, we have the joint distribution for each class immediately after estimation of the mean vectors and dispersion matrix. Consequently, obtaining independent components is not necessary for SPRT and MSPRT. We can directly sample the original (possibly dependent) components, resulting in new variants of SPRT and MSPRT. These novel variants of SPRT and MSPRT will be investigated in our future work.

Acknowledgements

This research was supported in part by the National Science Foundation, USA, under the grant CCF-0755373; and the National Institutes of Health, USA, under the grant R13LM008619.

Disclosure

This manuscript has been read and approved by all authors. This paper is unique and is not under consideration by any other publication and has not been published elsewhere. The authors and peer reviewers of this paper report no conflicts of interest. The authors confirm that they have permission to reproduce any copyrighted material.

References

1. Hastie T, Tibshirani R, Friedman JH. The elements of statistical learning, 2nd ed. Springer; 2009.
2. Fu KS. Sequential methods in pattern recognition and machine learning. Academic Press; 1968.
3. Mukhopadhyay N, de Silva BM. Sequential methods and their applications. CRC Press; 2009.
4. Little MA, McSharry PE, Hunter EJ, Spielman J, Ramig LO. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. Nature Precedings, 2008, hdl:10101/npre.2008.2298.1.
5. Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Bio Medical Engineering on Line*. June 26 2007; 6:23.
6. Alon U, Barkai N, Notterman DA, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Cell Biology*. 1999;96:6745–50.
7. Shevade SK, Keerthi SS. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*. 2003;19(17):2246–53.

8. Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*. 2002;97(457):77–87.
9. Chang CC, Lin CJ. LIBSVM: a library for support vector machines; 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
10. Mukhopadhyay N. Probability and statistical inference. Marcel Dekker; 2000.
11. Breiman L. Random forests. *Machine Learning*. 2001;45(1):5–32.