BLOCK DIAGONAL LINEAR DISCRIMINANT ANALYSIS WITH SEQUENTIAL EMBEDDED FEATURE SELECTION

Roger Piqué-Regí and Antonio Ortega

Signal and Image Processing Institute, Department of Electrical Engineering
Viterbi School of Engineering, University of Southern California
Email: {piquereg, ortega}@sipi.usc.edu

ABSTRACT

Model selection and feature selection are usually considered two separate tasks. For example, in a Linear Discriminant Analysis (LDA) setting, a modeling assumption is typically made first (e.g., a full or a diagonal covariance matrix can be chosen) and then with this model the feature subset providing the best prediction performance is selected. If limited training data is available, then the number of parameters of a model that can be reliably estimated will also be limited. In the context of LDA, model selection basically entails simplifying the covariance matrix by setting to zero some of this components. This leads to different *block diagonal* matrix structures (e.g., full / diagonal) which involve different sets of features and require different parameters to be estimated. In this paper we argue that LDA feature and parameter selection should be done jointly; and we propose a greedy algorithm for joint selection of features and of a *block diagonal* structure for the covariance matrix. To the best of our knowledge this is the first time such a joint design has been proposed in the context of LDA. The choice of a block diagonal structure is motivated by microarray classification problems, where we have a very large amount of features, i.e., genes, that are expected to be corregulated in small groups. Results obtained with artificial datasets show that the algorithm can flexibly choose an adequate covariance matrix structure according to the size of the training set and the generating distribution. Our results consistently outperform those achieved with other LDA based techniques.

1. INTRODUCTION

In statistical pattern recognition problems, Bayes decision techniques provide optimal classification performance as long as the distribution of the samples is known[1]. In many practical cases, these distributions are not known and they must be learned from training data. We focus on the case where the training data is in fact very limited, as compared to the number of features. In particular our work is motivated by classification in the context of genomic applications.

Consider the process of selecting the right model to represent the training data. Two steps are involved: a model has to be selected and then parameters of the model have to be estimated. Because only limited training data is available, we argue that these two steps have to be performed jointly in order to achieve better performance.

To illustrate why, consider two extreme cases. First, if we select a relatively simple model structure (and correspondingly few parameters) the overall model estimation variance will be low, but the risk is that the model oversimplifies the characteristics of the training data, thus leading to large *model bias*. Conversely, if we choose a relatively complex model, a better match to the underlying training data characteristics may be achieved, but the small training set leads to increased *model variance*. Thus the optimal performance will be given for a certain bias/variance trade-off, which has to be found in the process of model selection (see Chapter 7 in [2]).

In this paper we focus on Linear Discriminant Analysis (LDA) techniques. These assume that samples have a multivariate normal distribution, where each class has its own vector mean but all classes have a common covariance matrix. Thus, in the most general case we need to estimate class vector means and the covariance matrix. Clearly, an LDA approach is only feasible when the number of training samples, n, is much larger than the number of features, p, otherwise the covariance matrix will be ill-conditioned (Chpt. 3 [1]).

When n and p are comparable, different authors [3] have proposed a regularized solution for the problem by assuming some structure in the covariance matrix (e.g., a diagonal covariance matrix). This has the advantage of reducing the number of parameters that need to be estimated (only diagonal terms). However, when n is much smaller than p regularization alone is not enough to achieve reliable classification and it is necessary to further simplify the model by discarding features, i.e., by selecting a reduced feature set. Feature selection is in fact almost always needed in the context of microarray genomic classification, where p is in the order of tens of thousands of genes while n corresponds to a few hundred tissue samples. Taking cancer as an example, it is typically expected that only a few genes will be associated with the disease. Thus, feature (i.e., gene) selection serves the dual purpose of i) reducing the effect of a small training set on classification performance, and ii) identifying concrete genes that are more likely to be associated with the disease.

There are three major approaches to classifier design and feature selection [4]; namely, (i) **filter**, (ii) **wrapper**, and (iii) **embedded**. In **filter** approaches, features are first ranked using a statistical score, such as a t-test. Then the classifier is built by selecting the highest ranking features. This is the most popular method in microarray classification problems, due primarily to its simplicity. Note, however, that it completely ignores interactions among genes.

In **wrapper** approaches [5] a classifier is constructed with different candidate feature subsets, the performance is measured (using, for example, cross validation), and finally the feature subset that achieves the maximum performance is chosen. This is a combinatorial optimization problem and a full search would be very complex, requiring 2^p different evaluations , and prone to overfitting. For this reason, only greedy search strategies using different heuristics are feasible. In the context of microarrays and LDA, wrapper approaches have been proposed using full [6] or diagonal [7, 8] co-

Roger Piqué-Regí's work has been supported by a "La Caixa" fellowship. The author's thank Dr. Shahab Asgharzadeh, from Children's Hospital, Los Angeles, for his collaboration in the development of new techniques for microarray data analysis.

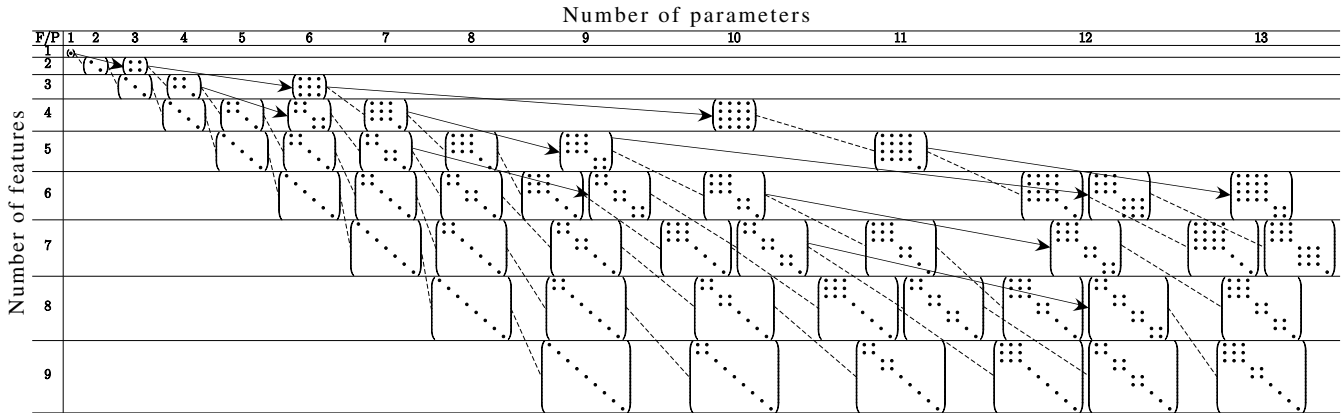

Table 1. Sequential generation of candidate covariance matrix models for LDA. Starting with an empty list, we add one feature at a time (namely, the one that maximizes a statistical score) using two possible operations: (i) *Block expansion* (solid lines), where a new feature is added to an existing block grouping already chosen features in the correlation structure. (ii) *Independent feature addition* (dashed lines), where a feature is added ignoring correlations (i.e., independent of existing blocks of variables in the correlation structure). The best among all these models is selected using crossvalidation.

variance matrices and different search strategies.

Finally, **embedded** approaches [9] consider jointly the classifier design and the feature subset selection. This is in contrast to the wrapper approaches that consider the classifier as a black box that induces a prediction rule once the feature subset is chosen. Guyon et al. proposed an embedded approach[10] for Support Vector Machines. To the best of our knowledge no embedded design techniques have been proposed in the context of LDA.

In this paper we present a novel LDA *embedded* approach for joint feature and model selection. In the LDA context model selection is essentially a choice of a structure for the covariance matrix. Thus a simple method would perform feature selection for both diagonal and a full covariance matrix structures and pick the best of them. In a diagonal model, the number of parameters to estimate, l, equals p, while in a full matrix $l = \frac{1}{2}p\,(p+1)$. We propose to further increase the number of available models by including a whole range of block diagonal matrix structures, as shown in Table 1.

Applying the bias/variance tradeoff principle in this setting implies that the more parameters we estimate the less bias we will have, but at the cost of increasing the variance. For this reason, the LDA performance is limited primarily by the number of parameters to estimate (rather than by the number of features). We use this insight to develop novel efficient techniques to embed feature and model selection, which are based on searching for the best feature set and covariance model *for a given number of parameters*.

Thus, for a given number of parameters, more features can be used with a diagonal covariance model than with a full covariance matrix, but correlation among features will be completely ignored. For uncorrelated features this model will perform best, but there might be correlations present that could be exploited to get better performance with fewer features. Exploring all possible feature subsets and possible block diagonal structures is not feasible. Thus, we propose a sequential greedy algorithm, *SeqBDLDA*, for finding at the same time a feature subset and a block diagonal structure.

This paper is organized as follows, in Section 2 we present our proposed greedy algorithm, SeqDBLDA, in Section 3 the algorithm is evaluated and compared to two related methods, and finally in Section 4 we draw our conclusions.

2. GREEDY FEATURE AND MODEL SELECTION FOR BLOCK DIAGONAL LDA

Linear Discriminant Analysis (LDA) [1, 2] for two classes is defined by a linear function $g(\mathbf{x})$ called discriminant that partitions the feature space into two regions:

$$g\left(\mathbf{x}\right) = \mathbf{w}^t\mathbf{x} - b \begin{cases} \geq 0 \Rightarrow \text{ClassA} \\ < 0 \Rightarrow \text{ClassB} \end{cases} \tag{1}$$

where $\mathbf{x}$ is the feature vector of the sample to classify, $\mathbf{w}$ is a vector of weights orthogonal to the hyperplane that jointly with the scalar b define the decision boundary $g(\mathbf{x}) = 0$.

If the class conditional distribution is multivariate normal — $f_A\left(\mathbf{x}\right) \sim N\left(\mathbf{m}_A, \mathbf{K}\right)$, $f_B\left(\mathbf{x}\right) \sim N\left(\mathbf{m}_B, \mathbf{K}\right)$ —, then the optimal parameters in the decision rule (1) are:

$$\mathbf{w} = \mathbf{K}^{-1}\mathbf{d} \qquad \mathbf{d} = \mathbf{m}_A - \mathbf{m}_B \tag{2}$$

$$b = \ln\left(\frac{\pi_A}{\pi_B}\right) - \mathbf{w}^t\,\frac{1}{2}\left(\mathbf{m}_A + \mathbf{m}_B\right) \tag{3}$$

We consider cases where the mean vectors $\mathbf{m}_A$, $\mathbf{m}_B$; the covariance matrix $\mathbf{K}$; and the prior class probabilities π_A, π_B have to be estimated from training data. If we use the maximum likelihood (ML) estimators then $\hat{\mathbf{w}}$ coincides with the Fisher canonical variate computed as the direction which maximizes covariance between/within ratio [2]:

$$J_{\hat{\mathbf{K}}}\left(\mathbf{w}\right) = \frac{\left(\hat{\mathbf{d}}^t\mathbf{w}\right)^2}{\mathbf{w}^t\hat{\mathbf{K}}\mathbf{w}} \tag{4}$$

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} J_{\hat{\mathbf{K}}}\left(\mathbf{w}\right) = \hat{\mathbf{K}}^{-1}\hat{\mathbf{d}} \tag{5}$$

Our proposed greedy algorithm for feature and model selection (see Algorithm 1), adds features to the model sequentially, one at a time. The process starts by selecting the best feature measured with the J score of (4). Then, at each stage, we have two options: i) adding another feature to be considered independent of all previously selected features, thus leading to a new block in the block-diagonal structure, and ii) growing the current block in the matrix structure by adding one more feature to it. These two options are marked with

dashed and solid line, respectively, in Table 1 and can be used alternatively to produce feature subsets with different block diagonal covariance structures. In both operations the current set of features, $\mathcal{A}$, is "inherited" from the parent node; in order to determine which is the best new feature for a given structure we use the scoring procedure discussed in Section 2.1. After obtaining one feature subset $\mathcal{A}_m$ for each of the models in Table 1, we are interested in finding which is the more reliable model if the number of parameters is limited. To do so we use leave-one-out cross-validation (see Section 2.2).

Algorithm 1 Greedy feature subset and model construction

1: Create first model with best feature: $i = \arg\max_{j \in S} \frac{d_j}{\sigma_j}$
2: **for all** Model m in Table 1 **do**
3: $\mathcal{A} \leftarrow$ Feature set of the parent node
4: $j^* \leftarrow$ ADDFEATURE($\mathcal{A}$,m) ▷ Find the best feature to add
5: $\mathcal{A}_m \leftarrow \mathcal{A} \cup \{j^*\}$
6: $\epsilon_m \leftarrow$ EVALUATEMODEL($\mathcal{A}_m$,m) ▷ Using crossvalidation
7: **end for**
8: $l \leftarrow$ Number of parameters
9: $m^* \leftarrow \arg \max_{m:|m|=l} \epsilon_m$ ▷ Find the best model with l parameters

2.1. Feature addition scoring procedure

Assume that we have already chosen a subset of features $\mathcal{A}$, with sample covariance matrix $\hat{\mathbf{K}}_{\mathcal{A}}$ and difference of sample means $\hat{\mathbf{d}}_{\mathcal{A}}$. Then, from (2) the LDA classifier with a model m is constructed using the following weights:

$$\mathbf{w}_{\mathcal{A}} = \hat{\mathbf{K}}_{\mathcal{A},m}^{-1} \hat{\mathbf{d}}_{\mathcal{A}}, \tag{6}$$

where $\hat{\mathbf{K}}_{\mathcal{A},m}$ is obtained from $\hat{\mathbf{K}}_{\mathcal{A}}$ by zeroing out those terms that are zero in model m (see examples in Table 1). Then, using (5), the best new feature to add to the model $j \in \mathcal{A}^C$ (where $\mathcal{A}^C$ is the complement of $\mathcal{A}$ in the original feature set) will be:

$$j* = \arg\max_{j \in \mathcal{A}^C} \frac{\left(\hat{\mathbf{d}}_{\mathcal{A}_j}^t \mathbf{w}_{\mathcal{A}_j}\right)^2}{\mathbf{w}_{\mathcal{A}_j}^t \hat{\mathbf{K}}_{\mathcal{A}_j} \mathbf{w}_{\mathcal{A}_j}} \quad \mathcal{A}_j = \mathcal{A} \cup \{j\} \tag{7}$$

In our greedy procedure, the new feature is always added in the lower right corner of the matrix, either as an independent block (i.e., ignoring correlations), or by increasing the size of the lower right block by one. In finding the best feature, significant computational savings can be achieved by exploiting the block structure of the matrix in (6), and the fact that only certain blocks in vectors and matrices in (7) change with j.

2.2. Model selection with cross-validation

Since we used the J score (4) to guide the search for the feature subset we cannot use it to decide which model to select. This is because it is a biased estimate of performance of the classifier that can be used to compare alternative models with same number of parameters and features, but does not provide a reliable way to compare models with different structures. Cross-validation [2] is an unbiased procedure to estimate the probability of error of a classifier. In leave-one-out crossvalidation, one sample is left out and we train with the remaining $n-1$ samples. Then the sample that has been left out is classified. The entire training procedure is repeated n times for each of the samples and the error rate ϵ_m is estimated as the total number of misclassified samples divided by n. In our case, if the number of parameters is limited to l, we will select the model in the column l of Table 1 with the lowest cross-validation error.

2.3. Relationship with other LDA methods and applications

Table 1 contains several models that have been proposed in the literature: models "grown" by following *only* solid lines, correspond to "full matrix" LDA with forward feature selection (SeqLDA, [6]). Alternatively models grown by following only dashed lines correspond to forward selection using the Diagonal LDA (SeqDLDA, [7, 8]) model. Thus both "full matrix" LDA and SeqDLDA are part of the space of solutions being searched. Note also that if some a priori knowledge was available about the structure of the covariance matrix this could be exploited to reduce the complexity of the search by removing some of the paths in Table 1 from consideration. For example, if it is believed that features will tend to be correlated in small groups, it is very easy to set limits on the maximum size of the blocks to be explored by our algorithm.

3. EXPERIMENTAL RESULTS

We have extensively analyzed our algorithm with artificial data for two basic reasons. First this allows us to control the covariance matrix and so evaluate the ability of the algorithm to select a model close to actual one. Second, evaluation is simplified, since for a given LDA-trained model we can exactly compute the probability of error without having to estimate it.

The training data is generated by drawing n samples with distributions $f_A(\mathbf{x}) \sim N(\mathbf{m}_A, \mathbf{K})$, $f_B(\mathbf{x}) \sim N(\mathbf{m}_B, \mathbf{K})$. The two basic generating parameters are $\mathbf{K}$, and $\mathbf{d} = \mathbf{m}_A - \mathbf{m}_B$. We have experimented with several covariance matrix structures and randomly permuted the features, so that in general two contiguous features are not necessarily correlated. In the experiments presented here $\mathbf{d}$ was fixed so that the SNR of the features is exponentially decreasing with parameter γ:

$$\left|\frac{d_j}{\sigma_j}\right| = e^{-\gamma j} \quad \left(\sigma_j^2\right)_j = \mathrm{diag}(\mathbf{K}) \tag{8}$$

The number of features that will be optimal for the classifier will usually be between $1/\gamma$ and $4/\gamma$ approximately, increasing with the sample size n and decreasing with p. When n and p are constant, if γ is small, a large number of features will be required for the classifier and a diagonal matrix model will be preferred over a full matrix one.

After training the weight vector $\mathbf{w}$, the probability of error is

$$P_{e|\mathbf{w}} = 1 - \Phi\left(\frac{1}{2}\sqrt{\frac{J_{\mathbf{K}}(\mathbf{w})}{1+1/n}}\right) \quad J_{\mathbf{K}}(\mathbf{w}) = \frac{(\mathbf{d}^t\mathbf{w})^2}{\mathbf{w}^t\mathbf{K}\mathbf{w}} \tag{9}$$

where $\Phi(x)$ is the standard normal cumulative distribution function and $1+1/n$ takes into account the cost of estimating the b parameter in (1). We repeat the training and evaluation T times and the average P_e is estimated as:

$$\hat{P}_e = \frac{1}{T}\sum_{t=1}^{T} P_{e|\hat{\mathbf{w}}_t} \tag{10}$$

These results are reported for our proposed algorithm (SeqBDLDA) along with the two related not embedded methods SeqDLDA and SeqLDA described in Section 2.3. Finally, 95% confidence intervals asses the statistical significance of our findings.

3.1. Toeplitz symmetric covariance matrix

A Toeplitz symmetric matrix arises from AR processes, in which contiguous features are locally correlated. This is exploited by several classifying algorithms [3], which will, however, fail if the features are permuted. Our proposed algorithm avoids this problem since it is invariant to feature permutation. This comes indirectly from our original design assumption that no prior knowledge exists about correlation between features.

In our experiments the more diagonally dominant the matrix is, the better the diagonal model will be. While if the training data is limited, the full-matrix approach quickly fails as we increase the number of parameters. Figure 1 illustrates this with the following covariance matrix:

$$\mathbf{K} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{8} & \frac{1}{10} & 0 & \cdots \\ -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & \frac{1}{10} & \ddots \\ \frac{1}{10} & -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & \ddots \\ 0 & \frac{1}{10} & -\frac{1}{8} & \frac{1}{4} & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \tag{11}$$

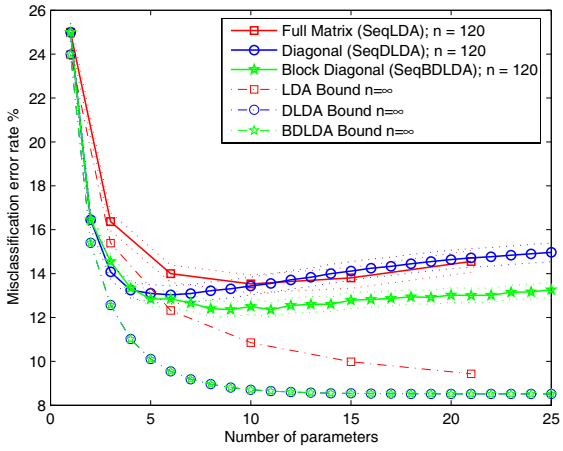

Fig. 1. Classification performance for $p = 200$, $n = 120$, $\mathbf{K}$ as in (11), $\gamma = 0.2$. Solid and dotted lines represent the mean $\hat{P}_e$ and its 95 % confidence interval for 100 trainings.

3.2. Block diagonal covariance matrices

We have tested our algorithm with block diagonal matrices. Figure 2 shows the results for the following covariance matrix structure:

$$\mathbf{K} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D} \end{pmatrix} \tag{12}$$

$$\mathbf{A} = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{3} & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ -\frac{1}{3} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} \frac{1}{2} & 0 & \cdots \\ 0 & \frac{1}{2} & \ddots \\ \vdots & \ddots & \frac{1}{2} \end{pmatrix}$$

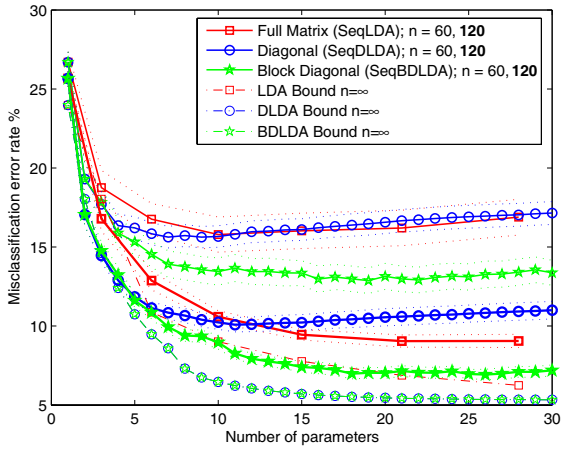

Fig. 2. Classification performance for $p = 200$, $n = 60$ (thin line),120 (thick line), $\mathbf{K}$ as in (12) , $\gamma = 0.1$. Solid and dotted lines represent the mean and its 95 % confidence interval for P_e of 100 trainings

Figure 2 shows that when training data is very limited, e.g., $n = 60$, a diagonal structure (SeqDLDA) outperforms a full matrix approach (SeqLDA), while as n increases the full matrix approach becomes better. Our technique approach is able to choose SeqLDA or SeqDLDA for number of parameters for which these perform well, and is also capable of choosing intermediate block-diagonal alternatives that outperform both of them in other cases.

4. CONCLUSIONS

This paper proposes a new method for performing Linear Discriminant Analysis in which the feature subset selection, and the covariance matrix structure, are jointly selected. The proposed approach is greedy but it is computationally feasible and can be seen to outperform existing LDA-based techniques. Furthermore, among the models explored by our approach are two standard techniques: SeqDLDA [7, 8] and SeqLDA [6]. When one of these techniques provides the best performance, our algorithm is capable of selecting the corresponding model among the solutions it searches. In general we are capable of outperforming these two standard techniques. Further work will explore different methods of guiding the search and deciding which feature subsets to explore, e.g., by using compound operations [5] in which more than one feature added at each iteration in order to speed up the search.

5. REFERENCES

[1] Richard Duda, Peter Hart, and David Stork, *Pattern Classification*, John Wiley and Sons, 2001, 0-471-05669-3.

[2] Trevor Hastie, Robert Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, Springer, July 2001.

[3] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, pp. 165–175, 1989.

[4] Isabelle Guyon and André Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[5] Ron Kohavi and George H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[6] M Xiong, W Li, J Zhao, L Jin, and E Boerwinkle, "Feature (gene) selection in gene expression-based tumor classification.," *Mol Genet Metab*, vol. 73, no. 3, pp. 239–47, 2001.

[7] Trond Bo and Inge Jonassen, "New feature subset selection procedures for classification of expression profiles.," *Genome Biol*, vol. 3, no. 4, pp. RESEARCH0017, 2002.

[8] R. Piqué-Regí, A. Ortega, and S. Asgharzadeh, "Sequential diagonal linear discriminant analysis (SeqDLDA) for microarray classification and gene identification," in *Computational Systems and Bioinformatics*, Aug. 2005.

[9] T.N. Lal, O. Chapelle, J. Weston, and A. Elisseeff, *Embedded methods*, Springer, 2005.

[10] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1-3, pp. 389–422, 2002.