Machine Learning, 46, 389–422, 2002

Gene Selection for Cancer Classification using Support Vector Machines

ISABELLE GUYON isabelle@barnhilltechnologies.com
JASON WESTON
STEPHEN BARNHILL
Barnhill Bioinformatics, Savannah, Georgia, USA

VLADIMIR VAPNIK vlad@research.att.com
AT&T Labs, Red Bank, New Jersey, USA

Editor: Nello Cristianini

Abstract. DNA micro-arrays now permit scientists to screen thousands of genes simultaneously and determine whether those genes are active, hyperactive or silent in normal or cancerous tissue. Because these new micro-array devices generate bewildering amounts of raw data, new analytical methods must be developed to sort out whether cancer tissues have distinctive signatures of gene expression over normal tissues or other types of cancer tissues.

In this paper, we address the problem of selection of a small subset of genes from broad patterns of gene expression data, recorded on DNA micro-arrays. Using available training examples from cancer and normal patients, we build a classifier suitable for genetic diagnosis, as well as drug discovery. Previous attempts to address this problem select genes with correlation techniques. We propose a new method of gene selection utilizing Support Vector Machine methods based on Recursive Feature Elimination (RFE). We demonstrate experimentally that the genes selected by our techniques yield better classification performance and are biologically relevant to cancer.

In contrast with the baseline method, our method eliminates gene redundancy automatically and yields better and more compact gene subsets. In patients with leukemia our method discovered 2 genes that yield zero leave-one-out error, while 64 genes are necessary for the baseline method to get the best result (one leave-one-out error). In the colon cancer database, using only 4 genes our method is 98% accurate, while the baseline method is only 86% accurate.

Keywords: diagnosis, diagnostic tests, drug discovery, RNA expression, genomics, gene selection, DNA micro-array, proteomics, cancer classification, feature selection, support vector machines, recursive feature elimination

1. Introduction

The advent of DNA micro-array technology has brought to data analysts broad patterns of gene expression simultaneously recorded in a single experiment (Fodor, 1997). In the past few months, several data sets have become publicly available on the Internet. These data sets present multiple challenges, including a large number of gene expression values per experiment (several thousands to tens of thousands), and a relatively small number of experiments (a few dozen).

The data can be analyzed from many different viewpoints. The literature already abounds in studies of gene clusters discovered by unsupervised learning techniques (see e.g. Eisen,

1998; Perou, 1999; Alon, 1999; Alizadeh, 2000). Clustering is often done along the other dimension of the data. For example, each experiment may correspond to one patient carrying or not carrying a specific disease (see e.g. Golub, 1999). In this case, clustering usually groups patients with similar clinical records. Recently, supervised learning has also been applied, to the classification of proteins (Brown, 2000) and to cancer classification (Golub, 1999).

This last paper on leukemia classification presents a feasibility study of diagnosis based solely on gene expression monitoring. In the present paper, we go further in this direction and demonstrate that, by applying state-of-the-art classification algorithms (Support Vector Machines (Boser, 1992; Vapnik, 1998), a small subset of highly discriminant genes can be extracted to build very reliable cancer classifiers. We make connections with related approaches that were developed independently, which either combine (Furey, 2000; Pavlidis, 2000) or integrate (Mukherjee, 1999; Chapelle, 2000; Weston, 2000) feature selection with SVMs.

The identification of discriminant genes is of fundamental and practical interest. Research in Biology and Medicine may benefit from the examination of the top ranking genes to confirm recent discoveries in cancer research or suggest new avenues to be explored. Medical diagnostic tests that measure the abundance of a given protein in serum may be derived from a small subset of discriminant genes.

This application also illustrates new aspects of the applicability of Support Vector Machines (SVMs) in knowledge discovery and data mining. SVMs were already known as a tool that discovers informative patterns (Guyon, 1996). The present application demonstrates that SVMs are also very effective for discovering informative features or attributes (such as critically important genes). In a comparison with several other gene selection methods on Colon cancer data (Alon, 1999) we demonstrate that SVMs have both quantitative and qualitative advantages. Our techniques outperform other methods in classification performance for small gene subsets while selecting genes that have plausible relevance to cancer diagnosis.

After formally stating the problem and reviewing prior work (Section 2), we present in Section 3 a new method of gene selection using SVMs. Before turning to the experimental section (Section 5), we describe the data sets under study and provide the basis of our experimental method (Section 4). Particular care is given to evaluate the statistical significance of the results for small sample sizes. In the discussion section (Section 6), we review computational complexity issues, contrast qualitatively our feature selection method with others, and propose possible extensions of the algorithm.

2. Problem description and prior work

2.1. Classification problems

In this paper we address classification problems where the input is a vector that we call a "pattern" of n components which we call "features". We call F the n-dimensional feature space. In the case of the problem at hand, the features are gene expression coefficients and patterns correspond to patients. We limit ourselves to two-class classification problems. We

identify the two classes with the symbols (+) and (−). A training set of a number of patterns $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_k, \ldots \mathbf{x}_\ell\}$ with known class labels $\{y_1, y_2, \ldots y_k, \ldots y_\ell\}$, $y_k \in \{-1, +1\}$, is given. The training patterns are used to build a decision function (or discriminant function) $D(\mathbf{x})$, that is a scalar function of an input pattern $\mathbf{x}$. New patterns are classified according to the sign of the decision function:

$$D(\mathbf{x}) > 0 \Rightarrow \mathbf{x} \in \text{class } (+)$$
$$D(\mathbf{x}) < 0 \Rightarrow \mathbf{x} \in \text{class } (-)$$
$$D(\mathbf{x}) = 0, \text{ decision boundary.}$$

Decision functions that are simple weighted sums of the training patterns plus a bias are called linear discriminant functions (see e.g. Duda, 1973). In our notations:

$$D(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + \mathbf{b}, \tag{1}$$

where $\mathbf{w}$ is the weight vector and b is a bias value.

A data set is said to be "linearly separable" if a linear discriminant function can separate it without error.

2.2. *Space dimensionality reduction and feature selection*

A known problem in classification specifically, and machine learning in general, is to find ways to reduce the dimensionality n of the feature space F to overcome the risk of "overfitting". Data overfitting arises when the number n of features is large (in our case thousands of genes) and the number ℓ of training patterns is comparatively small (in our case a few dozen patients). In such a situation, one can easily find a decision function that separates the training data (even a linear decision function) but will perform poorly on test data. Training techniques that use regularization (see e.g. Vapnik, 1998) avoid overfitting of the data to some extent without requiring space dimensionality reduction. Such is the case, for instance, of Support Vector Machines (SVMs) (Boser, 1992; Vapnik, 1998; Cristianini, 1999). Yet, as we shall see from experimental results (Section 5), even SVMs benefit from space dimensionality reduction.

Projecting on the first few principal directions of the data is a method commonly used to reduce feature space dimensionality (see, e.g. Duda, 73). With such a method, new features are obtained that are linear combinations of the original features. One disadvantage of projection methods is that none of the original input features can be discarded. In this paper we investigate pruning techniques that eliminate some of the original input features and retain a minimum subset of features that yield best classification performance. Pruning techniques lend themselves to the applications that we are interested in. To build diagnostic tests, it is of practical importance to be able to select a small subset of genes. The reasons include cost effectiveness and ease of verification of the relevance of selected genes.

The problem of feature selection is well known in machine learning. For a review of feature selection, see e.g. (Kohavi, 1997). Given a particular classification technique, it is conceivable to select the best subset of features satisfying a given "model selection" criterion

by exhaustive enumeration of all subsets of features. For a review of model selection, see e.g. (Kearns, 1997). Exhaustive enumeration is impractical for large numbers of features (in our case thousands of genes) because of the combinatorial explosion of the number of subsets. In the discussion section (Section 6), we shall go back to this method that can be used in combination with another method that first reduces the number of features to a manageable size.

Performing feature selection in large dimensional input spaces therefore involves greedy algorithms. Among various possible methods feature-ranking techniques are particularly attractive. A fixed number of top ranked features may be selected for further analysis or to design a classifier. Alternatively, a threshold can be set on the ranking criterion. Only the features whose criterion exceeds the threshold are retained. In the spirit of Structural Risk Minimization (see e.g. Vapnik, 1998; Guyon, 1992) it is possible to use the ranking to define nested subsets of features $F_1 \subset F_2 \subset \cdots \subset F$, and select an optimum subset of features with a model selection criterion by varying a single parameter: the number of features. In the following, we compare several feature-ranking algorithms.

2.3. Feature ranking with correlation coefficients

In the test problems under study, it is not possible to achieve an errorless separation with a single gene. Better results are obtained when increasing the number of genes. Classical gene selection methods select the genes that individually classify best the training data. These methods include correlation methods and expression ratio methods. They eliminate genes that are useless for discrimination (noise), but they do not yield compact gene sets because genes are redundant. Moreover, complementary genes that individually do not separate well the data are missed.

Evaluating how well an individual feature contributes to the separation (e.g. cancer vs. normal) can produce a simple feature (gene) ranking. Various correlation coefficients are used as ranking criteria. The coefficient used in Golub (1999) is defined as:

$$w_i = (\mu_i(+) - \mu_i(-))/(\sigma_i(+) + \sigma_i(-)) \tag{2}$$

where μ_i and σ_i are the mean and standard deviation of the gene expression values of gene i for all the patients of class $(+)$ or class $(-)$, $i = 1, \ldots n$. Large positive w_i values indicate strong correlation with class $(+)$ whereas large negative w_i values indicate strong correlation with class $(-)$. The original method of Golub (1999) is to select an equal number of genes with positive and with negative correlation coefficient. Others (Furey, 2000) have been using the absolute value of wi as ranking criterion. Recently, in Pavlidis (2000), the authors have been using a related coefficient $(\mu_i(+) - \mu_i(-))^2/(\sigma_i(+)^2 + \mu_i(-)^2)$, which is similar to Fisher's discriminant criterion (Duda, 1973).

What characterizes feature ranking with correlation methods is the implicit orthogonality assumptions that are made. Each coefficient w_i is computed with information about a single feature (gene) and does not take into account mutual information between features. In the next section, we explain in more details what such orthogonality assumptions mean.

2.4. *Ranking criterion and classification*

One possible use of feature ranking is the design of a class predictor (or classifier) based on a pre-selected subset of features. Each feature that is correlated (or anti-correlated) with the separation of interest is by itself such a class predictor, albeit an imperfect one. This suggests a simple method of classification based on weighted voting: the features vote proportionally to their correlation coefficient. Such is the method being used in Golub (1999). The weighted voting scheme yields a particular linear discriminant classifier:

$$D(\mathbf{x}) = \mathbf{w} \cdot (\mathbf{x} - \mu) \tag{3}$$

where $\mathbf{w}$ is defined in Eq. (2) and $\mu = (\mu(+) + \mu(-))/2$.

It is interesting to relate this classifier to Fisher's linear discriminant. Such a classifier is also of the form of Eq. (3), with

$$\mathbf{w} = S^{-1}(\mu(+) - \mu(-)),$$

where S is the (n, n) within class scatter matrix defined as

$$S = \sum_{x \in X(+)} (\mathbf{x} - \mu(+))(\mathbf{x} - \mu(+))^T + \sum_{x \in X(-)} (\mathbf{x} - \mu(-))(\mathbf{x} - \mu(-))^T$$

And where μ is the mean vector over all training patterns. We denote by $X(+)$ and $X(-)$ the training sets of class $(+)$ and $(-)$. This particular form of Fisher's linear discriminant implies that S is invertible. This is not the case if the number of features n is larger than the number of examples ℓ since then the rank of S is at most ℓ. The classifier of Golub (1999) and Fisher's classifier are particularly similar in this formulation if the scatter matrix is approximated by its diagonal elements. This approximation is exact when the vectors formed by the values of one feature across all training patterns are orthogonal, after subtracting the class mean. It retains some validity if the features are uncorrelated, that is if the expected value of the product of two different feature is zero, after removing the class mean. Approximating S by its diagonal elements is one way of regularizing it (making it invertible). But, in practice, features are usually correlated and therefore the diagonal approximation is not valid.

We have just established that the feature ranking coefficients can be used as classifier weights. Reciprocally, the weights multiplying the inputs of a given classifier can be used as feature ranking coefficients. The inputs that are weighted by the largest value influence most the classification decision. Therefore, if the classifier performs well, those inputs with the largest weights correspond to the most informative features. This scheme generalizes the previous one. In particular, there exist many algorithms to train linear discriminant functions that may provide a better feature ranking than correlation coefficients. These algorithms include Fisher's linear discriminant, just mentioned, and SVMs that are the subject of this paper. Both methods are known in statistics as "multivariate" classifiers, which means that they are optimized during training to handle multiple variables (or features) simultaneously. The method of Golub (1999), in contrast, is a combination of multiple "univariate" classifiers.

2.5. *Feature ranking by sensitivity analysis*

In this section, we show that ranking features with the magnitude of the weights of a linear discriminant classifier is a principled method. Several authors have suggested to use the change in objective function when one feature is removed as a ranking criterion (Kohavi, 1997). For classification problems, the ideal objective function is the expected value of the error, that is the error rate computed on an infinite number of examples. For the purpose of training, this ideal objective is replaced by a cost function J computed on training examples only. Such a cost function is usually a bound or an approximation of the ideal objective, chosen for convenience and efficiency reasons. Hence the idea to compute the change in cost function $DJ(i)$ caused by removing a given feature or, equivalently, by bringing its weight to zero. The OBD algorithm (LeCun, 1990) approximates $DJ(i)$ by expanding J in Taylor series to second order. At the optimum of J, the first order term can be neglected, yielding:

$$DJ(i) = (1/2)\frac{\partial^2 J}{\partial w_i^2}(Dw_i)^2 \tag{4}$$

The change in weight $Dw_i = w_i$ corresponds to removing feature i. The authors of the OBD algorithm advocate using $DJ(i)$ instead of the magnitude of the weights as a weight pruning criterion. For linear discriminant functions whose cost function J is a quadratic function of w_i these two criteria are equivalent. This is the case for example of the mean-squared-error classifier (Duda, 1973) with cost function $J = \sum_{x \in X} \|\mathbf{w} \cdot \mathbf{x} - y\|^2$ and linear SVMs (Boser, 1992; Vapnik, 1998; Cristianini, 1999), which minimize $J = (1/2)||w||^2$, under constrains. This justifies the use of $(w_i)^2$ as a feature ranking criterion.

2.6. *Recursive Feature Elimination*

A good feature ranking criterion is not necessarily a good feature subset ranking criterion. The criteria $DJ(i)$ or $(w_i)^2$ estimate the effect of removing one feature at a time on the objective function. They become very sub-optimal when it comes to removing several features at a time, which is necessary to obtain a small feature subset. This problem can be overcome by using the following iterative procedure that we call Recursive Feature Elimination:

1. Train the classifier (optimize the weights w_i with respect to J).
2. Compute the ranking criterion for all features ($DJ(i)$ or $(w_i)^2$).
3. Remove the feature with smallest ranking criterion.

This iterative procedure is an instance of backward feature elimination (Kohavi, 2000 and references therein). For computational reasons, it may be more efficient to remove several features at a time, at the expense of possible classification performance degradation. In such a case, the method produces a feature subset ranking, as opposed to a feature ranking. Feature subsets are nested $F_1 \subset F_2 \subset \cdots \subset F$.

If features are removed one at a time, there is also a corresponding feature ranking. However, the features that are top ranked (eliminated last) are not necessarily the ones that are individually most relevant. Only taken together the features of a subset F_{m} are optimal in some sense.

In should be noted that RFE has no effect on correlation methods since the ranking criterion is computed with information about a single feature.

3. Feature ranking with Support Vector Machines

3.1. Support Vector Machines (SVM)

To test the idea of using the weights of a classifier to produce a feature ranking, we used a state-of-the-art classification technique: Support Vector Machines (SVMs) (Boser, 1992; Vapnik, 1998). SVMs have recently been intensively studied and benchmarked against a variety of techniques (see for instance, Guyon, 1999). They are presently one of the best-known classification techniques with computational advantages over their contenders (Cristianini, 1999).

Although SVMs handle non-linear decision boundaries of arbitrary complexity, we limit ourselves, in this paper, to linear SVMs because of the nature of the data sets under investigation. Linear SVMs are particular linear discriminant classifiers (see Eq. (1)). An extension of the algorithm to the non-linear case can be found in the discussion section (Section 6). If the training data set is linearly separable, a linear SVM is a maximum margin classifier. The decision boundary (a straight line in the case of a two-dimensional separation) is positioned to leave the largest possible margin on either side. A particularity of SVMs is that the weights w_i of the decision function $D(\mathbf{x})$ are a function only of a small subset of the training examples, called "support vectors". Those are the examples that are closest to the decision boundary and lie on the margin. The existence of such support vectors is at the origin of the computational properties of SVM and their competitive classification performance. While SVMs base their decision function on the support vectors that are the borderline cases, other methods such as the method used by Golub et al. (1999) base their decision function on the average case. As we shall see in the discussion section (Section 6), this has also consequences on the feature selection process.

In this paper, we use one of the variants of the soft-margin algorithm described in Cortes (1995). Training consists in executing the following quadratic program:

Algorithm SVM-train:

Inputs: Training examples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_k, \ldots \mathbf{x}_\ell\}$ and class labels $\{\mathrm{y}_1, \mathrm{y}_2, \ldots \mathrm{y}_k, \ldots \mathrm{y}_\ell\}$.

$$\begin{cases} \text{Minimize over } \alpha_{\mathrm{k}}: \\ \mathrm{J} = (1/2) \displaystyle\sum_{hk} \mathrm{y_h}\, \mathrm{y_k}\, \alpha_{\mathrm{h}}\, \alpha_{\mathrm{k}}\, (\mathbf{x}_{\mathrm{h}} \cdot \mathbf{x}_{\mathrm{k}} + \lambda \delta_{\mathrm{hk}}) - \displaystyle\sum_{k} \alpha_{\mathrm{k}} \\ \text{subject to:} \\ 0 \leq \alpha_{\mathrm{k}} \leq \mathrm{C} \quad \text{and} \quad \displaystyle\sum_{k} \alpha_{\mathrm{k}}\, \mathrm{y_k} = 0 \end{cases} \tag{5}$$

Outputs: Parameters α_k.

The summations run over all training patterns $\mathbf{x}_k$ that are n dimensional feature vectors, $\mathbf{x}_h \cdot \mathbf{x}_k$ denotes the scalar product, y_k encodes the class label as a binary value $+1$ or -1, δ_{hk} is the Kronecker symbol ($\delta_{hk} = 1$ if $h = k$ and 0 otherwise), and λ and C are positive constants (soft margin parameters). The soft margin parameters ensure convergence even when the problem is non-linearly separable or poorly conditioned. In such cases, some of the support vectors may not lie on the margin. Most authors use either λ or C. We use a small value of λ (of the order of 10^{-14}) to ensure numerical stability. For the problems under study, the solution is rather insensitive to the value of C because the training data sets are linearly separable down to just a few features. A value of $C = 100$ is adequate.

The resulting decision function of an input vector $\mathbf{x}$ is:

$$D(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + \mathrm{b}$$

with

$$\mathbf{w} = \sum_k \alpha_k \, \mathrm{y}_k \, \mathbf{x}_k \quad \text{and} \quad b = \langle \mathrm{y}_k - \mathbf{w} \cdot \mathbf{x}_k \rangle$$

The weight vector $\mathbf{w}$ is a linear combination of training patterns. Most weights α_k are zero. The training patterns with non-zero weights are support vectors. Those with weight satisfying the strict inequality $0 < \alpha_k < C$ are marginal support vectors. The bias value b is an average over marginal support vectors.

Many resources on support vector machines, including computer implementations can be found at: http://www.kernel-machines.org.

3.2. SVM Recursive Feature Elimination (SVM RFE)

SVM RFE is an application of RFE using the weight magnitude as ranking criterion.

We present below an outline of the algorithm in the linear case, using *SVM-train* in Eq. (5). An extension to the non-linear case is proposed in the discussion section (Section 6).

Algorithm SVM-RFE:

Inputs:

Training examples

$$\mathbf{X}_0 = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_\mathrm{k}, \ldots \mathbf{x}_\ell]^\mathrm{T}$$

Class labels

$$\mathbf{y} = [\mathrm{y}_1, \mathrm{y}_2, \ldots \mathrm{y}_\mathrm{k}, \ldots \mathrm{y}_\ell]^\mathrm{T}$$

Initialize:

Subset of surviving features

$$\mathbf{s} = [1, 2, \ldots \mathrm{n}]$$

Feature ranked list

$$r = [\,]$$

Repeat until $\mathbf{s} = [\,]$

Restrict training examples to good feature indices

$$X = X_0(:, \mathbf{s})$$

Train the classifier

$$\alpha = \textit{SVM-train}(X, \mathbf{y})$$

Compute the weight vector of dimension length(**s**)

$$\mathbf{w} = \sum_k \alpha_k \, y_k \, \mathbf{x}_k$$

Compute the ranking criteria

$$c_i = (w_i)^2, \quad \text{for all } i$$

Find the feature with smallest ranking criterion

$$f = \textit{argmin}(\mathbf{c})$$

Update feature ranked list

$$\mathbf{r} = [\mathbf{s}(f), \mathbf{r}]$$

Eliminate the feature with smallest ranking criterion

$$\mathbf{s} = \mathbf{s}(1{:}f-1, f+1{:}\text{length}(\mathbf{s}))$$

Output:
Feature ranked list **r**.

As mentioned before the algorithm can be generalized to remove more than one feature per step for speed reasons.

4. Material and experimental method

4.1. Description of the data sets

We present results on two data sets both of which consist of a matrix of gene expression vectors obtained from DNA micro-arrays (Fodor, 1997) for a number of patients. The first set was obtained from cancer patients with two different types of leukemia. The second set was obtained from cancerous or normal colon tissues. Both data sets proved to be relatively easy to separate. After preprocessing, it is possible to find a weighted sum of a set of only a few genes that separates without error the entire data set (the data set is linearly separable). Although the separation of the data is easy, the problems present several features of difficulty, including small sample sizes and data differently distributed between training and test set (in the case of leukemia).

One particularly challenging problem in the case of the colon cancer data is that "tumor" samples and "normal" samples differ in cell composition. Tumors are generally rich in epithelial (skin) cells whereas normal tissues contain a variety of cells, including a large fraction of smooth muscle cells. Therefore, the samples can easily be split on the basis of cell composition, which is not informative for tracking cancer-related genes.

4.1.1. Differentiation of two types of Leukemia. In Golub (1999), the authors present methods for analyzing gene expression data obtained from DNA micro-arrays in order

to classify types of cancer. Their method is illustrated on leukemia data that is available on-line.

The problem is to distinguish between two variants of leukemia (ALL and AML). The data is split into two subsets: A training set, used to select genes and adjust the weights of the classifiers, and an independent test set used to estimate the performance of the system obtained. Their training set consists of 38 samples (27 ALL and 11 AML) from bone marrow specimens. Their test set has 34 samples (20 ALL and 14 AML), prepared under different experimental conditions and including 24 bone marrow and 10 blood sample specimens. All samples have 7129 features, corresponding to some normalized gene expression value extracted from the micro-array image. We retained the exact same experimental conditions for ease of comparison with their method.

In our preliminary experiments, some of the large deviations between leave-one-out error and test error could not be explained by the small sample size alone. Our data analysis revealed that there are significant differences between the distribution of the training set and the test set. We tested various hypotheses and found that the differences can be traced to differences in the data sources. In all our experiments, we followed separately the performance on test data from the various sources. However, since it ultimately did not affect our conclusions, we do not report these details here for simplicity.

4.1.2. Colon cancer diagnosis. In Alon (1999), the authors describe and study a data set that is available on-line. Gene expression information was extracted from DNA micro-array data resulting, after pre-processing, in a table of 62 tissues $\times$ 2000 gene expression values. The 62 tissues include 22 normal and 40 colon cancer tissues. The matrix contains the expression of the 2000 genes with highest minimal intensity across the 62 tissues. Some genes are non-human genes.

The paper of Alon et al. provides an analysis of the data based on top down hierarchical clustering, a method of unsupervised learning. They show that most normal samples cluster together and most cancer samples cluster together. They explain that "outlier" samples that are classified in the wrong cluster differ in cell composition from typical samples. They compute a so-called "muscle index" that measures the average gene expression of a number of smooth muscle genes. Most normal samples have high muscle index and cancer samples low muscle index. The opposite is true for most outliers.

Alon et al. also cluster genes. They show that some genes are correlated with the cancer vs. normal separation but do not suggest a specific method of gene selection. Our reference gene selection method will be that of Golub et al. that was demonstrated on leukemia data (Golub, 1999). Since there was no defined training and test set, we split randomly the data into 31 samples for training and 31 samples for testing.

4.2. Assessment of classifier quality

In Golub (1999), the authors use several metrics of classifier quality, including error rate, rejection rate at fixed threshold, and classification confidence. Each value is computed both on the independent test set and using the *leave-one-out method* on the training set. The leave-one-out procedure consists of removing one example from the training set, constructing the

decision function on the basis only of the remaining training data and then testing on the removed example. In this fashion one tests all examples of the training data and measures the fraction of errors over the total number of training examples.

In this paper, in order to compare methods, we use a slightly modified version of these metrics. The classification methods we compare use various decision functions $D(\mathbf{x})$ whose inputs are gene expression coefficients and whose outputs are a signed number. The classification decision is carried out according to the sign of $D(\mathbf{x})$. The magnitude of $D(\mathbf{x})$ is indicative of classification confidence.

We use four metrics of classifier quality (see figure 1):

- **Error** (B1 + B2) = number of errors ("bad") at zero rejection.
- **Reject** (R1 + R2) = minimum number of rejected samples to obtain zero error.
- **Extremal margin** (E/D) = difference between the smallest output of the positive class samples and the largest output of the negative class samples (rescaled by the largest difference between outputs).

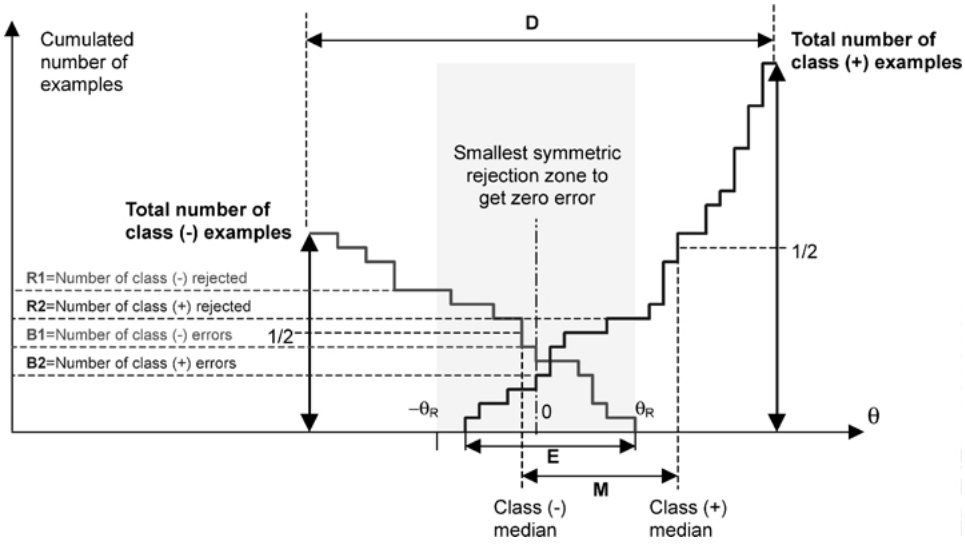

Figure 1. Metrics of classifier quality. The gray and black curves represent example distributions of two classes: class (−) and class (+). Gray: Number of examples of class (−) whose decision function value is larger than or equal to θ. Black: Number of examples of class (+) whose decision function value is smaller than or equal to θ. The number of errors B1 and B2 are the ordinates of $\theta = 0$. The number of rejected examples R1 and R2 are the ordinates of $-\theta_R$ and θ_R in the gray and black curves respectively. The decision function value of the rejected examples is smaller than θ_R in absolute value, which corresponds to examples of low classification confidence. The threshold θ_R is set such that all the remaining "accepted" examples are well classified. The extremal margin E is the difference between the smallest decision function value of class (+) examples and the largest decision function value of class (−) examples. On the example of the figure, E is negative. If the number of classification error is zero, E is positive. The median margin M is the difference in median decision function value of the class (+) density and the class (−) density.

– **Median margin** (M/D) = difference between the median output of the positive class samples and the median output of the negative class samples (re-scaled by the largest difference between outputs).

Each value is computed both on the training set with the leave-one-out method and on the test set.

The **error rate** is the fraction of examples that are misclassified (corresponding to a diagnostic error). It is complemented by the **success rate**. The **rejection rate** is the fraction of examples that are rejected (on which no decision is made because of low confidence). It is complemented by the **acceptance rate**. Extremal and median margins are measurements of classification confidence.

Notice that this notion of margin computed with the leave-one-out method or on the test set differs from the margin computed on training examples sometimes used in model selection criteria (Vapnik, 1998).

4.3. Accuracy of performance measurements with small sample sizes

Because of the very small sample sizes, we took special care in evaluating the statistical significance of the results. In particular, we address:

1. How accurately the test performance predicts the true classifier performance (measured on an infinitely large test set).
2. With what confidence we can assert that one classifier is better than another when its test performance is better than the other is.

Classical statistics provide us with error bars that answer these questions (for a review, see e.g. Guyon, 1998). Under the conditions of our experiments, we often get 1 or 0 error on the test set. We used a z-test with a standard definition of "statistical significance" (95% confidence). For a test sample of size $t = 30$ and a true error rate $p = 1/30$, the difference between the observed error rate and the true error rate can be as large as 5%. We use the formula $\varepsilon = z_\eta \,\mathrm{sqrt}(p(1-p)/t)$, where $z_\eta = \mathrm{sqrt}(2)\,\mathrm{erfinv}(-2(\eta - 0.5))$, and erfinv is the inverse error function, which is tabulated. This assumes i.i.d. errors, one-sided risk and the approximation of the Binomial law by the Normal law. This is to say that the absolute performance results (question 1) should be considered with extreme care because of the large error bars.

In contrast, it is possible to compare the performance of two classification systems (relative performance, question 2) and, in some cases, assert with confidence that one is better than the other is. For that purpose, we shall use the following statistical test (Guyon, 1998).

With confidence $(1 - \eta)$ we can accept that one classifier is better than the other, using the formula:

$$\begin{aligned} (1-\eta) &= 0.5 + 0.5\,\mathrm{erf}(z_\eta/\mathrm{sqrt}(2)) \\ z_\eta &= \varepsilon\, t/\mathrm{sqrt}(\nu) \end{aligned} \tag{6}$$

where t is the number of test examples, ν is the total number of errors (or rejections) that only one of the two classifiers makes, ε is the difference in error rate (or in rejection rate), and erf is the error function $\mathrm{erf}(x) = \int_0^x \exp(-t^2)\,dt$.

This assumes i.i.d. errors, one-sided risk and the approximation of the Binomial law by the Normal law.

5. Experimental results

5.1. The features selected matter more than the classifier used

In a first set of experiments, we carried out a comparison between the method of Golub et al. and SVMs on the leukemia data. We de-coupled two aspects of the problem: selecting a good subset of genes and finding a good decision function. We demonstrated that the performance improvements obtained with SVMs could be traced to the SVM feature (gene) selection method. The particular decision function that is trained with these features matters less.

As suggested in Golub (1999) we performed a simple preprocessing step. From each gene expression value, we subtracted its mean and divided the result by its standard deviation. We used the Recursive Feature Elimination (RFE) method, as explained in Section 3. We eliminated chunks of genes at a time. At the first iteration, we reached the number of genes, which is the closest power of 2. At subsequent iterations, we eliminated half of the remaining genes. We thus obtained nested subsets of genes of increasing informative density. The quality of these subsets of genes was then assessed by training various classifiers, including a linear SVM, the Golub et al. classifier, and Fisher's linear discriminant (see e.g. Duda, 1973).

The various classifiers that we tried did not yield significantly different performance. We report the results of the classifier of Golub (1999) and a linear SVM. We performed several cross tests with the baseline method to compare gene sets and classifiers (figure 2 and Tables 1–4): SVMs trained on SVM selected genes or on baseline genes and baseline classifier trained on SVM selected genes or on baseline genes. Baseline classifier refers to the classifier of Eq. (3) described in Golub (1999). Baseline genes refer to genes selected according to the ranking criterion of Eq. (2) described in Golub (1999). In figure 2, the larger the colored area, the better the classifier. It is easy to see that a change in classification method does not affect the result significantly whereas a change in gene selection method does.

Table 5 summarizes the best results obtained on the test set for each combination of gene selection and classification method. The classifiers give identical results, given a gene selection method. This result is consistent with Furey (2000) who observed on the same data no statistically significant difference in classification performance for various classifiers all trained with genes selected by the method of Golub (1999). In contrast, the SVM selected genes yield consistently better performance than the baseline genes for both classifiers. This is a new result compared to Furey (2000) since the authors did not attempt to use SVMs for gene selection. Other authors also report performance improvements for SVM selected genes using other algorithms (Mukherjee, 1999; Chapelle, 2000; Weston, 2000). The details are reported in the discussion section (Section 6).

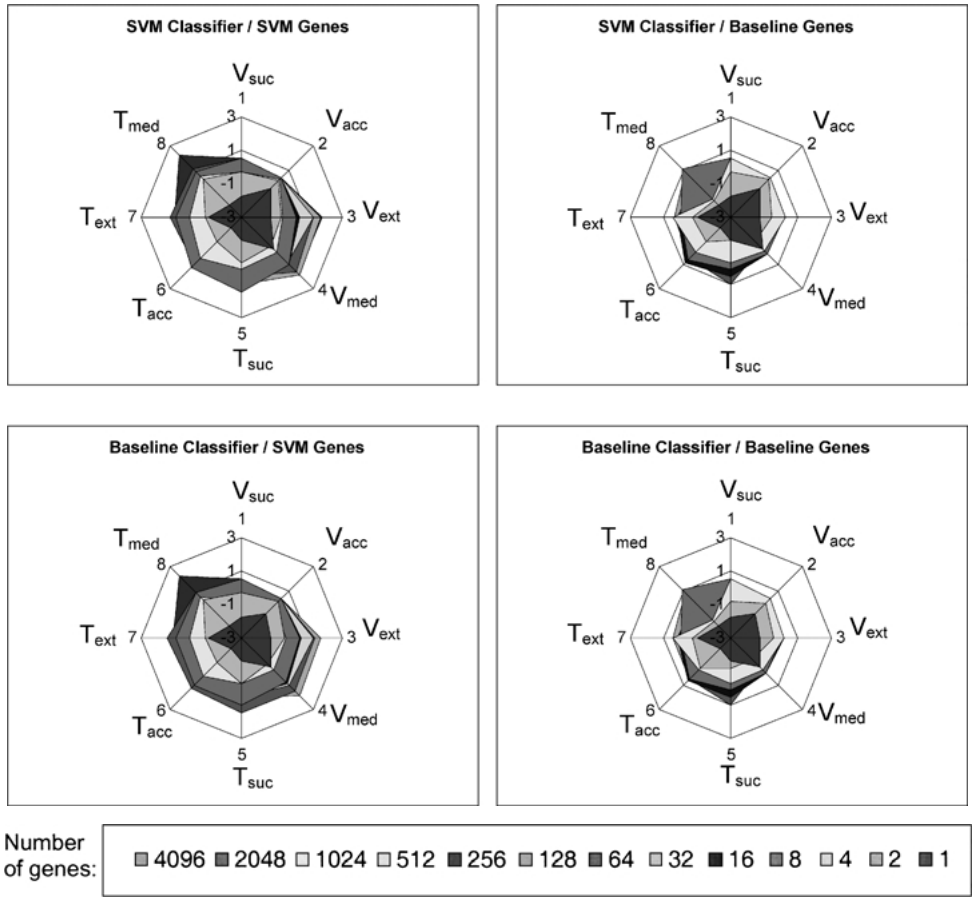

Figure 2. Performance comparison between SVMs and the baseline method (Leukemia data). Classifiers have been trained with subsets of genes selected with SVMs and with the baseline method (Golub, 1999) on the training set of the Leukemia data. The number of genes is color coded and indicated in the legend. The quality indicators are plotted radially: channel 1–4 = cross-validation results with the leave-one-out method; channels 5–8 = test set results; suc = success rate; acc = acceptance rate; ext = extremal margin; med = median margin. The coefficients have been rescaled such that each indicator has zero mean and variance 1 across all four plots. For each classifier, the larger the colored area, the better the classifier. The figure shows that there is no significant difference between classifier performance on this data set, but there is a significant difference between the gene selections.

We tested the significance of the difference in performance with Eq. (6). Whether SVM or baseline classifier, SVM genes are better with 84.1% confidence based on test error rate and 99.2% based on the test rejection rate.

To compare the top ranked genes, we computed the fraction of common genes in the SVM selected subsets and the baseline subsets. For 16 genes or less, at most 25% of the genes are common.

We show in figure 3(a) and (c) the expression values of the 16-gene subsets for the training set patients. At first sight, the genes selected by the baseline method look a lot more orderly.

Table 1. SVM classifier trained on SVM genes obtained with the RFE method (Leukemia data).

	Training set (38 samples)				Test set (34 samples)			
Number of genes	V_{suc}	V_{acc}	V_{ext}	V_{med}	T_{suc}	T_{acc}	T_{ext}	T_{med}
All (7129)	0.95	0.87	0.01	0.42	0.85	0.68	−0.05	0.42
4096	0.82	0.05	−0.67	0.30	0.71	0.09	−0.77	0.34
2048	0.97	0.97	0.00	0.51	0.85	0.53	−0.21	0.41
1024	1.00	1.00	0.41	0.66	0.94	0.94	−0.02	0.47
512	0.97	0.97	0.20	0.79	0.88	0.79	0.01	0.51
256	1.00	1.00	0.59	0.79	0.94	0.91	0.07	0.62
128	1.00	1.00	0.56	0.80	0.97	0.88	−0.03	0.46
64	1.00	1.00	0.45	0.76	0.94	0.94	0.11	0.51
32	1.00	1.00	0.45	0.65	0.97	0.94	0.00	0.39
16	**1.00**	**1.00**	**0.25**	**0.66**	**1.00**	**1.00**	**0.03**	**0.38**
8	**1.00**	**1.00**	**0.21**	**0.66**	**1.00**	**1.00**	**0.05**	**0.49**
4	0.97	0.97	0.01	0.49	0.91	0.82	−0.08	0.45
2	0.97	0.95	−0.02	0.42	0.88	0.47	−0.23	0.44
1	0.92	0.84	−0.19	0.45	0.79	0.18	−0.27	0.23

The success rate (at zero rejection), the acceptance rate (at zero error), the external margin and the median margin are reported for the leave-one-out method on the 38 sample training set (V results) and the 34 sample test set (T results). We outline in boldface the classifiers performing best on test data reported in Table 5. For comparison, we also show the results on all genes (no selection).

This is because they are strongly correlated with either AML or ALL. There is therefore a lot of redundancy in this gene set. In essence, all the genes carry the same information. Conversely, the SVM selected genes carry complementary information. This is reflected in the output of the decision function (figure 3(b) and (d)), which is a weighted sum of the 16 gene expression values. The SVM output separates AML patients from ALL patients more clearly.

5.2. SVMs select relevant genes

In another set of experiments, we compared the effectiveness of various feature selection techniques. Having established in Section 5.1 that the features selected matter more than the classifier, we compared various feature selection techniques with the same classifier (a linear SVM). The comparison is made on Colon cancer data because it is a more difficult data set and therefore allows us to better differentiate methods.

In this section, unless otherwise stated, we use Recursive Feature Elimination (RFE) by eliminating one gene at a time. We use a number of preprocessing steps that include: taking the logarithm of all values, normalizing sample vectors, normalizing feature vectors, and passing the result through a squashing function of the type $f(x) = c\,\mathrm{antan}(x/c)$ to diminish the importance of outliers. Normalization consists in subtracting the mean over all training values and dividing by the corresponding standard deviation.

Table 2. SVM classifier trained on baseline genes (Leukemia data).

	Training set (38 samples)				Test set (34 samples)			
Number of genes	V_{suc}	V_{acc}	V_{ext}	V_{med}	T_{suc}	T_{acc}	T_{ext}	T_{med}
All (7129)	0.95	0.87	0.01	0.42	0.85	0.68	−0.05	0.42
4096	0.92	0.18	−0.43	0.29	0.74	0.18	−0.68	0.36
2048	0.95	0.95	−0.09	0.32	0.85	0.38	−0.25	0.33
1024	1.00	1.00	0.09	0.34	0.94	0.62	−0.13	0.34
512	1.00	1.00	0.08	0.39	0.94	0.76	−0.06	0.37
256	1.00	1.00	0.08	0.40	0.91	0.79	−0.04	0.42
128	1.00	1.00	0.09	0.39	0.94	0.82	−0.04	0.49
64	**0.97**	**0.97**	**0.01**	**0.44**	**0.97**	**0.82**	**−0.09**	**0.44**
32	1.00	1.00	0.07	0.46	0.91	0.88	−0.07	0.42
16	1.00	1.00	0.16	0.52	0.94	0.91	−0.07	0.39
8	1.00	1.00	0.17	0.52	0.91	0.85	−0.10	0.51
4	1.00	1.00	0.21	0.48	0.88	0.68	−0.03	0.28
2	0.97	0.97	0.00	0.36	0.79	0.47	−0.22	0.27
1	0.92	0.84	−0.19	0.45	0.79	0.18	−0.27	0.23

The success rate (at zero rejection, the acceptance rate (at zero error), the extremal margin and the median margin are reported for the leave-one-out method on the 38 sample training set (V results) and the 34 sample test set (T results). We outline in boldface the classifiers performing best on test data reported in Table 5. For comparison, we also show the results on all genes (no selection).

We first conducted a preliminary set of experiments using the data split of 31 training samples and 31 test samples. We summarize the results of the comparison between the SVM method and the baseline method (Golub, 1999) in Table 6. According to the statistical test of Eq. (6) computed on the error rate, the SVM method (SVM classifier trained on SVM genes) is significantly better than the baseline method (baseline classifier trained on baseline genes).

On the basis of this test, we can accept that the SVM is better than the baseline method with 95.8% confidence. In addition, the SVM achieves better performance with fewer genes.

Yet, the rejection rate reveals that some of the misclassified examples are very far from the decision boundary: most of the examples must be rejected to yield zero error. We examined the misclassified examples. As mentioned previously, the tissue composition of the samples is not uniform. Most tumor tissues are rich in epithelial (skin) cells and most normal samples are rich in muscle cells. The muscle index is a quantity computed by Alon et al. (1999) that reflects the muscle cell contents of a given sample. Most misclassified examples have an inverted muscle index (high for tumor tissues and low for normal tissues). An analysis of the genes discovered reveals that on such a small training data set both methods rank first a smooth muscle gene (gene J02854). Therefore, the separation is made on the basis of tissue composition rather than the distinction cancer vs. normal. We conjectured that the size of the training set was insufficient for the SVM to eliminate

Table 3. Baseline classifier trained on SVM genes obtained with the RFE method (Leukemia data).

	Training set (38 samples)				Test set (34 samples)			
Number of genes	V_{suc}	V_{acc}	V_{ext}	V_{med}	T_{suc}	T_{acc}	T_{ext}	T_{med}
All (7129)	0.89	0.47	−0.25	0.28	0.85	0.35	−0.24	0.34
4096	0.97	0.97	0.01	0.41	0.88	0.59	−0.12	0.40
2048	1.00	1.00	0.29	0.56	0.88	0.76	−0.07	0.45
1024	1.00	1.00	0.44	0.67	0.94	0.82	0.01	0.47
512	1.00	1 00	0.39	0.81	0.91	0.88	0.07	0.55
256	1.00	1.00	0.55	0.76	0.94	0.94	0.09	0.62
128	1.00	1.00	0.56	0.81	0.94	0.82	0.02	0.45
64	**1.00**	**1.00**	**0.47**	**0.74**	**1.00**	**1.00**	**0.14**	**0.49**
32	1.00	1.00	0.44	0.66	0.94	0.79	0.01	0.40
16	1.00	1.00	0.27	0.63	0.94	0.91	0.03	0.39
8	1.00	1.00	0.25	0.62	0.97	0.94	0.05	0.50
4	0.95	0.89	0.04	0.45	0.88	0.76	−0.09	0.45
2	0.97	0.95	0.03	0.39	0.88	0.44	−0.23	0.44
1	0.92	0.76	−0.17	0.43	0.79	0.18	−0.27	0.23

The success rate (at zero rejection), the acceptance rate (at zero error), the extremal margin and the median margin are reported for the leave-one-out method on the 38 sample training set (V results) and the 34 sample test set (T results). We outline in boldface the classifiers performing best on test data reported in Table 5. For comparison, we also show the results on all genes (no selection).

tissue composition related genes that are presumably irrelevant to the cancer vs. normal separation.

In a second set of experiments, to increase the training set size, we placed all the Colon cancer data into one training set of 62 samples. We used the leave-one-out method to assess performance.

The best leave-one-out performance is 100% accuracy for the SVMs (SVM classifier trained on SVM genes) and only 90% for the baseline method (baseline classifier trained on baseline genes). Using the statistical test of Eq. (6), we can assert with 99.3% confidence that SVMs are better than the baseline method. An analysis of the genes discovered reveals that the first smooth muscle gene ranks 5 for the baseline method and only 41 for SVMs. SVMs seem to be able to avoid relying on tissue composition related genes to do the separation. As confirmed by biological data presented in Section 5.3, the top ranking genes discovered by SVMs are all plausibly related to the cancer vs. normal separation. In contrast, the baseline method selects genes that are plausibly related to tissue composition and not to the distinction cancer vs. normal in its top ranking genes.

It is instructive to examine the support vectors to understand the mechanism of gene selection used by SVM RFE. The α's do not vary a lot until the last few iterations. The number of support vectors goes through a minimum at 7 genes for 7 support vectors (it is coincidental that the two numbers are 7). At this point, the leave-one-out error is zero. In Table 7, we show the "muscle index" values of these 7 support vectors. We remind that

Table 4. Baseline classifier trained on baseline genes (Leukemia data).

	Training set (38 samples)				Test set (34 samples)			
Number of genes	V_{suc}	V_{acc}	V_{ext}	V_{med}	T_{suc}	T_{acc}	T_{ext}	T_{med}
All (7129)	0.89	0.47	−0.25	0.28	0.85	0.35	−0.24	0.34
4096	0.95	0.76	−0.12	0.33	0.85	0.44	−0.20	0.37
2048	0.97	0.97	0.02	0.36	0.85	0.53	−0.13	0.37
1024	1.00	1.00	0.11	0.36	0.94	0.65	−0.11	0.37
512	1.00	1 00	0.11	0.39	0.94	0.79	−0.05	0.40
256	1.00	1.00	0.11	0.40	0.91	0.76	−0.02	0.43
128	1.00	1.00	0.12	0.39	0.94	0.82	−0.02	0.50
64	**1.00**	**1.00**	**0.07**	**0.43**	**0.97**	**0.82**	**−0.08**	**0.45**
32	1.00	1.00	0.11	0.44	0.94	0.85	−0.07	0.42
16	1.00	1.00	0.18	0.50	0.94	0.85	−0.07	0.40
8	1.00	1.00	0.15	0.50	0.91	0.82	−0.10	0.51
4	1.00	1.00	0.18	0.45	0.88	0.62	−0.03	0.28
2	0.95	0.92	0.02	0.33	0.82	0.59	−0.22	0.27
1	0.92	0.76	−0.17	0.43	0.79	0.18	−0.27	0.23

The success rate (at zero rejection), the acceptance rate (at zero error), the extremal margin and the median margin are reported for the leave-one-out method on the 38 sample training set (V results) and the 34 sample test set (T results). We outline in boldface the classifiers performing best on test data reported in Table 5. For comparison, we also show the results on all genes (no selection).

Table 5. Best classifiers on test data (Leukemia data).

Selection method	SVM RFE			Baseline feature selection			No feature selection		
classifier	#genes	Error # (0 rej.)	Reject # (0 error)	#genes	Error # (0 rej.)	Reject # (0 error)	#genes	Error # (0 rej.)	Reject # (0 error)
SVM classifier	8, 16	0 {}	0 {}	64	1 {28}	6 {4, 16, 22, 23, 28, 29}	7129	5 {16, 19, 22, 23, 28}	11 {2, 4, 14, 16, 19, 20, 22, 23, 24, 27,28}
Baseline classifier	64	0 {}	0 {}	64	1 {28}	6 {4, 16, 22, 23, 28, 29}	7129	5 {16, 19, 22, 27, 28}	22 {1, 2, 4, 5, 7, 11, 13, 14, 16–20, 22–29, 33}

The performance of the classifiers performing best on test data (34 samples) are reported. The baseline method is described in Golub (1999) and SVM RFE is used for feature (gene) selection (see text). For each combination of SVM or Baseline genes and SVM or Baseline classifier, the corresponding number of genes, the number of errors at zero rejection and the number of rejections at zero error are shown in the table. The number of genes refers to the number of genes of the subset selected by the given method yielding best classification performance. The patient id numbers of the classification errors are shown in brackets. For comparison, we also show the results with no gene selection.

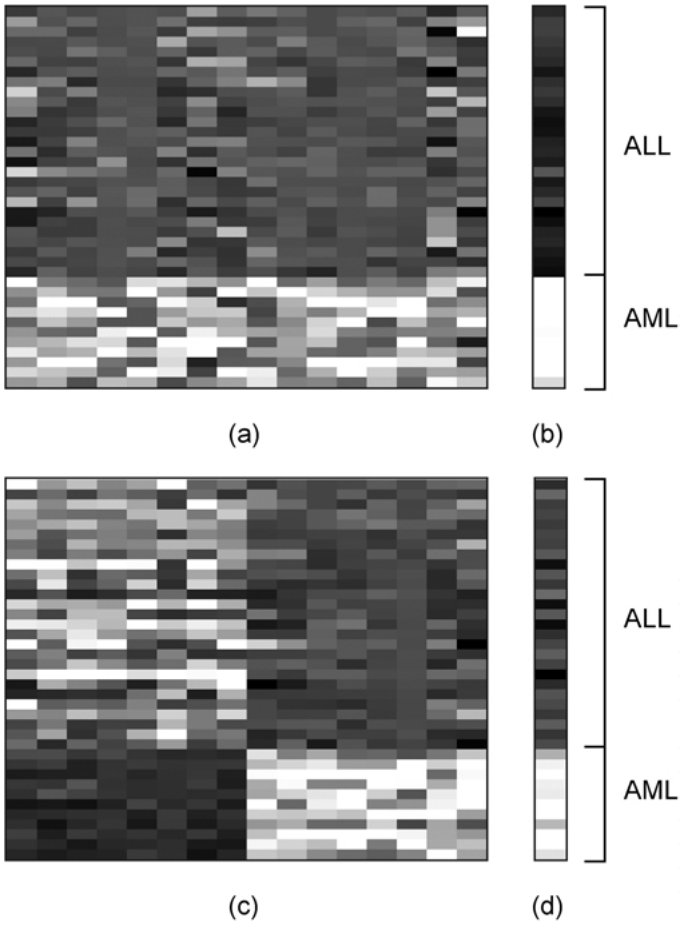

Figure 3. Best sets of 16 genes (Leukemia data). In matrices (a) and (c), the columns represent different genes and the lines different patients from the training set. The 27 top lines are ALL patients and the 11 bottom lines are AML patients. The gray shading indicates gene expression: the lighter the stronger. (a) SVM best 16 genes. Genes are ranked from left to right, the best one at the extreme left. All the genes selected are more AML correlated. (b) Weighted sum of the 16 SVM genes used to make the classification decision. A very clear ALL/AML separation is shown. (c) Baseline method (Golub, 1999) 16 genes. The method imposes that half of the genes are AML correlated and half are ALL correlated. The best genes are in the middle. (d) Weighted sum of the 16 baseline genes used to make the classification decision. The separation is still good, but not as contrasted as the SVM separation.

the muscle index is a quantity computed by Alon et al. (1999) that reflects the muscle cell contents of a given sample. Most normal samples have a higher muscle index than tumor samples. However, the support vectors do not show any such trend. There is a mix of normal and cancer samples with either high or low muscle index.

Table 6. Best classifiers on test data (Leukemia data).

Selection method / classifier	SVM RFE #genes	SVM RFE Error # (0 rej.)	SVM RFE Reject # (0 error)	Baseline feature selection #genes	Baseline feature selection Error # (0 rej.)	Baseline feature selection Reject # (0 error)	No feature selection #genes	No feature selection Error # (0 rej.)	No feature selection Reject # (0 error)
SVM classifier	8	3 {36, 34, −36}	29	8	5 {28, 36, 11, 34, −36}	24	2000	5 {36, 34, −36, −30, −2}	30
Baseline classifier	32	4 {36, 34, −36, −30}	21	16	6 {8, 36, 34, −36, −30, 2}	21	2000	7 {8, 36, 34, −37, −36, −30, −2}	21

The performance of the classifiers performing best on test data (31 samples) are reported. The baseline method is described in Golub (1999) and SVM RFE is used for feature (gene) selection (see text). For each combination of SVM or Baseline genes and SVM or Baseline classifier, the corresponding number of genes, the number of errors at zero rejection and the number of rejections at zero error are shown in the table. The number of genes refers to the number of genes of the subset selected by the given method yielding best classification performance. The list of errors is shown between brackets. The numbers indicate the patients. The sign indicates cancer (negative) or normal (positive). For comparison, we also show the results with no gene selection.

Table 7. Muscle index of 7 the support vectors of an SVM trained on the top 7 genes selected by SVM RFE (Colon cancer data).

Support vector samples	−6 (**T**)	8 (**N**)	34 (**N**)	−37 (**T**)	9 (**N**)	−30 (**T**)	−36 (**T**)
Muscle index	0.009	0.2	0.2	0.3	0.3	0.4	0.7

Samples with a negative sign are tumor tissues (T). Samples with positive signs are normal tissues (N). We ranked samples in ordered of increasing muscle index. In most samples in the data set, normal tissues have higher muscle index than tumor tissues because tumor tissues are richer in epithelial (skin) cells. This is not the case for support vectors which show a mix of all possibilities. This particular gene subset selected by SVM RFE corresponds to the smallest number of support vectors (seven). Coincidentally, it also corresponds the smallest number of genes (seven) that yields zero training error and zero leave-one-out error.

As a feature selection method, SVM RFE differs from the baseline method in two respects:

– The mutual information between features is used by SVMs (SVMs are multivariate classifiers) whereas the baseline method makes implicit orthogonality assumptions (it can be considered as a combination of univariate classifiers).
– The decision function is based only on support vectors that are "borderline" cases as opposed to being based on all examples in an attempt to characterize the "typical" cases.

We assume that the use of support vectors is critical in eliminating irrelevant tissue composition related genes. To verify experimentally that hypothesis, we compared SVM RFE with RFE methods using other multivariate linear discriminant functions that do not make orthogonality assumptions but attempt to characterize the "typical" cases.

We chose two discriminant functions:

– The Fisher linear discriminant also called Linear Discriminant Analysis (LDA) (see e.g. Duda, 1973) because the baseline method approximates Fisher's linear discriminant by making orthogonality assumptions. We compute LDA by solving a generalized eigenvalue problem (Duda, 1973).

– The Mean-Squared-Error (MSE) linear discriminant computed by Pseudo-inverse (Duda, 1973), because when all training examples are support vectors the pseudo-inverse solution is identical to the SVM solution. The MSE discriminant is obtained by calculating $[\mathbf{w}, b]^T = [X, 1]^T([X, 1][X, 1]^T)^{-1}\mathbf{y}$, where $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_k, \ldots \mathbf{x}_\ell]^T$, $\mathbf{y} = [y_1, y_2, \ldots y_k, \ldots y_\ell]^T$, and $\mathbf{1}$ is an ℓ dimensional vector of ones. This requires only the inversion of an (ℓ, ℓ) matrix.

We show the result of comparison in figure 4. All multivariate methods outperform the baseline method and reach 100% leave-one-out accuracy for at least one value of the number of genes. LDA may be at a slight disadvantage on these plots because, for computational reasons, we used RFE by eliminating chunks of genes that decrease in size by powers of two. Other methods use RFE by eliminating one gene at a time.

Down to 4 genes, SVM RFE shows better performance than all the other methods. We examined the genes ranking 1 through 64 for all the methods studied. The first gene that

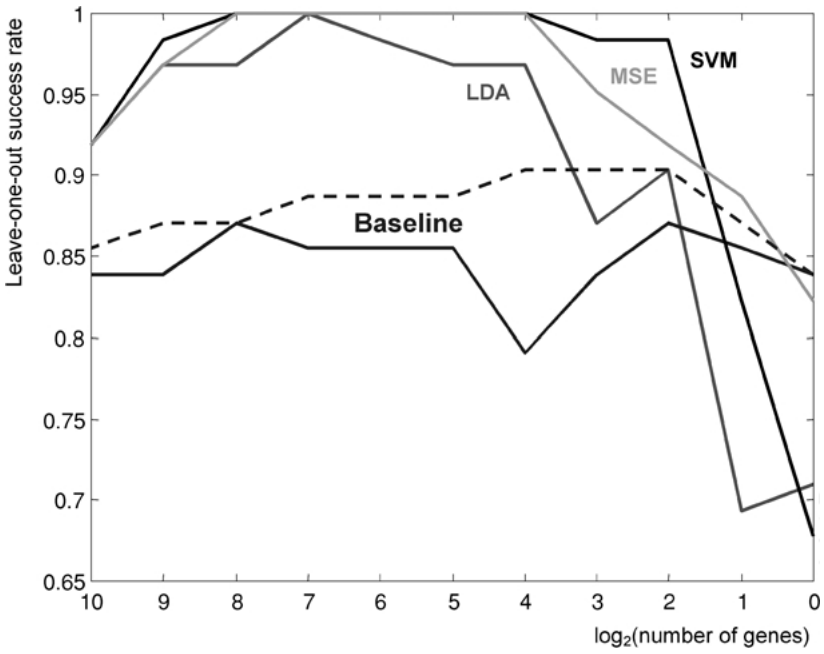

Figure 4. Comparison of feature (gene) selection methods (Colon cancer data). We varied the number of genes selected by Recursive Feature Elimination (RFE) with various methods. Training was done on the entire data set of 62 samples. The curves represent the leave-one-out success rate for the various feature selection methods. Top black: SVM RFE. Darkgray: Linear Discriminant Analysis RFE. Lightgray: Mean Squared Error (Pseudo-inverse) RFE. Bottom black: Baseline method (Golub, 1999). The solid line indicates that the classifier used is an SVM. The dashed line indicates that the classifier used is the same as the one used to select the genes. This corresponds to a single experiment for SVM. For MSE, the dashed and solid lines overlap. For LDA we could not compute the dashed line, for computational reasons. The baseline method performs slightly better when used with its own classifier. SVM RFE gives the best results down to 4 genes.

is related to tissue composition and mentions "smooth muscle" in its description ranks 5 for the baseline method, 4 for LDA, 1 for MSE and only 41 for SVM. Therefore, this is an indication that SVMs might make a better use of the data than the other methods via the support vector mechanism. They are the only method tested that effectively eliminates tissue composition related genes while providing highly accurate separations with a small subset of genes. In the discussion section (Section 6), we propose a qualitative explanation of the SVM feature selection mechanism.

5.3. Results validation with the biology literature

In Section 5.2, we found that SVM RFE eliminates from its top ranked genes smooth muscle genes that are likely to be tissue composition related. In this section, we individually checked the seven top ranked genes for relevance in Colon cancer diagnosis (Table 8).

Table 8. SVM RFE to ranked genes (Colon cancer data).

Rk	Expression	GAN	Description	Possible function/relation to Colon cancer
7	C > N	H08393	Collagen alpha 2(XI) chain (Homo sapiens)	Collagen is involved in cell adhesion. Colon carcinoma cells have collagen degrading activity as part of the metastatic process (Karakiulakis, 1997)
6	C > N	M59040	Human cell adhesion molecule (CD44) mRNA, complete cds	CD44 is upregulated when colon adenocarcinoma tumor cells transit to the metastatic state (Ghina, 1998)
5	C > N	T94579	Human chitotriosidase precursor mRNA, complete cds.	Another chitinase (BRP39) was found to play a role in breast cancer. Cancer cells overproduce this chitinase to survive apoptosis (Aronson, 1999).
4	N > C	H81558	Procyclic form specific polypeptide B1-alpha precursor (Trypanosoma brucei brucei)	Clinical studies report that patients infected by Trypanosoma (a colon parasite) develop resistance against colon cancer (Oliveira, 1999).
3	N > C	R88740	ATP synthase coupling factor 6, mitochondrial precursor (human)	ATP synthase is an enzyme that helps build blood vessels that feed the tumors (Mozer, 1999).
2	C > N	T62947	60S ribosomal protein L24 (Arabidopsis thaliana)	May play a role in controlling cell growth and proliferation through the selective translation of particular classes of mRNA.
1	N > C	H64807	Placental folate transporter (Homo translation of sapiens)	Diminished status of folate has been with enhanced risk of colon cancer (Walsh, 1999).

The entire data set of 62 samples was used to select genes with SVM RFE. Genes are ranked in order of increasing importance. The first ranked gene is the last gene left after all other genes have been eliminated. Expression: C > N indicates that the gene expression level is higher in most cancer tissues; N > C indicates that the gene expression level is higher in most normal tissues. GAN: Gene Accession Number. All the genes in this list have some plausible relevance to the cancer vs. normal separation.

The number seven corresponds to the minimum number of support vectors, a criterion sometimes used for "model selection". We did not attempt to find whether it was particularly relevant. Our findings are based on a bibliographic search and have been reviewed by one of us who is a medical doctor. However, they have not been subject to any experimental verification.

The role of some of the best-ranked genes can be easily explained because they code for proteins whose role in colon cancer has been long identified and widely studied. Such is the case of CD44, which is upregulated when colon adenocarcinoma tumor cells become metastatic (Ghina, 1998) and collagen, which is involved in cell adhesion. Colon carcinoma cells are known to have collagen degrading activity as part of the metastatic process (Karakiulakis, 1997). The presence of some other genes in our list can be explained by very recently published studies. For example, the role of ATP synthase, as an enzyme that helps build blood vessels (tumor angiogenesis), to feed the tumors was published only a year ago (Mozer, 1999).

Diminished status of folate has been associated with enhanced risk of colon cancer in a recent clinical study (Walsh, 1999). To this date, however, no known biochemical mechanism explained the role of folate in colon cancer. Knowing that gene H64807 (Placental folate transporter) was identified as one of the most discriminant genes in the colon cancer vs. normal separation can give researchers a clue where to start to investigate such mechanisms.

In the case of human chitotriosidase, one needs to proceed by analogy with another homologous protein of the same family whose role in another cancer is under study: another chitinase (BRP39) was found to play a role in breast cancer. Cancer cells overproduce this chitinase to survive apoptosis (Aronson, 1999). An important increase in chitotriosidase activity has been noticed in clinical studies of Gauchers disease patients, an apparently unrelated condition. To diagnose Gauchers disease the chitotriosidase enzyme can be very sensitively measured. The plasma or serum prepared from less than a droplet of blood is sufficient for the chitotriosidase measurement (Aerts, 1996). This opens the door to a possible new diagnosis test for colon cancer as well.

In addition we identified 60S ribosomal protein L24 (Arabidopsis thaliana). This non-human protein is homologous to a human protein located on chromosome 6. Like other ribosomal proteins, it may play a role in controlling cell growth and proliferation through the selective translation of particular classes of mRNA.

Finally, one of our most intriguing puzzles has been the "procyclic form specific polypeptide B1-alpha precursor (Trypanosoma brucei brucei)". We first found out that Trypanosoma is a tiny parasitic protozoa indigenous to Africa and South America. We thought this may be an anomaly of our gene selection method or an error in the data, until we discovered that clinical studies report that patients infected by Trypanosoma (a colon parasite) develop resistance against colon cancer (Oliveira, 1999). With this discovered knowledge there may be the possibility for developing a vaccine for colon cancer.

To complete our study, we proceeded similarly with the Leukemia data by running our gene selection method on the entire data set of 72 samples. We examined the four top ranked genes (Table 9). The number four corresponds to the minimum number of support vectors

Table 9. SVM RFE top ranked genes (Leukemia data).

Rk	Expression	GAN	Description	Possible function/relation to Leukemia
4	AML > ALL	U59632	Cell division control related protein (hCDCrel-1) mRNA	hCDCrel-1 is a partner gene of MLL in some leukemias (Osaka, 1999).
3	AML > ALL	U82759	GB DEF = Homeodomain protein HoxA9 mRNA	Hoxa9 collaborates with other genes to produce highly aggressive acute leukemic disease (Thorsteinsdottir, 1999).
2	ALL > AML	HG1612	MacMarcks	Tumor necrosis factor-alpha rapidly stimulate Marcks gene transcription in human promyelocytic leukemia cells (Harlan, 1991).
1	AML > ALL	X95735	Zyxin	Encodes a LIM domain protein localized at focal contacts in adherent erythroleukemia cells (Macalma, 1996).

The entire data set of 72 samples was used to select genes with SVM RFE. Genes are ranked in order of increasing importance. The first ranked gene is the last gene left after all other genes have been eliminated. Expression: ALL > AML indicates that the gene expression level is higher in most ALL samples; AML > ALL indicates that the gene expression level is higher in most AML samples; GAN: Gene Accession Number. All the genes in this list have some plausible relevance to the AML vs. ALL separation.

(5 in this case). All four genes have some relevance to leukemia and deserve a more detailed analysis to understand their exact role in discriminating between AML and ALL variants.

In this last experiment, we also noted that the smallest number of genes that separate the whole data set without error is two (Zyxin and MacMarcks). For this set of genes, there is also zero leave-one-out error. In contrast, the baseline method (Golub, 1999) always yields at least one training error and one leave-one-out error. One training error can be achieved with a minimum of 16 genes and one leave-one-out error with a minimum of 64 genes.

6. Other explorations and discussion

6.1. Computational considerations

The fastest methods of feature selection are correlation methods: for the data sets under study, several thousands of genes can be ranked in about one second by the baseline method (Golub, 1999) with a Pentium processor.

The second fastest methods use as ranking criterion the weights of a classifier trained only once with all the features. Training algorithms such as SVMs or Pseudo-inverse/MSE require first the computation of the (ℓ, ℓ) matrix H of all the scalar products between the ℓ training patterns. The computation of H increases linearly with the number of features (genes) and quadratically with the number of training patterns. After that, the training time is of the order of the time required to invert matrix H. For optimized SVM algorithms, training may be faster than inverting H, if the number of support vectors is small compared to ℓ. For the data sets under study, the solution is found in a couple of seconds on a Pentium processor, with non-optimized Matlab code.

Recursive Feature Elimination (RFE) requires training multiple classifiers on subsets of features of decreasing size. The training time scales linearly with the number of classifiers to be trained. Part of the calculations can be reused. Matrix H does not need to be re-computed entirely. The partial scalar products of the eliminated features can be subtracted. Also, the coefficients α can be initialized to their previous value. Our Matlab implementation of SVM RFE on a Pentium processor returns a gene ranking in about 15 minutes for the entire Colon dataset (2000 genes, 62 patients) and 3 hours on the Leukemia dataset (7129 genes, 72 patients). Given that the data collection and preparation may take several months or years, it is quite acceptable that the data analysis takes a few hours.

All our feature selection experiments using various classifiers (SVM, LDA, MSE) indicated that better features are obtained by using RFE than by using the weights of a single classifier (see Section 6.2 for details). Similarly, better results are obtained by eliminating one feature at a time than by eliminating chunks of features. However, there are only significant differences for the smaller subset of genes (less than 100). This suggests that, without trading accuracy for speed, one can use RFE by removing chunks of features in the first few iterations and then remove one feature at a time once the feature set reaches a few hundreds. This may become necessary if the number of genes increases to millions, as is expected to happen in the near future.

The scaling properties of alternative methods that have been applied to other "feature selection" problems are generally not as attractive. In a recent review paper (Blum, 1997), the authors mention that "few of the domains used to date have involved more than 40 features". The method proposed in Shürmann (1996), for example, would require the inversion of a (n, n) matrix, where n is the total number of features (genes).

6.2. *Analysis of the feature selection mechanism of SVM-RFE*

6.2.1. Usefulness of RFE. In this section, we question the usefulness of the computationally expensive Recursive Feature Elimination (RFE). In figure 5, we present the performance of classifiers trained on subsets of genes obtained either by "naively" ranking the genes with $(w_i)^2$, which is computationally equivalent to the first iteration of RFE, or by running RFE. RFE consistently outperforms the naïve ranking, particularly for small gene subsets.

The naïve ranking and RFE are qualitatively different. The naïve ranking orders features according to their individual relevance. The RFE ranking is a feature subset ranking. The nested feature subsets contain complementary features not necessarily individually most relevant. This is related to the relevance vs. usefulness distinction (Kohavi, 1997).

The distinction is most important in the case of correlated features. Imagine, for example, a classification problem with 5 features, but only 2 distinct features both equally useful: x_1, x_1, x_2, x_2, x_2. A naïve ranking may produce weight magnitudes $x_1(1/4), x_1(1/4), x_2(1/6), x_2(1/6), x_2(1/6)$, assuming that the ranking criterion gives equal weight magnitudes to identical features. If we select a subset of two features according to the naïve ranking, we eliminate the useful feature x_2 and incur possible classification performance degradation. In contrast, a typical run of RFE would produce:

first iteration $x_1(1/4), x_1(1/4), x_2(1/6), x_2(1/6), x_2(1/6)$,
second iteration $x_1(1/4), x_1(1/4), x_2(1/4), x_2(1/4)$

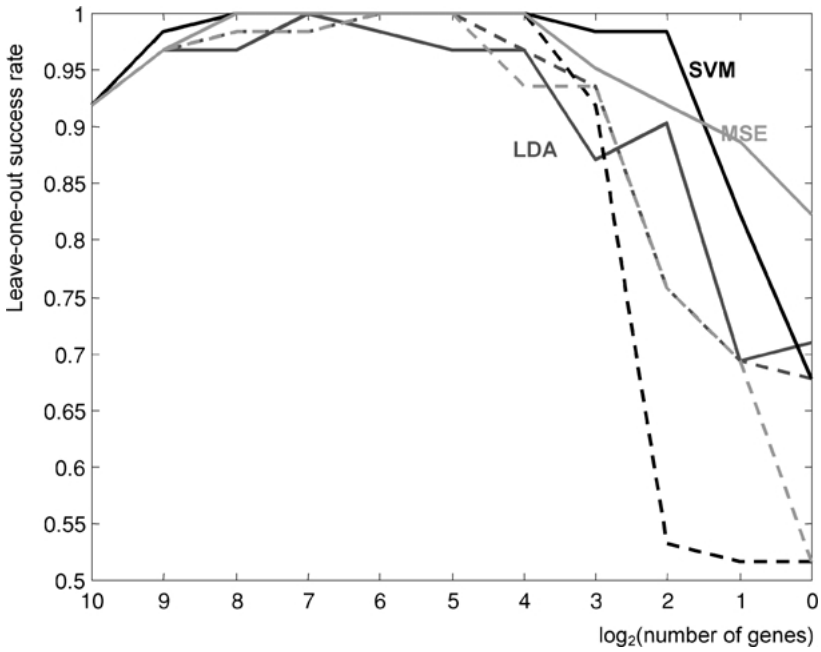

Figure 5. Effect of Recursive Feature Elimination (Colon cancer data). In this experiment, we compared the ranking obtained by RFE with the naïve ranking obtained by training a single classifier and using the magnitude of the weights as ranking coefficient. We varied the number of top ranked genes selected. Training was done on the entire data set of 62 samples. The curves represent the leave-one-out success rate for the various feature selection methods, using an SVM classifier. The colors represent the classifier used for feature selection. Top black: SVM. Darkgray: Linear Discriminant Analysis. Lightgray: Mean Squared Error (Pseudo-inverse). We do not represent the baseline method (Golub, 1999) since RFE and the naïve ranking are equivalent for that method. The solid line corresponds to RFE. The dashed line corresponds to the naïve ranking. RFE consistently outperforms the naïve ranking, for small gene subsets.

third iteration $x_2(1/2)$, $x_1(1/4)$, $x_1(1/4)$
fourth iteration $x_1(1/2)$, $x_2(1/2)$
fifth iteration $x_1(1)$

Therefore if we select two features according to RFE, we obtain both x_1 and x_2, as desired.

The RFE ranking is not unique. Our imagined run produced: $x_1\, x_2\, x_1\, x_2\, x_2$, corresponding to the sequence of eliminated genes read backwards. Several other sequences could have been obtained because of the symmetry of the problem, including $x_1\, x_2\, x_2\, x_1\, x_2$ and $x_2\, x_1\, x_2\, x_1\, x_2$. We observed in real experiments that a slight change in the feature set often results in a completely different RFE ordering. RFE alters feature ordering only for multivariate classification methods that do not make implicit feature orthogonality assumptions. The method of Golub (1999) yields the same ordering for the naïve ranking and RFE.

6.2.2. Feature selection mechanism of SVMs. In the experimental section (Section 5), we conjectured that SVMs have a feature selection mechanism relying on support vectors

that distinguishes them from "average case" methods. In this section, we illustrate on an example what such a mechanism could be.

In figure 6, we constructed a two-dimensional classification example. The dots were placed such that feature x_2 separates almost perfectly all examples with a small variance, with the exception of one outlier. Feature x_1 separates perfectly all examples but has a higher variance. We think of feature x_1 as the relevant feature (a cancer-related gene) and as feature x_2 as the irrelevant feature (a tissue composition related gene): most examples are very well separated according to tissue composition, but one valuable outlier contradicts this general trend. The baseline classifier (Golub, 1999) prefers feature x_2. But the SVM prefers feature x_1.

Therefore the SVM feature selection critically depends on having clean data since the outliers play an essential role. In our approach, the two problems of selection of useful patterns (support vectors) and selection of useful features are tightly connected. Other approaches consider the two problems independently (Blum, 1997).

6.2.3. Importance of preprocessing. The preprocessing used in Section 5 has a strong impact on SVM-RFE. From each feature, we subtract its mean and divide the result by its standard deviation. This ensures that feature scales are comparable. Such preprocessing is unnecessary if scaling is taken into account in the cost function or in the similarity measure.

6.3. Generalization of SVM RFE to the non-linear case and other kernel methods

The method of eliminating features on the basis of the smallest change in cost function described in Section 2.5 can be extended to the non-linear case and to all kernel methods in general (Weston, 2000(b)). One can make computations tractable by assuming no change in the value of the α's. Thus one avoids having to retrain a classifier for every candidate feature to be eliminated.

Specifically, in the case of SVMs (Boser, 1992; Vapnik, 1998; Cristianini, 1999), the cost function being minimized (under the constraints $0 \leq \alpha_k \leq C$ and $\sum_k \alpha_k \, y_k = 0$) is:

$$J = (1/2)\alpha^T H \alpha - \alpha^T \mathbf{1},$$

where H is the matrix with elements $y_h y_k K(\mathbf{x}_h, \mathbf{x}_k)$, K is a kernel function that measures the similarity between $\mathbf{x}_h$ and $\mathbf{x}_k$, and 1 is an ℓ dimensional vector of ones. An example of such a kernel function is $K(\mathbf{x}_h, \mathbf{x}_k) = \exp(-\gamma \|\mathbf{x}_h - \mathbf{x}_k\|^2)$.

To compute the change in cost function caused by removing input component i, one leaves the α's unchanged and one re-computes matrix H. This corresponds to computing $K(\mathbf{x}_h(-i), \mathbf{x}_k(-i))$, yielding matrix $H(-i)$, where the notation $(-i)$ means that component i has been removed. The resulting ranking coefficient is:

$$DJ(i) = (1/2)\alpha^T H \alpha - (1/2)\alpha^T H(-i)\alpha$$

The input corresponding to the smallest difference $DJ(i)$ shall be removed. The procedure can be iterated to carry out Recursive Feature Elimination (RFE).

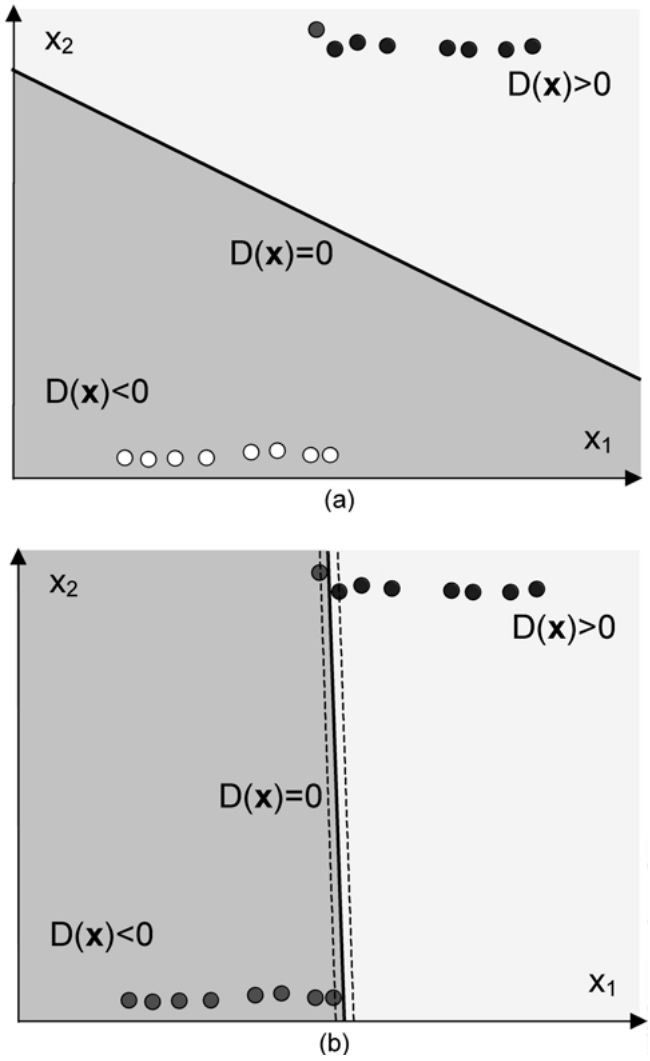

Figure 6. Feature selection and support vectors. This figure contrasts on a two dimensional classification example the feature selection strategy of "average case" type methods and that of SVMs. The white and black dots represent examples of class $(-)$ and $(+)$ respectively. The decision boundary $D(\mathbf{x}) = 0$ seperates the plane into two half planes $D(\mathbf{x}) < 0 \Rightarrow \mathbf{x}$ in class $(-)$, and $D(\mathbf{x}) > 0 \Rightarrow \mathbf{x}$ in class $(+)$. There is a simple geometric interpretation of the feature ranking criterion based on the magnitude of the weights: for slopes larger than 45 degrees, the preferred feature is x_1, otherwise it is x_2. The example was constructed to demonstrate the qualitative difference of the methods. Feature x_2 separates almost perfectly all examples with a small variance, with the exception of one outlier. Feature x_1 seperates perfectly all examples but has a higher variance. (a) Baseline classifier (Golub, 1999). The preferred feature is x_2. (b) SVM. The preferred feature is x_1.

Table 10. Non-linear SVM RFE.

Training set size	Top 2 features selected	No feature selection
30	0.1775 ± 0.1676	0.4691 ± 0.0233
40	0.1165 ± 0.1338	0.4654 ± 0.0236
100	0.0350 ± 0.0145	0.4432 ± 0.0259
1000	0.0036 ± 0.0024	0.3281 ± 0.0218

Fifty extra noisy dimensions were added to a 2 by 2 two checker board (XOR problem). All 52 dimensions were generated with uniform distribution on -0.5 to 0.5. The table shows averaged performance and standard deviation on a test set of 500 examples over 30 random trials using a polynomial classifier of degree 2.

In the linear case, $K(\mathbf{x}_h, \mathbf{x}_k) = \mathbf{x}_h \cdot \mathbf{x}_k$ and $\alpha^T H \alpha = \|\mathbf{w}\|^2$. Therefore $DJ(i) = (1/2)(wi)^2$. The method is identical the one we proposed and studied in the previous sections for linear SVMs.

Computationally, the non-linear kernel version of SVM RFE is a little more expensive than the linear version. However, the change in matrix H must be computed for support vectors only, which makes it affordable for small numbers of support vectors. Additionally, parts of the calculation such as the dot products $\mathbf{x}_h \cdot \mathbf{x}_k$ between support vectors can be cached.

We performed preliminary experiments on a classical example of non-linear classification problem, the XOR problem, which indicate that the method is promising. We added to a 2 by 2 two checker board 50 extra noisy dimensions, all 52 dimensions generated with uniform distribution on -0.5 to 0.5. We averaged performance over 30 random trials with a polynomial kernel of degree 2, $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^2$. Using our feature ranking method we selected the top two features. For 30 training examples the correct two features were selected 22/30 times. For 40 training examples the correct two features were selected 28/30 times. We also compared classification performance with and without feature selection (Table 10). The SVM performs considerably better with feature selection on this problem.

The idea of keeping the α's constant to compute DJ can be extended to the multi-class problem (Bredensteiner, 1999) other kernel methods such as KPCA (Schölkopf, 1998), non-classification problems such as regression, density estimation (see e.g. Vapnik, 1998) and clustering (Ben-Hur, 2000). It is not limited to RFE type of search. It can be used, for instance, for forward feature selection (progressively adding features), a strategy recently used, for example, in Smola (2000).

6.4. Other SVM feature selection methods

6.4.1. Penalty-based methods. We explored other methods of feature selection using SVMs. One idea is to formulate the optimization problem in a way that a large number of weights will be forced to zero. We tried the following linear programming formulation

of the problem:

$$\begin{cases} \text{Minimize over } w_i,\ w_i^* \text{and } \xi_k \\ \sum_i (w_i + w_i^*) + C \sum_k \xi_k \\ \text{subject to:} \\ y_k[(\mathbf{w}^* - \mathbf{w}) \cdot \mathbf{x}_k + b] \geq 1 - \xi_k \\ w_i > 0 \\ w_i^* > 0 \\ i = 1 \cdots n, k = 1 \cdots \ell \end{cases}$$

where C is a positive constant (penalty parameter).

Although the idea is quite attractive, we did not obtain in our experiment results that matched the performance of SVM RFE. Similar ideas have been proposed and studied by other authors (Bradley, 1998(a) and (b)). One drawback of penalty-based methods is that the number of features chosen is an indirect consequence of the value of the penalty parameter.

6.4.2. Feature scaling methods. The magnitude of the weights of a linear discriminant function is a scaling factor of the inputs. The idea of ranking features according to scaling factors subject to training therefore generalizes the scheme that we have been using. Non-linear discriminant functions such as neural networks and kernel methods can incorporate such scaling factors. Several authors (Mukherjee, 2000; Jebera, 2000; Chapelle, 2000; Weston, 2000(a)) have recently proposed and studied feature selection methods for SVMs that incorporate scaling factors into the kernel:

$$K_A(\mathbf{x}, \mathbf{y}) = K(A\mathbf{x}, A\mathbf{y})$$

where A is a diagonal matrix of scaling factors $a_1, a_2, \ldots a_n$. Training is performed by iterating:

1. Optimize the α's for a fixed A (regular SVM training).
2. Optimize A for fixed α's by gradient descent.

There are various flavors of the method depending on which cost function is being optimized in step 2 and which optimization method is used. The scaling factors are used to assess feature relevance. It is possible to set a threshold on feature relevance or select a given number of most relevant features.

In Mukherjee (2000), the authors report on the leukemia data (Golub, 1999) zero error with no rejects on the test set using the top 40 genes. They were able to classify 32 of 34 cases correctly using 5 genes. In Chapelle (2000), the authors achieve 1 error on the test set with 5 genes using the same data set. In Weston (2000(a)), the authors report on the Leukemia data zero error with 20 genes and 1 error with 5 genes. On the colon cancer data (Alon, 1999), the same authors report 12.8% average error of 50 splits of the data into 50 training examples and 12 test examples.

We note that the least relevant feature(s) could be eliminated and the process iterated as in RFE, but no results on this computationally expensive approach have been reported.

One drawback of feature scaling methods is that they rely on gradient descent. As such, they are sensitive to the choice of the gradient step, prone to falling in local minima and may be slow for a large number of features.

6.4.3. Wrapper methods and other search techniques. SVM RFE improves feature selection based on feature ranking by eliminating the orthogonality assumptions of correlation methods. Yet, it remains a greedy sub-optimal method. It generates nested subsets of features. This means that the selected subset of m features is included in the subset of $m + 1$ features. But, assume that we found a feature singleton that provides the best possible separation. There is no guarantee that the best feature pair will incorporate that singleton. Feature ranking methods miss that point.

Combinatorial search is a computationally intensive alternative to feature ranking. To seek an optimum subset of m features or less, all combinations of m features or less are tried. The combination that yields best classification performance (on a test set or by cross-validation) is selected. The classifier is used as a so-called "wrapper" in the feature selection process (Kohavi, 1997).

We tried to refine our optimum feature set by combinatorial search using SVMs in a wrapper approach. We started from a subset of genes selected with SVM RFE. We experimented with the leukemia data, using the training/test data split. We could easily find a pair of genes that had zero leave-one-out error and a very wide positive extremal margin. Yet, the error rate on the test set was very poor (13/34 errors).

The failure of these explorations and the success of RFE indicate that RFE has a built in regularization mechanism that we do not understand yet that prevents overfitting the training data in its selection of gene subsets. Other authors have made similar observations for other greedy algorithms (Kohavi, 1997). Placing constraints on the search space undoubtedly contributes to reduce the complexity of the learning problem and prevent overfitting, but a more precise theoretical formulation is still missing.

As a compromise between greedy methods and combinatorial search, other search methods could be used such as beam search or best first search (Kohavi, 1997).

7. Conclusions and future work

SVMs lend themselves particularly well to the analysis of broad patterns of gene expression from DNA micro-array data. They can easily deal with a large number of features (thousands of genes) and a small number of training patterns (dozens of patients). They integrate pattern selection and feature selection in a single consistent framework.

We proposed and applied the SVM method of Recursive Feature Elimination (RFE) to gene selection. We showed experimentally on two different cancer databases that taking into account mutual information between genes in the gene selection process impacts classification performance. We obtained significant improvements over the baseline method that makes implicit orthogonality assumptions. We also verified the biological relevance of the genes found by SVMs. The top ranked genes found by SVM all have a plausible relation

to cancer. In contrast, other methods select genes that are correlated with the separation at hand but not relevant to cancer diagnosis.

The RFE method was demonstrated for linear classifiers, including SVMs. This simple method allows us to find nested subsets of genes that lend themselves well to a model selection technique that finds an optimum number of genes. Our explorations indicate that RFE is much more robust to data overfitting than other methods, including combinatorial search.

Further work includes experimenting with the extension of the method to non-linear classifiers, to regression, to density estimation, to clustering, and to other kernel methods. We envision that linear classifiers are going to continue to play an important role in the analysis of DNA micro-array because of the large ratio number of features over number of training patterns.

Feature ranking methods do not dictate the optimum number of features to be selected. An auxiliary model selection criterion must be used for that purpose. The problem is particularly challenging because the leave-one-out error by itself is of little use since it is zero for a large number of gene subsets. Possible criteria that we have explored include the number of support vectors and a combination of the four metrics of classifier quality (error rate, rejection rate, extremal margin, and median margin) computed with the leave-one-out procedure. We have also explored with adding penalties for large numbers of features, using bounds on the expected error rate. Finding a good model selection criterion is an important avenue of experimental and theoretical research.

Greedy methods such as RFE are known by experimentalists to be less prone to overfitting than more exhaustive search techniques. A learning theoretic analysis of the regularization properties of SVM RFE remains to be done.

Finally, we have directed our attention to feature selection methods that optimize the feature subset for a given family of classifier (e.g. linear discriminant). More generally, the simultaneous choice of the learning machine and the feature subset should be addressed, an even more complex and challenging model selection problem.

Acknowledgments

The authors are grateful to the authors of Matlab code who made their source code available through the Internet. Our implementation grew from code written by Steve Gunn (http://www.isis.ecs.soton.ac.uk/resources/svminfo) Dick De Ridder and Malcolm Slaney http://valhalla.ph.tn.tudelft.nl/feature_extraction/source/svc/). We would also like to thank Trey Rossiter for technical assistance and useful suggestions, and the reviewers for their thorough work.

References

Aerts, H. (1996). Chitotriosidase—New biochemical marker. *Gauchers News*.

Alizadeh, A. et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, *403:3*, 503–511.

Alon, U. et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays. *PNAS*, *96*, 6745–6750, Cell Biology. The data is available on-line at http://www.molbio.princeton.edu/colondata.

Aronson, N. (1999). Remodeling the mammary GI and at the termination of breast feeding: Role of a new regulator protein BRP39. *The Beat*, University of South Alabama College of Medecine, July, 1999.

Ben Hur, A., Horn, D., Siegelman, H., & Vapnik, V. (2000). A support vector method for clustering. *Advances in Neural Information Processing Systems 13*, Cambridge, MA: MIT Press.

Blum, A. & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*, 245–271.

Boser, B., Guyon, I., & Vapnik, V. (1992). An training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (pp. 144–152). Pittsburgh: ACM.

Bradley, P. & Mangasarian, O. (1998). Feature selection via concave minimization and support vector machines. In *Proceedings of the 13th International Conference on Machine Learning* (pp. 82–90). San Francisco, CA.

Bradley, P., Mangasarian, O., & Street, W. (1998). Feature selection via mathematical programming. Technical Report. *INFORMS Journal on Computing*, *10*, 209–217.

Bredensteiner, E. & Bennett, K. (1999). Multicategory classification for support vector machines. *Computational Optimizations and Applications, 12*, 53–79.

Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., Jr., & Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS, 97:1*, 262–267.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2000). Choosing kernel parameters for support vector machines. AT&T Labs Technical Report.

Cortes, C. & Vapnik, V. (1995). Support vector networks. *Machine Learning, 20:3*, 273–297.

Cristianini, N. & Shawe-Taylor, J. (1999). *An introduction to support vector machines*. Cambridge, MA: Cambridge University Press.

Duda, R. O. & Hart, P. E. (1973). Pattern classification and scene analysis. New York: Wiley.

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *PNAS, 95*, 14863–14868.

Fodor, S. A. (1997). Massively parallel genomics. *Science, 277*, 393–395.

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

Ghigna, C., Moroni, M., Porta, C., Riva, I., & Biamonti, G. (1998). Altered expression of heterogeneous nuclear ribonucleoproteins and SR factors in human. *Cancer Research, 58*, 5818–5824.

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., & Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science, 286*, 531–537. The data is available on-line at http://www.genome.wi.mit. edu/MPR/data_set_ALL_AML.html.

Guyon, I. (1999). SVM Application Survey: http://www.clopinet.com/SVM.applications.html.

Guyon, I., Makhoul, J., Schwartz, R., & Vapnik, V. (1998). What size test set gives good error rate estimates? *PAMI, 20:1*, 52–64, IEEE.

Guyon, I., Matic, N., & Vapnik, V. (1996). Discovering informative patterns and data cleaning. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy, (Eds.). *Advances in knowledge discovery and data mining* (pp. 181–203). Cambridge, MA: MIT Press.

Guyon, I., Vapnik, V., Boser, B., Bottou, L., & Solla, S. A. (1992). Structural risk minimization for character recognition. In J. E. Moody et al. (Ed), *Advances in neural information processing systems 4 (NIPS 91)*, (pp. 471–479). San Mateo CA: Morgan Kaufmann.

Harlan, D. M., Graff, J. M., Stumpo, D. J., Eddy Jr, R. L., Shows, T. B., Boyle, J. M., & Blackshear, P. J. (1991). The human myristoylated alanine-rich C kinase substrate (MARCKS) gene (MACS). Analysis of its gene product, promoter, and chromosomal localization. *Journal of Biological Chemistry, 266:22*, 14399–14405.

Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Ross, D., Scherf, U., Weinstein, J., Alisadeh, A., Staudt, L., & Botstein, D. (2000). Gene shaving: A new class of clustering methods for expression arrays. Stanford Technical Report.

Jebara, T. & Jaakkola, T. (2000). Feature selection and dualities in maximum entropy discrimination. In *16th*

Conference on Uncertainty in Artificial Intelligence, UAI 2000, July 2000.

Karakiulakis, G., Papanikolaou, C., Jankovic, S. M., Aletras, A., Papakonstantinou, E., Vretou, E., & Mirtsou-Fidani, V. (1997). Increased type IV collagen-degrading activity in metastases originating from primary tumors of the human colon. *Invasion and Metastasis, 17:3*, 158–168.

Kearns, M., Mansour, Y., Ng, A. Y., & Ron, D. (1997). An experimental and theoretical comparison of model selection methods. *Machine Learning, 27*, 7–50.

Kohavi, R. & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97:12*, 273–324.

LeCun, Y., Denker, J. S., & Solla, S. A. (1990). Optimum brain damage. In D. Touretzky (Ed.). *Advances in neural information processing systems 2* (pp. 598–605). San Mateo, CA: Morgan Kaufmann.

Macalma, T., Otte, J., Hensler, M. E., Bockholt, S. M., Louis, H. A., Kalff-Suske, M., Grzeschik, K. H., von der Ahe, D., & Beckerle, M. C. (1996). Molecular characterization of human zyxin. *Journal of Biological Chemistry, 271:49*, 31470–31478.

Moser, T. L., Sharon Stack, M., Asplin, I., Enghild, J. J., Højrup, P., Everitt, L., Hubchak, S., William Schnaper, H., & Pizzo, S. V. (1999). Angiostatin binds ATP synthase on the surface of human endothelial cells. *PNAS, 96:6*, 2811–2816.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Messirov, J. P., & Poggio, T. (2000). Support vector machine classification of microarray data. AI memo 182. CBCL paper 182. MIT. Can be retrieved from ftp://publications.ai.mit.edu.

de Oliveira, E. C. (1999). Chronic Trypanosoma cruzi infection associated to colon cancer. An experimental study in rats. Resumo di Tese. *Revista da Sociedade Brasileira de Medicina Tropical, 32:1*, 81–82.

Osaka, M., Rowley, J. D., & Zeleznik-Le, N. J. (1999). MSF (MLL septin-like fusion), a fusion partner gene of MLL, in a therapy-related acute myeloid leukemia with at (11; 17)(q23; q25). *PNAS, 96:11*, 6428–6433.

Pavlidis, P., Weston, J., Cai, J., & Grundy, W. N. (2000). Gene functional analysis from heterogeneous data. Submitted for publication.

Perou, C. M. et al. (1999). Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *PNAS, 96*, 9212–9217.

Schölkopf, B., Smola, A., & Muller, K.-R. (1998). Non-linear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*, 1299–1319.

Shürmann, J. (1996). *Pattern classification*. Wiley Interscience.

Smola, A. & Schölkopf, B. (2000). Sparce greedy matrix approximation for machine learning. In *Proceedings of the 17th International Conference on Machine Learning* (pp. 911–918).

Thorsteinsdottir, U., Krosl, J., Kroon, E., Haman, A., Hoang, T., & Sauvageau, G. (1999). The oncoprotein E2A-Pbx1a collaborates with Hoxa9 to acutely transform primary bone marrow cells. *Molecular Cell Biology*, *19:9*, 6355–6366.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley Interscience.

Walsh, J. H. (1999). Epidemiologic evidence underscores role for folate as foiler of colon cancer. *Gastroenterology, 116*, 3–4.

Weston, J., Muckerjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for SVMs. In *Proceedings of NIPS 2000*, to appear.

Weston, J. & Guyon, I. (2000b). Feature selection for kernel machines using stationary weight approximation. In preparation.

Received March 28, 2000
Revised March 9, 2001
Accepted March 9, 2001
Final manuscript March 9, 2001