Analysis of Gene Expression Data **Spring Semester, 2007**

Lecture 7: June 7, 2007

Lecturer: Ron Shamir *Scribe: Hadas Zur and Roy Navon* [1]

7.1 Support Vector Machines

7.1.1 Introduction

The theory of support vector machines (**SVM**), has its origins in the late seventies, in the work of Vapnik [16] on the theory of statistical learning. Lately it has been receiving increasing attention, and many applications as well as important theoretical results are based on this theory. In fact, Support Vector Machines are arguably the most important discovery in the area of machine learning. The main area in which the methods described below had been used is in the field of pattern recognition. By way of motivation, we summarize some recent applications and extensions of support vector machines: For the pattern recognition case, SVMs have been used for [3]:

- Isolated handwritten digit recognition.
- Object recognition.
- Speaker identification.
- Face detection in images.
- Text categorization.

For the regression estimation case, SVMs have been compared on:

- Benchmark time series prediction tests.
- PET operator inversion problem.

The main idea of Support Vector Machines is to find a decision surface - a hyperplane in feature space (a line in the case of two features) - which separates the data into two classes. SVMs are extremely successful, robust, efficient, and versatile. In addition, there are good theoretical indications as to why they generalize well. In most of the cases SVM *generalization performance* either matches or is significantly better than that of competing methods. In the next section we will describe in detail the usage of SVM in the analysis of microarray gene expression data. This exposition is based mainly on [3, 4, 9].

[1] Based in part on a scribe by Simon Kamenkovich and Erez Greenstein May 2002 and on a scribe by Daniela Raijman and Igor Ulitsky Mars 2005

7.1.2 General motivation

One of the basic notions in the theory of SVM is the *capacity* of the machine, i.e. the ability of the machine to learn any training set of data without error. A machine with too much capacity is like a botanist with a photographic memory - who, when presented with a new tree, concludes that it is not a tree since it has a different number of leaves from anything she has seen before. On the other hand, a machine with too little capacity is like the botanist lazy brother - who declares that if it is green it is a tree. Neither can generalize well. Roughly speaking, for a given learning task with a given amount of training data, the best generalization performance will be achieved if the right balance is found, between the accuracy attained on the particular training set and the capacity of the machine.

The main idea of support vector machines is:

- Map the data to a predetermined very high-dimensional space via a kernel function.

- Find the hyperplane that maximizes the margin between the two classes - i.e. that separates the two classes.

- If data are not separable find the hyperplane that maximizes. the margin and minimizes the (weighted average of the) misclassifications - i.e. perform a soft separation allowing errors.

Three derivatives of this idea are:

- Define what an optimal hyperplane is (taking into account that it needs to be computed efficiently): maximize margin.

- Generalize to non-linearly separable problems: have a penalty term for misclassifications.

- Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space.

7.1.3 General mathematical setting

Suppose we are given l observations. Each observation consists of a pair - a vector $x_i \in \mathbf{R}^n$, $1 \leq i \leq l$, and an associated label y_i, given to us by a trusted source. In the tree recognition problem, the vector x_i might be some representation of an image, e.g., by its pixel values, and y_i would be $+1$ if the image contains a tree, and 0 otherwise. Furthermore, it is assumed that there exists some unknown probability distribution $P(x,y)$, from which the above observations are assumed to be independently drawn and identically distributed.

Now suppose we have a deterministic machine whose task is to learn the mapping $x_i \rightarrow y_i$. The machine is actually defined by a set of possible mappings $x_i \rightarrow f(x,\alpha)$, where the function f depends on a parameter α, i.e, a choice of the parameter α specifies one particular machine from the set $f(x,\alpha)$. For a particular choice of the parameter, the machine will be called a *trained machine.*

7.1.4 The VC dimension

The VC dimension is a property of a set of functions $\{f(\alpha)\}$. It can be defined in a more general manner, but we will assume families of functions that obtain binary values.

If a given set of l points can be labeled in all possible 2^l ways, and for each labeling, a member of the set $\{f(\alpha)\}$ can be found which correctly assigns those labels, we say that the set of points is **shattered** by the set of functions. The VC dimension for the set of functions $\{f(\alpha)\}$ is defined as the maximum size of a set of points that can be shattered by $\{f(\alpha)\}$. In other words, if the VC dimension is h there exists at least one set of h points that can be shattered.

Example: Shattering points with oriented lines in $\mathbf{R}^2$.
Suppose that the data are points in $\mathbf{R^2}$, and the set $\{f(\alpha)\}$ consists of oriented straight lines, such that for a given line all points on one side are assigned the value 1, and all points on the other are assigned the value 0.
It is possible to find a set of three points that can be shattered by the oriented lines (see 7.1), but it is not possible to shatter a set of 4 points with the set of oriented lines. Thus, the VC dimension of the set of oriented lines in $\mathbf{R}^2$ is 3. In the more general case the VC dimension of a set of oriented hyperplanes in $\mathbf{R}^n$ is $n + 1$.

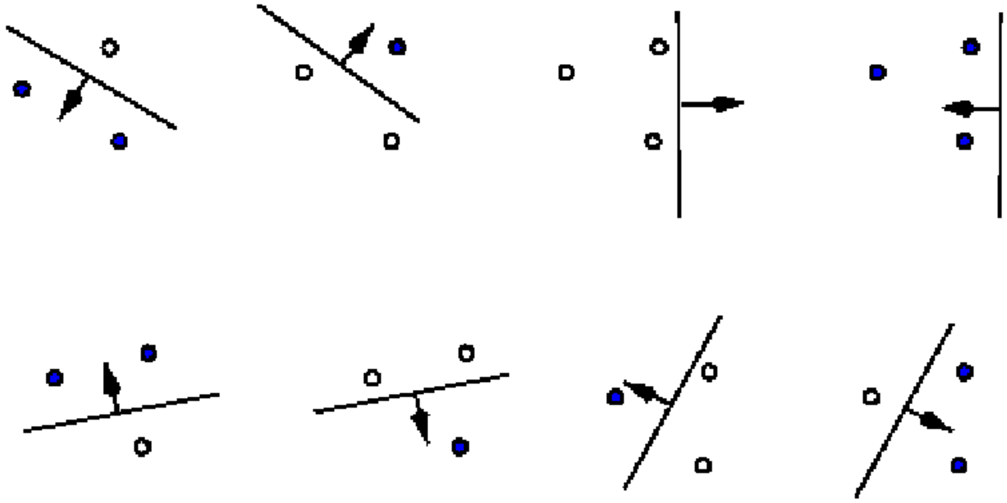


Figure 7.1: Three points in the plane shattered by oriented lines.

7.1.5 Support Vector Machines

We will start with the simplest case, linear machines trained on separable data. Given a training set $x_i, y_i, x_i \in \mathbf{R}^n, y_i \in \{-1, 1\}$, we assume that the data is linearly separable, i.e., there exists a separating hyperplane which separates the positive examples ($y_i = 1$) from the negative ones ($y_i = -1$). The points x which lie on the hyperplane satisfy $w \cdot x + b = 0$, where w is a normal to the hyperplane and $\frac{|b|}{\|w\|}$ is the perpendicular distance from the hyperplane to the origin. Let $d_+(d_-)$ be the shortest distance from the separating hyperplane to the closest positive (negative) example. Define the *margin* of a separating hyperplane to be $d_+ + d_-$. For the linearly separable case, the support vector algorithm simply looks for the separating hyperplane with largest margin.

Thus the goal is to find the optimal linear classifier (a hyperplane), such that it classifies every training example correctly, and maximizes the classification margin. The above description can be formulated in the following way: Suppose that all the training data satisfy the following constraints:

$$x_i \cdot w + b \geq +1, y_i = +1 \tag{7.1}$$

$$x_i \cdot w + b \leq -1, y_i = -1 \tag{7.2}$$

Now consider the points for which the equality in (7.1) holds. These points lie on the hyperplane $H_1 : x_i \cdot w + b = 1$. Similarly, the points for which the equality in (7.2) holds lie on the hyperplane $H_2 : x_i \cdot w + b = -1$. In this case $d_+ = d_- = \frac{1}{\|w\|}$, therefore the margin is $\frac{2}{\|w\|}$. Note that H_1 and H_2 are parallel (they have the same normal) and that no training points fall between them. Thus we can find the pair of hyperplanes which gives the maximum margin by minimizing $\|w\|^2$, subject to the above constraints, because keeping this norm small will also keep the VC-dimension small.

$$\textit{minimize} \quad \frac{1}{2}\|w\|^2 \tag{7.3}$$

$$s.t. \quad y_i(w \cdot x_i + b) \geq 1, i = 1 \ldots l \tag{7.4}$$

or

$$\textit{maximize} \quad \frac{2}{\|w\|} \tag{7.5}$$

$$s.t. \quad y_i(w \cdot x_i + b) \geq 1, i = 1 \ldots l \tag{7.6}$$

Linear, Hard-Margin SVM Formulation

Find w, b that solve equation (7.3) under comstraint (7.4)

The above minimization problem is convex, therefore there exists a unique global minimum value (when feasible), and there is a unique minimizer, i.e. weight and b value that provides the minimum (given that the data is indeed linearly separable). There is no solution if the data are not linearly separable. At this point, since the problem is convex, it can be solved using standard Quadratic Programming (QP) optimization techniques which is not very complex since the dimensionality is $N + 1$. Since N is the dimension of the input space, this problem is more or less tractable for real applications. Nevertheless, in order to easily explain the extension to nonlinear decision surfaces (which will be described in section 7.2.8), we look at the dual problem, and use the technique of Lagrange Multipliers.

Lagrange formulation of the problem

We will now switch to a Lagrangian formulation of the problem. There are two reasons for doing this. The first is that the constraints (7.1),(7.2) will be replaced by constraints on the Lagrange multipliers themselves, which will be much easier to handle. The second is that in this reformulation of the problem, the training data will only appear (in the actual training and test algorithms) in the form of dot products between vectors. This is a crucial property which will allow us to generalize the procedure to the nonlinear case.

We introduce positive Lagrange multipliers $\alpha_i, i = 1, \ldots, l$ one for each of the inequality constraints . We form the Lagrangian:

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{l} \alpha_i$$

We have to minimize L_P with respect to w, b and simultaneously require that the derivatives of L_P with respect to all the α_i vanish. This is a convex quadratic problem, which has a dual formulation - maximize L_P, subject to the constraint that the gradients of L_P with respect to w and b vanish, and subject to the constraints that $\alpha_i \geq 0$. This gives:

$$maximize \qquad L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{7.7}$$

$$s.t. \qquad \sum_{i=1}^{l} \alpha_i y_i = 0, \alpha_i \geq 0 \tag{7.8}$$

Support vector training (for the separable, linear case) therefore amounts to maximizing L_D with respect to the α_i, subject to above constraints and non-negativity of the α_i, with solution given by $w = \sum_{i=1}^{l} \alpha_i y_i x_i$. Notice that there is a Lagrange multiplier α_i for every training point. In the solution, those points for which $\alpha_i > 0$ are called *support vectors*, and lie on one of the hyperplanes H_1 or H_2. All other training points have $\alpha_i = 0$ and lie to the side of H_1 or H_2 such that strict inequality holds. For these machines, the support vectors are the critical elements of the training set. They lie closest to the decision boundary. If all other training points were removed (or moved around, but so as not to cross H_1 or H_2), and training was repeated, the same separating hyperplane would be found. This remains a convex quadratic equation, and therefore quadratic programming still applies.

7.1.6 The soft margin hyperplane: Linear non separable case

When applied to non-separable data, the above algorithm will find no feasible solution. This will be evidenced by the objective function (i.e. the dual Lagrangian) growing arbitrarily

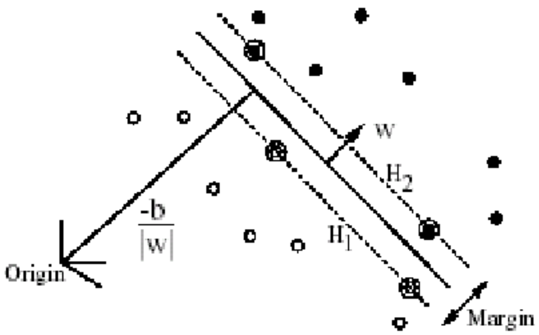


Figure 7.2: Linear separating hyperplanes for the separable case. The support vectors are circled

large. So how can we extend these ideas to handle non-separable data? We would like to relax the constraints:

$$x_i \cdot w + b \geq +1, y_i = +1 \tag{7.9}$$

$$x_i \cdot w + b \leq -1, y_i = -1 \tag{7.10}$$

but only when necessary, that is, we would like to introduce a further cost (i.e. an increase in the primal objective function) for doing so. This can be done by introducing positive slack variables $\xi_i, i = 1, \ldots, l$ in the constraints (Cortes and Vapnik, 1995), which then become:

$$x_i \cdot w + b \geq +1 - \xi_i, y_i = +1 \tag{7.11}$$

$$x_i \cdot w + b \leq -1 + \xi_i, y_i = -1 \tag{7.12}$$

Thus, for an error to occur, the corresponding ξ_i must exceed unity, so $\sum_i \xi_i$ is an upper bound on the number of training errors. Hence a natural way to assign an extra cost for errors is to change the objective function to be minimized from $\frac{1}{2}\|w\|^2$ to $\frac{1}{2}\|w\|^2 + C\sum_i^l \xi_i$, where C is a parameter to be chosen by the user. A larger C corresponds to assigning a higher penalty to errors. This is again a convex quadratic programming problem. Thus the dual formulation becomes:

$$maximize \qquad L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{7.13}$$

$$s.t \qquad 0 \leq \alpha_i \leq C, \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{7.14}$$

The only difference from the optimal hyperplane case is that the α_i have an upper bound of C, which is the penalty for misclassification. The parameter C controls the range of the α_i and avoids over emphasizing some examples. C is called the complementary slackness when C tends to infinity we return to the separable case. The parameter C controls the range of λ_i, i.e.

avoids over emphasizing some examples. ξ_i(C - λ_i) = 0, indicates complementary "slacknes". The parameter C can be extended to be case dependant. The weight $\lambda_i : \lambda_i < C \rightarrow \xi_i = 0$, meaning the i-th example is correctly classified is not quite important. $\lambda_i < C \rightarrow \xi_i$ can be non-zero, i.e. the i-th training example may be misclassified, which is very important.

This algorithm tries to keep ξ_i at zero while maximizing the margin. It does not minimize the number of misclassifications (which is an NP-complete problem) but the sum of distances from the margin hyperplanes. There are some formulations which use ξ_i^2 instead.

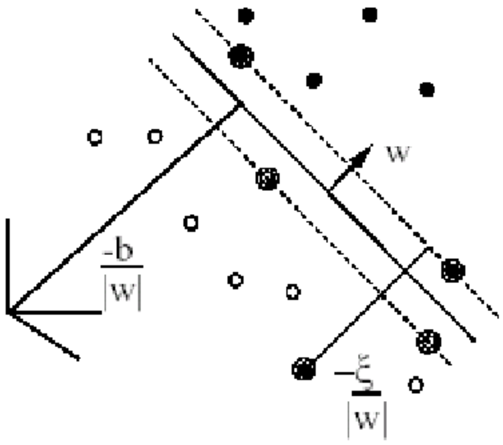


Figure 7.3: Linear Separating hyperplanes for the non separable case.

7.1.7 Soft vs hard margin SVM

Even when the data can be linearly separated, we might benefit from using a soft margin, allowing us to get a much wider margin at a small cost (See figure 7.4). Using a Soft-margin we can always obtain a solution, since the method is more robust to outliers (smoother surfaces in the non-linear case). However, it requires us to guess the cost parameter, as opposed to the hard-margin method, which does not require any parameters.

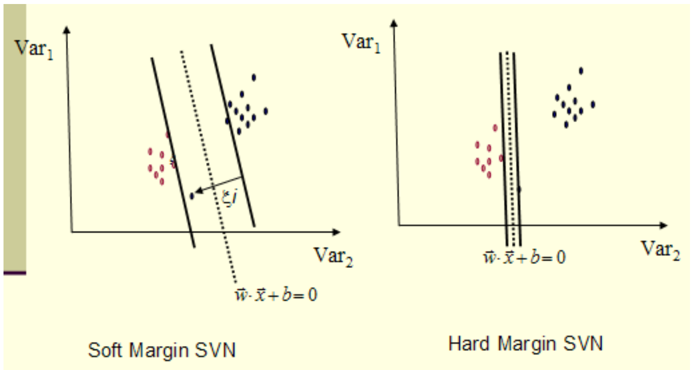


Figure 7.4: Robustness of Soft vs Hard Margin SVMs.

7.1.8 Disadvantages of linear decision surfaces

Linear decision surfaces may sometimes be unable to seperate the requested data.

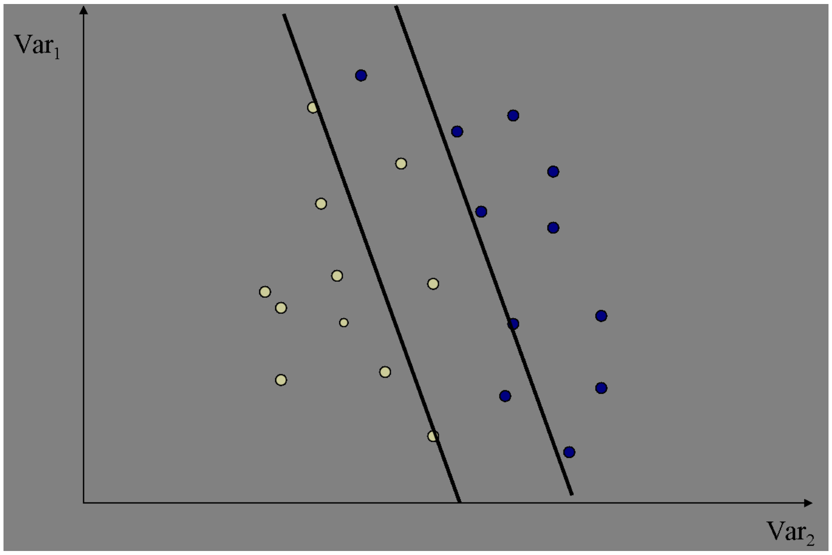


Figure 7.5: Disadvantages of linear decision surfaces - the two presented classes can not be seperated by a linear decision surface.

7.1.9 Advantages of Non-linear decision surfaces

Non-linear decision durfaces are more robust.

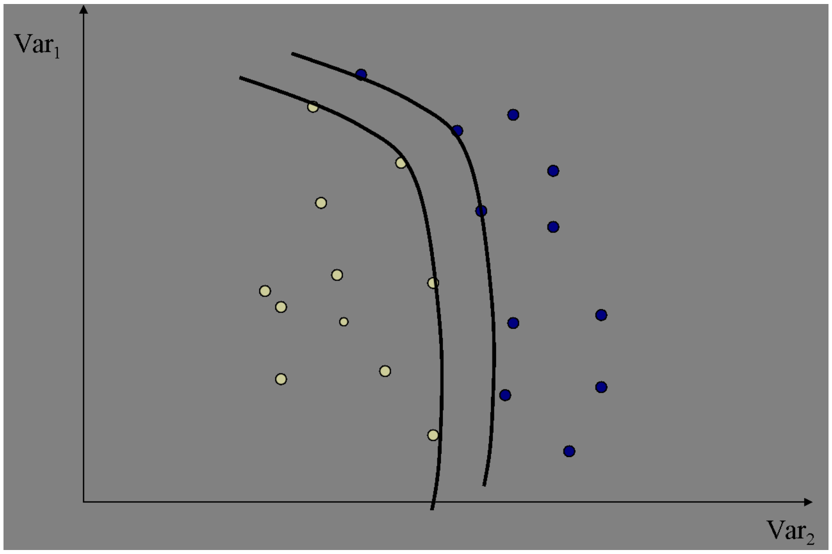
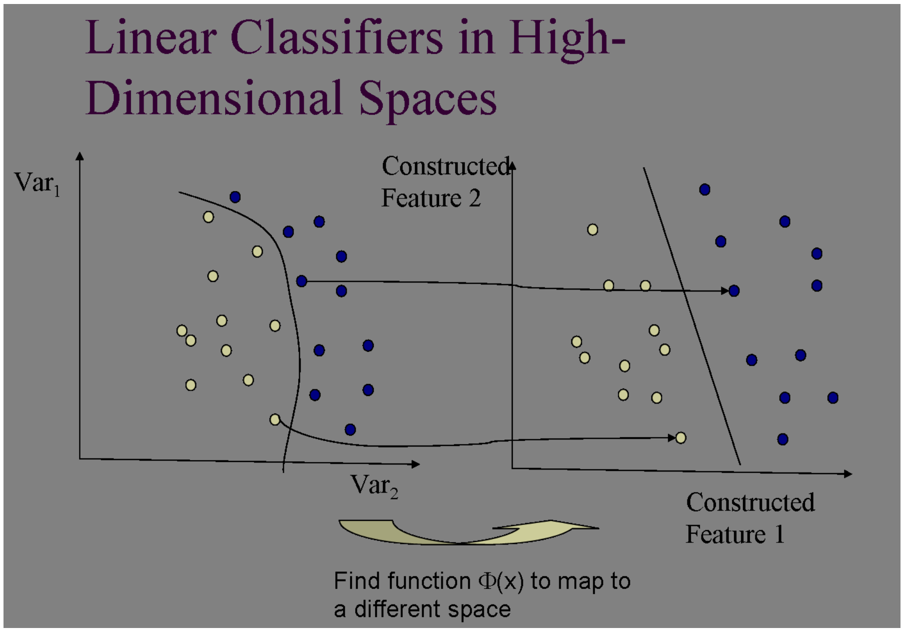


Figure 7.6: Advantages of Non-linear decision surfaces - seperation of non-linearly seperable classes is enabled, due to the curved nature of non-linear hyperplanes.

7.1.10 Non linear case

In some cases the data requires a more complex, non-linear separation. When coming to generalize the above ideas to the non-linear case - the idea is to use the same techniques as above for linear machines . Since finding a linear machine is not possible in the original space of the training set, we first map the training set to an Euclidean space with a higher dimension (even of infinite dimension), this higher-dimensional space is called the *feature space*, as opposed to the *input space* occupied by the training set. With an appropriately chosen feature space of sufficient dimensionality, any consistent training set can be made separable. However, translating the training set into a higher-dimensional space incurs both computational and learning-theoretic costs. (see Figure 7.7)



Linear Classifiers in High-Dimensional Spaces

Find function Φ(x) to map to a different space

Figure 7.7: **Linear Classifiers in High-Dimensional Spaces.**

Suppose we first map the data to some other space H, using a mapping $\Phi : R^d \to H$. Then the SVM formulation becomes:

$$minimize\ \frac{1}{2}\|w\|^2 + C\sum_i^l \xi_i \tag{7.15}$$

$$\text{s.t. } y_i(w \cdot \Phi(x_i) + b) \geq 1 - \xi_i, \forall x_i,\ \xi_i \geq 0 \tag{7.16}$$

Data now appear as $\Phi(x_i)$. Weights are also mapped to the new space. However, if $\Phi(x_i)$ is very high dimensional, explicit mapping is very expensive. Therefore, we would like to solve the problem without explicitly mapping the data. The key idea is to notice that in the dual representation of the above problems - the training data appeared only in the form of dot products. Now suppose we first map the data to some other space H, using a mapping $\Phi : R^d \to H$. Then the training algorithm would only depend on the data through dot products in H, i.e. on functions of the form $\Phi(x_i) \cdot \Phi(x_j)$.

7.1.11 The Kernel Trick

All we need in order to perform training in H is a function that satisfies $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$, i.e., the image of the inner product of the data is the inner product of the images of the data. This type of function is called a *kernel function.* The kernel function is used in the higher dimension space as a dot product, so we do not need to explicitly map the data into the high-dimensional space. Classification can else be done without explicitly mapping the new instances to the higher dimension, as we take advantage of the fact that $sgn(wx + b) = sgn(\sum_i \alpha_i y_i K(x_i, x) + b)$ where b solves $\alpha_j(y_j \sum_i \alpha_i y_i K(x_i, x_j) + b - 1) = 0$ for any j with $\alpha_j \neq 0$.

Examples of kernel functions are:

- $K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$ - radial basis kernel.
- $K(x_i, x_j) = e^{\frac{x_i - x_j}{\sigma^2}}$ - gaussian kernel.
- $K(x_i, x_j) = (x_i \cdot x_j + 1)^k$ - polynomial kernel.

In general, using the kernel trick provides huge computational savings over explicit mapping.

7.1.12 The Mercer Condition [3, 4]

For which kernels does there exist a pair H,Φ, with the properties described above, and for which does there not? The answer is given by Mercer's condition: There exists a mapping Φ and an expansion:

$$K(x, y) = \sum i\Phi(x_i)\Phi(y_i) \tag{7.17}$$

if and only if, for any $g(x)$ such that

$$\int g(x)^2 dx \tag{7.18}$$

is finite then

$$\int K(x,y)g(x)g(y)dxdy \geq 0 \qquad (7.19)$$

Note that for specific cases, it may not be easy to check whether Mercer's condition is satisfied. Equatio (7.19) must hold for every g with finite L2 norm (i.e. which satisfies equation (7.18)). However, we can easily prove that the condition is satisfied for positive integral powers of the dot product: $K(x;y) = (x \cdot y)^p$. We must show that

$$\int (\sum x_{i=1}^{d} x_i y_i)^p g(x)g(y)dxdy \geq 0 \qquad (7.20)$$

The typical term in the multinomial expansion of $(\sum x_{i=1}^{d} x_i y_i)^p$ contributes a term of the form

$$\frac{p!}{r_1!_2! \cdots (p - r_i - r_2 \cdots)!} \int x_1^{r_1} x_2^{r_2} \cdots y_1^{r_1} y_2^{r_2} \cdots g(x)g(y)dxdy \qquad (7.21)$$

to the left hand side of (7.19), which factorizes:

$$\frac{p!}{r_1!_2! \cdots (p - r_i - r_2 \cdots)!} (\int x_1^{r_1} x_2^{r_2} \cdots g(x)dx)^2 \geq 0 \qquad (7.22)$$

One simple consequence is that any kernel which can be expressed as $K(x,y) = \sum p = 0^{\infty} c_p (x \cdot y)^p$ where the c_p are positive real coefficients and the series is uniformly convergent, satisfies Mercer's condition, a fact also noted in.

A number of observations are in order:

- Vapnik (1995) uses the condition (2) above to characterize the kernels that can be used in SVM.
- There is another result similar to Mercer's one, but more general: the kernel is positive definite if and only if:

$$\int_{\Omega} dxdyK(x,y)g(x)g(y) \geq 0 \qquad \forall g \in L_1(\Omega) \qquad (7.23)$$

- The kernels K that can be used to represent a scalar product in the feature space are closely related to the theory of Reproducing Kernel Hilbert Spaces (RKHS).

7.1.13 Other Types of Kernel Methods

- SVMs that perform regression
- SVMs that perform clustering

- v-Support Vector Machines: maximize margin while bounding the number of margin errors
- Leave One Out Machines: minimize the bound of the leave-one-out error
- SVM formulations that take into consideration difference in cost of misclassification for the different classes
- Kernels suitable for sequences of strings or other specialized kernels.

7.1.14 Feature Selection with SVMs

Recursive Feature Elimination:

- Train a linear SVM
- Remove the x% of variables with the lowest weights (those variables affect classification the least) Retrain the SVM with remaining variables and repeat until classification is reduced.

  This method is very successful. Other formulations exist where minimizing the number of variavles is folded into the optimization problem. The algorithms for non-linear SVMs are similar, and are quite successful.

7.1.15 Multi Class SVMs

In its basic form an SVM is able to classify test examples into only two classes: positive and negative. We say that SVM is a binary classifier. This means that training examples must also be only of the two kind: positive and negative. Even though an SVM is binary, we can combine several such classifiers to form a multi-class variant of an SVM.

- One-versus-all: Train n binary classifiers, one for each class against all other classes. Predicted class is the class of the most confident classifier.
- One-versus-one: Train $n(n-1)/2$ classifiers each discriminating between a pair of classes. Several strategies for selecting the final classification based on the output of the binary SVMs.
- Truly Multi Class SVMs: Generalize the SVM formulation to multiple categories.

7.1.16 Training SVM , Problems and Heuristics

There are several heuristics for solving the problem of training an SVM. As we have seen above the training of an SVM is a quadratic optimization problem. Zoutendijk's Method [7] aims to solve a linear programming problem: Find the direction that minimizes the objective function, and make the largest move along this direction ,while still satisfying all the constraints. Another problem in the training of SVM is that in the general case the computation of the kernel function $K(x_i, x_j)$ for each pair of elements might be computationally expensive. The solution is to use only part of the training data, using the assumption that only part of the data contributes to the decision boundary. Then we define a strategy to increase the objective function, while updating the set of data points contributing to the formation of the decision boundary.

7.1.17 Heuristic for training SVM with large data set

1. Divide the training examples into two sets A, B.
2. Use the set A of training examples to find out the optimal decision boundary.
3. Find an example x_i in A with no contribution to the decision boundary, $\alpha_i = 0$.
4. Find another example, x_m in B that can not be classified correctly by the current decision boundary.
5. Remove x_i from A and add x_m to A.
6. Repeat steps 2-5 till some stopping criterion is satisfied.

7.2 Knowledge-based analysis of microarray gene expression data

7.2.1 Introduction

After laying the theoretic foundations for SVM, we explore its applications in the analysis of microarray gene expression data. The work described here is due to [8]. In this work the authors applied an SVM for functional annotation of genes. The idea is to begin with a set of genes that have a common function: for example, genes coding for ribosomal proteins or genes coding for components of the proteasome. In addition, a separate set of genes that are known not to be members of the functional class is specified. These two sets of genes are combined to form a set of training examples in which the genes are labelled positively if they are in the functional class and are labelled negatively if they are known not to be in the functional class. A set of training examples can easily be assembled from literature and database sources. Using this training set, an SVM would learn to discriminate between the members and non-members of a given functional class based on expression data. Having learned the expression features of the class, the SVM could recognize new genes as members or as non-members of the class based on their expression data.

We describe here the use of SVM to classify genes based on gene expression. Analyzing expression data from 2,467 genes from the budding yeast *Saccharomyces cerevisiae* measured in 79 different DNA microarray hybridization experiments [2]. From these data, the authors learn to recognize five functional classes from the Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) (http://www.mips.biochem.mpg.de/proj/yeast).

In addition to SVM classification, the authors subject the data to analyses by four competing machine learning techniques, including Fisher's linear discriminant, Parzen windows, and two decision tree learners. The SVM method out-performed all other methods.

The work described here experimented with several kernel functions:
$K(X,Y) = (X \cdot Y + 1)^d, d = 1,2,3$
and:
$K(X,Y) = e^{-\|X-Y\|^2/2\sigma^2}$

7.2.2 Balancing positive and negative examples

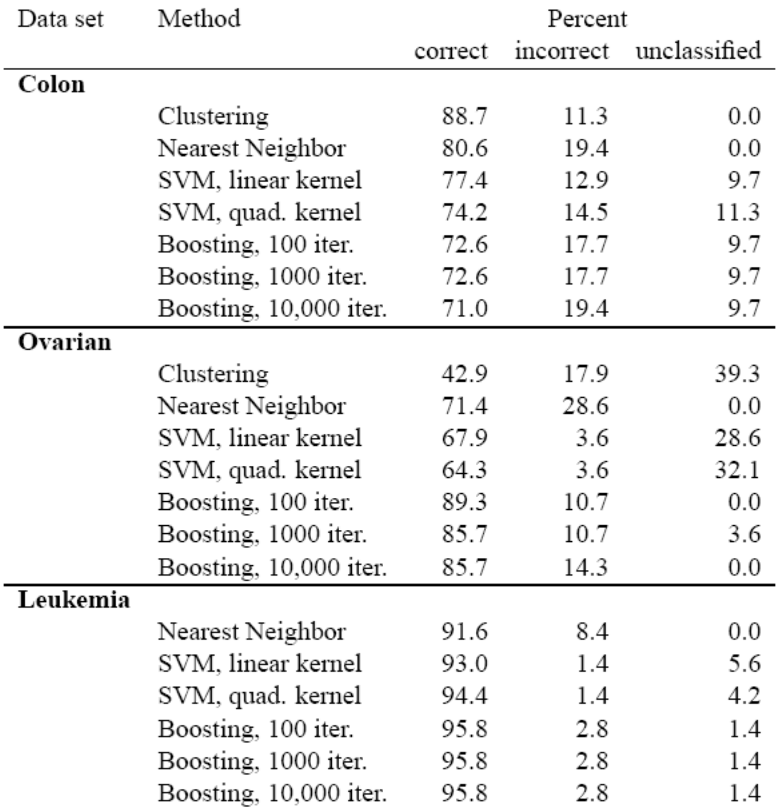
The gene functional classes examined here contain very few members relative to the total number of genes in the data set. This leads to an imbalance in the number of positive and negative training examples that, in combination with noise in the data, is likely to cause the SVM to make incorrect classifications. When the magnitude of the noise in the negative examples outweighs the total number of positive examples, the optimal hyperplane located by the SVM will be uninformative, classifying all members of the training set as negative examples. The authors overcame this problem by modifying the matrix of kernel values computed during SVM optimization. Let $X_1, \ldots, X_n$ be the genes in the training set, and let $\mathbf{K}$ be the matrix defined by the kernel function K on this training set i.e., $K_{ij} = K(X_i, X_j)$. X_i is the logarithm of the ratio of expression level E_i for gene $\mathbf{X}$ in experiment i to the expression level R_i of gene $\mathbf{X}$ in the reference state, normalized so that the expression vector $X = (X_1, \ldots, X_{79})$ has Euclidean length 1:

$$X_i = \frac{\log(E_i/R_i)}{\sqrt{\sum_{j=1}^{79} \log^2 E_j/R_j}}$$

By adding to the diagonal of the kernel matrix a constant whose magnitude depends on the class of the training example, one can control the fraction of misclassified points in the two classes. This technique ensures that the positive points are not regarded as noisy labels. For positive examples, the diagonal element is modified by $K_{ij} := K_{ij} + \lambda n^+/N$, where n^+ is the number of positive training examples, N is the total number of training examples, and λ is scale factor. A similar formula is used for the negative examples, with n^+ replaced by n^-. In the experiments reported here, the scale factor is set to 0.1.

7.3 Tissue Classification with Gene Expression Profiles [10]

Constantly improving gene expression profiling technologies are expected to provide understanding and insight into cancer related cellular processes. Gene expression data is also expected to significantly aid in the development of efficient cancer diagnosis and classification platforms. In this work the authors examine three sets of gene expression data measured across sets of tumor(s) and normal clinical samples: The first set consists of 2,000 genes, measured in 62 epithelial colon samples. The second consists of $\approx$ 100,000 clones, measured in 32 ovarian samples. The third set consists of $\approx$ 7,100 genes, measured in 72 bone marrow and peripheral blood samples (Golub99). They examine the use of scoring methods, measuring separation of tissue type (e.g., tumors from normals) using individual gene expression levels. These are then coupled with high dimensional classification methods to assess the classification power of complete expression profiles. They present results of performing leave-one-out cross validation (LOOCV) experiments on the three data sets, employing nearest neighbor classifier, SVM, AdaBoost and a novel clustering based classification technique.



Data set	Method	Percent		
		correct	incorrect	unclassified
Colon				
	Clustering	88.7	11.3	0.0
	Nearest Neighbor	80.6	19.4	0.0
	SVM, linear kernel	77.4	12.9	9.7
	SVM, quad. kernel	74.2	14.5	11.3
	Boosting, 100 iter.	72.6	17.7	9.7
	Boosting, 1000 iter.	72.6	17.7	9.7
	Boosting, 10,000 iter.	71.0	19.4	9.7
Ovarian				
	Clustering	42.9	17.9	39.3
	Nearest Neighbor	71.4	28.6	0.0
	SVM, linear kernel	67.9	3.6	28.6
	SVM, quad. kernel	64.3	3.6	32.1
	Boosting, 100 iter.	89.3	10.7	0.0
	Boosting, 1000 iter.	85.7	10.7	3.6
	Boosting, 10,000 iter.	85.7	14.3	0.0
Leukemia				
	Nearest Neighbor	91.6	8.4	0.0
	SVM, linear kernel	93.0	1.4	5.6
	SVM, quad. kernel	94.4	1.4	4.2
	Boosting, 100 iter.	95.8	2.8	1.4
	Boosting, 1000 iter.	95.8	2.8	1.4
	Boosting, 10,000 iter.	95.8	2.8	1.4

Figure 7.8: **Summary of classification performance of the different methods on the three data sets.**
The tables shows the precent of samples that were correctly classified, incorrectly classfied, and unclassfied by each method in the LOOCV evaluation. Unsupervised labels for margin based classifier were decided by a fixed threshold on classification margin: in SVM, 0.25, and in Adaboost, 0.05.

7.4 Molecular Classification of Cancer

7.4.1 Overview

Although cancer classification has improved over the past 30 years, there has been no general approach for identifying new cancer classes (class discovery) or for assigning tumors to known classes (class prediction). In [6] and [13] a generic approach to cancer classification based on gene expression monitoring by DNA microarrays is described and applied to human acute leukemias as a test case. A class discovery procedure automatically discovered the distinction between acute myeloid leukemia (AML) [2] and acute lymphoblastic leukemia (ALL) [3] without previous knowledge of these classes. An automatically derived class predictor was able to determine the class of new leukemia cases. The results demonstrate the feasibility of cancer classification based solely on gene expression monitoring and suggest a general strategy for discovering and predicting cancer classes for other types of cancers, independent of previous biological knowledge.

7.4.2 Introduction

Determination of cancer type helps assigning an appropriate treatment to a patient. Cancer classification is based primarily on location or on morphological appearance of the tumor, that requests experienced biologist to distinguish known forms. The morphology classification has the limitation, that tumors with similar histopathological [4] appearance can have significantly different clinical courses and response to therapy. In general, cancer classification has been difficult because it has historically relied on specific biological insights, rather than systematic and unbiased approaches for recognizing tumor subtypes. In [6] and [13] an approach based on global gene expression analysis is described.

The authors study the following three challenges:

1. Feature selection:
   Identifying the most informative genes for prediction. Clearly most genes are not relevant to cancer so we want to choose only the best features (criteria) for class prediction.

2. Class prediction or Classification:
   Assignment of particular tumor samples to already defined classes, which could reflect current states or future outcomes.

3. Class discovery or Clustering:
   Finding previously unrecognized tumor subtypes.

[2] AML affects various white blood cells including granulocytes, monocytes and platelets. Leukemic cells accumulate in the bone marrow, replace normal blood cells and spread to the liver, spleen, lymph nodes, central nervous system, kidneys and gonads.

[3] ALL is a cancer of immature lymphocytes, called lymphoblasts (sometimes referred to as blast cells). Normally, white blood cells repair and reproduce themselves in an orderly and controlled manner but in leukaemia the process gets out of control and the cells continue to divide, but do not mature.

[4] Histopathology is the science that studies pathologic tissues.

Thus, difference between classification and clustering is that clustering is unsupervised - we do not know anything about division, whereas classification is a supervised learning process, where division to subtypes is already known.

The authors studied the problem of classifying acute leukemia. Classification of acute leukemia began with the observation of variability in clinical outcome and subtle differences in nuclear morphology. Enzyme-based histochemical analysis, introduced in the 1960s, provided the first basis for classification of acute leukemias into acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Later ALL was divided into 2 subcategories: T-lineage ALL and B-lineage ALL. Some particular subtypes of acute leukemia have been found to be associated with specific chromosomal translocations.

Although the distinction between AML and ALL has been well established, no single test is currently sufficient to establish the diagnosis. Rather, current clinical practice involves an experienced hematopothologist's interpretation of the tumor morphology, histochemistry and immunophenotyping analysis performed in highly specialized laboratory. Although usually accurate, leukemia classification remains imperfect and errors do occur, for example when one type of cancer pretends to be another or when a mix of cancers accidently is identified as cancer of only one type.
The goal is to develop a systematic approach to cancer classification based on gene expression data.

Two data sets were taken:

- Learning set, containing 38 bone marrow samples (27 ALL and 11 AML), that were obtained at the same stage of the disease, but from different patients. On this set features will be learned and predictors will be developed to validate the test set.

- Test set, containing 34 leukemia samples (20 ALL and 14 AML), that consisted of 24 bone marrow and 10 peripheral blood samples.

RNA prepared from bone marrow mononuclear cells was hybridized to old generation Affymetrix oligonucleotide microarrays with 6817 human genes.

7.4.3 Feature Selection

The first goal was to find a set of predicting genes,whose typical expression pattern are strongly correlated with the class distinction to be predicted and have low variance within each class.

Let c = (1,1,1,1,0,0,0,0) be a binary class vector, containing the class assigned to each sample (0 or 1).

Let $gene_i = (e_1, e_2, e_3, ..., e_{12})$ is expression vector for $gene_i$, consisting of its expression levels in each of the tumor samples.

The authors scored a gene as a distinctor by $P(gene_i, c) = \frac{\mu_1 - \mu_0}{\sigma_1 + \sigma_0}$,where μ_i is the mean expression level of samples in class i and σ_i is standard deviation of the expression levels in these samples, $i = 0, 1$. The larger $P(gene, c)$, the better the gene distinction. Hence genes with highest $|P(g, c)|$ are chosen as predictor set.

Neighborhood analysis

The 6817 genes were sorted by their degree of correlation with distinction. To establish whether the observed correlation would be stronger than expected by chance, the researchers developed a method called "Neighborhood analysis". Assume that range of an expression levels is $[-1, 1]$, so expression vector of an ideal class distinctor would be represented by an "idealized expression pattern" c, in which the expression level is uniformly high in class 1 and uniformly low in class 2: $c = (1, 1, 1, 1, -1, -1, -1)$.

The idea of neighborhood analysis is simply look at a neighborhood of a fixed size around c, count the number of gene expression vectors within it, compare it to the number of expression vectors within the neighborhood of the same size around a random permutation of c - $\pi(c)$. Let $N(g)$ be a number of genes such that $P(g, c) > \alpha$, for some constant α. Let R(g) be a number of genes such that $P(g, \pi(c)) > \alpha$. By trying many random permutations we can determine if the neighborhood around c holds more gene expression vectors that we expect to see by chance. If we find that $N(g) \gg E(R(G))$, we can conclude that the class distinction represented by c is likely to be predictable from the expression data. This analysis is illustrated in Figure 7.9. Note that permuting preserves class sizes.

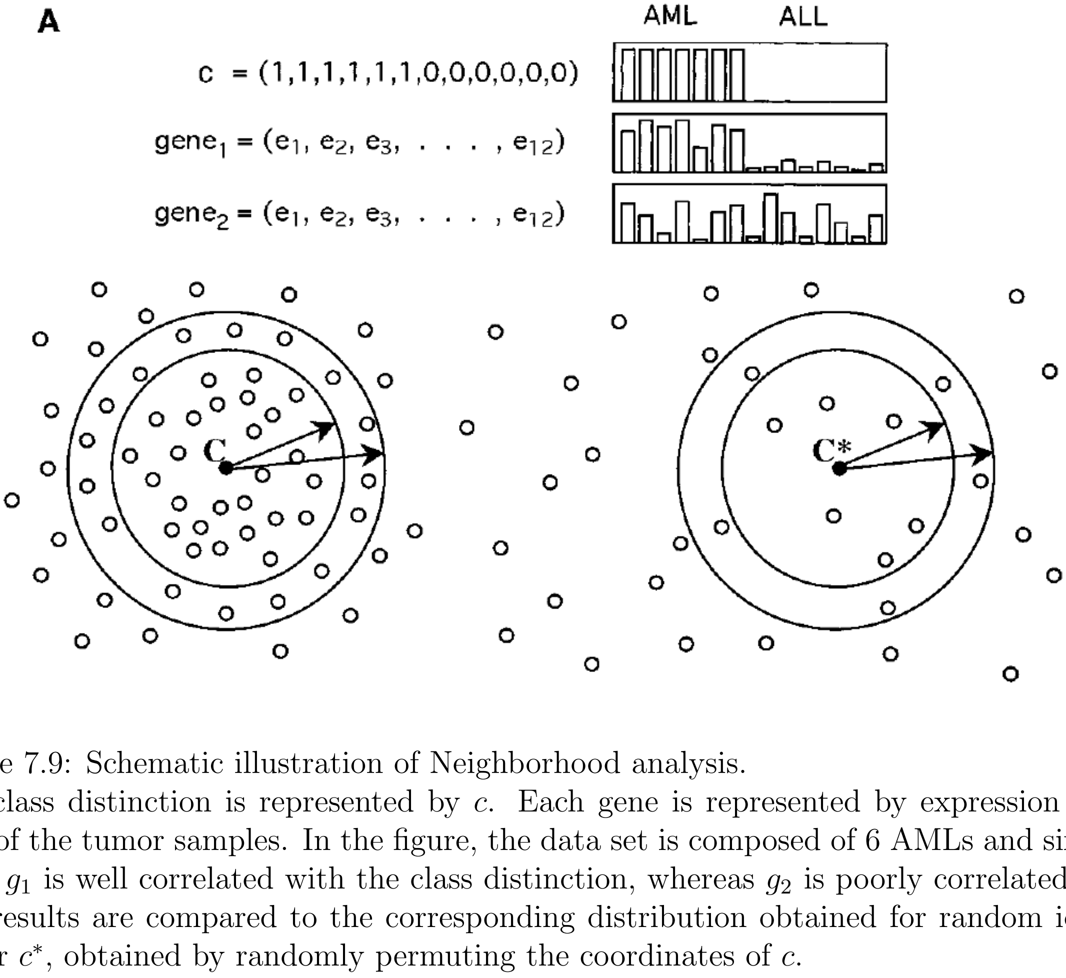

A
AML
ALL
c = (1,1,1,1,1,1,0,0,0,0,0,0)
gene$_1$ = (e$_1$, e$_2$, e$_3$, ... , e$_{12}$)
gene$_2$ = (e$_1$, e$_2$, e$_3$, ... , e$_{12}$)
C
C*

Figure 7.9: Schematic illustration of Neighborhood analysis.
The class distinction is represented by c. Each gene is represented by expression level in each of the tumor samples. In the figure, the data set is composed of 6 AMLs and six ALLs. Gene g_1 is well correlated with the class distinction, whereas g_2 is poorly correlated with c. The results are compared to the corresponding distribution obtained for random idealized vector c^*, obtained by randomly permuting the coordinates of c.

Another approach is to present neighborhoods, whose radii are a function of the expression value, i.e., for each class c we count genes with $P(g, c) > x$ as a function of x. In this approach we do not aspire to get more genes in the same circle, as in previous one, but to obtain small circles, containing many genes.

We can calculate this distribution for known subsets and for random permutations. In Figure 7.10 we can see how many genes (y axis) have as value of at least P on the x axis. If our data was random we would expect that observed data curve should be much closer to the median, but it's not so and for $P(g, c) > 0.3$ it's located far even from 1% significance level. In summary we have about 1000 more genes that are highly correlated with the AML-ALL class distinction, than what we could expect in random.

Since the neighborhood analysis has shown that there are genes significantly correlated with class distinction c, the authors used known samples to create a "class predictor" capable of assigning a new sample to one of two classes. The goal is to choose the k genes most closely correlated with AML-ALL distinction in the known samples.

Choosing a prediction set

We could simple choose the top k genes by the absolute value of $P(g, c)$, but this allows the possibility that, for example, all genes are expressed at a high level in class 1 and a low level(or not at all)in class 2. However, predictors do better when they include some gene expressed at high levels in each class, because to assign new sample to a class, it must correlate with highly expressed genes in this class, so we choose the top k_1 genes (highly expressed in class 1) and the bottom k_2 genes (highly expressed in class 2) so that:

- They must be roughly equal to prevent a sample, that is located somewhere between 2 classes, to be assigned to class 1 just because $k_1 > k_2$ or the opposite.
- The fewer genes we choose the more statistically significant they will be.
- The more genes we choose the robust the results will be obtained. We know that gene expression is different in different people ,different tissues etc., so it is not reasonable that one gene is enough to predict. Also we know that cancer is related to many biological processes, so we expect to find several genes to represent each of these processes.
- Too many genes are not helpful, because if all of them are significantly correlated with a class distinction, it is unlikely that they're all represent different biological mechanisms. Their expression patterns are probably dependent, so that the are unlikely to add information not already provided by the others.

Thus, generally we pick few tens of genes. Let S be the set of the chosen informative genes.

7.4.4 Class prediction

Prediction by weighted voting

We now describe the voting scheme presented in [6]. Each gene in S gets to cast its vote for exactly one class. The gene's vote on a new sample x is weighted by how closely its

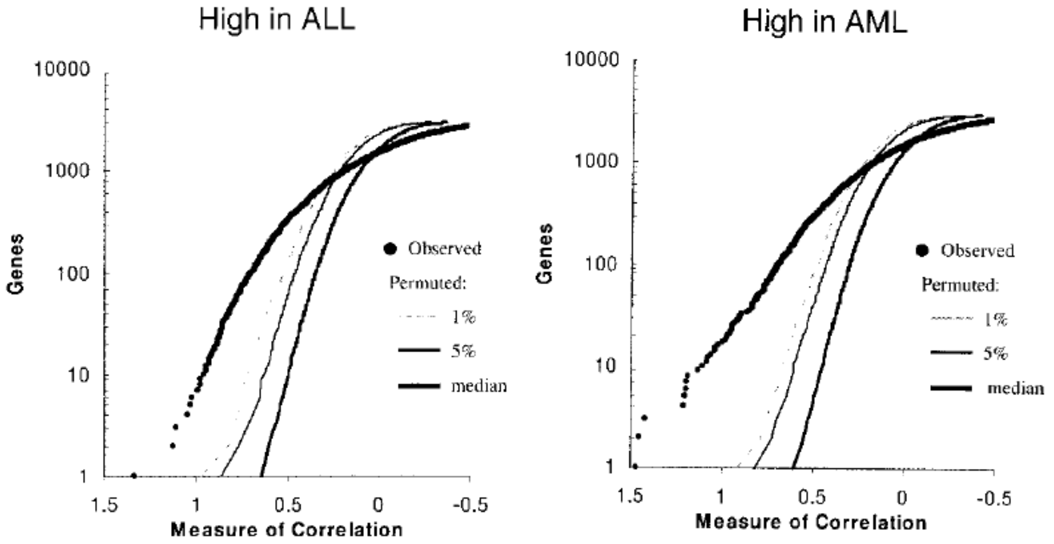


Figure 7.10: Neighborhood analysis: ALL versus AML.
For the 38 leukemia samples in the initial data set, the plot shows the number of genes within various "neighborhoods" of the ALL-AML class distinction together with curves showing the 5% and 1% significance levels for the number of genes within corresponding neighborhoods of the randomly permuted class distinction. Genes highly expressed in ALL compared to AML are shown in the left panel; those more highly expressed in ALL compared to AML are shown in the left panel. The large number of genes highly correlated with the class distinction is apparent. For example, in the left panel the number of genes with correlation $P(g,c) > 0.3$ was 709 for the AML-ALL distinction, but had a median of 173 genes for random class distinctions. $P(g,c) = 0.3$ is the point where the observed data intersect the 1% significance level, meaning that 1% of random neighborhoods contain as many points as the observed neighborhood around the AML-ALL distinction.

expression in the learning set correlates with c: $w(g) = P(g,c)$. The vote is the product of this weight and a measure of how informative the gene appears to be for predicting the new sample.

Intuitively, we expect the gene's level in x to look like that of either a typical class 1 sample or a typical class 2 sample in the learning set, so we compare expression in the new sample to the class means in the learning set. We define a "decision boundary" as halfway between the two class means: $b_g = \frac{\mu_1-\mu_2}{2}$. The vote corresponds to the distance between the decision boundary and the gene's expression in the new sample. So each gene casts a weighted vote $V = w(g)(x_g - b_g)$.

The weights are defined so that positive votes count as votes for membership in class 1, negative ones for membership in class 2. The votes for all genes in S are combined; V_+ is the sum of positive votes and V_- is the sum of negative votes. The winner is simply the class receiving the larger total absolute vote.

Intuitively, if one class receives most of the votes, it seems reasonable to predict this majority. However, if the margin of victory is small, a prediction for the majority class seems somewhat arbitrary and can only be done with low confidence. The authors therefore define the "prediction strength" to measure the margin of victory as: $PS = \frac{V_{winner} - V_{loser}}{V_{winner} + V_{loser}}$. Since V_{winner} is always greater then V_{loser}, PS varies between 0 and 1. Empirically, the researchers decided on a threshold of 0.3 i.e., if $PS > 0.3$, then x will be assigned to the winning class, otherwise x is left undetermined. Figure 7.11 shows a graphic presentation of the solution: Each gene g_i votes for either AML or ALL, depending on its expression level x_i in the sample. Summing separately votes for AML and ALL, shows that ALL wins.

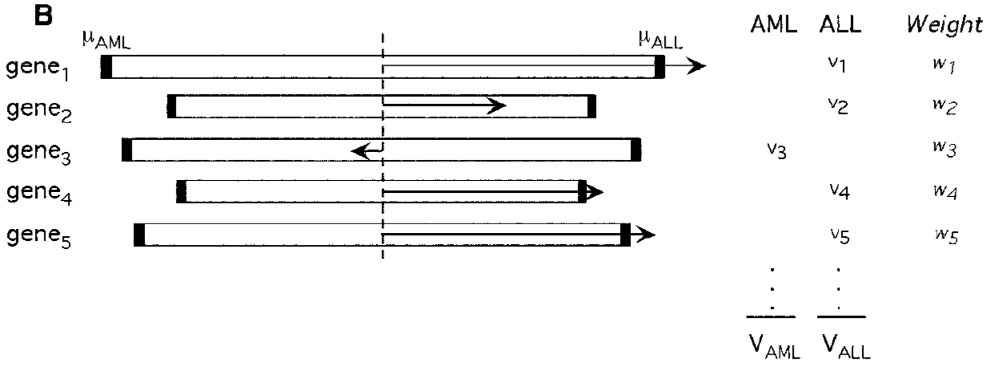


Figure 7.11: Class predictor.
The prediction of a new sample is based on the weighted votes of a set of informative genes. Each such gene g_i votes for either AML or ALL, depending on whether its expression level x_i is closer to μ_{ALL} or μ_{AML} The votes of each class are summed to obtain V_{AML} and V_{ALL}. The sample is assigned to the class with the higher total vote, provided that the prediction strength exceeds a predetermined threshold.

Testing class predictors

There are two possibilities to test the validity of class predictors:

1. Leave-one-out Cross Validation (LOOCV) on initial data set:
   Withhold a sample, choose informative features, build a predictor based on the remaining samples, and predict the class of the withheld sample. The process is repeated for each sample and the cumulative error rate is calculated.

2. Validation on an independent set:
   Assess the accuracy on an independent set of samples.

Generally, the two above procedures are carried out together. Testing on an independent is better, but we are forced to do LOOCV, when samples are very scarce.

The authors applied this approach to the acute leukemia samples. The set of informative gene to be used as predictors was to chosen to be the 50 genes most closely correlated with AML-ALL distinction in the known samples. The following results were obtained:

1. On learning set:
   These predictors assigned 36 of 38 samples as either AML or ALL and the remaining two as uncertain($PS < 0.3$); all predictions agreed with the patients' clinical diagnosis.

2. On independent test set:
   These predictors assigned 29 of 34 samples with 100% accuracy.

The success is notable, as the test set included samples from peripheral blood, from childhood AML patients and from different reference laboratories that used different sample preparation protocols.
Overall, as shown in Figure 7.12, the prediction strength was quite high. The median in cross validation was $PS = 0.77$. On the test the median was PS = 0.73. The average prediction strength was lower for samples from one laboratory, that used a very different protocol for sample preparation. This suggests that clinical implementation of such approach should include standardization of sample preparation.

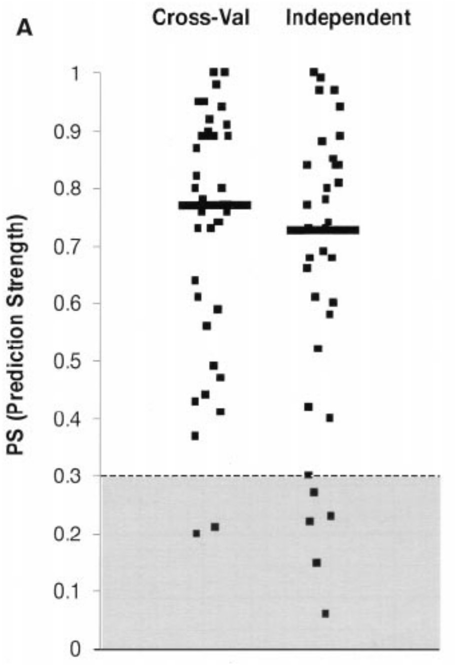


Figure 7.12: Prediction strengths.
The scatter-plots show the prediction strengths(PS) for the samples in cross-validation(left) and on the independent set(right). Median PS is denoted by a horizontal line. Predictors with PS less 0.3 are considered uncertain.

The choice to use 50 informative genes in the predictor was somewhat arbitrary.In fact, the results were insensitive to the particular choice: predictors based on between 10 to 200 genes were all found to be 100% accurate, reflecting the strong correlation with the AML-ALL distinction.

Informative genes

The list of informative genes used in the AML versus ALL predictor is highly instructive. Figure 7.13 show the list of used informative genes.

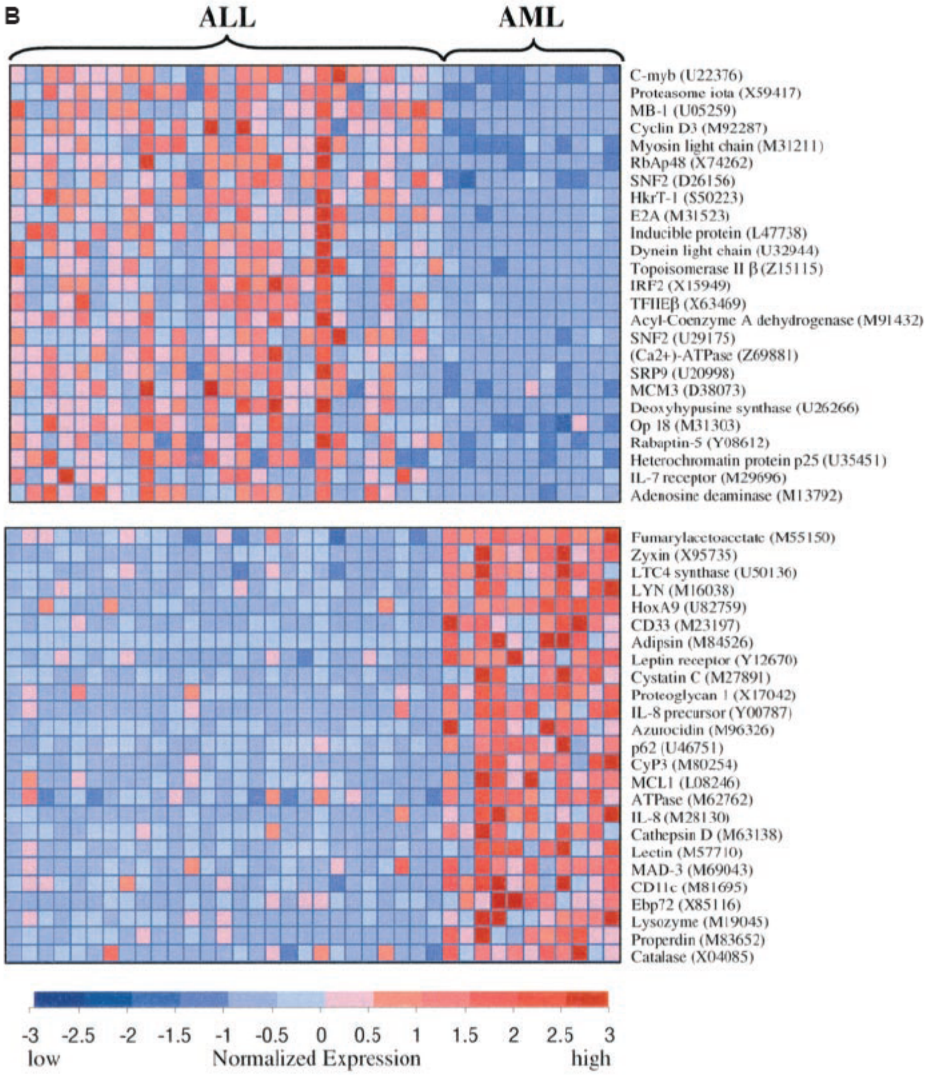


Figure 7.13: Genes distinguishing ALL from AML.
The 50 genes most highly correlated with the ALL-AML class are shown. Each row corresponds to a gene, with the columns corresponding to expression level in different samples. Expression levels for each gene are normalized across the samples such that the mean is 0 and the standard deviation(SD) is 1. Expressions level greater then the mean are shaded in red, and those below the mean are shaded in blue. The scale indicates SDs above or below the mean.

Some genes, including CD11c, CD33 and MB-1 encode cell surface protein useful in distinguishing lymphoid from myeloid lineage cells. Others provide new markers of acute

leukemia subtype, for example, the leptin receptor [5] , originally identified through its role in weight regulation, showed high relative expression in AML. These data suggest that genes useful for cancer class prediction may also provide insights into cancer pathogenesis and pharmacology.
The researchers explored the ability to predict response to chemotherapy among the 15 adult AML patients who had been treated and for whom long-term clinical follow-up was available. Eight patients failed to achieve remission after induction chemotherapy, while the remaining seven remained in remission for 46 to 84 months. Neighborhood analysis found no striking excess of genes correlated with the response to chemotherapy. Class predictors that used 10 to 50 genes were not highly accurate in cross-validation. Thus, no evidence of strong multigene expression signature was correlated with clinical outcome, although this could reflect the relatively small sample size.

7.4.5 Class Discovery

Next the researchers turn to the question of class discovery. They explored whether cancer classes could be discovered automatically. For example, if the AML-ALL distinction was not already known, could we discover it simply on the basis of gene expression.

The authors used a clustering algorithm for this part. To cluster tumors Golub et al. used the SOM (self-organizing maps) algorithm. At first a 2-cluster SOM was applied to cluster the 38 initial leukemia samples on the basis of the expression patterns of all 6817 genes. The clusters were evaluated by comparing them to the known AML-ALL classes (Figure 7.14A). The results were as follows: cluster A1 contained mostly ALL samples (24 of 25) and cluster A2 contained mostly AML samples (10 of 13).

On basis of the clusters the authors constructed predictors to assign new samples as “type A1” or “type A2” and tested their validity. Predictors that used a wide range of different numbers of informative genes preformed well in cross-validation. The cross-validation not only showed high accuracy, but actually refined the SOM-defined classes: the subset of samples accurately classified in cross-validation were those perfectly subdivided by SOM into ALL and AML. Then the class predictor of A1-A2 distinction was tested on the independent test set. The prediction strengths were quite high: the median PS was 0.61 and 74% of samples were above threshold (Figure 7.14B), indicating that the structure seen in the initial set is also seen in the test set. In contrast, random clusters consistently yielded predictors with poor accuracy in cross-validation and low PS on the independent data set.

On basis of such analysis, the A1-A2 distinction can be seen to be meaningful, rather then simply a statistical artifact of the initial data set. The results thus show that the AML-ALL distinction could have been automatically discovered and confirmed without previous biological knowledge.

Finally, the researchers tried to extend the class discovery by searching for finer subclasses of the leukemias. 4-cluster SOM divided the samples into four classes, that largely corresponded to AML, T-lineage ALL, B-lineage ALL and B-lineage ALL, respectively (Figure 7.14C). When these classes were evaluated by constructing class predictors, all pairs could be distinguished one from another, with exception of B3 versus B4 (Figure 7.14D). The

[5] leptin receptor is a molecule that identifies leptines

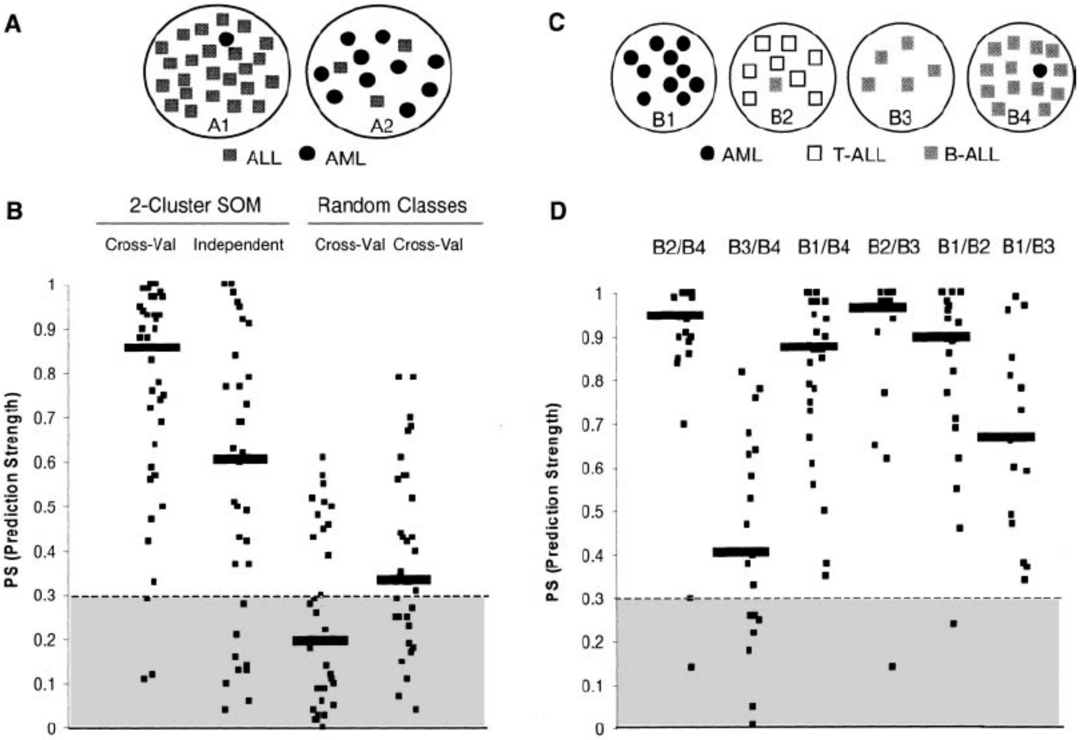


Figure 7.14: ALL-AML class discovery.
(A) Schematic presentation of 2-cluster SOM. A 2-cluster (2 by 1) SOM was generated from the 38 initial leukemia samples with a modification of the GENECLUSTER computer package. Cluster A1 contains the majority of ALL samples(grey squares) and cluster A2 contains the majority of AML samples(black circles).
(B) Prediction strength distribution. The scatter plots show the distribution of PS scores for class predictors.
(C) Schematic presentation of the 4-cluster SOM. ALL samples are shown as black circles, T - lineage ALL as open squares, and B-lineage ALL as grey squares.
(D) Prediction strength distribution for pairwise comparison among classes. Cross-validation studies show that the four classes could be distinguished with high prediction scores, with the exception of classes B3 and B4.

prediction tests thus confirmed the distinction corresponding to AML, B-ALL and T-ALL and suggested that it may be appropriate to merge classes B3 and B4, composed primarily of B-lineage ALL.

7.4.6 Class discovery of a new class

In the previous article the authors showed how they can discover classes which are already known. In Bittner et al [1] the authors discover new classes previously unknown in melanoma. They used the CAST clustering algorithm to cluster the data into different groups and found a subset which has a different gene signature. They later experimantally proved *in vitro* that this new subtype is related to invasive melanomas that form primitive tubular networks, a feature of some highly aggresive metastastatic melanomas.

7.5 Breast Cancer Classification

Breast cancer is one of the most common cancers found in women. Different breast cancer patients with the same stage of disease can have markedly different treatment responses and overall outcome. Current methods (such as lymph node status and histological grade) fail to classify accurately breast tumors and 70-80% of patients receiving chemotherapy or hormonal treatment would have survivaed with out it. In [15] the authors aim to find gene expression based classification methods that can predict the clinical outcome of breast cancer.

The authors used a data set containing 98 primary breast cancers divided into the following sub groups:

1. Sporadic patients:

   (a) 34 samples from patients who developed distant metastases within 5 years. These were called the poor prognosis group and the mean time to metastases was 2.5 years.

   (b) 44 samples from patients who continued to be disease-free after a period of at least 5 years. These were called the good prognosis group and had a mean follow-up time of 8.7 years.

   All sporadic patients were lymph node negative and under 55 years of age at diagnosis.

2. Carriers:

   (a) 18 samples from patients with BRCA1 germline mutations.

   (b) 2 samples from BRCA2 mutation carriers.

From each of the above patients 5μg of RNA was isolated from their tumor samples. The RNA was then hybridized on Agilent microarrays containing approximately 25,000 human genes. Out of these 25,000 genes, 5,000 were significantly regulated across the group of samples i.e. at least a twofold difference in more than five tumors.

7.5.1 Hierarchical clustering of the data

The first step performed by the authors was to use a non-supervised, hierarchical clustering algorithm to cluster the 98 tumors on the basis of their similarities measured over these 5,000 significant genes. Similarly, the 5,000 genes were clustered on the basis of their similarities measured over the group of 98 tumors. The results of this clustering can be seen in figure 7.15.

Looking at both parts of the figure we can see a connection between the two main clusters and the clinical markers. For example, looking at the metastases column (the right-most column) we can see that in the upper group only 34% of the sporadic patients were from the group who developed distant metastases within 5 years, whereas in the lower group 70% of the sporadic patients had progressive disease. Thus, using unsupervised clustering we can already, to some extent, distinguish between good prognosis and bad prognosis tumors.

Another connection we can see is the ER (estrogen receptor) status and lymphocytic infiltration. We see that the top group is enriched in positive ER tumors and negative

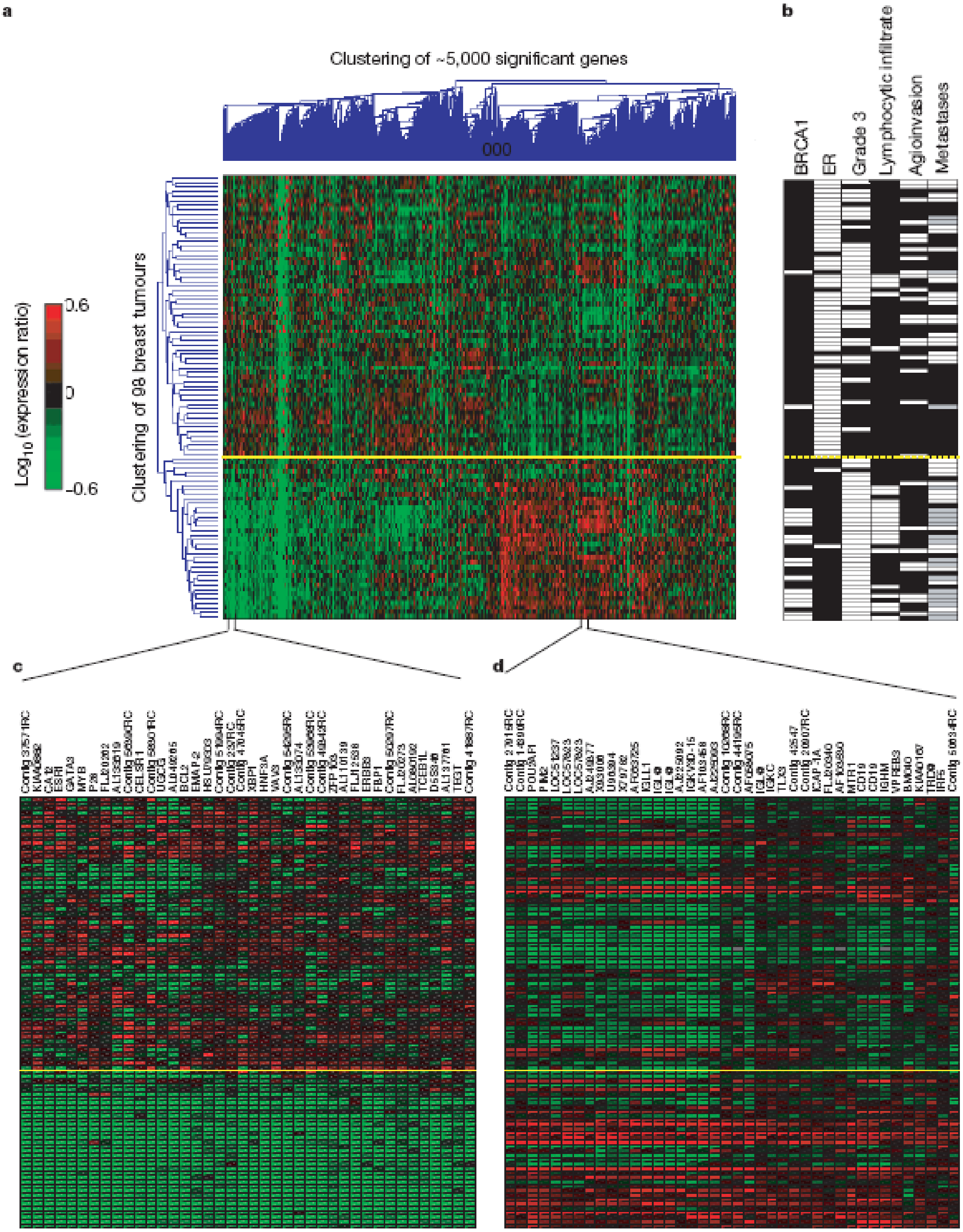


Figure 7.15: Unsupervised two-dimensional clustering analysis of 98 breast tumors and 5,000 genes. **a** Heatmap ordered according to the clustering. Each row represents one of the 98 tumors and each column represents one of the 5,000 significant genes. **b** Selected clinical data for the 98 patients. **c** Zoom-in on genes that co-regulate with the ER-α gene. **d** Zoom in on genes connected to lymphocytic infiltration

lymphocytic infiltration and the bottom group is enriched in the opposite phenotypes. This is consistent with previous reports which grouped breast cancer into two subgroups which differ in ER status and lymphocytic infiltration. Part c of the figure shows a zoom in on genes that co-regulate with the ER-α gene and part d of genes connected to lymphocytic infiltration. The difference between the two groups of genes is easily visible.

7.5.2 Classification

The next step the authors did was to try and create a classifier that can classify between the poor prognosis and good prognosis groups using gene expression values. Approximately 5,000 genes (significantly regulated in more than 3 tumors out of 78) were selected from the 25,000 genes on the microarray. The correlation coefficient of the expression for each gene with the disease outcome was calculated and 231 genes were found to be significantly associated with disease outcome ($|CC| > 0.3$) These genes were rank-ordered according to $|CC|$.

A classifier was built using the top 5 genes on this ordered list and then evaluated using LOOCV. In each iteration, 5 more genes from the top of the list were added to the classifier input and LOOCV was used again. The accuracy improved with each iteration until the optimal number of marker genes was reached at 70 genes. The expression pattern of the 70 genes in the 78 samples is shown in Figure 7.16 where tumors are ordered according to their correlation coefficients with the average good prognosis profile. The classifier predicted correctly the actual outcome of disease for 65 out of the 78 patients (83%). 5 poor prognosis and 8 good prognosis patients were assigned to the opposite category. The optimal accuracy threshold is shown as a solid line in part b of Figure 7.16. Since we are more worried about poor prognosis patients being misdiagnosed a more sensitive threshold was used (the dashed line in the figure). This optimized sensitivity threshold resulted in a total of 15 misclassifications: 3 poor prognosis tumors were classified as good prognosis (as opposed to 5 before) and 12 good prognosis tumors were classified as poor prognosis (as opposed to 8 in the previous threshold).

The authors note, that the functional annotation for the genes provides insight into the underlying biological mechanism leading to rapid metastases. Genes invlolved in cell cycle, invastion and metastasis, angiogenesis and signal transduction are significantly up regulated in the poor prognosis signature (for example, cyclin E2, MCM6, metalloproteinases MMP9 ad MP1, RAB6B, PK428, ESM1 and the VEGF receptor FLT1).

7.5.3 Validation

The classifier classifies tumors having a gene expression profile with a correlation coefficient above the 'optimized sensitivity' threshold (dashed line) as good prognosis signature, and below this threshold as a poor prognosis signature. To validate the prognosis classifier, an additional independent set of primary tumors from 19 young, lymph-node-negative breast cancer patients was selected. This group consisted of 7 patients who remained metastasis free for at least five years, and 12 patients who developed distant metastases within 5 years. The disease outcome was predicted by the 70-gene classifier and resulted in 2 out of 19

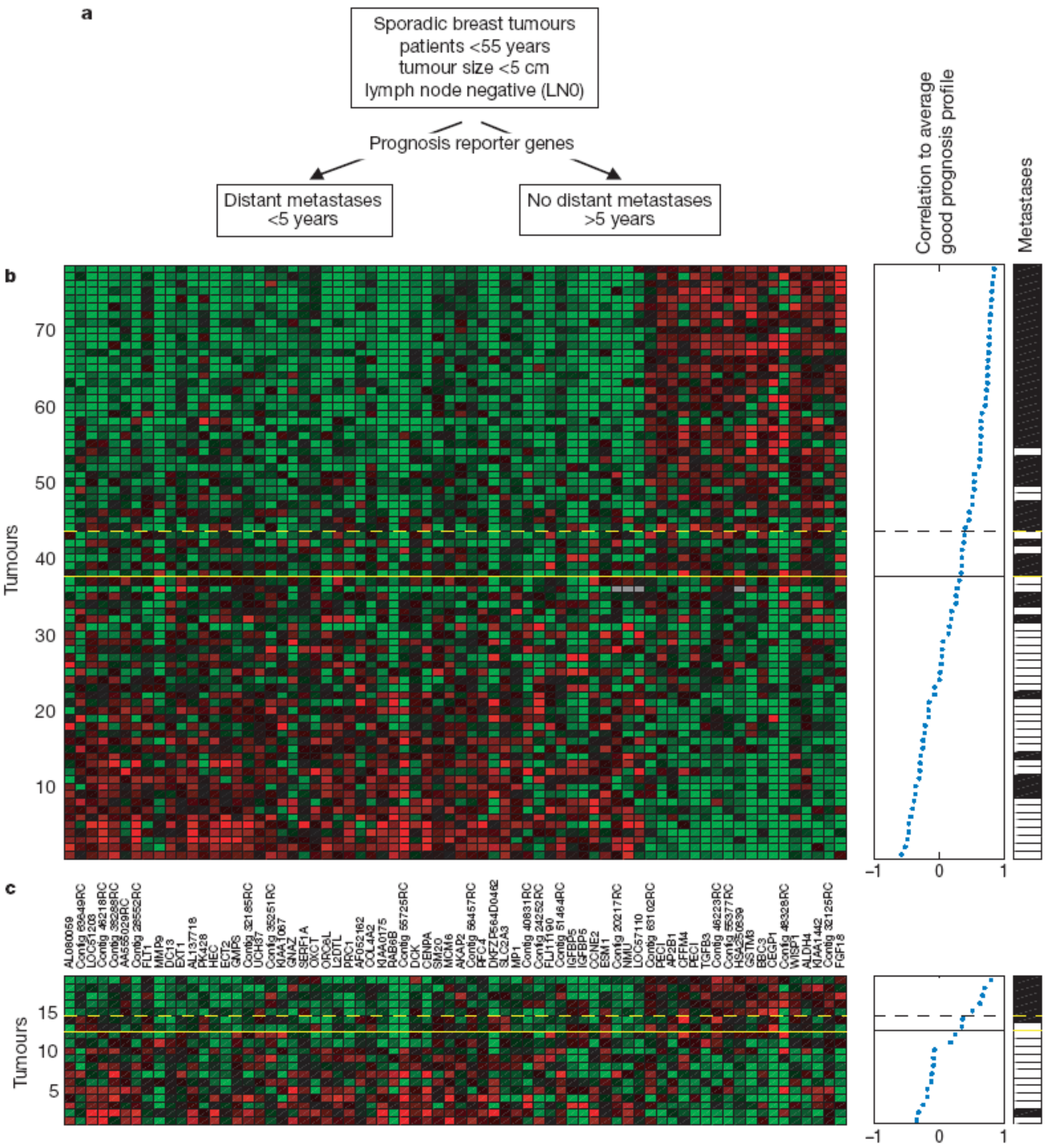


Figure 7.16: Supervised classification on prognosis signatures. **a** The classifier structure. **b** A heatmap of the expression values. Each row represents a tumor and each column a gene. The solid line is the optimal accuracy threshold and the dashed line the optimal sensitivity **c** Same as **b**, but the expression matrix is for tumors of 19 additional breast cancer patients.

incorrect classification using both the optimal accuracy threshold (solid line) and optimal sensitivity threshold (dashed line). The results are shown in part c of Figure 7.16.

The odds ratio that a woman under 55 years of age diagnosed with lymph-node-negative breast cancer that has a poor prognosis signature to develop a distant metastasis within 5 years compared with those that have the good prognosis signature is 15-fold. This is compared to previous methods which achieved only 2.4-6.4 fold.

7.5.4 Using the classifier to predict survival rate

In a different paper [14], the same group of authors tried to re-validate their classifier on a much larger population. This time 295 samples taken from patients with breast cancer were used. 151 of the patients had lymph-node-negative disease and 144 patients had lymph-node-positive disease.

The same classifier using the previous 70 genes signature and threshold was used to classify the patients. Among the 295 patients, 180 had a poor prognosis signature and 115 had a good-prognosis signature. The 10 year survival rate was 54.6±4.4 percent for the poor prognosis group and 94.5±2.6 percent for the good prognosis group. At 10 years, the probability of remaining free of distant metastases was 50.6±4.5 percent in the group with a poor-prognosis signature and 85.2±4.3 percent in the group with a good-prognosis signature.

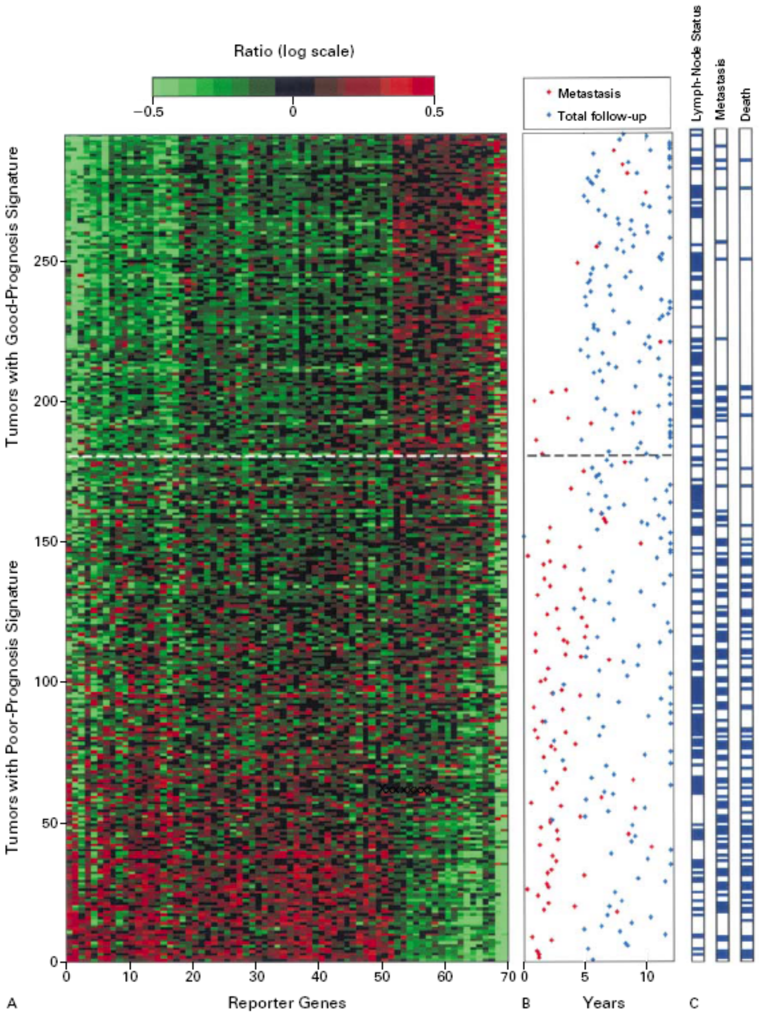
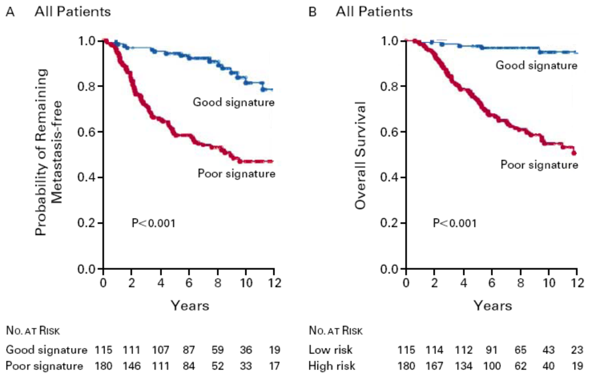


Figure 7.17: Pattern of expression of genes in 295 patients with breast cancer.

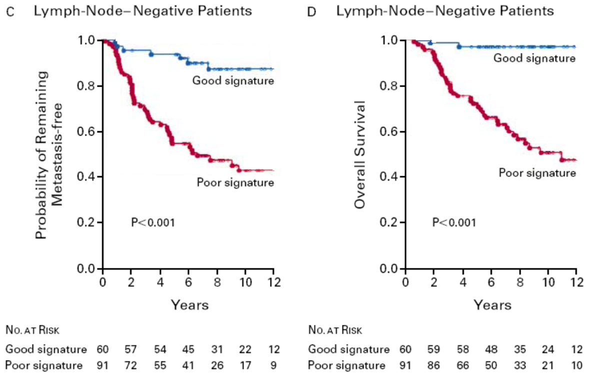
Figure 7.17 shows a heatmap for the 295 tumors and 70 genes. The tumors are ranked the same as in Figure 7.16 . Notice that the tumors classified as good-prognosis have a



No. at Risk							
Good signature	115	111	107	87	59	36	19
Poor signature	180	146	111	84	52	33	17

No. at Risk							
Low risk	115	114	112	91	65	43	23
High risk	180	167	134	100	62	40	19

Figure 7.18: Kaplan Meier Analysis of the probability that patients would remain free of metasteses and the probablility of overall survival among all patients.



No. at Risk							
Good signature	60	57	54	45	31	22	12
Poor signature	91	72	55	41	26	17	9

No. at Risk							
Good signature	60	59	58	48	35	24	12
Poor signature	91	86	66	50	33	21	10

Figure 7.19: Kaplan Meier Analysis of the probability that patients would remain free of metasteses and the probablility of overall survival among patients with lymph-node-negative disease.

much lower incidence of metastases and death. The authors also show Kaplan Meier plots to show the probability that patients would remain metastases free and the probability of overall survival among all patients. Figure 7.18 shows metastasis rates and survival rates amongst all patients. It is easily seen that as the years go by, the chances for patients from the good prognosis group to survive and remain metastasis free are significantly higher than those in the poor prognosis group. Figure 7.19 shows the same plots for lymph-node-negative patients.

7.5.5 Is the above gene set unique?

In 2005, Ein-Dor et al [5] set out to check if the 70-genes signature described before is unique in its ability to classify breast cancer. The gene set has little or no overlap with other published work. To test this the authors used the same training set used in [15] consisting of 77 sporadic patients and the same test set of 19 sporadic patients([15] used 78 samples for their training set but the authors in this paper removed one because it had more than 20% missing values). In a similar way to [15] the authors first created a sub-set of about 5,000 genes which are highly regulated and ranked the genes in this subset according to their correlation to survival. The correlation p-value was calculated using comparison to the correlation with 10^5 permutated survival vectors.

Next, a series of classifiers was built using consecutive groups of 70 genes. For each classifier, the training and the test error was measured, and seven other sets of 70 genes were found to produce classifiers with the same prognostic capabilities as those based on the top 70. The Kaplan-Meier plots for the classifier based on the top 70 genes, and the other seven classifiers are shown in figure 7.20. The figure shows that the new seven classifiers based on lower ranking genes than the original classifier produced similar results.

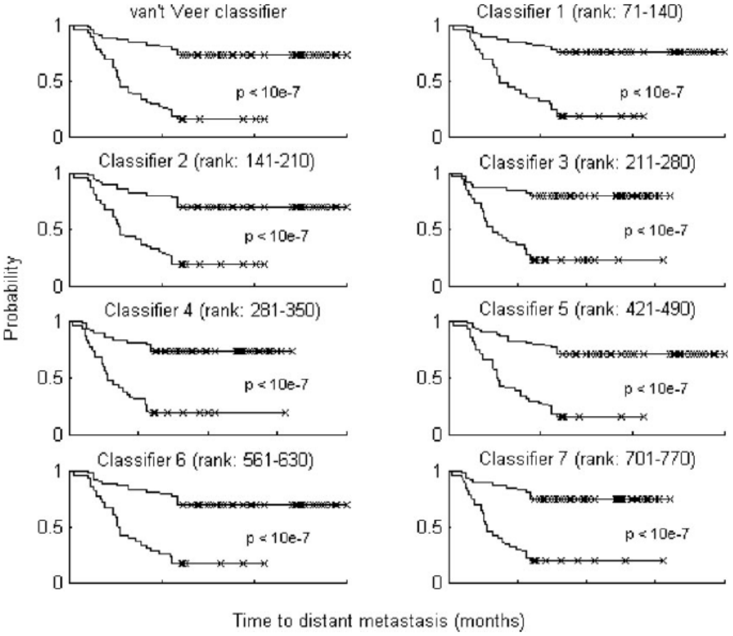


Figure 7.20: Kaplan-Meier analysis of van't classifier and of the seven alternative classifiers as obtained from classifying all 96 samples. Upper curves describe the probability of remaining free of metastasis in the group of samples classified as having a good prognosis signature, while the lower curves describe the poor prognosis group.

After showing that the 70-gene signature is not unique in its ability to predict survival rates, the authors tried to see if the gene list is dependent on the selection of the training set. Out of the 96 total samples 77 random patients were chosen to be the training set. The same ratio of good/poor prognosis (33/44) was kept, and repetitions (using bootstrapping) were used. Figure 7.21 shows the location of the top 70 genes (ranked by correlation). The genes in the figure are ranked according to the first training set. The figure shows that for each selection of the training set, the ranks of the top 70 genes change significantly, and therefore not robust to the selection of the training set.

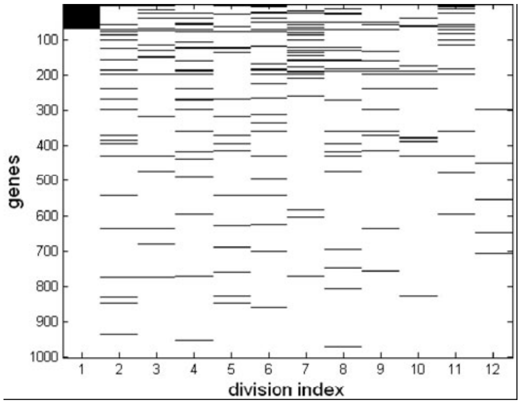


Figure 7.21: Ten sets of top 70 genes, identified in 10 randomly chosen training sets ofN = 77 patients. Each row represents a gene and each column a training set. The genes were ordered according to their correlation rank in the first training set (leftmost column). For each training set, the 70 top ranked genes are colored black. The genes that were top ranked in one training set can have a much lower rank when another training set is used. The two rightmost columns (columns 11 and 12) mark those of the 70 genes published by van't Veer et al. [15] and the 128 genes appearing in (Ramaswamy et al. [12]) that are among the top 1000 of our first training set.

In conclusion, the authors of [5] show that the 70-gene signature is neither unique in its ability to predict survival nor is it robust and is very dependent on the selection of the training set. Therefore, one should not try to gain insight on the biological behavior of cancer based on these 70 genes. Nevertheless, the 70-gene signature still produces good results in prognosis and can be used in the clinical world.

7.6 Classification into multiple classes

7.6.1 Introduction

We now describe the work of Ramaswamy et al. [11] that deals with multiclass cancer diagnosis.

To establish analytic methods capable of solving complex, multiclass gene expression-based classification problems the researchers created a gene expression database, containing the expression profiles of 218 tumor samples, representing 14 common human cancer classes, and 90 normal tissue samples. Hybridization targets were prepared with RNA from whole tumors. The targets were hybridized sequentially to oligonucleotide arrays, containing a total of 16063 probe sets. Expression values for each gene were calculated by using Affymetrix GENECHIP analysis software. Two fundamentally different approaches to data analysis were explored: clustering (unsupervised learning) and classification (supervised learning).

7.6.2 Clustering

As we already know, this approach allows the dominant structure in a dataset to dictate the separation of samples into clusters based on overall similarity in expression, without prior knowledge of sample identity.

Of 16,063 expression values considered, 11,322 passed some variation filter (see [11] for details) and were used for clustering. The dataset was normalized by standardizing each row (gene) to mean = 0 and variance = 1. Average-linkage hierarchical clustering was performed by using CLUSTER and TREEVIEW software. Self-organizing map analysis was performed by using GENECLUSTER analysis package. Figure 7.22 shows the results of both hierarchical and self-organizing map clustering of this data set.

7.6.3 Classification

To make multiclass distinctions the researchers devised an analytic scheme, depicted in Figure 7.23.

One vs. All (OVA) SVM scheme

For each known type a binary classifier is built. The classifier uses the SVM algorithm to define a hyperplane that best separates training samples into two classes: samples from this class vs. all other samples. An unknown test sample's position relative to the hyperplane determines its class, and the confidence of each SVM prediction is based on the distance of the test sample from the hyperplane; that distance is calculated in 16,063-dimensional gene space, corresponding to the total number of expression values considered.

Recursive feature elimination

Given microarray data with n genes per sample, each OVA SVM classifier outputs a hyperplane w, that can be thought of as a vector with n elements each corresponding to the expression of a particular gene. Assuming the expression values of each gene have similar

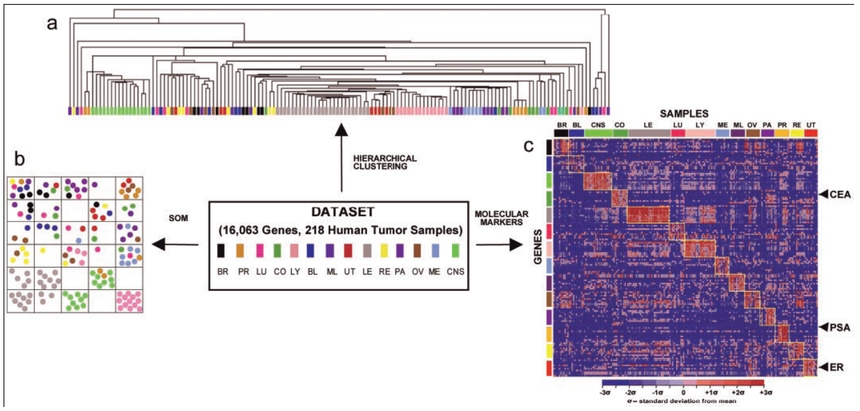


Figure 7.22: Clustering of tumor gene expression data and identification of tumor-specific molecular markers.
Hierarchical clustering (a) and a 5 x 5 self-organizing map (SOM) (b) were used to cluster 144 tumors spanning 14 tumor classes according to their gene expression patterns. (c) Gene expression values for class-specific OVA markers are shown. Columns represent 190 primary human tumor samples ordered by class. Rows represent 10 genes most highly correlated with each OVA distinction. Red indicates high relative level of expression, and blue represents low relative level of expression. The known cancer markers prostate-specific antigen (PSA), carcinoembryonic antigen (CEA), and estrogen receptor (ER) are identified. BR, breast adenocarcinoma; PR, prostate adenocarcinoma; LU, lung adenocarcinoma; CR, colorectal adenocarcinoma; LY, lymphoma; BL, bladder transitional cell carcinoma; ML, melanoma; UT, uterine adenocarcinoma; LE, leukemia; RE, renal cell carcinoma; PA, pancreatic adenocarcinoma; OV, ovarian adenocarcinoma; ME, pleural mesothelioma; CNS, central nervous system.

ranges, the absolute magnitude of each element in w determines its importance in classifying the sample, since the class label is $sign[f(x)]$. Each OVA SVM classifier is first trained with all genes, 10 % of the genes with least $|w_i|$ are removed , and each classifier is retrained with the smaller gene set. This procedure is repeated iteratively to study prediction accuracy as a function of gene number.

Prediction

Each test sample is presented sequentially to the 14 OVA classifiers, each of which either claims or rejects that sample as belonging to a single class with an associated confidence. Finally, each test sample is assigned to the class with the highest OVA classifier confidence. If the confidences were low no prediction is made.

Testing and Results

As we mentioned above, the number of genes contributing to the high accuracy of the SVM OVA classifier was also investigated. The SVM algorithm considers all 16063 input genes and naturally utilizes all genes that contain information for each OVA distinction.

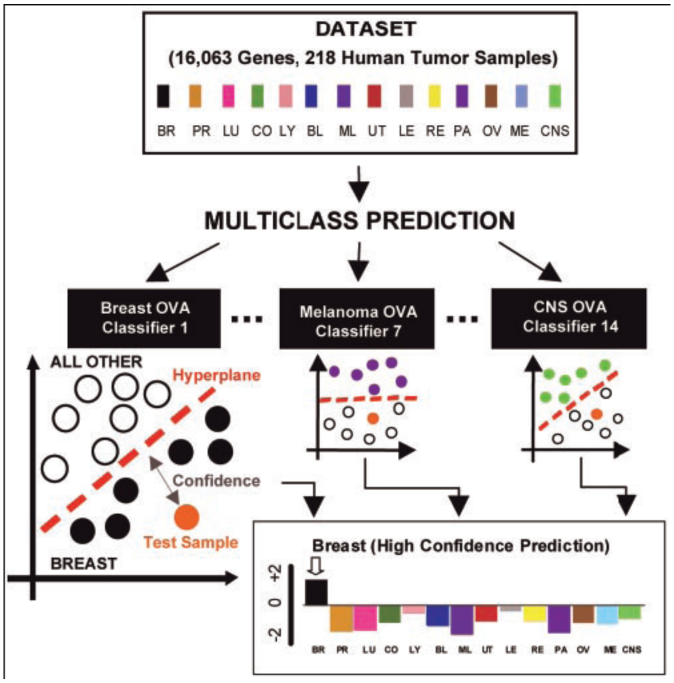

DATASET
(16,063 Genes, 218 Human Tumor Samples)
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS
MULTICLASS PREDICTION
Breast OVA Classifier 1
Melanoma OVA Classifier 7
CNS OVA Classifier 14
ALL OTHER
Hyperplane
Confidence
Test Sample
BREAST
Breast (High Confidence Prediction)
+2
0
-2
BR PR LU CO LY BL ML UT LE RE PA OV ME CNS

Figure 7.23: Multiclass classification scheme.
The multiclass cancer classification problem is divided into a series of 14 OVA problems, and each OVA problem is addressed by a different class-specific classifier (e.g., “breast cancer” vs. “not breast cancer”). Each classifier uses the SVM algorithm to define a hyperplane that best separates training samples into two classes. In the example shown, a test sample is sequentially presented to each of 14 OVA classifiers and is predicted to be breast cancer, based on the breast OVA classifier having the highest confidence.

Genes are assigned weights based on their relative contribution to the determination of each hyperplane, and genes that do not contribute to a distinction are weighted zero. Virtually all genes on the array were assigned weakly positive or negative weights in each OVA classifier, indicating that thousands of genes potentially carry information relevant for the 14 OVA class distinctions. To determine whether the inclusion of this large number of genes was actually required for the observed high-accuracy predictions, the authors examined the relationship between classification accuracy and gene number by using recursive feature elimination. As shown in Figure 7.25, maximal classification accuracy is achieved when the predictor utilizes all genes for each OVA distinction. Nevertheless, significant prediction can still be achieved by using smaller gene numbers.

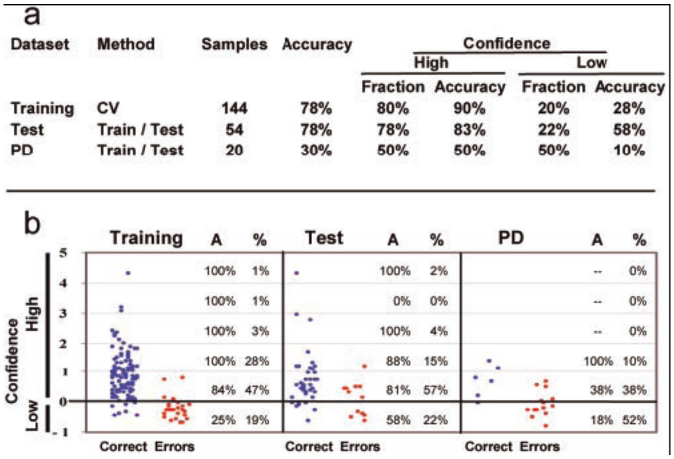
The accuracy of the multiclass SVM-based classifier in cancer diagnosis was first evaluated by leave-one-out cross-validation in a set of 144 training samples. As shown in Figure 7.24, the majority (80%) of the 144 calls was high confidence (defined as confidence > 0) and these had an accuracy of 90%, using the patient’s clinical diagnosis as the ”gold standard”. The remaining 20% of the tumors had low confidence calls (confidence 0), and these predictions had an accuracy of 28%. Overall, the multiclass prediction corresponded to the correct assignment for 78% of the tumors, far exceeding the accuracy of random classification(9%).

For half of the errors, the correct classification corresponded to the second- or third-most confident OVA prediction.

These results were confirmed by training the multiclass SVM classifier on the entire set of 144 samples and applying this classifier to an independent set of 54 tumor samples. Overall prediction accuracy on this test set was 78%.

Poorly differentiated samples yielded low-confidence prediction in cross-validation and could not be accurately classified according to the tissue origin, indicating that they are molecularly distinct entities with different gene expression patterns compared with their well differentiated counterparts.

Overall, these results demonstrate the feasibility of accurate multiclass molecular cancer classification and suggests a strategy for future clinical implementation of molecular cancer diagnosis.



a

Dataset	Method	Samples	Accuracy	Confidence			
				High		Low	
				Fraction	Accuracy	Fraction	Accuracy
Training	CV	144	78%	80%	90%	20%	28%
Test	Train / Test	54	78%	78%	83%	22%	58%
PD	Train / Test	20	30%	50%	50%	50%	10%

Figure 7.24: Multiclass classification results.
(a) Results of multiclass classification by using cross-validation on a training set (144 primary tumors) and independent testing with 2 test sets: Test (54 tumors; 46 primary and 8 metastatic) and PD (20 poorly differentiated tumors; 14 primary and 6 metastatic). (b) Scatter plot showing SVM OVA classifier confidence as a function of correct calls (blue) or errors (red) for Training, Test, and PD samples. A - accuracy of prediction; % - percentage of total sample number

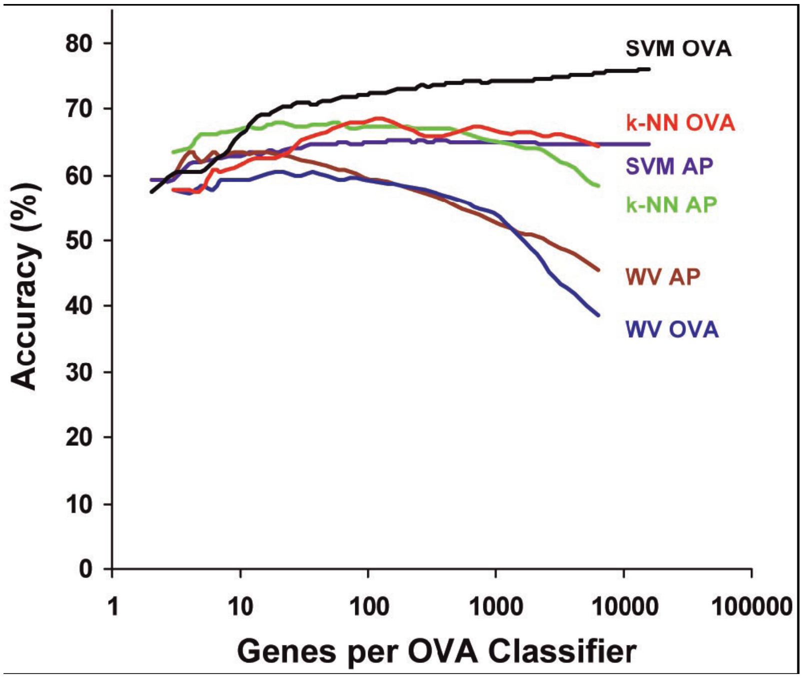


Figure 7.25: Multiclass classification as a function of gene number.
Training and test datasets were combined (190 tumors; 14 classes), then were randomly split into 100 training and test sets of 144 and 46 samples (all primary tumors) in a class-proportional manner. SVM OVA prediction was performed, and mean classification accuracy for the 100 splits was plotted as a function of number of genes used by each of the 14 OVA classifiers, showing decreasing prediction accuracy with decreasing gene number. Results using other algorithms (k-NN, k-nearest neighbors; WV, weighted voting) and classification schemes (AP, all-pairs) are also shown.

Bibliography

[1] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, and V. Sondak. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406(6795):536–40, 2000.

[2] M.B. Eisen P.T. Spellman P.O. Brown D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.*, 95(25):14863–14868, 1998.

[3] C.J.C Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:1–47, 1998.

[4] R. Freund E. Osuna and F. Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, MIT, 1996.

[5] Liat Ein-Dor, Itai Kela, Gad Getz, David Givol, and Eytan Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21(2):171–178, 2005.

[6] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999.

[7] G.Zoutendijk. *Methods of Feasible Directions: A study in linear and non-linear programming*. Elsevier, 1970.

[8] M.P. Brown W.N. Grundy D. Lin N. Cristianini C.W. Sugnet T.S. Fury M. Ares D Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci.*, 97(1):262–267, 2001.

[9] J. Shawe-Taylor N. Cristianini. *An Introduction to support vector machines and other Kernel based learning methods*. Cambridge University Press, 2000.

[10] A. Ben-Dor L. Bruhn I. Nachman M. Schummer N. Friedman and Z. Yakhini. Tissue classification with gene expression profiles. *J. Computational Biology*, 7:559–584, 2000.

[11] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci U S A*, 98(26):15149–15154, December 2001.

[12] Sridhar Ramaswamy, Ken N. Ross, Eric S. Lander, and Todd R. Golub. A molecular signature of metastasis in primary solid tumors. *Nature Genetics*, 33:49–54, 2002.

[13] Donna K. Slonim, Pablo Tamayo, Jill P. Mesirov, Todd R. Golub, and Eric S. Lander. Class prediction and discovery using gene expression data. In *RECOMB*, pages 263–272, 2000.

[14] M. J. van de Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E. T. Rutgers, S. H. Friend, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med*, 347(25):1999–2009, December 2002.

[15] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, January 2002.

[16] V.N.Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1999.