Gene Selection for Microarray Data

Sepp Hochreiter
hochreit@cs.tu-berlin.de

Klaus Obermayer
Technische Universität Berlin
Fakultät für Elektrotechnik und Informatik
Franklinstraße 28/29
10587 Berlin, Germany
oby@cs.tu-berlin.de

in

Kernel Methods in Computational Biology

Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert
MIT Press, pp. 319-355, 2004

Contents

1 Gene Selection for Microarray Data

Sepp Hochreiter
hochreit@cs.tu-berlin.de

Klaus Obermayer
Technische Universität Berlin
Fakultät für Elektrotechnik und Informatik
Franklinstraße 28/29
10587 Berlin, Germany
oby@cs.tu-berlin.de

In this chapter we discuss methods for gene selection on data obtained from the microarray technique. Gene selection is very important for microarray data, (a) as a preprocessing step to improve the performance of classifiers or other predictors for sample attributes, (b) in order to discover relevant genes, that is genes which show specific expression patterns across the given set of samples, and (c) to save costs, for example if the microarray technique is used for diagnostic purposes. We introduce a new feature selection method which is based on the support vector machine technique. The new feature selection method extracts a sparse set of genes, whose expression levels are important for predicting the class of a sample (for example "positive" vs. "negative" therapy outcome for tumor samples from patients). For this purpose the support vector technique is used in a novel way: instead of constructing a classifier from a minimal set of most informative samples (the so-called support vectors), the classifier is constructed using a minimal set of most informative features. In contrast to previously proposed methods, however, features rather than samples now formally assume the role of support vectors. We introduce a protocol for preprocessing, feature selection and evaluation of microarray data. Using this protocol we demonstrate the superior performance of our feature selection method on data sets obtained from patients with certain types of cancer (brain tumor, lymphoma, and breast cancer), where the outcome of a chemo- or radiation therapy must be predicted based on the gene expression profile. The feature selection method extracts genes (the so-called support genes) which are correlated with therapy outcome. For classifiers based on these genes, generalization performance is improved compared to previously proposed methods.

1.1 Introduction

Gene expression profiles obtained by the microarray technique provide a snapshot of the expression values of some thousand up to some ten thousand genes in a particular tissue sample. The advantage of the microarray method — namely to monitor a large number of variables of a cell's (or a piece of tissue's) state, however, often turns out to be difficult to exploit. The number of samples is small and the level of noise is high which makes it difficult to detect the small number of genes relevant to the task at hand. Therefore, specific gene selection methods must be designed in order to reliably extract relevant genes.

1.1.1 Microarray Technique

The microarray technique (Southern, 1988; Lysov et al., 1988; Drmanac et al., 1989; Bains and Smith, 1988) is a recent technique which allows to monitor the concentration of many kinds of messenger RNA (mRNA) simultaneously in cells of a tissue sample and provides a snapshot of the pattern of gene expression at the time of preparation (Wang et al., 1998; Gerhold et al., 1999). The so-called DNA microarrays allow for the first time the simultaneous measurement of several 1000 or 10,000 expression levels providing valuable information about whole genetic networks. DNA microarrays allow to search for genes related to certain properties of the sample tissue and to extract related genes via dependencies in their expression pattern.

Figure 1.1 depicts the microarray procedure. Messenger RNA is extracted from the samples (Step 1) and reversely transcribed to cDNA (Step 2). This "target" cDNA is then coupled to a fluorescent dye (Step 3). The target cDNA is then hybridized with a large number of probes of immobilized DNA (steps 4 and 5) which had been synthesized and fixed to different locations of the DNA chip during fabrication. The cDNA from the samples binds to their corresponding probes on the chip (Step 5). After cleaning, the chip is scanned with a confocal microscope and the strength of the fluorescent light is recorded (Step 6). Genes which are predominantly expressed in the sample give rise to bright spots of strong fluorescent light. No expression is indicated by weak fluorescent light. After segmentation of the stained locations on the chip and a correction for background intensity, intensity values are transformed to real numbers for every location (Step 7). After processing, the data from several experiments with different samples are collected and represented in matrix form, where columns correspond to tissue samples, rows correspond to genes, and matrix entries describe the result of a measurement of how strong a particular gene was expressed in a particular sample.

Microarray Noise

Expression values as measured by the DNA microarray technique are noisy. Firstly, there exists biological noise, because samples do not show the same "expression state" and exactly the same levels of mRNA even if they belong to the same class or the same experimental condition. Then there is noise introduced by the microarray measurement technique. Sources of noise include tolerances in chip properties which originate from the fabrication process, different efficiencies for the mRNA extraction and the reverse transcription process, variations in background intensities, nonuniform labeling of the cDNA targets (the dye may bind multiple times and with different efficiencies), variations in the dye concentration during labeling, pipette errors, temperature fluctuations and variations in the efficiency of hybridization, and scanner deviations. The effect of measurement noise can be reduced by averaging over multiple measurements using the same sample but usually remains large. Measurement noise is not always Gaussian. Hartemink et al. (2001) for example found that the measurement noise distribution of the logarithmic expression values has heavy tails.

1.1.2 Gene Selection for Microarrays

Gene selection aims at three goals:

Why Gene/Feature Selection?

- (a) data preprocessing in order to improve the prediction quality of machine learning approaches,
- (b) identification of indicator genes (this would aid the interpretation and understanding of the data), and
- (c) reducing costs, if microarray data are used for example for diagnostic purposes.

Item (a) is an important issue in machine learning if the input dimension is larger than the number of samples. Kohavi and John (1997) for example found that decision tree approaches like ID3 (Quinlan, 1986), CART (Breiman et al., 1984), and C4.5 (Quinlan, 1993), as well as instance-based (for example K-nearest neighbor) methods, degrade in performance when the number of features is larger than a minimal set of relevant features. The Naive-Bayes method is reported to be more robust to irrelevant features but the prediction accuracy decreases if correlated features are present. Also Kittler (1986) observed decreasing performance of machine learning methods for large feature sets.

Curse of Dimensionality

The reduction in performance for data sets with many attributes is known as the "curse of dimensionality" (Bellman, 1961). According to Stone (1980), the number of training examples has to increase exponentially with the number of dimensions in order to ensure that an estimator also performs well for higher dimensional data. Otherwise overfitting (high variance in model selection) occurs, that is fitting of the selected model to noise in the training data. On the other hand, if the model class is chosen to be smooth so that the variance of model selection is restricted (low overfitting), then underfitting (high bias of model selection) occurs, that is the training data is not approximated well enough. The later is shown by Friedman (1997) who demonstrated for K-nearest neighbor classifiers that the curse of dimensionality leads to large bias. Practical applications confirm the theory: many input dimensions lead to poor generalization performance. Fewer features on the other hand should improve generalization for equal training error.

Many, Noisy Features with Microarray Data

For microarray data the situation is especially difficult because the number of features (genes) is often more than 10 times larger than the number of examples (tissue samples). The high level of noise additionally complicates the selection of relevant genes of microarray data. Both facts, the large number of genes and the presence of noise, led **?**) to state that "the features selected matter more than the classifier used" for DNA microarray data classification, a fact which will be confirmed by our analysis later on.

Microarray Data Requires Gene Selection

Item (b) refers to the identification of genes whose expression values change with the sample class. Genes which show different expression values in a control condition when compared to the condition to analyze are useful to differentiate between these conditions and should be extracted (see Jäger et al. (2003)). The knowledge of the relevant genes can then be exploited in two ways. Firstly, cellular mechanisms can be understood and active pathways may be identified. Secondly, target genes or target proteins for causing or avoiding conditions can be detected. In medical applications both kinds of information are highly relevant for diagnosis and drug design. Note, however, that the selection of genes for prediction and the selection of genes whose expression levels are correlated lead to different sets. Redundant sets of genes, which are the outcome of the latter task, may lead to a reduced performance of the former task. On the other hand, genes selected for the purpose of prediction may not include genes strongly correlated to each other in order to keep the number of features small.

Identify Genes

Item (c) refers to the costs of large scale microarray studies, for example, for diagnostic purposes. Small gene ensembles lead to cheaper chips (fewer probes on a chip), to savings in manpower (fewer experiments), and to easier interpretable experiments (Jäger et al., 2003).

Reducing Costs

1.1.3 Feature Selection Methods to Extract Relevant Genes

In the previous subsection we stated three important reasons why it is necessary to reduce the number of genes, that is to reduce the number of a sample's expression values obtained by the microarray technique. These expression values are considered as features of the sample in the field of machine learning. In order to reduce the number of genes and to select the most important genes, machine learning methods, called "feature selection" methods, must be applied. Because of the special structure of the microarray data, namely the large number of noisy features, not all previously proposed feature selection methods are suited for the analysis of microarray data. Feature selection methods should be able to cope with many features but few samples, to remove redundancies, and to consider dependencies of whole subsets of features.

To address both the suitability for and the performance on microarray data, this chapter consists of two parts. In the first, methodological part we review previous approaches and then derive a feature selection method, which is particularly suited to the peculiarities of microarray data. In the second, application part we assess the algorithms' performance and provide benchmark results. More precise, this chapter is organized as follows. In Section 1.2 we review feature selection methods with respect to their ability to address the particular constraints of microarray data. Then we introduce the new feature selection method in Section 1.3. In Section 1.4 we describe a "gene selection protocol" which is then evaluated together with the new feature selection method in Section 1.5. Benchmark results on microarray data are provided using several previously described approaches.

1.2 Review of Feature Selection Methods

For simplicity let us consider a classification task where the objects to classify are described by vectors with a fixed number of components (the features). The training set consists of vectors which are labeled by whether the according object belongs to a class or not and — again for reasons of simplicity — we assume that there are only two classes. Given the training data, a classifier should be selected which assigns correct class labels to the feature vectors. The goal of machine learning methods is not only to select a classifier which performs well on the training set, but which also correctly classifies new examples, that is which correctly predicts events of the future.

Feature Selection vs. Feature Construction

There are two classes of preprocessing methods which are commonly used to improve machine learning techniques: *feature selection* and *feature construction* methods. Feature construction methods compute new features as a combination of the original ones and are often used for dimensionality reduction. Many popular methods for feature construction are based on linear combinations of the original features, that is on projections of data points into low dimensional spaces, like *projection pursuit* (for example Friedman and Tukey (1974); Friedman and Stuetzle (1981); Huber (1985)), *principal component analysis* (PCA, for example Oja (1982); Jolliffe (1986); Jackson (1991)), or *independent component analysis* (ICA, for example Cardoso and Souloumiac (1993); Jutten and Herault (1991); Bell and Sejnowski (1995); Hochreiter and Schmidhuber (1999); Hyvärinen et al. (2001)). More recently nonlinear feature construction algorithms based on kernel methods (Cristianini et al., 2002) and the information bottleneck idea (Tishby et al., 1999; Tishby, 2001) have been proposed.

Feature selection methods, on the other hand, choose a subset of the input components which are supposed to be relevant for solving a task and leave it to a subsequent stage of processing to combine their values in a proper way[1]. In the following we focus on feature selection, that is on the task of choosing a subset of "informative" input components, that is components which are relevant for predicting the class labels. The classifier is then selected using the reduced feature vectors as the objects' description. Therefore, only feature selection techniques address items (b) and (c) from the previous section, that is the extraction of

[1] This combination can also be done during feature selection.

indicator genes and reducing costs[2].

Overview Feature Selection

Review articles on feature selection have been published in a "special issue on relevance" of the journal *Artificial Intelligence* (Kohavi and John, 1997; Blum and Langley, 1997) and a "special issue on variable and feature selection" of the *Journal of Machine Learning Research* (Guyon and Elisseeff, 2003) to which we refer the reader for more details. The book of Liu and Motoda (1998) also gives an overview on feature selection.

1.2.1 Feature Selection Using Class Attributes

This subsection gives a general overview on feature selection techniques which have been used for gene selection on microarray data, whereas the next subsection focuses on the more recently developed kernel based methods. In both sections we consider feature selection techniques which extract features according to their dependencies with the sample classes, hence we assume that the class labels are available for the training set. Methods which exploit the additional information given by the class labels are in general superior to methods not using this information: features which can be removed without changing the conditional probability of class labels with respect to all features are irrelevant for classification. Without explicit class labels feature selection must be based on the distribution of feature values, for example on entropy or saliency measures. Those methods are not considered here.

Feature Ranking and Subset Selection

Feature selection methods perform either feature ranking or subset selection. In feature ranking an importance value is assigned to every feature while subset selection attempts at constructing an optimal subset of features. While some feature ranking methods do not consider dependencies between features, subset selection methods usually do and may even include features which have low correlations with the class label if justified by a good classification performance. The latter usually happens if dependencies between features (and not between class label and a certain feature) are important for prediction. In those cases the selection of interacting features is important, but it is also difficult to achieve (see Turney (1993a,b)).

Feature selection methods fall into one of two categories (Langley (1994); Kohavi and John (1997); John et al. (1994); Das (2001); Liu and Motoda (1998); **?**)):

Filter vs. Wrapper Methods

- (a) *filter methods* or
- (b) *wrapper methods.*

Filter Methods

Ad. (a) Filter methods: filter methods extract features whose values show dependencies with class labels without explicitly relying on a predictor (classifier).

One example are statistical methods which compute the statistical dependencies between class labels and features and select features where the dependencies are

[2] Note that item (b) may not be fully addressed by feature selection approaches because redundant features are avoided and not all indicator genes are extracted. However, the missing genes can be extracted by correlation analysis in a subsequent step.

Ranking Methods, Statistics and Information Theory

strong. The calculation of dependencies is based on Pearson's correlation coefficient, Wilcoxon statistics, t-statistics, Fisher's criterion or signal-to-noise ratios (see Hastie et al. (2001); **?**); Furey et al. (2000); Tusher et al. (2001)). Statistical methods are fast and robust, but assume certain data or class distributions and cannot recognize dependencies between features. In principle, statistical methods can serve as subset selection methods if the dependency between a whole feature subset and the class label is computed. For example, the mutual information between feature sets and class labels has been considered in (Koller and Sahami, 1996). However the number of possible subsets increases exponentially with the number of features which makes these approaches unattractive. Therefore, the method in (Koller and Sahami, 1996) is only tractable if approximations are made.

The "relief" methods (Kira and Rendell (1992); Rendell and Kira (1992); Kononenko (1994); Robnik-Sikonja and Kononenko (1997)) are another approach which assign relevance values to features. Values are assigned according to the average separation of data vectors belonging to different classes minus the average separation of data points belonging to the same class. The averages are computed by randomly selecting a data point and determining its nearest data points from the same class and the opposite class. The "relief" methods are fast, can detect feature dependencies but — again — do not remove redundant features.

Subset Selection Methods

Combinatorial search procedures are able to remove redundant features from the selected set. These methods exhaustively test all feature subsets for their ability to separate the classes, that is whether two training vectors have the same values on the selected feature subset but different class labels. After testing, the minimal subset necessary to predict the class label is chosen (for example FOCUS (Almuallim and Dietterich, 1991) or the probabilistic approach in (**?**)). Combinatorial search methods, however, suffer from high computational costs and can only be applied to a small number of features. They are prone to overfitting through noise but on the other hand they will find the best solution in the noiseless case. Another feature subset selection which — like FOCUS — searches for a minimal necessary feature subset to separate the classes is based on decision trees (Cardie, 1993). The decision tree is used for separating the classes but not as a classifier. This method, however, is not applicable for small training sets because only $\log_2 m$ features are selected if m training examples are available. Since the sample size for microarray data is usually below 100, only $\log_2 100 \approx 7$ genes are typically selected. These are too few genes.

Wrapper Methods

Ad. (b) Wrapper methods: wrapper methods (see Kohavi and John (1997); John et al. (1994)) use a classifier as the objective function for the evaluation of a subset of features. The classifier is obtained through a model selection (training) method which minimizes the classification error on the training data. The classifier is then used to compute the prediction error on a validation set. Typical classifiers are decision trees, for example ID3 (Quinlan, 1986), CART (Breiman et al., 1984), and C4.5 (Quinlan, 1993), or instance-based classifiers like K-nearest neighbor.

Well known wrapper methods are the nested subset methods which are based on greedy strategies like hill-climbing (for example SLASH Caruana and Freitag (1994) and the random mutation hillclimbing — random mutation of feature presence map — described in Skalak (1994)). Nested subset methods perform either "forward selection" (Cover and Campenhout (1977)) or "backward elimination" (Marill and Green (1963)). Forward selection works in the underfitting regime. It starts from an empty set of features and adds features step by step which lead to the largest reduction of the generalization error. Backward elimination, on the other hand, works in the overfitting regime. It starts with the set of all features and removes unimportant features step by step in order to maximally reduce the generalization error. The major shortcoming of these methods is that they do not consider all possible combinations of features (Cover and Campenhout (1977)). If, for example, only the XOR of two features is important, these features would not be recognized by a forward selection procedure which adds only a single feature at a time. The backward selection procedure suffers from a similar problem. Assume that one feature conveys the information of two other features and vice versa. The best strategy would be to remove these two features to obtain a minimal set but backward selection may keep these two features and remove the single one. Another problem of backward selection is to determine good candidate features for deletion because overfitting makes it hard to distinguish between label noise fitting and true dependencies with class labels.

Forward vs. Backward Selection

Hill-Climbing and Search Methods

Other search strategies are computationally more expensive but explore more possible feature sets. Such search methods include beam and bidirectional search (Siedlecki and Sklansky, 1988), best-first search (Xu et al., 1989), and genetic algorithms (Vafaie and Jong, 1993, 1992; Bala et al., 1995).

1.2.2 Kernel Based Methods

Recently kernel based feature selection methods which use the support vector machine (SVM) approach have shown good performance for feature selection tasks (see for example the review in Guyon and Elisseeff (2003)). Kernel based feature selection methods are emphasized through this subsection since they are especially suited for microarray data due to the fact that they have shown good results in high dimensional data spaces and have been successfully applied to noisy data. These methods use either one of two feature selection technique, which were already known in the field of neural networks:

Feature Selection During or After Learning

- (a) feature selection by pruning irrelevant features after a classifier has been learned or
- (b) adding a regularization term to the training error which penalizes the use of uninformative features during learning a classification task.

Ad. (a) Feature selection after learning.

?) proposed a feature selection method for support vector learning of linear classifiers where the features with the smallest squared weight values are pruned after learning is complete. This method is a special case of the "Optimal Brain Surgeon" (OBS, Hassibi and Stork (1993)) or "Optimal Brain Damage" (LeCun et al., 1990) techniques for dependent (in case of OBS) or independent (in case of OBD) feature values under the assumption that feature values have variance one. OBS is based on a Taylor expansion of the mean squared error around its minimum, and the increase in training for a pruned feature is estimated by the Hessian[3]. Intuitively, the support vector method corresponds to projecting the normal vector of the separating hyperplane into the subspace perpendicular to the less important directions. The features for which these values are lowest are then deleted. **?**) also describe an iterative version of this feature selection procedure where the feature with the smallest absolute weight is removed after each SVM optimization step. This method is then called "Recursive Feature Elimination" (RFE) and is an example of backward elimination of features. It has recently been extended by Rakotomamonjy (2003) for nonlinear kernels. Note, however, that these methods which prune features after learning cannot detect redundant features and that they are sensitive to outliers.

Recursive Feature Elimination (RFE)

Ad. (b) Feature selection during learning.

1-Norm SVMs

0-Norm SVMs

Regularization techniques have been proposed for support vector machines in order to improve prediction performance by selecting relevant features. The first set of techniques directly favors SVMs with sparse weight vectors. This can be done by using the 1-norm in the SVM objective function, a technique, which is known as the linear programming (LP) machine (Schölkopf and Smola, 2002; Smola et al., 1999; Frieß and Harrison, 1998). This approach leads to many zero components of the weight vector and to the removal of the corresponding features. In (Bradley and Mangasarian, 1998; Bi et al., 2003) these methods are utilized together with backward elimination. In (Bradley and Mangasarian, 1998) the 0-norm of the weight vector is considered as an objective to select a classifier. The 0-norm counts the non-zero components of the weight vector which leads to a discrete and NP-hard optimization problem. Approximations can be made but they are sensitive to the choice of parameters (see Weston et al. (2003)) and the optimization is still computationally complex in high dimensions. Weston et al. (2003) propose an improved approximation of the 0-norm, which reduces to a method which iteratively solves 1-norm SVMs and adjusts scaling factors for the different features. In (Perkins et al., 2003) both the 0-norm and the 1- or 2-norm are used for feature selection, where the 1- or 2-norm serves for regularization and the 0-norm selects features.

R2W2

The second set of techniques is based on the proper choice of scaling factors for the different features. Weston et al. (2000) applies scaling factors to the 2-norm SVM approach ("R2W2"). Two phases are performed iteratively. First the SVM is optimized and a bound for the generalization error is computed. Secondly, the scaling factors are determined by a gradient descent method minimizing the bound.

[3] For a linear classifier the Hessian of the mean squared error is equal to the estimated covariance matrix.

This method has the advantage, that it can be extended to nonlinear kernels, where the scaling factors are put into the kernel function. On the other hand, this method is computationally expensive because two optimization problems (SVM solution and error bound) have to be solved for every iteration and the kernel matrix must be evaluated for every step. Additionally, the gradient based optimization suffers from convergence to local optima.

Statistical methods are so far the most common choice for selecting relevant genes from microarray data (for example Pomeroy et al. (2002)). However, support vector machine based methods have recently been applied with good success (Shipp et al., 2002).

1.3 The Potential Support Vector Machine for Feature Selection

In this section we introduce a new feature selection method which is based on the support vector machine technique (see also Hochreiter and Obermayer (2003a)). Feature selection and classification are performed simultaneously. The main differences to previous approaches are:

- *Sphering.* In order to judge the relevance of feature components, the variance should be normalized, that is the data should be sphered (whitened). Therefore, an objective is formulated according to which the classifier is selected by maximizing the margin after sphering. It turns out that sphering has two additional advantages for the SVM technique. Firstly, the derived new support vector machine approach is invariant to linear transformation of the data — as are the margin bounds. Secondly, tighter margin bounds can be obtained.
- *New constraints.* The constraints of the optimization problem are modified in order to ensure that the classifier is optimal with respect to the mean squared error between the classification function and the labels. In contrast to previous approaches where one constraint is associated with each of the m training examples, each constraint is now associated with one feature and the number of new constraints is equal to the number N of features.
- *Support features.* The combination of the new objective with the new constraints allows to assign support vector weights to features, and the normal vector of the classification boundary is expanded in terms of these weights rather than in terms of support vector data points. This allows feature selection according to whether a feature is a support vector or not. As a side effect the dual optimization problem can be efficiently solved using a technique similar to sequential minimal optimization Platt (1999).

In summary, a classifier is selected from the set of all classifiers with minimal mean squared error which yields the largest margin after sphering the data. The new support vector machine removes irrelevant features, that are features which lead to a minimal increase of the mean squared error when removed. More formally, feature selection is done by assigning support vector weights to features — the features which are support vectors are selected.

In the following subsections, we first briefly review the classical support vector machine (SVM). Then we introduce a new objective for achieving scale-invariant SVMs, present new constraints for correct classification, and combine the new objective and the new constraints into one framework. Finally, a summary of the new technique is given.

1.3.1 The Classical Support Vector Machine

Let us consider a set of m objects, which are described by feature vectors $\mathbf{x} \in \mathbb{R}^N$, and let us represent this data set by the matrix $\mathbf{X} := (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$. We furthermore assume that every object belongs to one of two classes, and that class membership is denoted by a binary label $y \in \{+1, -1\}$. The labels for the m objects are summarized by a label vector $\mathbf{y}$, where y_i is the label of $\mathbf{x}_i$.

The goal is to construct a linear classifier based on the feature vectors $\mathbf{x}$. In the standard support vector machine approach (see Chapter **??**) this classifier is defined by taking the sign of the classification function

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b , \tag{1.1}$$

where the weight vector $\mathbf{w}$ has been normalized such that the margin ρ, that is the distance between the classification boundary and the closest data point, is $\rho = \|\mathbf{w}\|^{-1}$.

Classical support vector machines construct a classification function which maximize the margin under the constraint that the training data is classified correctly:

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2} \|\mathbf{w}\|^2 \\ &\text{s.t.} \ \ y_i \ ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 \ . \end{aligned} \tag{1.2}$$

VC Dimension

Here we assume that the data $\mathbf{X}$ with label vector $\mathbf{y}$ is linearly separable, otherwise slack variables have to be used. If the number of training examples m is larger than the Vapnik-Chervonenkis (VC) dimension h (a capacity measure for classifiers, see for example **?**)), then one obtains the following bound on the generalization error $R(f)$ of f (also called "risk of f") (**?**Schölkopf and Smola, 2002):

$$R(f) \leq R_{\text{emp}}(f) + \sqrt{\frac{1}{m}\left(h\left(\ln\left(\frac{2\,m}{h}\right) + 1\right) - \left(\frac{\ln(\delta)}{4}\right)\right)} \tag{1.3}$$

Worst Case Bounds

which holds with probability $1 - \delta$. δ denotes the probability, that a training set $\mathbf{X}$ of size m has been randomly drawn from the underlying distribution, for which the bound eq. (1.3) does not hold. $R_{\text{emp}}(f)$ denotes the training error of f (also called the "empirical risk of f"). For the set of all linear classifiers defined on $\mathbf{X}$, for which $\rho \geq \rho_{\min}$ holds, one obtains

$$h \leq \min\left\{\left[\frac{\mathrm{R}^2}{\rho_{\min}^2}\right] , N\right\} + 1 \tag{1.4}$$

(see **?**Schölkopf and Smola (2002)), where $[\cdot]$ denotes the integer part and R is the radius of the smallest sphere in the data space, which contains all the training data. The fact that the bounds[4] become smaller for increasing ρ and decreasing N, motivates the maximum margin principle, eq. (1.2), as well as the concept of feature selection (minimizing N). Bounds on the *expected* generalization error can also be derived (cf. **?**Schölkopf and Smola (2002)). They also become smaller for increasing ρ and decreasing N.

1.3.2 A Scale Invariant Objective Function

Both the selection of a classifier using the maximum margin principle and the values obtained for the bounds on the generalization error described in the last section suffer from the problem that they are not invariant under linear transformations. This problem is illustrated in Fig. 1.2. The figure shows a two dimensional classification problem, where the data points from the two classes are indicated by triangles and circles. In the left figure, both classes are separated by the hyperplane with the largest margin (solid line). In the right figure, the same classification problem is shown, but scaled along the vertical axis by a factor s. Again, the solid line denotes the support vector solution, but when the classifier is scaled back to $s = 1$ (dashed line in the left figure) it does no longer coincide with the original SVM solution. Therefore, the optimal hyperplane is not invariant under scaling, hence predictions of class labels may change if the data is rescaled before learning. In the legend of Fig. 1.2 it is shown that the ratio $\frac{\mathrm{R}^2}{\rho^2}$, which bounds the VC dimension (see eq. 1.4) and determines an upper bound on the generalization error (see eq. 1.3) has also changed. If, however, the classifier depends on scale factors, the question arises, which scale factors should be used for classifier selection.

Here we suggest to scale the training data such that the margin ρ remains constant while the radius R of the sphere containing all training data becomes as small as possible. This scaling results in a new sphere with radius $\tilde{\mathrm{R}}$ which still contains all training data and which leads to a tight margin-based bound for the generalization error. Optimality is achieved when all directions orthogonal the normal vector $\mathbf{w}$ are scaled to zero and $\tilde{\mathrm{R}} = \min_{t \in \mathbb{R}} \max_i |(\hat{\mathbf{w}} \cdot \mathbf{x}_i) + t| \leq \max_i |(\hat{\mathbf{w}} \cdot \mathbf{x}_i)|$, where $\hat{\mathbf{w}} := \frac{\mathbf{w}}{\|\mathbf{w}\|}$. Note that with offset b of the classification function the sphere must not be centered at the origin (**?**). Unfortunately, above formulation does not lead to a handable optimization problem. Therefore, we suggest to minimize the upper bound:

$$\frac{\tilde{\mathrm{R}}^2}{\rho^2} = \tilde{\mathrm{R}}^2 \|\mathbf{w}\|^2 \leq \max_i (\mathbf{w} \cdot \mathbf{x}_i)^2 \leq \sum_i (\mathbf{w} \cdot \mathbf{x}_i)^2 = \left\|\mathbf{X}^\top \mathbf{w}\right\|^2 . \tag{1.5}$$

In (Hochreiter and Obermayer, 2003b) it is shown that replacing the objective

[4] Note that these bounds can be improved using the concepts of covering numbers and the fat shattering dimension (Shawe-Taylor et al., 1996, 1998; Schölkopf and Smola, 2002).

New Objective

function in eqs. (1.2) by the upper bound

$$\mathbf{w}^\top \mathbf{X}\, \mathbf{X}^\top \mathbf{w} \;=\; \left\| \mathbf{X}^\top \mathbf{w} \right\|^2 \;, \tag{1.6}$$

Improved Error Bound by Sphering

of eq. (1.5) on $\frac{\tilde{\mathrm{R}}^2}{\rho^2}$ corresponds to the integration of sphering (whitening) and SVM learning into one framework. The resulting classifier is called "sphered support vector machine" (S-SVM). Minimizing the new objective leads to normal vectors which tend to point in directions of low variance of the data. If the data has already been sphered, then the covariance matrix is given by $\mathbf{X}\,\mathbf{X}^\top \;=\; \mathbf{I}$ and we recover the classical SVM[5]. In (Hochreiter and Obermayer, 2003b) a new error bound based on covering numbers is derived according to the considerations above, which additionally motivates the new objective function eq. (1.6). There it is also shown, that the new objective is well defined for cases where $\mathbf{X}\;\mathbf{X}^\top$ or/and $\mathbf{X}^\top\;\mathbf{X}$ is singular.

Invariant Under Linear Transformations

The new objective leads to separating hyperplanes which are invariant to linear transformations of the data. Consequently, the bounds and the performance of the derived classifier no longer depend on scale factors. Note, that the kernel trick carries over to the S-SVM as shown in (Hochreiter and Obermayer, 2003b). The S-SVM can also be applied for kernels which are not positive definite, that is which are not Mercer kernels (Hochreiter and Obermayer, 2002).

1.3.3 New Constraints

To assign support vector weights to the feature components the m constraints enforcing correct classification have to be transformed into N constraints associated with the features. The idea of the transformation is to compute the correlation between the residual error and a feature component. If these correlations are zero, the empirical risk is minimal.

Residual Error

We define a residual error r_i for a data point $\mathbf{x}_i$ as the difference between its class label y_i and the value of the classification function f, $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$:

$$r_i = (\mathbf{w} \cdot \mathbf{x}_i) \;+\; b \;-\; y_i \quad . \tag{1.7}$$

For every feature component j we then compute the mixed moments σ_j,

$$\sigma_j \;=\; \frac{1}{m} \sum_{i=1}^{m} \; (\mathbf{x}_i)_j \;\; r_i \;, \tag{1.8}$$

between the residual error r_i and the measured values $(\mathbf{x}_i)_j$. These mixed moments σ_j should be made small (or zero). The rationale behind minimal values for σ_j is that — given quadratic loss functions — they lead to an optimal classifier. Consider the quadratic loss function

$$c(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) \;=\; \frac{1}{2} r_i^2 \tag{1.9}$$

[5] In general, however, sphering is not possible as a preprocessing step if a kernel is used.

and the empirical loss (the mean squared error)

$$R_{\mathrm{emp}}(f_{\mathbf{w}}) \;=\; \frac{1}{m}\sum_{i=1}^{m} c\left(\mathbf{x}_i, y_i, f(\mathbf{x}_i)\right) \quad . \tag{1.10}$$

The mixed moments σ_j are equal to the derivative of the empirical loss with respect to $(\mathbf{w})_j$:

$$\sigma_j \;=\; \frac{\partial R_{\mathrm{emp}}(f)}{\partial\,(\mathbf{w})_j} \quad . \tag{1.11}$$

Constraint Assures Minimal Empirical Risk

That is the empirical error is minimal if

$$\sigma_j \;=\; \frac{1}{m}\,\sum_i (\mathbf{x}_i)_j \;\, r_i \;=\; 0\;. \tag{1.12}$$

Note that there exists only one minimum since the squared error is a convex function in the parameters $\mathbf{w}$.
These considerations motivate a new set of constraints

$$\mathbf{X}\,\mathbf{r} \;=\; \mathbf{X}\;\left(\mathbf{X}^{\top}\,\mathbf{w} \;+\; b\mathbf{1} \;-\; \mathbf{y}\right) \;=\; \mathbf{0} \quad , \tag{1.13}$$

which an optimal classifier must fulfill, because

$$(\mathbf{X}\,\mathbf{r})_j \;=\; \sum_{i=1}^{m}\;(\mathbf{x}_i)_j\;\, r_i \;=\; m\,\sigma_j \;=\; m\,\frac{\partial R_{\mathrm{emp}}(f)}{\partial\,(\mathbf{w})_j}\;. \tag{1.14}$$

However, measurement noise may lead to high values of σ_j which — when minimized — would lead to strong overfitting. Therefore, we introduce a "correlation threshold" ϵ which separates the noise from the signal part, and we modify the constraints in eq. (1.13) according to

Correlation Threshold ϵ

$$\begin{aligned} &\mathbf{X}\;\left(\mathbf{X}^{\top}\,\mathbf{w}\;+b\mathbf{1}\;-\;\mathbf{y}\right)\;-\;\epsilon\,\mathbf{1}\;\leq\;\mathbf{0}\,, \\ &\mathbf{X}\;\left(\mathbf{X}^{\top}\,\mathbf{w}\;+b\mathbf{1}\;-\;\mathbf{y}\right)\;+\;\epsilon\,\mathbf{1}\;\geq\;\mathbf{0}\;\;. \end{aligned} \tag{1.15}$$

This formulation is analogous to the ϵ-insensitive loss (Schölkopf and Smola, 2002).

If measurements of some features have larger variance then others, a global (independent of the feature j) correlation threshold ϵ cannot distinguish between high σ_j values resulting from high correlation between the r_i and $(\mathbf{x}_i)_j$ and high σ_j values resulting from large variance of the values $(\mathbf{x}_i)_j$. A global, that is feature independent, ϵ would lead to an undesired preference of highly varying features even if the do not convey information about the class label. Therefore, the variance of the values $(\mathbf{x}_i)_j$ should be taken into account. For example, a more appropriate measure would be Pearson's correlation coefficient

$$\hat{\sigma}_j \;=\; \frac{\sum_{i=1}^{m}\;\left((\mathbf{x}_i)_j\;-\;\bar{x}_j\right)\;(r_i\;-\;r)}{\sqrt{\sum_{i=1}^{m}\;\left((\mathbf{x}_i)_j\;-\;\bar{x}_j\right)^2}\;\sqrt{\sum_{i=1}^{m}\;(r_i\;-\;r)^2}} \quad , \tag{1.16}$$

where $r \;=\; \frac{1}{m}\;\sum_{i=1}^{m}\; r_i$ is the mean residual and $\bar{x}_j \;=\; \frac{1}{m}\;\sum_{i=1}^{m}\;(\mathbf{x}_i)_j$ is the mean of the jth feature. In order to utilize the correlation coefficient $\hat{\sigma}_j$ for a global ϵ, we assume that the data vectors $\left((\mathbf{x}_1)_j, (\mathbf{x}_2)_j, \ldots, (\mathbf{x}_m)_j\right)$ are normalized to zero mean and unit variance:

$$\frac{1}{m}\sum_{i=1}^{m} \left((\mathbf{x}_i)_j \; - \; \bar{x}_j\right)^2 \; = \; 1 \;\; \text{and} \;\; \bar{x}_j \; = \; \frac{1}{m} \sum_{i=1}^{m} \; (\mathbf{x}_i)_j \; = \; 0 \; . \tag{1.17}$$

This normalization assumption is sufficient for a global ϵ because σ_j,

$$\sigma_j \; = \; \frac{1}{m}\sum_{i=1}^{m} \; (\mathbf{x}_i)_j \; r_i \;\; = \; \hat{\sigma}_j \; \frac{1}{\sqrt{m}} \; \|\mathbf{r} \; - \; r\mathbf{1}\| \;\; , \tag{1.18}$$

is linear in the correlation coefficient $\hat{\sigma}_j$ and otherwise independent of component j. If the noise is large, random correlations will occur more often, and the value of ϵ must be increased. If the strength of the measurement noise were known, the correct value of ϵ can be determined a priori. Otherwise, it takes the role of a hyperparameter and must be adapted using model selection techniques.

ϵ Bounds the Error Increase

Besides the important interpretation of ϵ as a noise parameter, there is a second interpretation in terms of bounding the increase of the residual error when a feature is removed. If we change $\mathbf{w}$ in the direction $\mathbf{e}_j$ by an amount of β, the new residual error r_i^{new} is

$$r_i^{\text{new}} \; = \; ((\mathbf{w} + \beta \; \mathbf{e}_j) \cdot \mathbf{x}_i) \; + \; b \; - \; y_i \;\; , \tag{1.19}$$

where $\mathbf{e}_j$ is the unit vector parallel to the jth feature axis. We obtain

$$\begin{aligned} \sum_i \left(r_i^{\text{new}}\right)^2 &= \sum_i \left(r_i^{\text{old}} \; + \; \beta\left(\mathbf{e}_j \cdot \mathbf{x}_i\right)\right)^2 \\ &= \sum_i \left(r_i^{\text{old}}\right)^2 \; + \; 2\,\beta \; \sum_i r_i^{\text{old}} \left(\mathbf{x}_i\right)_j \; + \; \sum_i \beta^2 \left(\mathbf{x}_i\right)_j^2 \\ &= \sum_i \left(r_i^{\text{old}}\right)^2 \; + \; 2\,\beta \;\; m\,\sigma_j \; + \; m\,\beta^2 \; . \end{aligned} \tag{1.20}$$

Because the constraints ensure that $|\sigma_j| \; m \; \leq \; \epsilon$, the increase on the residual error after the elimination the jth feature is bounded by

$$2 \; \epsilon \; |(\mathbf{w})_j| \; + \; m \; (\mathbf{w})_j^2 \; , \tag{1.21}$$

where we set $\beta = - \; (\mathbf{w})_j$.

1.3.4 The Potential Support Vector Machine

Now we combine both the new objective from equation (1.6) and the new constraints from equation (1.15) and call the new procedure of selecting a classifier the **Potential Support Vector Machine (P-SVM)**. As we will see, the combination of new objective and new constraints leads to an expansion of the normal vector of the classification boundary into a sparse set of features, hence allows to express the relevance of features via support vector weights. Combining eq. (1.6) and eqs. (1.15) we obtain

Potential Support Vector Machine (P-SVM): Primal

$$\begin{aligned} \min_{\mathbf{w},b} \ & \frac{1}{2} \|\mathbf{X}^\top \mathbf{w}\|^2 \\ \text{s.t.} \ & \mathbf{X} \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y}\right) + \epsilon \mathbf{1} \geq \mathbf{0} \\ & \mathbf{X} \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y}\right) - \epsilon \mathbf{1} \leq \mathbf{0} \ . \end{aligned} \tag{1.22}$$

Why ϵ?

If the row vectors of $\mathbf{X}$ are normalized to mean zero, then $\mathbf{X1} = \mathbf{0}$ and the term containing b vanishes. The parameter ϵ serves two important purposes:

- Large values of ϵ lead to a sparse expansion of the weight vector through the support features.
- If $\mathbf{X}\,\mathbf{X}^\top$ is singular and $\mathbf{w}$ is not uniquely determined, ϵ enforces a unique solution, which is characterized by the most sparse representation through features.

The interpretation of ϵ as a sparseness property is known from support vector regression (Schölkopf and Smola, 2002).

Potential Support Vector Machine (P-SVM): Dual

Optimization is usually performed using the Wolfe dual of eq. (1.22) given by (see Appendix 1.A)

$$\begin{aligned} \min_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-} \ & \frac{1}{2} \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right)^\top \mathbf{X}\,\mathbf{X}^\top \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) - \\ & \mathbf{y}^\top \mathbf{X}^\top \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) + \epsilon\, \mathbf{1}^\top \left(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-\right) \\ \text{s.t.} \ & \mathbf{1}^\top \mathbf{X}^\top \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) = 0 \ , \\ & \mathbf{0} \leq \boldsymbol{\alpha}^+ \ , \ \mathbf{0} \leq \boldsymbol{\alpha}^- \ , \end{aligned} \tag{1.23}$$

Fast Optimization by SMO

where the quantities $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-)$ denote the Lagrange parameters. $\mathbf{1}^\top \mathbf{X}^\top (\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) = 0$ is automatically satisfied, if the row vectors of $\mathbf{X}$ are normalized to zero mean. The dual problem is solved by a Sequential Minimal Optimization (SMO) technique, see (Platt, 1999). A fast solver for the P-SVM is described in (Hochreiter and Obermayer, 2003b), where advantage can be taken of the fact that the equality constraint vanishes for zero mean row vectors of $\mathbf{X}$. The SMO technique is important if the number of features is large because the optimization problem of the P-SVM is quadratic in the number of features rather than in the number of data points. In contrast to standard SVMs with a linear kernel it is the correlation matrix $\mathbf{X}\,\mathbf{X}^\top$ and **not** the Gram matrix $\mathbf{X}^\top \mathbf{X}$ which enters the SVM objective.

Finally, the classification function f has to be constructed using the optimal values of the Lagrange parameters $\boldsymbol{\alpha}$. In Appendix 1.A we show that

$$\mathbf{w} = \boldsymbol{\alpha} \ . \tag{1.24}$$

Computing b

In contrast to the standard SVM expansion of $\mathbf{w}$ by its support vectors $\mathbf{x}$, the weight vector $\mathbf{w}$ is now expanded into a sparse set of features components which serve as the support vectors in this case. The value for b can be computed from the condition that the average residual error r is equal to zero:

$$b = -\frac{1}{m} \sum_{i=1}^{m} \left((\mathbf{w} \cdot \mathbf{x}_i) - y_i \right) \ . \tag{1.25}$$

Note that b is chosen so that

$$\frac{\partial R_{\mathrm{emp}}(f)}{\partial b} \;=\; \frac{1}{m}\sum_{i} r_i \;=\; b \;+\; \frac{1}{m}\sum_{i}\left((\mathbf{w}\cdot\mathbf{x}_i) \;-\; y_i\right) \;=\; 0 \;\;. \tag{1.26}$$

That means b takes on its optimal value for minimization of the empirical risk (as was already ensured for $\mathbf{w}$ through the constraints). If the row vectors of $\mathbf{X}$ are normalized to zero, we obtain

$$b \;=\; \frac{1}{m}\sum_{i=1}^{m} y_i \;. \tag{1.27}$$

The classification function is then given by

$$f(\mathbf{u}) \;=\; (\mathbf{w}\cdot\mathbf{u}) \;+\; b \;=\; \sum_{j=1}^{n}\alpha_j\;(\mathbf{u}\cdot\mathbf{e}_j) \;+\; b \;=\; \sum_{j=1}^{n}\alpha_j\;(\mathbf{u})_j + \;b\;. \tag{1.28}$$

The classifier based on eq. (1.28) depends on the weighting coefficients α_j and b, which were determined during optimization, and it depends on the measured values $(\mathbf{u})_j$ of the selected features for the object to be classified. The weighting coefficients α_j can be interpreted as class indicators, because they separate the features into features which are relevant for class 1 and class -1, according to the sign of $\alpha_j \;=\; \alpha_j^+ \;-\; \alpha_j^-$. If the value of ϵ is large enough during learning, the expansion eq. (1.24) of the weight vector contains only a few "most informative" features, hence most of the components of $\mathbf{w}$ are zero. The other features are discarded because of being too noisy or not conveying information about the class labels. Sparseness can be attributed to the term $\epsilon\;\mathbf{1}^\top\,(\boldsymbol{\alpha}^+ \;+\; \boldsymbol{\alpha}^-)$ (or $\epsilon\;\|\boldsymbol{\alpha}\|_1$) in the dual objective function eq. (1.23). For large enough values of ϵ, this term pushes all α_j towards zero except for the features most relevant for classification.

Class Indicators

Sparseness

Feature Selection

One may wonder, whether the P-SVM is similar to the 1-norm SVM because sparseness of $\mathbf{w}$ is enforced through $\epsilon\;\|\boldsymbol{\alpha}\|_1$. However in contrast to the 1-norm SVM, the P-SVM still contains a quadratic part which enforces a large Euclidean margin on the training data. The 1-norm term originates from the ϵ insensitive loss in the primal optimization problem. The P-SVM is in this sense similar to support vector regression (SVR) (Schölkopf and Smola, 2002), where the vector $\boldsymbol{\alpha}$ is also regularized by a quadratic and an 1-norm part. However, all terms of the P-SVM are different from the corresponding terms in SVR: the quadratic matrix, the linear term, and the constraints. In contrast to both the 1-norm SVM and the SVR the value ϵ which weights the sparseness has a noise interpretation (measurement noise) and ϵ can be used to bound the residual error if a feature component is deleted.

1.3.5 Summary of the P-SVM and Application to Microarray Data

The Potential Support Vector Machine is a method for selecting a classification function, where the classification boundary depends on a small number of "relevant" features. The method can be used for feature selection, and it can also be used for the subsequent prediction of class labels in a classification task. The optimal P-SVM classifier is a classifier with minimal empirical risk but with the largest margin after sphering the data. Feature selection can be interpreted as removing features from the optimal classifier but bounding the increase in mean squared error through the value ϵ.

The first observation is that optimization (that is the selection of the proper values for $\boldsymbol{\alpha}$ and b) only involves the measurement matrix $\mathbf{X}$ and the label vector $\mathbf{y}$. In order to apply the P-SVM method to the analysis of microarray data, we identify the objects with samples, the features with genes, and the matrix $\mathbf{X}$ with the matrix of expression values. Due to the term $\mathbf{X}\ \mathbf{X}^\top$ in the dual objective, the optimization problem is well defined for measurement matrices $\mathbf{X}$ of expression values. From a conceptual point of view, however, it is advantageous to interpret the matrix $\mathbf{X}$ of observations (here: expression values) itself as a dot product matrix whose values emerge as a result of the application of a measurement kernel. This view is taken in (Hochreiter and Obermayer, 2003a,b) and briefly summarized in Appendix 1.B.

No Kernel Only Measurements and Labels

The second observation is that an evaluation of the classifier for new samples i only involves the measurement of its expression values $(\mathbf{x}_i)_j$ for the selected "support" genes j. The number of "support genes" depends on the value of a noise parameter, the correlation threshold ϵ. If the value of ϵ is large during learning, only a few "most informative" genes are kept. The other genes are discarded because of being too noisy or not conveying information about the class labels.

Classification with Few "Support Genes"

The set of all genes for which the weighting coefficients α_j are non-zero (the set of support genes) is the selected feature set. The size of this set is controlled by the value of ϵ, and if the P-SVM is applied for feature selection, the value of ϵ should be large.

P-SVM Gene Selection

1.4 The Gene Selection Protocol

In this section we describe the protocol for extracting meaningful genes from a given set of expression values for the purpose of predicting labels of the sample classes. The protocol includes data preprocessing, the proper normalization of the expression values, the feature selection and ranking steps, and the final construction of the predictor. We use the protocol together with our feature selection procedure which was described in Section 1.3. The protocol, however, can also be applied for other feature selection or ranking methods.

Additional Information by Labels

Note that our feature selection method requires class labels which must be supplied together with the expression values of the microarray experiment. When this technique is applied to the classification of tumor samples in Subsection 1.5.2 we are provided with binary class labels, but real values which are associated with the different samples may also be used. For the following, however, we assume that the task is to select features for classification and that m labeled samples are given for training the classifier.

1.4.1 Description of the Protocol

Gene Selection Protocol

1. **Expression values vs. ratios.** Before data analysis starts it is necessary to choose an appropriate representation of the data. Common representations are based on the ratio $T_j = \frac{R_j}{G_j}$ of expression values between the value R_j (red) of a gene j in the sample to analyze and the value G_j (green) in the control sample, and the log ratio $L_j = \log_2(T_j)$. We, however, suggest to use the original expression values R_j because our experimental findings on different datasets (for example the GIST data from Allander et al. (2001)) showed increased classification performance when the original values were used.

Present Call

2. **Present call.** The present call is usually the first step in the analysis of microarray data. During this step genes are identified for which the confidence is high, that they are actually expressed in at least one of the samples. Genes for which this confidence is low are excluded from further processing in order to suppress noise.

For this purpose an error model has to be constructed for the expression values or their ratios (sometimes before, sometimes after averaging across multiple measurements of the same sample — see Tseng et al. (2001); Schuchhardt et al. (2000); Kerr et al. (2000); Hartemink et al. (2001)). This error model accounts for both measurement specific noise (for example background fluctuations), which affects all expression values in a similar way, and gene specific noise (for example the binding efficiency of the dye), which affects expression values for different genes in a different way. Using this error model one assigns a P-value, which gives the probability that the observed measurement is produced by noise, to every measurement of an expression level. If the P-value is smaller than a threshold q_1 (typical values are 5%, 2%, or 1%), the expression level is marked "reliable". If this happens for a minimum number q_2 (typical values range from 3 to 20) of samples, the corresponding gene is selected and a so-called present call has been made.

Normalization

3. **Normalization.** Before further processing, the expression values are normalized to mean zero and unit variance across all training samples and separately for every gene. Normalization accounts for the fact that expression values may differ by orders of magnitudes between genes and allows to assess the importance of genes also for genes with small expression values. Sometimes more advanced normalization techniques are used (Schuchhardt et al., 2000; Hill et al., 2001; Durbin et al., 2002; Yang et al., 2002; Huber et al., 2002).

Gene Ranking, Determining Hyperparameters and Number of Genes

4. **Gene ranking and gene selection.** Here we assume that a feature selection method has been chosen where the size of the set of selected genes is controlled by a hyperparameter which we call ϵ in the following. Although we propose to use the P-SVM method any other features selection method can be used with this gene selection protocol.

In this step we perform two loops: An "inner loop" and an "outer loop" (the leave-one-out loop). The inner loop serves two purposes. It ranks features if only a subset method like the P-SVM is available and it makes feature selection more robust against variations due to initial conditions of the selection method. The outer loop also serves also two purposes. It makes the selection robust against outlier samples and allows to determine the optimal number of selected genes together with the optimal values of hyperparameters for the later prediction of class labels. In order to do this, a predictor must be constructed. Here we suggest to use a ν-SVM where the value of ν is optimized by the outer loop. In order to implement the outer (leave-one-out) loop, m different sets of samples of size $m - 1$ are constructed by leaving out one sample for validation. For each of the m sets of reduced size, we perform gene selection and ranking using the following "inner loop".

Inner Loop

Inner loop. The subset selection method is applied multiple times to every reduced set of samples for different values of ϵ. For every set of samples multiple sets of genes of different size are obtained, one for every value of ϵ. If the value of ϵ is large, the number of selected genes is small and vice versa. The inner loop starts with values of ϵ which are fairly large in order to obtain few genes only. Gradually the value is reduced to obtain more genes per run. Genes obtained for the largest value of ϵ obtain the highest rank, the second highest rank is given to genes which additionally appear for the second largest value of ϵ, etc. The values of ϵ are constant across sample sets. The minimal value should be chosen, such that the number of extracted genes is approximately the total number m of samples. The maximal value should be chosen such that approximately five to ten genes are selected. The other values are distributed uniformly between these extreme values. In the numerical experiments in Section 1.5 a total of three to five different values were used.

Outer Loop

Outer loop. The results of the inner loops are then combined across the m different sets of samples. A final ranking of genes is obtained according to how often genes are selected in the m leave-one-out runs of the inner loop. If a gene is selected in many leave-one-out runs, it is ranked high, else it is ranked low. Genes which are selected equally often are ranked according to the average of their rank determined by the inner loops. The advantage of the leave-one-out procedure is that a high correlation between expression values and class labels induced by a single sample is scaled down if the according sample is removed. This makes the procedure more robust against outliers.

The outer loop is also used for selecting an optimal number of genes and other hyperparameters. For this purpose, ν-SVMs are trained on each of the m sets of samples for different values of the hyperparameter ν and the number F of high ranking genes (ranking is obtained by the inner loop). Then the average error is calculated on the left out samples. Since the leave-one-out error as a function of the number F of selected genes is noisy, the leave-one-out error for F is replaced by the average of the leave-one-out errors for F, $F + a$, and $F - a$. Then the values of the hyperparameter ν and the number of genes F which give rise to the lowest error are selected. This completes the feature selection procedure.

1.4.2 Comments on the Protocol and on Gene Selection

Corrections to the outer, leave-one-out loop. The samples which were removed from the data in the outer loop when constructing the m reduced subsets for the gene ranking should not be considered for the present call and for determining the normalization parameters. Both steps should be done individually for each of the m sets of sample, otherwise feature or hyperparameter selection may not be optimal.

Computational Costs. The feature selection protocol requires $m \times n_\epsilon$ feature selection runs, where n_ϵ is the number of different values of the ϵ parameter. However the computational effort is justified by the increased robustness against correlation by chance (see next item) and the elimination of single sample correlations.

Correlation by Chance

Correlations by chance. "Correlations by chance" refers to the fact, that noise induced spurious correlations between genes and class labels may appear for a small sample size if the level of noise is high. If the number of selected genes is small compared to the total number of probes (genes) on the chip, spurious correlations may become a serious problem. Monte-Carlo simulations of **?**) on randomly chosen expression values for a data set of 78 samples and 5000 genes resulted in 36 "genes" which had noise induced correlation coefficients larger than 0.3. In order to avoid large negative effects of abovementioned spurious correlations the number of selected genes should not be too small, and one should extract a few tens of genes rather than a few genes only to decrease the influence of single spurious correlated genes. The random correlation effect can also be reduced, by increasing q_2, the minimum number of "reliable" expression values for making a present call. This avoids the selection of genes for which too few samples contribute to the correlation measure. However as explained in the next paragraph too many genes should be avoided as well.

Many Genes are Better than Few Genes

Redundancy

Redundancy. Redundant sets of genes, that is sets of genes with correlated expression patterns should be avoided in order to obtain good machine learning results (Jäger et al., 2003). Selection of too many genes with redundant information may lead to low generalization performance (cf. Section 1.1). The P-SVM described in Section 1.3 extracts a sparse set of genes hence reduces redundancy. Another reason for avoiding redundancy is that not all causes which imply the conditions may be recognized. This may happen if the set has to be kept small while redundant genes are included (redundant genes indicate the same cause, see experiments in Section 1.5.1). Reducing redundancy does not preclude the extraction of coexpressed clusters of genes: corregulated genes can be extracted in a subsequent processing step, for example based on classical statistical analysis.

Finally, one may wonder why redundant information does not help to decrease the noise level of otherwise informative genes. Empirically one finds that nonredundant feature selection methods (P-SVM and R2W2) outperform feature selection methods which include redundant genes (Fisher correlation and RFE), see Section 1.5.1. It seems as if the detrimental effects of a larger number of features are stronger.

1.4.3 Classification of Samples

In order to construct a predictor for the class labels of new samples a classifier is trained on all the m samples using the optimal set of genes and the optimal value of the hyperparameter (here: ν, cf. Step 4). The generalization performance of the classifier can again be estimated using a cross-validation procedure. This procedure must involve performing the full gene selection procedure including all preprocessing steps (for example normalization and feature selection) separately on all m cross-validation subsets. Otherwise a bias is introduced in the estimate. Note that this also requires to perform the "inner loop" of Step 4 on sets of $m-2$ samples.

Before the classifier is applied to new data, the expression values for the new sample must be scaled according to the parameters derived from the training set. As a consequence we may observe expression values which are larger than the ones which occur in the training data. We set the expression values exceeding the maximal value in the training set to this maximal value. With this procedure we may underestimate certain expression levels but the robustness against unexpected deviations from the training data is increased.

1.5 Experiments

1.5.1 Toy Data

We compare different methods on data analogous to, but more difficult than the ones used in (Weston et al., 2000). Compared to the data in (Weston et al., 2000), the number of features is 10 times larger, the ratio of the number of relevant features to the number of all features is smaller, and the noise in the data (measured through misclassification rate on data vectors where all irrelevant features are set to zero) is larger, too. The methods which are chosen for comparison are Fisher score, Recursive Feature Elimination (RFE) (**?**), R2W2 according to (Weston et al., 2000), and the P-SVM. The benchmark methods have all been successfully applied to microarray data: Fisher score in (Pomeroy et al., 2002; **?**), RFE in (**?**), and R2W2 in (Shipp et al., 2002).

For the toy experiments we simulate microarray data by assuming two or more causes ("modes") for the class labels. Each mode is characterized by a few indicators which means that a cause is reflected by an expression pattern across a few genes, for example, genes belonging to the same pathway. Most features have no dependencies with the label. All features are very noisy, and only a few samples are given, because microarray data typically suffer from small sample sizes.

1.5.1.1 Weston Data 1

We randomly chose 600 data points (samples) with probabilities 0.5 from class 1 ($y_i = 1$) and 0.5 from class 2 ($y_i = -1$). 100 data points are available for feature selection and training and the remaining 500 data points are used for testing the classifier. Next, we generated the 2000 attributes which simulate expression values of 2000 genes. Each data point was generated according to one of two modes to simulate two class determining causes for every sample.

Mode 1 was chosen with probability 0.7 and mode 2 with probability 0.3. In mode 1 the first 10 features indicate mode 1 and are generated according $(\mathbf{x}_i)_l \sim y_i \cdot \mathrm{N}(l, 10)$, $1 \leq l \leq 10$, where $\mathrm{N}(\mu, \sigma)$ denotes a normal distribution with mean μ and standard deviation σ. The features from 11 to 20 are chosen according to $(\mathbf{x}_i)_l \sim \mathrm{N}(0, 10)$, $11 \leq l \leq 20$. In mode 2 the features from 11 to 20 indicate mode 2 and are generated according to $(\mathbf{x}_i)_l \sim y_i \cdot \mathrm{N}(l - 10, 10)$, $11 \leq l \leq 20$. The first 10 features are drawn from $(\mathbf{x}_i)_l \sim \mathrm{N}(0, 10)$, $1 \leq l \leq 10$. The remaining 1980 features are chosen according to $(\mathbf{x}_i)_l \sim \mathrm{N}(0, 20)$, $21 \leq l \leq 2000$, for both modes.

Table 1.1 Classification performance for the Weston Data 1 data set. The values are the fractions of misclassification averaged over 10 runs on different test set for classifiers trained on the selected features. The original data has 2000 features of which only 20 are relevant. Features 1 to 10 were class indicators in 70 % and features 11 to 20 in 30 % of the data points. The table shows the results using the top ranked 5, 10, 15, 20, and 30 features. The methods are Fisher score, Recursive Feature Elimination (RFE) according to (**?**), R2W2 according to (Weston et al., 2000), and the P-SVM. The new P-SVM, performed best in all cases.

features	5	10	15	20	30
Fisher	0.21	0.23	0.26	0.26	0.28
RFE	0.26	0.28	0.28	0.28	0.27
R2W2	0.23	0.24	0.24	0.23	0.24
P-SVM	0.21	0.20	0.22	0.22	0.23

Table 1.1 shows the fraction of misclassification on the test set which is averaged over 10 different runs with different training and test sets. Features were selected using the Fisher score, Recursive Feature Elimination (RFE) (**?**), R2W2 (Weston et al., 2000), and the P-SVM method. Numbers are reported for classifiers using the 5, 10, 15, 20, and 30 top–ranked features. With these features we trained a classical C-SVM on the training set and validated the performance on a test set with 500 data points. The hyperparameter C was selected through 5-fold cross-validation on the training set from the set $\{0.01, 0.1, 1, 10, 100\}$ (0.1 was chosen in most cases) for all methods. To ensure a fair comparison of the methods the hyperparameter C was not determined in the outer loop of our protocol for the P-SVM. Because we neither imposed label noise nor generated extreme outliers in our data, the SVM was not sensitive to the hyperparameter choice and the C-SVM performed as well as the ν-SVM which we proposed in the protocol.

The success of feature selection depends on how many noisy, irrelevant features are wrongly selected and whether all modes which influence classification performance are represented sufficiently well. The result of a C-SVM for using all features is 0.37 (no feature selection) and for using the relevant 20 features (the perfect selection) is 0.11. The P-SVM shows the best results in all cases (see Table 1.1).

1.5.1.2 Weston Data 2

Here we extend the number of modes to five to make the task more difficult. Each mode was chosen with equal probability 0.2. For every mode $r,\ 1 \leq r \leq 5$, we draw the values for the 4 associated features. We first choose a mode r then we draw the feature values for the 4 associated features according to $(\mathbf{x}_i)_l \sim y_i \cdot \mathrm{N}\,(2, 0.5\ \tau)$, where $1 \leq \tau \leq 4$ and $l = (r-1) \cdot 5 + \tau$. The remaining features from 1 to 20 (that is excluding the features associated with r) are chosen according to $(\mathbf{x}_i)_l \sim \mathrm{N}\,(0, 1)$. The remaining 1980 features are for all modes always chosen according to $(\mathbf{x}_i)_l \sim \mathrm{N}\,(0, 20)\,,\ 21 \leq l \leq 2000$. All other parameters and the hyperparameter selection scheme were similar to the previous experiments (Weston Data 1). This data set is more challenging because there is a high chance to miss indicators for one mode, especially if the set of selected features is small. Missing a mode leads to differences in performance.

Table 1.2 Classification performance for the Weston Data 2 data set. Parameters and procedures are described in the legend of Table 1.1.

features	5	10	15	20	30
Fisher	0.31	0.28	0.26	0.25	0.26
RFE	0.33	0.32	0.32	0.31	0.32
R2W2	0.29	0.28	0.28	0.27	0.27
P-SVM	0.28	0.23	0.24	0.24	0.26

Table 1.2 shows the fractions of misclassification on the test set averaged over 10 different combinations of training and test set. It is instructive to compare the results of Table 1.2 with values obtained for the 20 relevant features (perfect selection), which leads to a fractional error of 0.10, and for all 2000 features (no selection), which leads to a fractional error of 0.38. Feature selection improves the classification result but does not quite reach the performance of "perfect selection" case because not all relevant genes were selected. R2W2 with the weighing coefficients instead of selecting features has an error of 0.26, that means R2W2 in the non-selection mode is better than in the selection mode. P-SVM shows the best results in all cases.

In Table 1.3 we printed the numbers of the top 30 selected features for a typical single trial, listed according to their rank. P-SVM found 11, R2W2 9, Fisher statistic 10, and RFE 7 relevant features (numbers printed in boldface). All other features are spurious. All five modes were detected by P-SVM, R2W2, Fisher statistics, and RFE using the 10, 18, 15, and 23 most highly ranked features. P-SVM detected indicator genes corresponding to all five modes using the smallest features set from all the methods tested, which explains in part the better performance of classifiers based on the P-SVM feature set.

Table 1.3 Numbers of the top 30 selected features for a typical single trial, listed according to their rank. Relevant features are printed in boldface.

P-SVM:	**7**	837	**2**	**18**	1248	**5**	**6**	**12**	**20**	**14**
	1562	980	664	1110	**11**	1404	1822	668	525	**9**
	80	1205	997	1228	1331	289	1605	621	1277	1987
R2W2:	837	**2**	980	**7**	**20**	**11**	1277	**6**	45	**5**
	18	1822	**12**	621	398	664	289	**14**	1110	587
	1605	1833	1331	1248	1752	525	1060	1443	820	997
Fisher:	980	**7**	**5**	837	**6**	**18**	1562	**12**	**2**	837
	20	1248	**8**	1404	**14**	1110	**11**	1228	80	664
	1987	1275	1331	668	263	640	621	1954	1774	1605
RFE:	837	**7**	1987	1277	**2**	753	**20**	1110	1774	997
	219	1636	**12**	398	**6**	1472	536	820	**18**	314
	974	525	**14**	877	621	1516	540	654	1331	664

1.5.2 Microarray Data

1.5.2.1 Data Sets

In this subsection we apply the P-SVM to the DNA microarray data published in (Pomeroy et al., 2002),(Shipp et al., 2002), and (**?**).

1. *Brain tumor data set (Pomeroy et al., 2002).* In our first dataset embryonal tumors in the central nervous system are investigated. The response to a therapy for the malignant brain tumor medulloblastoma should be predicted. Patients have a highly variable response to therapy, which made it difficult for classical methods to predict the therapy outcome. A good machine based prognosis, however, is highly desirable. Negative prognoses may indicate the necessity of an alternative therapy while positive prognoses may lead to a therapy with reduced toxicity.
Data is provided (supplementary information to (Pomeroy et al., 2002)) for 60 samples of human tissue taken from patients with different brain tumors of the medulloblastoma kind before treatment. The patients were treated in a similar way by chemotherapy and radiation. The clinical follow-up was monitored and the samples were labeled according to treatment outcome. From the snap frozen tumor

samples RNA was isolated and hybridized to an array contained 7129 probes.
The data was generated by the Affimetrix software and numbers denote "perfect match minus mismatch". "Perfect match" probes are oligonucleotides which are the probes with highest hybridization efficiency (the identifying base sequence) for a cDNA and "mismatch" probes are oligonucleotides with a small difference to the perfect match probe (one base in the middle of the identifying base sequence changed). Therefore, the specialization and the efficiency of the probe can be normalized by subtracting the expression value of mismatch probe from the probe's expression value. For more details see (Pomeroy et al., 2002).

2. *Lymphoma data set (Shipp et al., 2002).* Lymphoma tumors (diffuse large B-cell lymphoma DLBCL) show a positive response to therapy in less than 50 % of the cases. Previous methods were not sufficient to reliably predict treatment outcome, hence new approaches are necessary. Good predictions would allow to identify high risk patients in order to observe them closer and to control them more intensively and would certainly improve existing treatments.
Samples and clinical follow-ups for 58 DLBCL patients are available. For all patients the chemotherapy is equal and the labels denote the treatment outcome: positive or negative. From the snap frozen tumor samples RNA was isolated and hybridized to an array containing 7129 probes resulting in an 58×7129 matrix of "perfect match minus mismatch". For more details see (Shipp et al., 2002).

3. *Breast cancer data set (**?**).* 70 – 80 % of breast cancer patients receiving chemotherapy or hormonal therapy would have survived without treatment (**?**), because metastasis appears only in 20 – 30 % of the cases. Because therapy has strong side–effects, it would be important to predict beforehand, whether a patient would benefit from a particular therapy or not. Therefore, tumor samples were analyzed using the DNA microarray technique in order to search for gene expression patterns indicating the development of metastasis and the need for stronger medication. Clinical indicators however fail to predict the treatment outcome. The data set is a collection of 78 patients and expression values of 24481 genes. All patients were treated with modified radical mastectomy or breast-conserving treatment. The treatment outcome was monitored and the tumor samples were labeled according to whether the outcome was positive or negative. The data was given as log expression ratios — for more details see (**?**).

1.5.2.2 Common Setting in All Following Experiments

We normalized the rows of the data matrix $\mathbf{X}$ to mean zero and variance one. In Step 4 of our protocol we used $a = 5$ for the first and third, and $a = 3$ for the second experiment for estimating the optimal number of features because the number of features was smaller in the second experiment.

To classify the tissue samples after selecting the relevant genes, we applied a *linear* ν-SVM (Schölkopf and Smola, 2002). We found that the choice of the classifier does not matter much (also C-SVM and K-nearest neighbor worked) but the ν-SVM was the most robust against variations in the hyperparameter ν which allowed a simpler optimization scheme in the outer loop of the protocol: ν was chosen from the set $\{0.2, 0.3, 0.4, 0.5\}$. For all ν-SVM classifiers we fixed the threshold value b to zero, because initial experiments showed that these reduced sets of classifiers led to better generalization performance compared to the full set ($b \neq 0$). Table 1.4 summarizes the parameters used in the experiments.

Table 1.4 Summary of parameter values used in the numerical experiments. TOP TABLE: the first column ("data set") gives the number of the data set (1. brain tumor, 2. lymphoma, and 3. breast cancer). The second column ("samples") reports the number of tumor samples (patients). The third column ("genes") gives the number of probes (genes) in the original data. The fourth column ("n_ϵ") shows the number n_ϵ of ϵ runs in our protocol. The fifth column ("ϵ") lists the ϵ values for the runs. BOTTOM TABLE: the second column ("step 2") tells whether step 2 of our protocol is performed, that is whether the P-values were available. The third column ("q_1") shows the P-value threshold for step 1 of our protocol. The fourth column ("q_2") gives the minimal number of samples which must possess a P-value below the threshold in step 1 of our protocol to select the gene. The fifth column in the second part ("N") is the number of features after step 1 of our protocol. The sixth column in the second part ("a") lists the number of features which are subtracted and added to the actual feature number to build an average. The seventh column in the second part ("F") shows the average number of features used for classification.

data set	samples	genes	n_ϵ	ϵ
1	60	7129	3	0.25, 0.15, 0.05
2	58	7129	3	0.23, 0.13, 0.03
3	78	24481	4	0.1, 0.07, 0.03, 0.01

data set	step 2	q_1	q_2	N	a	F
1	no	–	–	7129	5	45
2	no	–	–	7129	3	18
3	yes	0.02	20	3623	5	30

1.5.2.3 Benchmark Methods

We compare the result of the P-SVM method in combination with the ν-SVM to results which have already been reported in the literature for (selection method / classification method): known important gene / one gene classification (Pomeroy et al., 2002), SPLASH / likelihood ratio classifier (Califano et al., 1999), signal-to-noise-statistics / K-nearest neighbor, signal-to-noise-statistics / weighted voting, Fisher-statistics / weighted voting, and R2W2. Except for the P-SVM results are taken from the corresponding literature. For the ν-SVM and the C-SVM (toy data) we used LIBSVM (**?**) whereas the P-SVM was implemented in C.

1.5.2.4 Results for the Brain Tumor Data Set

The data set from (Pomeroy et al., 2002) was processed according to the protocol from Section 1.4, except for Step 2 because of the missing P-values.

Table 1.5 Brain tumor data set: comparison of different approaches to prediction of therapy outcome based on the DNA microarray data (for explanations see text). The table shows the leave-one-out error given by the number of wrong classifications ("E") for a given number of selected features ("F"). For R2W2 "*" means that there is no "number of features" (R2W2 scales features and does not select features). For the P-SVM / ν-SVM the protocol determined $\nu = 0.4$. Features were selected using signal-to-noise-statistics ("statistics"), R2W2 (Weston et al., 2000), SPLASH (Califano et al., 1999), and P-SVM. Data with statistical feature selection are provided for "TrkC"-Gene classification, weighted voting, K-nearest neighbor (KNN), combined SVM/TrkC/KNN (Comb). For SPLASH the classifier is a likelihood ratio classifier ("LRC"). The ν-SVM is used as a classifier after feature selection with P-SVM. Except for our method (P-SVM / ν-SVM), results were taken from (Pomeroy et al., 2002).

Feature Selection / Classification	# F	# E
TrkC (one gene)	1	20
SPLASH / LRC		15
R2W2	*	15
statistics / weighted voting		14
statistics / KNN	8	13
Comb		12
P-SVM / ν-SVM	45	4

For the P-SVM method, the optimal number of features was 45 (average over the leave-one-out runs). Table 1.5 shows the number of misclassifications obtained from a leave-one-out cross-validation procedure. The P-SVM is compared with "TrkC"-Gene classification (one gene classification), R2W2, "weighted voting" classification (the sum of the features weighted by their correlation to class labels according to the signal-to-noise-statistics), K-nearest neighbor (KNN), and combined SVM/TrkC/KNN classifier. Feature selection was based on the correlation of features with classes using signal-to-noise-statistics (**?**) for KNN and weighted voting.

The other feature selection method is called SPLASH developed by Califano et al. (1999). SPLASH is a greedy subset selection method (wrapper method) and the subsequent classifier is a likelihood ratio classifier (“LRC”) based on density estimation for each gene. For the R2W2 SVM technique (Weston et al., 2000) see Section 1.2.2. The results show that the P-SVM method clearly outperforms standard methods — the number of misclassifications is down by a factor of 3.

1.5.2.5 Results for the Lymphoma Data Set

The data set from (Shipp et al., 2002) was processed according to the protocol from Section 1.4, except for Step 2 because of the missing P-values. For the P-SVM the optimal number of features was 18.

Table 1.6 Lymphoma data set: comparison of different approaches to prediction of therapy outcome based on the DNA microarray data (for explanations see text). The columns are as in Table 1.5. The outer loop of the protocol yielded $\nu = 0.5$ for the P-SVM / ν-SVM method. Feature selection is done by signal-to-noise-statistics (“statistic”), R2W2, and the P-SVM. The classifiers are K-nearest neighbor (KNN), weighted voting, and R2W2. Except for P-SVM, results were taken from (Shipp et al., 2002).

Feature Selection / Classification	# F	# E
statistic / KNN	8	16
statistic / weighted voting	13	14
R2W2	*	13
P-SVM / ν-SVM	18	12

Table 1.6 summarizes the results. The P-SVM is compared with weighted voting, K-nearest neighbor (KNN), and the R2W2 technique. The signal-to-noise-statistics was used to select feature for weighted voting and KNN. The new P-SVM selected more features than the selection methods taken from (Shipp et al., 2002). The increased number of features in this experiment was not too surprising because sometimes “many genes are better than few genes” in order to reduce the impact of “correlations by chance” (see Subsection 1.4.2). The P-SVM method yields comparable to slightly better results than the best methods from (Shipp et al., 2002).

1.5.2.6 Results for the Breast Cancer Data Set

Before further processing of the data set from (**?**) the log-ratios of the expression values were transformed according to

$$S_j = \mathrm{sgn}\ (L_j)\ 2^{|L_j|}\ = \begin{cases} \frac{R_j}{G_j} & \text{for } R_j\ \geq\ G_j \\ \\ -\ \frac{G_j}{R_j} & \text{otherwise .} \end{cases} \qquad (1.29)$$

This transformation was performed in order to scale up the ratios into magnitudes of the original R_j (see Step 1 of the protocol). Because P-values were given, Step 2 of our protocol was performed and we set the parameters q_1 and q_2 (see Table 1.4) to pick between 3000 and 8000 genes (however, we did not optimize these values). 3623 were selected after Step 2 for further processing. For the P-SVM the optimal number of features was 30.

In (**?**) the results for different classification threshold values are published in the supplementary information report. That allowed us to compare classifiers according to the Receiver Operating Characteristic (ROC) curve. The ROC curve consists of points whose x-components (distance to the left) denote the false positive rate (class 2 misclassification rate), that is wrongly positive classified negatives divided by the overall number of negatives. The y-components denote the true positive rate, that is correctly classified positives divided by the overall number of positives. Note that $(1 - y)$ (distance to the top) is the false negative rate (class 1 misclassification rate) and that $n\ x\ +\ p\ (1 - y)$ is the overall misclassification rate, where n is the fraction of class 2 (negative) examples and p the fraction of class 1 (positive) examples. For this experiment we observe $n = 0.44$ and $p = 0.56$, therefore, the overall misclassification rate is approximately $x + (1 - y)$. A high performance classifier has an ROC curve which is close to the left upper corner ($x = 0$ and $y = 1$ — no misclassification). The ROC curve gives more information on the quality of the classifier especially if it is required to work in different regimes, for example, under the requirements that class 1 or class 2 misclassifications should be below a given threshold. The ROC allows, for example, to optimize for

- (a) the selection of patients for adjuvant therapy where negative therapy outcome should not be misclassified, that is a small false positive rate required (small x-values should have large y-values: curve steeply increases at the left hand side),
- (b) the selection of patients for alternative treatment where positive treatment outcome should not be misclassified, that is a small false negative rate required (large y-values should have small x-values: curve should not decrease starting from the right upper corner), and
- (c) the selection of good indicator genes (indicated by both the minimal misclassification error given by the minimal distance $x + (1 - y)$ of the ROC curve to the left upper corner and the area under the ROC curve).

Table 1.7 reports the results for the breast cancer data set. The ROC curves

Table 1.7 Breast cancer data set: comparison of different approaches to predict therapy outcome based on the DNA microarray data (for explanations see text). Features are selected by Fisher-statistics ("statistics") and the P-SVM. The classifiers are weighted voting and ν-SVM where weighted voting results were taken from (?). The number F of selected features, the number E of minimal misclassifications over the threshold range, and the area under the ROC curve are shown. The protocol chose $\nu = 0.2$ for the ν-SVM.

Feature Selection /	#	min.	ROC
Classification	F	# E	area
statistics / weighted voting	70	20	0.77
P-SVM / ν-SVM	30	12	0.88

are shown in Figure 1.3 where the threshold b of the ν-SVM classifier was varied to produce the ROC curve for the P-SVM / ν-SVM method. For comparison these figures also contain the weighted voting result from (?) (supplementary information). Item (a) is the goal described in (?) (distance of the left part of the ROC curve to the top), where the poor prognosis patients should be recognized. In contrast to (a), in (b) positive prognosis patients should be recognized (distance of the top of the ROC curve to the left). The ROC curve judges all the regimes between (a) and (b). For (c) we suggest two indicators, the minimum leave-one-out error (indicated by the distance $x + (1 - y)$ of the ROC curve to the left upper corner) and the area under the ROC curve, where both indicator must use all genes in an optimal way. For (a) the poor prognosis indicators are more important and in (b) the positive prognosis indicators are the most relevant. The P-SVM results are comparable with (?) for (a) but the results are better for (b) and (c). Overall the P-SVM method performed better than weighted voting in (?) which is expressed though the larger values for the ROC areas. Larger values of the area under the ROC curve ("ROC area") mean higher performance where the minimum is 0.5 and the maximum 1.0.

Table 1.7 shows that the P-SVM method identified a smaller number of genes. Here a small number of genes is desirable because we already discarded genes with present calls based on less than 20 reliable values which reduces the risk of random correlations.

1.6 Summary

We have introduced a new feature selection method based on the support vector machine technique and we have applied it to the analysis of DNA microarray data. In contrast to previous approaches, features become support vectors and determine the classification boundary. This allows to select and rank the features through the support vector weights. Because the set of support vectors is sparse and avoids redundancy, the P-SVM approach is to be preferred over statistical methods which cannot recognize redundant information. We have described a data analysis protocol for the extraction of relevant genes from microarray data which can be used with the P-SVM as well as with other feature selection techniques. We have compared our feature selection approach on toy problems with state of the art features selection techniques and showed that our method gave the best results. Finally we have applied the P-SVM method to three data sets where the outcome of a chemo- or radiation therapy for cancer or tumors has to be predicted on the basis of gene expression profiles obtained from the microarray technique. The P-SVMs outperform previously used algorithms due to an improved selection of relevant genes.

Acknowledgments

We thank Cyril Minoux, Raman Sanyal, Merlyn Albery-Speyer, and Christoph Büscher for their help with the numerical simulations. This work was funded by the DFG (SFB 618).

1.A Derivation of the Dual Optimization Problem for the P-SVM

The primal optimization problem of the P-SVM is

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2} \left\| \mathbf{X}^\top \mathbf{w} \right\|^2 \\ &\text{s.t.} \quad \mathbf{X} \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y} \right) + \epsilon \, \mathbf{1} \geq \mathbf{0} \\ &\qquad\ \ \mathbf{X} \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y} \right) - \epsilon \, \mathbf{1} \leq \mathbf{0} \ . \end{aligned} \tag{1.30}$$

Following standard techniques of constrained optimization, we now derive the dual formulation of the optimization problem. The Lagrangian L is given by

$$\begin{aligned} L = &\frac{1}{2} \mathbf{w}^\top \mathbf{X} \mathbf{X}^\top \mathbf{w} \ - \\ &\left(\boldsymbol{\alpha}^+\right)^\top \left(\mathbf{X} \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y} \right) + \epsilon \, \mathbf{1} \right) + \\ &\left(\boldsymbol{\alpha}^-\right)^\top \left(\mathbf{X} \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y} \right) - \epsilon \, \mathbf{1} \right) \ , \end{aligned} \tag{1.31}$$

where the vectors $\boldsymbol{\alpha}^+ \ \geq \ \mathbf{0}$ and $\boldsymbol{\alpha}^- \ \geq \ \mathbf{0}$ are the Lagrange multipliers for the constraints in eqs. (1.30). The optimality conditions (Schölkopf and Smola, 2002) require that the following derivatives with respect to the primal variables of the Lagrangian L are zero:

$$\begin{aligned} \nabla_{\mathbf{w}} L &= \mathbf{X} \mathbf{X}^\top \mathbf{w} \ - \ \mathbf{X} \mathbf{X}^\top \boldsymbol{\alpha} \ = \ \mathbf{0} \ , \\ \frac{\partial L}{\partial b} &= \mathbf{1}^\top \mathbf{X}^\top \boldsymbol{\alpha} \ = \ 0 \ \ , \end{aligned} \tag{1.32}$$

where we used the abbreviation $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{+} - \boldsymbol{\alpha}^{-}$ $(\alpha_i = \alpha_i^{+} - \alpha_i^{-})$. In order to ensure the first condition $\mathbf{X}\,\mathbf{X}^{\top}\,\mathbf{w} = \mathbf{X}\,\mathbf{X}^{\top}\,\boldsymbol{\alpha}$ we set

$$\mathbf{w} = \boldsymbol{\alpha} \ . \tag{1.33}$$

We then obtain the dual optimization problem of the P-SVM:

$$\begin{aligned} \min_{\boldsymbol{\alpha}^{+},\boldsymbol{\alpha}^{-}} \quad & \frac{1}{2}\left(\boldsymbol{\alpha}^{+} - \boldsymbol{\alpha}^{-}\right)^{\top}\,\mathbf{X}\,\mathbf{X}^{\top}\,\left(\boldsymbol{\alpha}^{+} - \boldsymbol{\alpha}^{-}\right) - \\ & \mathbf{y}^{\top}\,\mathbf{X}^{\top}\,\left(\boldsymbol{\alpha}^{+} - \boldsymbol{\alpha}^{-}\right) + \epsilon\,\mathbf{1}^{\top}\left(\boldsymbol{\alpha}^{+} + \boldsymbol{\alpha}^{-}\right) \\ \text{s.t.} \quad & \mathbf{1}^{\top}\,\mathbf{X}^{\top}\,\left(\boldsymbol{\alpha}^{+} - \boldsymbol{\alpha}^{-}\right) = 0\ , \\ & \mathbf{0} \leq \boldsymbol{\alpha}^{+}\ , \quad \mathbf{0} \leq \boldsymbol{\alpha}^{-} \ . \end{aligned} \tag{1.34}$$

Normalization of $\mathbf{X}$ to zero mean during preprocessing automatically leads to the satisfaction of $\mathbf{1}^{\top}\,\mathbf{X}^{\top}\,(\boldsymbol{\alpha}^{+} - \boldsymbol{\alpha}^{-}) = 0$.

If vectors $\mathbf{u} \neq \mathbf{0}$ exist for which $\mathbf{X}^{\top}\mathbf{u} = 0$ holds, then the solution of eq. (1.33) and of the primal optimization problem eq. (1.30) is not unique (in the primal problem $\mathbf{w}$ appears only in context "$\mathbf{X}^{\top}\,\mathbf{w}$"). For positive values of ϵ, however, this degeneracy does not matter and a vector $\mathbf{w}$ is chosen, which is most sparse in its components, that is which has the largest number of zero components. The sparseness is due to $\mathbf{w} = \boldsymbol{\alpha}$ and the dual problem eqs. (1.34), where $\boldsymbol{\alpha}$ appears only in context "$\mathbf{X}^{\top}\,\boldsymbol{\alpha}$" except for the ϵ-part which enforces sparseness.

1.B Measurements of Complex Features

Here we consider the case that objects are fully described by a feature vector $\mathbf{x}$, but that we have measurement devices at hand that do not allow us to measure all of its individual components. Instead we assume that a measurement apparatus allows us to determine the values of a limited set of $\tilde{N}$ complex features $\mathbf{v}$. The complex features $\mathbf{v}$ are linear combinations of the elementary features $(\mathbf{x})_l$, $1 \leq l \leq N$, and the value of a complex feature j for an object i is given by the dot product

Complex Features

$$K_{ij} = (\mathbf{x}_i \cdot \mathbf{v}_j) \ . \tag{1.35}$$

If we define the matrix $\mathbf{V} := (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{\tilde{N}})$, our (incomplete) knowledge about the set $\mathbf{X}$ of objects can be summarized by the measurement or data matrix $\mathbf{K}$,

Measurement Matrix

$$\mathbf{K} = \mathbf{X}^{\top}\,\mathbf{V} \ . \tag{1.36}$$

For an application to microarray data, for example, we would identify the measured matrix $\mathbf{K}$ with the matrix of expression values obtained by a microarray experiment.

The complex features $\mathbf{v}$ span a subspace of the original feature space, but we do not require them to be orthogonal, normalized, or linearly independent. Due to the measurements, all objects are now described by an $\tilde{N}$-dimensional feature vector $(K_{i1}, K_{i2}, \ldots, K_{i\tilde{N}}) = ((\mathbf{x}_i \cdot \mathbf{v}_1), (\mathbf{x}_i \cdot \mathbf{v}_2), \ldots, (\mathbf{x}_i \cdot \mathbf{v}_{\tilde{N}}))$. If the number $\tilde{N}$ of different measurements is large, overfitting may occur. In order to obtain good generalization performance, feature selection must be performed on the set $\mathbf{V}$ of complex features.

Using the same line of arguments as in Section 1.3.3, the following constraints can be derived:

$$\begin{aligned} &\mathbf{K}^\top \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y}\right) - \epsilon\, \mathbf{1} \leq \mathbf{0}\,, \\ &\mathbf{K}^\top \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y}\right) + \epsilon\, \mathbf{1} \geq \mathbf{0}\ . \end{aligned} \tag{1.37}$$

Again we normalize the vectors which correspond to single genes,

$$\sum_{i=1}^{m} K_{ij} = 0 \text{ and } \frac{1}{m}\sum_{i=1}^{m} K_{ij}^2 = 1\,, \tag{1.38}$$

and — together with the P-SVM objective of Section 1.3.2 — we obtain the primal optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b}\ & \frac{1}{2}\,\|\mathbf{X}^\top \mathbf{w}\|^2 \\ \text{s.t.}\ & \mathbf{K}^\top \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y}\right) + \epsilon\, \mathbf{1} \geq \mathbf{0} \\ & \mathbf{K}^\top \left(\mathbf{X}^\top \mathbf{w} + b\mathbf{1} - \mathbf{y}\right) - \epsilon\, \mathbf{1} \leq \mathbf{0}\ . \end{aligned} \tag{1.39}$$

The corresponding dual formulation is

$$\begin{aligned} \min_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-}\ & \frac{1}{2}\left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right)^\top \mathbf{K}^\top \mathbf{K}\ \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) - \\ & \mathbf{y}^\top \mathbf{K}\ \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) + \epsilon\, \mathbf{1}^\top \left(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-\right) \\ \text{s.t.}\ & \mathbf{1}^\top \mathbf{K}\ \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) = 0\,, \\ & \mathbf{0} \leq \boldsymbol{\alpha}^+\,,\ \mathbf{0} \leq \boldsymbol{\alpha}^-\ , \end{aligned} \tag{1.40}$$

where the normal vector $\mathbf{w}$ has now been expanded with respect to the complex features $\mathbf{v}$,

$$\mathbf{w} = \mathbf{V}\,\boldsymbol{\alpha}\ . \tag{1.41}$$

The offset is again obtained by

$$b = -\frac{1}{m}\sum_{i=1}^{m}\left((\mathbf{w}\cdot\mathbf{x}_i) - y_i\right)\,, \tag{1.42}$$

from which we obtain the classification function

$$f(\mathbf{u}) = (\mathbf{w}\cdot\mathbf{u}) + b = \sum_{j=1}^{\tilde{N}} \alpha_j\ (\mathbf{u}\cdot\mathbf{v}_j) + b = \sum_{j=1}^{\tilde{N}} \alpha_j\ K_{i_u j} + b\ . \tag{1.43}$$

Note that $\mathbf{K}$ is neither required to be positive semidefinite nor square, because only the quadratic part $\mathbf{K}^\top\mathbf{K}$ appears in the objective function of eqs. (1.40). Therefore, we may consider $\mathbf{K}$ as the Gram matrix of a kernel which is not positive definite, that is a kernel which is not a Mercer kernel. Indeed, it has been shown that kernels which are not positive definite can be used for classification without any loss of generalization performance (Hochreiter and Obermayer, 2003b).

Matrix Data

The new interpretation allows to treat data in matrix form, where the matrix entries express the relationships between two sets of objects ("row objects" and "column objects"), and also allows to apply machine learning methods like classification, regression, or clustering to this data. The matrix originates from a dot product of representations of these objects in some feature space. An example of such matrix data is the drug-gene matrix in (Scherf et al., 2000), where a drug-cell matrix and a gene-cell matrix (expression values from a microarray experiment) are multiplied to obtain a drug-gene matrix. In this example the matrices $\mathbf{X}$ and $\mathbf{V}$ can be identified; however, the proposed framework allows to relate these matrices not only by a plain dot product but by some kernel evaluation. In this case our new interpretation holds as long as the kernel represents a dot product in some space (see kernel/dot product considerations below).

Pairwise Data

A special case of data in matrix form occurs if the row objects are identical to the column objects. This case is called "pairwise data", and the data matrix is usually interpreted as a similarity matrix $\mathbf{K}$. The advantage of the interpretation put forward in this appendix is that the P-SVM framework can still be applied by setting $\mathbf{V} = \mathbf{X}$. Pairwise data are common in bioinformatics, for example when considering the similarity measures for protein sequences (Lipman and Pearson, 1985), functional similarities of proteins (Sigrist et al., 2002; Falquet et al., 2002), chromosome location similarities of genes (Cremer et al., 1993; Lu et al., 1994), or coexpression data for genes (Heyer et al., 1999).

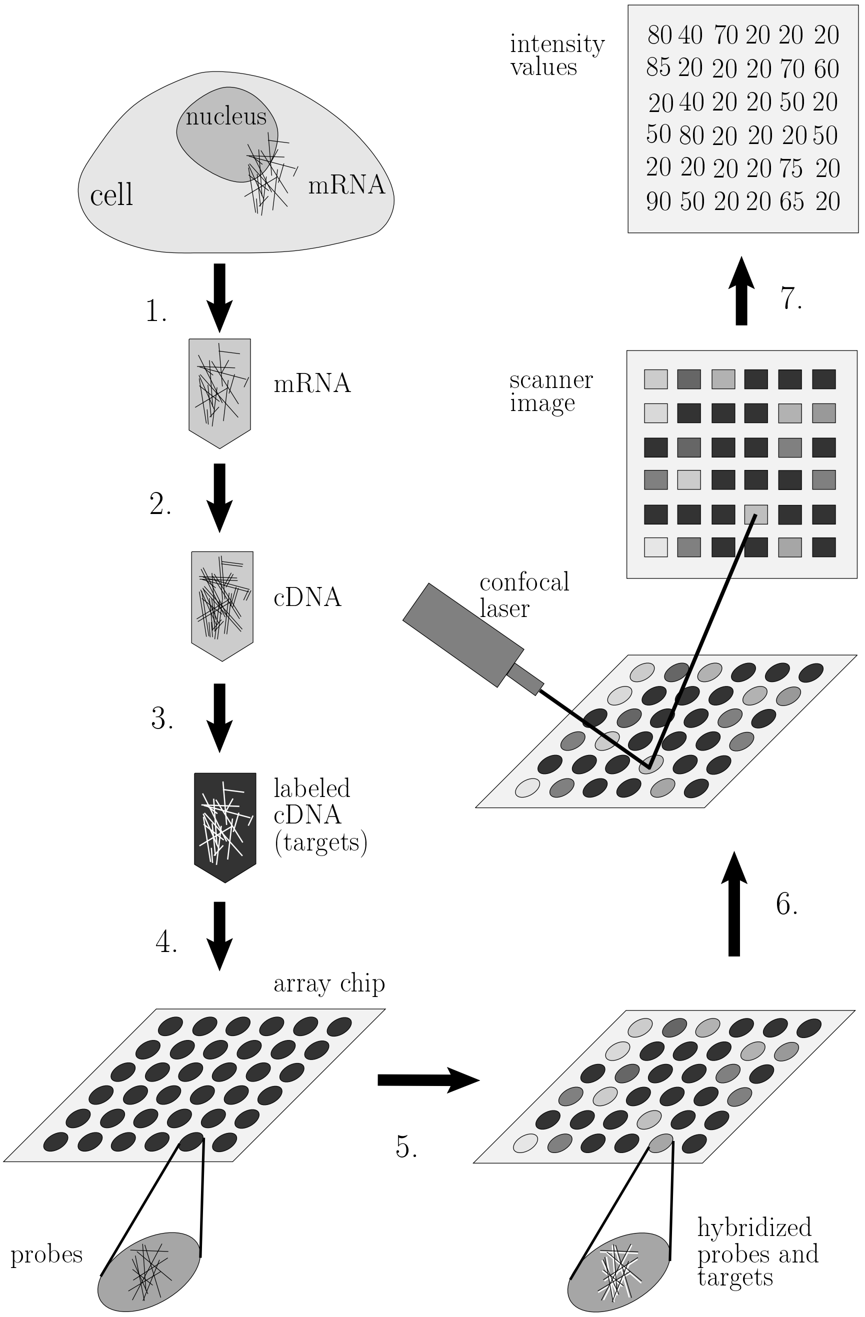
One issue, however, still remains open. Under what conditions is it possible to interpret a matrix of measured values as a dot product matrix $\mathbf{K}$? There is no full answer to this question from a theoretical viewpoint, practical applications have to confirm (or disprove) the chosen ansatz and data model. However, the question whether it is possible to describe a measurement by a dot product can be replaced by the question whether or not the following three conditions hold (see Hochreiter and Obermayer (2003a)):

- (1) Column objects ("samples") to classify are elements of a Hilbert space H_1, i.e. given a basis, these objects can be described by (possibly infinite dimensional) vectors.
- (2) Row objects ("complex features") are elements of a Hilbert space H_2, i.e. given a basis, these features can be described by (possibly infinite dimensional) vectors.

Measurement Device as Kernel

- (3) The measurement process can be expressed via the evaluation of a kernel.

Condition (3) equates the evaluation of a kernel as known from standard SVMs with physical measurements. As the kernel matrix is measured, no model selection has to be performed w.r.t. the kernel.



80	40	70	20	20	20
85	20	20	20	70	60
20	40	20	20	50	20
50	80	20	20	20	50
20	20	20	20	75	20
90	50	20	20	65	20

Figure 1.1 The microarray technique (see text for explanation).

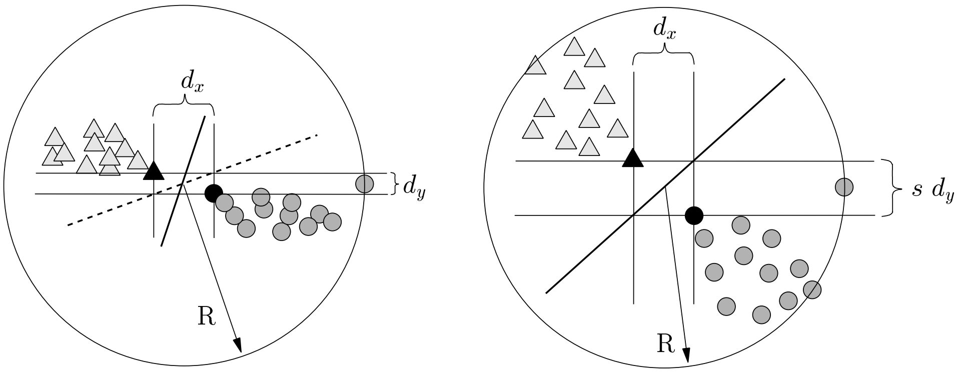


Figure 1.2 LEFT: data points from two classes (triangles and circles) are separated by the largest margin hyperplane (solid line) according to the support vector approach. The two support vectors (black symbols) are separated by d_x along the horizontal and by d_y along the vertical axis, from which we obtain $\rho = \frac{1}{2}\sqrt{d_x^2 + d_y^2}$ and $\frac{\mathrm{R}^2}{\rho^2} = \frac{4\ \mathrm{R}^2}{d_x^2 + d_y^2}$. The dashed line indicates the classification boundary of the classifier shown on the right, scaled back by a factor of $\frac{1}{s}$. RIGHT: the same data but scaled along the vertical axis by the factor s. The data points still lie within the sphere of radius R. The solid line denotes the support vector hyperplane, whose scaled version is shown on the left (dashed line). We obtain $\rho = \frac{1}{2}\sqrt{d_x^2 + s^2\ d_y^2}$ and $\frac{\mathrm{R}^2}{\rho^2} = \frac{4\ \mathrm{R}^2}{d_x^2 + s^2\ d_y^2}$. For $s \neq 1$ and $d_y \neq 0$ both the margin ρ and the term $\frac{\mathrm{R}^2}{\rho^2}$ change with scaling (see text for further explanation).

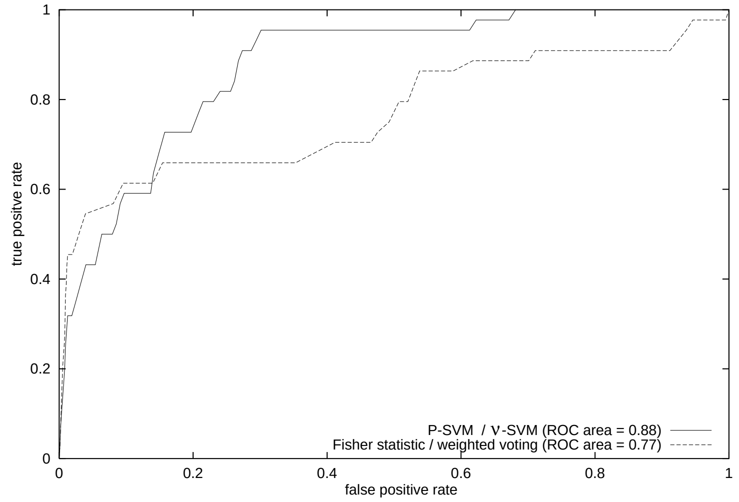


Figure 1.3 Breast cancer data set: the Receiver Operating Characteristic (ROC) curve is shown for the P-SVM feature selection followed by a ν-SVM classifier (solid line) and for the weighted voting approach of **?**) (dotted line). The number of selected features was F=30 for the P-SVM. The P-SVM outperformed the weighted voting approach of **?**) except for small numbers of false positives (left).

Index

References

S. V. Allander, N. N. Nupponen, M. Ringner, G. Hostetter, G. W. Maher, N. Goldberger, Y. Chen, J. Carpten, A. G. Elkahloun, and P. S. Meltzer. Gastrointestinal stromal tumors with KIT mutations exhibit a remarkably homogeneous gene expression profile. *Cancer Research*, pages 8624–8628, 2001. Download: `http://research.nhgri.nih.gov/microarray/gist_data.txt`.

H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, volume 2, pages 547–552, Anaheim, California, 1991. AAAI Press.

W. Bains and G. Smith. A novel method for nucleic acid sequence determination. *Journal of Theoretical Biology*, 135:303–307, 1988.

J. Bala, K. A. D. Jong, J. Haung, H. Vafaie, and H. Wechsler. Hybrid learning usnig genetic algorithms and decision trees for pattern classification. In C. S. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 719–724. Morgan Kaufmann Publishers, Inc, 1995.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

R. E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.

J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3:1229–1243, 2003. Special Issue on Variable and Feature Selection.

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

P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, pages 82–90, San Francisco, California, 1998. Morgan Kaufmann. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.Z.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.

A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarrays for phenotype classification. In *Proceedings of the Eighth International Conference* *on Intelligent Systems for Molecular Biology*, pages 75–85, 1999.

C. Cardie. Using decision trees to improve case–based learning. In *Proceedings of the 10th International Conference on Machine Learning*, pages 25–32. Morgan Kaufmann Publishers, Inc., 1993.

J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

R. Caruana and D. Freitag. Greedy attribute selection. In *International Conference on Machine Learning*, pages 28–36, 1994.

T. M. Cover and J. M. V. Campenhout. On the possible orderings in the measurement selection problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(9):657–661, 1977.

T. Cremer, A. Kurz, R. Zirbel, S. Dietzel, B. Rinke, E. Schröck, M. R. Speichel, U. Mathieu, A. Jauch, P. Emmerich, H. Schertan, T. Ried, C. Cremer, and P. Lichter. Role of chromosome territories in the functional compartmentalization of the cell nucleus. *Cold Spring Harbor Symp. Quant. Biol.*, 58:777–792, 1993.

N. Cristianini, H. Lodhi, and J. Shawe-Taylor. Latent semantic kernels. *Journal of Intelligent Information Systems (JJIS)*, 18(2-3):127–152, 2002.

S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the 18th International Conference on Machine Learning*, pages 74–81. Morgan Kaufmann, San Francisco, CA, 2001.

R. Drmanac, I. Labat, I. Brukner, and R. Crkvenjakov. Sequencing of megabase plus DNA by hybridization: theory of the method. *Genomics*, 4:114–128, 1989.

B. Durbin, J. Hardin, D. Hawkins, and D. M. Rocke. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 18(Supplement 1):105–110, 2002.

L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. J. Sigrist, K. Hofmann, and A. Bairoch. The PROSITE database, its status in 2002. *Nucleic Acids Research*, 30:235–238, 2002.

J. H. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, December 1981.

J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9):881–890, 1974.

T.-T. Frieß and R. F. Harrison. Linear programming support vector machines for pattern classification and regression estimation and the set reduction algorithm. TR RR-706, University of Sheffield, Sheffield, UK, 1998.

T. S. Furey, N. Duffy, N. Cristianini, D. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

D. Gerhold, T. Rushmore, and C. T. Caskey. DNA chips: promising toys have become powerful tools. *Trends in Biochemical Science*, 24(5):168–173, 1999.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. 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*, 286:531–537, 1999.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. Special Issue on Variable and Feature Selection.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Maximum likelihood estimation of optimal scaling factors for expression array normalization. In *Microarrays: Optical Technologies and Informatics, Proceedings of SPIE*, volume 4266, 2001.

B. Hassibi and D. G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In J. D. Cowan S. J. Hanson and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 164–171. San Mateo, CA: Morgan Kaufmann, 1993.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer Verlag, New York, 2001.

L.J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Research*, 11:1106–1115, 1999.

A. A. Hill, E. L. Brown, M. Z. Whitley, G. Tucker-Kellogg, C. P. Hunter, and D. K. Slonim. Evaluation of normalization procedures for oligonucleotide array data based on spiked cRNA controls. *Genome Biology*, 3(1):research0055.1–0055.13, 2001.

S. Hochreiter and K. Obermayer. Classification of pairwise proximity data with support vectors. In Y. LeCun and Y. Bengio, editors, *The Learning Workshop.* Computational & Biological Learning Society, Snowbird, Utha, 2002.

S. Hochreiter and K. Obermayer. Feature selection and classification on matrix data: From large margins to small covering numbers. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 889–896. MIT Press, Cambridge, MA, 2003a.

S. Hochreiter and K. Obermayer. Feature selection and matrix data. Technical report, Technische Universität Berlin, Fakultät für Elektrotechnik und Informatik, 2003b.

S. Hochreiter and J. Schmidhuber. LOCOCODE performs nonlinear ICA without knowing the number of sources. In J.-F. Cardoso, C. Jutten, and P. Loubaton, editors, *Proceedings of the First International Workshop on Independent Com-*

ponent Analysis and Signal Separation, Aussois, France, pages 149–154, 1999.

P. J. Huber. Projection pursuit. *Annals of Statistics*, 13(2):435–475, 1985.

W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(Supplement 1):96–104, 2002.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, New York, 2001.

J. E. Jackson. *A User's Guide to Principal Components*. Wiley, New York, 1991.

J. Jäger, R. Sengupta, and W. L. Ruzzo. Improved gene selection for classification of microarrays. In *Biocomputing – Proceedings of the 2003 Pacific Symposium*, pages 53–64, 2003.

G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129, 1994.

I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, New York, 1986.

C. Jutten and J. Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

M. K. Kerr, M. Martin, and G. A. Churchill. Analysis of variance for gene expression microarray data. *Journal of Computational Biology*, 7:819–837, 2000.

K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 129–134. MIT Press, 1992.

J. Kittler. Feature selection and extraction. In T. Y. Young and K.-S. Fu, editors, *Handbook of Pattern Recognition and Image Processing*, pages 59–83. Academic Press Inc., 1986.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

D. Koller and M. Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.

I. Kononenko. Estimating attributes: Analysis and extensions of Relief. In F. Bergadano and L. D. Raedt, editors, *Proceedings of the European Conference on Machine Learning*, 1994.

P. Langley. Selection of relevant features in machine learning. In *AAAI Fall Symposium on Relevance*, pages 140–144, 1994.

Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 598–605. San Mateo, CA: Morgan Kaufmann, 1990.

D. Lipman and W. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435–1441, 1985.

H. Liu and H. Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective.* Kluwer Academic Publishers, Boston, 1998.

H. Liu and R. Setiono. A probabilistic approach to feature selection – a filter solution. In *Proceedings of the 13th International Conference on Machine Learning*, pages 319–327. Morgan Kaufmann, 1996.

Q. Lu, L. L. Wallrath, and S. C. R. Elgin. Nucleosome positioning and gene regulation. *Journal of Cellular Biochemistry*, 55:83–92, 1994.

Y. Lysov, V. Florent'ev, A. Khorlin, K. Khrapko, V. Shik, and A. Mirzabekov. DNA sequencing by hybridization with oligonucleotides. *Doklady Academy Nauk USSR*, 303:1508–1511, 1988.

T. Marill and D. M. Green. On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory*, 9:11–17, 1963.

E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.

S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3: 1333–1356, 2003. Special Issue on Variable and Feature Selection.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, 2002.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

A. Rakotomamonjy. Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, 3:1357–1370, 2003. Special Issue on Variable and Feature Selection.

L. A. Rendell and K. Kira. A practical approach to feature selection. In *International Conference on machine learning*, pages 249–256, 1992.

M. Robnik-Sikonja and I. Kononenko. An adaptation of Relief for attribute estimation in regression. In *Proceedings of the 14th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1997.

U. Scherf, D. T. Ross, M. Waltham, L. H. Smith, J. K. Lee, L. Tanabe, K. W. Kohn, W. C. Reinhold, T. G. Myers, D. T. Andrews, D. A. Scudiero, M. B. Eisen, E. A. Sausville, Y. Pommier, D. Botstein, P. O. Brown, and J. N. Weinstein. A gene expression database for the molecular pharmacology of cancer. *Nature Genetics*, 24(3):236–244, 2000.

B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

J. Schuchhardt, D. Beule, A. Malik, E. Wolski, H. Eickhoff, H. Lehrach, and H. Herzel. Normalization strategies for cDNA microarrays. *Nucleic Acids Research*, 28(10):E47, 2000.

J. Shawe-Taylor, P. L. Bartlett, R. Williamson, and M. Anthony. A framework for structural risk minimization. In *Proceedings of the 9th Annual Conference on Computational Learning Theory*, pages 68–76, New York, 1996. Association for Computing Machinery.

J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, R. C. T. Aguiar J. L. Kutok, M. Gaasenbeek, M. Angelo, M. Reich, T. S. Ray G. S. Pinkus, M. A. Koval, K. W. Last, A. Norton, J. Mesirov T. A. Lister, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002.

W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

C. J. Sigrist, L. Cerutti, N. Hulo, A. Gattiker, L. Falquet, M. Pagni, A. Bairoch, and P. Bucher. PROSITE: A documented database using patterns and profiles as motif descriptors. *Brief Bioinformatics*, 3:265–274, 2002.

D. B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *International Conference on Machine Learning*, pages 293–301, 1994.

A. J. Smola, T. Frieß, and B. Schölkopf. Semiparametric support vector and linear programming machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 585–591, Cambridge, MA, 1999. MIT Press.

E. Southern. United Kingdom patent application GB8810400, 1988.

C. J. Stone. Optimal rates of convergence for nonparametric estimators. *Annals of Statistics*, 8(6):1348–1360, 1980.

N. Tishby. Relevant coding and information bottlenecks: A principled approach to multivariate feature selection. In *NIPS'2001 Workshop on Variable and Feature Selection*, 2001.

N. Tishby, F.C. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003. Special Issue on Variable and Feature Selection.

G. Tseng, M. Oh, L. Rohlin, J. Liao, and W. Wong. Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects. *Nucleic Acids Research*, 29:2549–2557, 2001.

P. D. Turney. Exploiting context when learning to classify. In *Proceedings of the European Conference on Machine Learning*, pages 402–407, 1993a. ftp://ai.iit.nrc.ca/pub/ksl-papers/NRC-35058.ps.Z.

P. D. Turney. Robust classification with context-sensitive features. In *Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 268–276, 1993b. ftp://ai.iit.nrc.ca/pub/ksl-papers/NRC-35074.ps.Z.

V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Science"*, 98:5116–5121, 2001.

H. Vafaie and K. De Jong. Genetic algorithms as a tool for feature selection in machine learing. In *Proceedings of the Fourth Conference on Tools for Artificial Intelligence*, pages 200–203. IEEE Computer Society Press, 1992.

H. Vafaie and K. De Jong. Robust feature selection algorithms. In *Proceedings of the Fifth Conference on Tools for Artificial Intelligence*, pages 356–363. IEEE Computer Society Press, 1993.

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

V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

D. G. Wang, J.-B. Fan, and C.-J. Siao. Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome. *Science*, 280:1077–1082, 1998.

J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3: 1439–1461, 2003. Special Issue on Variable and Feature Selection.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, Cambridge, MA, 2000.

L. Xu, P. Zan, and T. Chang. Best first strategy for feature selection. In *Ninth International Conference on Pattern Recognition*, pages 706–708. IEEE Computer Society Press, 1989.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed.

Normalization for cDNA microarray data: A robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4):e15, 2002.