Learning and Feature Selection Using the Set Covering Machine with Data-Dependent Rays on Gene Expression Profiles

Hans A. Kestler[1,2], Wolfgang Lindner[3], and André Müller[2]

[1] Neural Information Processing, University of Ulm, 89069 Ulm
hans.kestler@uni-ulm.de
[2] Internal Medicine I, University Hospital Ulm, Robert-Koch-Str. 8, 89081 Ulm
lindner@informatik.uni-ulm.de
[3] Theoretical Computer Science, University of Ulm, 89069 Ulm, Germany
andre.mueller@uniklinik-ulm.de

Abstract. Microarray technologies are increasingly being used in biological and medical sciences for high throughput analyses of genetic information on the genome, transcriptome and proteome levels. The differentiation between cancerous and benign processes in the body often poses a difficult diagnostic problem in the clinical setting while being of major importance for the treatment of patients. In this situation, feature reduction techniques capable of reducing the dimensionality of data are essential for building predictive tools based on classification. We extend the set covering machine of Marchand and Shawe-Taylor to data dependent rays in order to achieve a feature reduction and direct interpretation of the found conjunctions of intervals on individual genes. We give bounds for the generalization error as a function of the amount of data compression and the number of training errors achieved during training. In experiments with artificial data and a real world data set of gene expression profiles from the pancreas we show the utility of the approach and its applicability to microarray data classification.

1 Background

Microarray technologies are increasingly being used in biological and medical sciences for high throughput analyses of genetic information on the genome, transcriptome, and proteome levels. Gene-expression microarrays permit the estimation of mRNA concentrations for a large number of genes in parallel. These types of analysis generate vast amounts of data, often in the form of large lists of genes differentially expressed between different sample sets being insufficient for class prediction purposes. One challenge involves finding biologically meaningful subgroups of genes that are congruently expressed in multiple experiments (e.g. cell lines under different conditions or tissues from different disease states). Especially the differentiation between cancerous and benign processes in the body often poses a difficult diagnostic problem in the clinical setting while being of major importance for the treatment of patients. In this situation techniques

F. Schwenker and S. Marinai (Eds.): ANNPR 2006, LNAI 4087, pp. 286–297, 2006.

for generating simple rules on the expression values are of major importance. This usually implies a reduction of the dimensionality of the data either with principal component approaches or with feature selection methods. The latter is clearly more desirable, since these methods retain a direct interpretability if the subsequent classification concept is not too complex.

One of the few learning algorithms which provably performs well in the presence of many irrelevant attributes is the algorithm for learning conjunctions of few Boolean variables due to Haussler [1]. Given a sample of size m whose examples are labeled according to a conjunction with at most s out of n Boolean variables, the algorithm finds a consistent conjunction with at most $s \log m$ variables in polynomial time in n and m. By a standard Occam's Razor bound based on the cardinality of the hypothesis class it follows that a sample size m which is linear in s but only logarithmic in the dimension n suffices to guarantee a small generalization error with high confidence. This means that the algorithm performs well even when the dimension n is exponential in the sample size m, provided that sample is labeled by a conjunction which depends only on very few attributes.

The set covering machine (SCM) of Marchand and Shawe-Taylor [2] is a generalization of the algorithm of Haussler where the Boolean variables are replaced by a set of *features*, where each feature is a Boolean-valued function on the example space. In general, the example space may be an arbitrary subset of R^n, each feature may depend on all of the n attributes, and the set of features used by the SCM may depend on the given sample. The algorithm of Haussler is then used to find a conjunction of few features which makes only few errors on the sample to allow a trade-off between accuracy and complexity. Marchand and Shawe-Taylor further consider the specific set of so-called data-dependent balls as features and bound the generalization error of the corresponding SCM in terms of the amount of data compression the SCM achieves during training. The bound is obtained by a technique first used by Littlestone and Warmuth [3]. Subsequently, Marchand et al. [4] consider data-dependent half-spaces as an alternative to data-dependent balls.

In [5], Marchand and Shah consider rays as features and an algorithm similar to the SCM to classify gene expression data. Rays are simple threshold functions which depend on a single attribute. The algorithm is guided by a PAC-Bayesian style analysis of the generalization error as initiated by McAllester [6]. This approach amounts to specify a prior distribution P on the set of conjunctions of rays. Then the error bound applies to the Bayesian classifier which is the weighted majority vote over all binary hypothesis and where the weight corresponds to the posterior distribution Q induced by the prior distribution P and the given sample S. The resulting algorithm attains good theoretical and experimental results on high-dimensional gene expression data. However, the Bayesian classifier can no longer be expressed as a simple conjunction of rays and this might significantly aggravate the interpretation of the classifier in the clinical setting.

In this paper we use the SCM with *data-dependent rays* as features. The resulting classifier is then a simple conjunction of a small number of rays. We give bounds on the generalization error based on both an analysis of the VC

dimension of the (data-independent) hypotheses space as well as on the amount of data compression the SCM achieves during training. Finally we apply the proposed algorithm to gene expression data with the task of distinguishing between malignant and inflammatory tumors of the pancreas.

2 The Set Covering Machine with Rays as Features

In order to describe the SCM of Marchand and Shawe-Taylor [2] we will briefly review the algorithm for learning conjunctions with few Boolean variables of Haussler [1]. Suppose that we are given a sample S of examples from the Boolean hypercube $X = \{0,1\}^n$ where the examples are labeled according to some conjunction with at most r variables. Let P and N denote the subsets of all positive and negative examples from S, respectively. We say that a variable x_j is *consistent* with a labeled example $(\mathbf{x}, y)$ if the j-th coordinate of $\mathbf{x}$ coincides with its label y. The aim is to find a conjunction which is consistent with S and posses as few variables as possible.

The algorithm is based on the observation that a conjunction

$$\bigwedge_{x_j \in R} x_j$$

is consistent with the sample S if and only if the set of variables R possess the following two properties.

1. Every variable $x_j \in R$ is consistent with each positive example $\mathbf{x} \in P$.
2. For every negative example $\mathbf{x} \in N$ there is at least one variable $x_j \in R$ which is consistent with $\mathbf{x}$.

These two properties can be equivalently expressed in terms of the collections of sets R_j and Q_j, where R_j is the set of all positive examples $\mathbf{x} \in P$ such that the variable x_j is not consistent with $\mathbf{x}$, and Q_j is the set of all negative examples $\mathbf{x} \in N$ such that the variable x_j is consistent with $\mathbf{x}$ (R is a set of variables, whereas R_j and Q_j are sets of examples). Then the first property is equivalent to the fact that $R_j = \emptyset$ for all $x_j \in R$, and the second property is equivalent to saying that the union of sets Q_j for $x_j \in R$ *covers* the sets N in the sense that $\bigcup_{x_j \in R} Q_j = N$.

The algorithm of finding such a set of variables can be described by the following two steps:

1. Find a set of variables that is consistent with all positive examples
2. Cover this set with as few subsets of variables, that are consistent with the negative examples.

The task of finding the *smallest* such set R can be easily transformed into an instance of the *Minimal Set Cover Problem*, a well-known NP-complete problem [7] and is thus intractable. There is, however, a simple greedy strategy to efficiently find an approximate solution. Here we successively select the variables

x_k for which $R_k = \emptyset$ and $|Q_k|$ is maximal and thus Q_k covers as many negative examples as possible. After the selection of x_k for inclusion in R we discard Q_k from all Q_j and repeat this process with the remaining variables until there is no negative example left to cover. It is not hard to see that after at most $r \log m$ selected variables we have found a cover of N and thus the resulting conjunction is consistent with S. Note that the size of the conjunction is only by the factor $\log m$ larger than the optimal solution of size r. Furthermore, the running time is polynomial in n and m. For a more detailed description of the algorithm we refer to [8].

Now let us turn to the SCM algorithm. In contrast to the Boolean setting we are now given a sample S of labeled examples from an example space X which may be any subset of the n-dimensional real space R^n. The Boolean variables are replaced by a set of *features* H, where each feature $h_j \in H$ is an arbitrary Boolean-valued function on X. The set of features H is assumed to be finite but may depend on the sample S. The collections R_j and Q_j are defined analogously. That is, R_j is the subset of positive examples from P that are misclassified by h_j, and Q_j is the subset of negative examples from N that are correctly classified by h_j. The aim is now to find a small subset of features $R \subseteq H$ such that the conjunction

$$h(\mathbf{x}) = \bigwedge_{h_j \in R} h_j(\mathbf{x})$$

is consistent with S. This will be done in a greedy manner similarly as above.

In order to allow a trade-off between accuracy and complexity we are given additionally a *early stopping* value s and a *penalty* parameter p. Rather than solely based on the the cardinality $|Q_j|$ under the constraint that $R_j = \emptyset$ as above, now the greedy strategy selects the features h_j according to their *usefulness*

$$U_j = |Q_j| - p|R_j| \ .$$

Furthermore, the SCM algorithm stops as soon as the number of selected features reaches the value of s. Thus, the parameter s bounds the number of selected features, and the parameter p controls the fraction of errors on the positive examples among all errors on S. Note that when p and s are both set to ∞, then the SCM algorithm corresponds precisely to Haussler's algorithm in the Boolean setting where H is just the set of variables x_j for $j = 1, \ldots, n$. A more formal description of the algorithm can be found in Figure 1.

2.1 VC Dimension

We first consider the SCM with *data-independent* rays as its set of features and bound the generalization error of the corresponding SCM in terms of the VC dimension of its hypotheses space (cf. [9]). A *ray* over the example space $X = \mathrm{R}^n$ is a simple threshold function of the form

$$h_j^t(\mathbf{x}) = \begin{cases} 1 & \text{if } (\mathbf{x})_j \geq t \\ 0 & \text{otherwise} \end{cases}$$

Algorithm SCM(S, H, s, p)

1. Initially set $P \leftarrow \{\mathbf{x} \mid (x,1) \in S\}$, $N \leftarrow \{\mathbf{x} \mid (x,0) \in S\}$ and $R \leftarrow \emptyset$.
2. For each $h_j \in H$ let $Q_j = \{\mathbf{x} \in N \mid h_j(\mathbf{x}) = 0\}$ and $R_j = \{\mathbf{x} \in P \mid h_j(\mathbf{x}) = 0\}$.
3. Select the feature $h_k \in H$ with largest usefulness $U_k = |Q_k| - p|R_k|$. If $Q_k = \emptyset$ then goto step 7.
4. Set $R \leftarrow R \cup \{h_k\}$.
5. For each $h_j \in H$ set $Q_j \leftarrow Q_j \setminus Q_k$ and $R_j \leftarrow R_j \setminus R_k$.
6. If $\bigcup_{h_j \in R} Q_j = N$ or $|R| = s$ then go to step 7. Else go to step 3.
7. Return $h = \bigwedge_{h_j \in R} h_j$

Fig. 1. The SCM Algorithm

where $(\mathbf{x})_j$ denotes the j-th coordinate of the vector $x \in X$ and t is a real-valued threshold. The only constraint we impose on the set of rays when used as the set of features of a SCM is that the thresholds are taken form a finite set of values and consequently also the corresponding set of all rays over $X = \mathrm{R}^n$ is finite.

We now bound the VC dimension of the hypotheses space of all conjunctions over a bounded number of rays (with no constraint on the number of admissible thresholds) as follows.

Theorem 1. *Let H_n^r denote the hypotheses space of all conjunctions of at most r rays over the example space $X = \mathrm{R}^n$. Then,*

$$r \log\left(\frac{n}{r}\right) \leq \mathrm{VCdim}(H_n^r) \leq 2r \log\left(\frac{n}{r}\right) + 6r \ .$$

It is well-known that the generalization error $\mathrm{err}_D(h)$ of hypothesis h produced by a learning algorithm can be bounded in terms of the VC dimension of its hypotheses space [10,9]. Recall that the generalization error $\mathrm{err}_D(h)$ is the probability that $h(\mathbf{x}) \neq y$ for some labeled example $(\mathbf{x}, y)$ drawn according to D. We want to bound the generalization error in terms of the number of features r and the number of errors k of the hypothesis h produced by the SCM with rays as its set of features. Note that both r and k are quantities which may depend on S. For this reason we use the following bound of [11]. Let $H_1 \subseteq H_2 \subseteq \cdots \subseteq H_M$ be a nested sequence of hypotheses classes such that $\mathrm{VCdim}(H_i) = d_i$ for $i = 1, \ldots, M$, let D be any probability distribution on $X \times \{0,1\}$, and let S be a random sample of m labeled examples drawn independently according to D. Then with probability $1 - \delta$, if a learning algorithm finds a hypothesis $h \in H_i$ which makes k errors on the training set S, then the generalization error $\mathrm{err}_D(h)$ is at most

$$\frac{2k}{m} + \frac{4}{m}\left(d_i \log\left(\frac{2em}{d_i}\right) + \log\left(\frac{4M(m+1)}{\delta}\right)\right)$$

provided that $d \leq m$. Applying Theorem 1 we get the following bound for the SCM with rays as its set of features.

Corollary 1. *Let D be any probability distribution on $X \times \{0,1\}$, and and let S be a random sample of m labeled examples drawn independently according to D.*

Suppose that the SCM algorithm with any finite set of rays as its set of features on the sample S produces a hypothesis $h = \bigwedge_{h_j^t \in R} h_j^t$ with $|R| = r$ and such that h makes k errors on S. Then with probability $1 - \delta$ the generalization error $\mathrm{err}_D(h)$ *is at most*

$$\frac{2k}{m} + \frac{4}{m}\left(\left(2r\log\left(\frac{n}{r}\right) + 6r\right)\log\left(\frac{2em}{r\log(n/r)}\right) + \log\left(\frac{4n(m+1)}{\delta}\right)\right)$$

provided that $r \leq \frac{m}{2\log n + 6}$.

2.2 Sample Compression

Bounds on the generalization error based on the VC dimension are generally rather pessimistic. Better bounds can sometimes be achieved by considering the amount of data compression a learning algorithm achieves during training. For this purpose we consider *data-dependent* rays as features for the SCM algorithm. That is, for a given sample

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$$

with examples from $X = \mathrm{R}^n$, each feature $h_j^i \in H$ has the form

$$h_j^i(\mathbf{x}) = \begin{cases} 1 & \text{if } (\mathbf{x})_j \geq (\mathbf{x}_i)_j \\ 0 & \text{otherwise} \end{cases}$$

for some position $j \in \{1, \ldots, n\}$ and some index $i \in \{1, \ldots, m\}$ of a positive example $\mathbf{x}_i \in P$. Recall that $(\mathbf{x})_j$ denotes the j-th coordinate of any vector $x \in X$. Thus, the final hypothesis has the form

$$h = \bigwedge_{h_j^i \in R} h_j^i \ .$$

Let us now see how the corresponding SCM algorithm can be regarded as a compression scheme for the sample S. Let A denote the SCM algorithm with data-dependent rays as its set of features and with parameters s and p. Then A can be decomposed into a *compression* function f and a *reconstruction* function g as follows. The function f maps the sample S to the *compression set* S_I and an additional set of positions J, where $I = \{i \mid h_j^i \in R\}$ and $J = \{j \mid h_j^i \in R\}$ and S_I denotes the subsequence of S which consists only of those positive examples $\mathbf{x}_i \in P$ with indices $i \in I$. The reconstruction function g takes S_I and J as inputs and returns the hypothesis

$$g(S_I, J) = \bigwedge_{j \in J} h_j^{t_j}$$

where the threshold t_j of each ray $h_j^{t_j}$ is defined as $t_j = \min\{(\mathbf{x})_j \mid \mathbf{x} \in S_I\}$.

When A is run with penalty $p = \infty$, then each feature h_j^i selected by A is consistent with all positive examples $\mathbf{x} \in P$. This means that for each $h_j^i \in R$ we have $\mathbf{x}_i \in S_I$ and $(\mathbf{x})_j \geq (\mathbf{x}_i)_j$ for all $\mathbf{x} \in S_I$ and, hence, $(\mathbf{x}_i)_j = \min\{(\mathbf{x})_j \mid x \in S_I\}$. It follows that the reconstructed hypothesis $g(S_I, J)$ coincides with the hypothesis h produced by A and thus $A(S) = g(f(S))$. Note that if h makes no errors on S then the labels of all examples from S can be determined solely from the compression set S_I and from the additional information J. In this sense A indeed can be regarded as a compression scheme for S.

When A is run with penalty $p < \infty$, then a selected feature h_j^i might be inconsistent with some positive examples $x \in S_I$. For this reason we slightly modify the SCM algorithm A by considering a feature h_j^i for inclusion in the current set R only if $(\mathbf{x}_i)_j \leq (\mathbf{x})_j$ for all examples $\mathbf{x}$ in the current compression set S_I. This modification implies that all features h_j^i in the final set R satisfy $\mathbf{x}_i \in S_I$ and $(\mathbf{x})_j \geq (\mathbf{x}_i)_j$ for all $\mathbf{x} \in S_I$ as in the case of $p = \infty$ above. Thus for the modified algorithm A we have $A(S) = g(f(S))$ also in the case $p < \infty$.

By using similar arguments as in [12,2] we can now bound the generalization error in terms of the size of the compression set S_I, the number of features used in the hypothesis h, and the number of errors h makes on S as follows.

Theorem 2. *Let D be a probability distribution on $X \times \{0,1\}$, and let S be a random sample of m labeled examples drawn independently according to D. Suppose the SCM algorithm with data-dependent rays as its set of features on the sample S finds a hypothesis $h = \bigwedge_{h_j^i \in R} h_j^i$ which makes k errors on S, such that $|I| = d$ and $|J| = r$ for the sets $I = \{i \mid h_j^i \in R\}$ and $J = \{j \mid h_j^i \in R\}$. Then with probability $1 - \delta$ the error $\mathrm{err}_D(h)$ is at most*

$$\varepsilon(d, r, k) = 1 - \exp\left(-\frac{1}{m - d - k} \ln\left(\frac{B(d, r, k)}{\delta(d, r, k)}\right)\right)$$

where

$$B(d, r, k) = \binom{m}{d}\binom{n}{r}\binom{m-d}{k}$$

and

$$\delta(d, r, k) = \delta\left(\frac{\pi^2}{6}\right)^{-3} ((d+1)(r+1)(k+1))^{-2}$$

Proof. Let $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ and let f and g be the compression scheme as described above. Recall that $f(S) = (S_I, J)$ and $g(S_I, J) = h$. Further let $K = \{i \mid h(\mathbf{x}_i) \neq y_i\}$ be the set of k indices of examples $\mathbf{x}_i$ which are misclassified by h. Note that the compression set S_I is always correctly classified by h and hence we may assume that the sets I and K are disjoint. We want to bound the probability

$$\Pr_{S \sim D^m}\left(\mathrm{err}_D(h) > \varepsilon(d, r, k)\right)$$
$$= \sum \Pr_{S \sim D^m}\left(\mathrm{err}_D(h) > \varepsilon(d_0, r_0, k_0), I = I_0, J = J_0, K = K_0\right)$$

where the sum is taken over all possible values $0 \leq d_0 \leq m$, $0 \leq r_0 \leq n$ and $0 \leq k_0 \leq m - d_0$, and all possible sets $I_0 \subseteq \{1, \ldots, m\}$, $J_0 \subseteq \{1, \ldots, n\}$ and $K_0 \subseteq \{1, \ldots, m\} \setminus I_0$ with $|I_0| = d_0$, $|J_0| = r_0$ and $|K_0| = k_0$.

To bound the probability of $\mathrm{err}_D(h) > \varepsilon(d_0, r_0, k_0)$ with respect to fixed sets $I = I_0$, $J = J_0$ and $K = K_0$ of cardinalities d_0, r_0, k_0, first observe that $h = g(S_{I_0}, J_0)$ and hence the hypothesis h is fixed as soon as the examples in the subsequence S_{I_0} are drawn. Since the examples in S are drawn independently according to D, we may further assume that the $d_0 + k_0$ examples from the subsequences S_{I_0} and S_{K_0} are drawn first. Next the remaining $m - d_0 - k_0$ examples of S are drawn. If $\mathrm{err}_D(h) > \varepsilon$ then the probability that a single example drawn according to D is consistent with h is less than $1 - \varepsilon$. It follows that

$$\Pr_{S \sim D^m} \left(\mathrm{err}_D(h) > \varepsilon(d_0, r_0, k_0), I = I_0, J = J_0, K = K_0 \right) < (1 - \varepsilon(d_0, r_0, k_0))^{m - d_0 - k_0} = \frac{\delta(d_0, r_0, k_0)}{B(d_0, r_0, k_0)}$$

Note that $B(d_0, r_0, k_0)$ is just the number of possible ways to choose the sets I_0, J_0 and K_0 of cardinalities d_0, r_0 and k_0. Hence, by summing up over all possible d_0, r_0, k_0, I_0, J_0 and K_0 we get

$$\Pr_{S \sim D^m} \left(\mathrm{err}_D(h) > \varepsilon(d, r, k) \right) < \sum \frac{\delta(d_0, r_0, k_0)}{B(d_0, r_0, k_0)} < \delta$$

where for the last inequality we additionally used the fact that $\sum_{i \geq 1} 1/i^2 = \pi^2/6$. □

The bound of Theorem 2 is 0.27 for a data set with $n = 20$ dimensions $m = 100$ examples, 2 resulting features and 1 error on the training set ($\delta = 0.01$). The PAC bound from Corollary 1 is 7.23 in this case.

3 Experimental Results

The hypothesis space consisting of conjunctions of left-open intervals (the quadrant space) is often too restrictive - the concept could lie as well on the right side of a threshold - the union of the left open and the right open quadrant concept space is always used in the following.

3.1 Learning Algorithms

The following algorithms were applied to artificial and microarray data.

SCM.1 Choose $p = \infty$ so that only left- and right-open rays consistent with the positive examples were used. The SCM was trained until no errors on the training set remained ($s = \infty$). The possible ties in step 3. (Figure 1) were recursively broken to generate equivalent solutions (at most 50 distinct solutions were sought).

SCM.2 As SCM.1 but this time the classifier with the lowest theoretical generalization bound according to Theorem 2 was taken. A run of SCM with $s = \infty$ defines at step 3. (Figure 1) a sequence of features $i_1, i_2 \dots i_{s'}$ (in this order), that are combined by conjunctions, such that the classifiers $(C_k)_1^{s'}$, C_k including the hypotheses $h_{i_1} \dots h_{i_k}$, form a sequence of increasing complexity $C_1 \subset C_2 \dots \subset C_{s'}$. The C_k with the smallest generalization bound was taken.

SCM.3 For this classifier p was varied over every fifth value of the set $\{i/j \mid i = 1 \dots m_n, j = 1 \dots m_p\}$ sorted in ascending order and $p = \infty$. All possible $\leq m$ rays were generated for each feature. The theoretical bound from Theorem 2 was used to select the optimal classifier(s).

SVM A support vector machine with linear kernel was used (the support vector implementation from the R [13] package e1071 [14] was called with *kernel="linear", scale=FALSE*.

1-nn One-nearest-neighbor with Euclidean distances.

Cart Classification tree with pruning (R package rpart [15] with Gini impurity measure).

3.2 Artificial Data

Artificial data was generated with a randomly chosen concept C from the union of the left-open and right-open quadrant hypothesis space $Q_k \cup Q_k^c$ on the domain $\mathcal{X} = [0,1]^k$ for $k < n$ relevant coordinates for $k = 5$. Once generated the concept C was held constant and was used to generate $m_p = 50$ positive samples and $m_n = 50$ negative samples with a uniform distribution of the coordinates in C and in C^c. The remaining $n-k$ coordinates were filled up with uniformly distributed random numbers on $[0,1]^{n-k}$ in order to create n-dimensional random vectors with k relevant and $n-k$ irrelevant features. The number of dimensions n were varied from $\{20, 100, 200, 3000\}$.

Data Set / n	SCM.1 ($p=\infty$)				SCM.2 ($p=\infty$)				SCM.3					
	TR	TE	size	num	TR	TE	size	num	TR	TE	size	num	s	p
rnd/20	0	31	2	1	0	31	2	1	0	31	2	1	2	3
rnd/100	0	55.75	3	8	1	43	2	1	1	43	2	1	2	4
rnd/200	0	78.42	4	38	2	59	2	1	2	49.78	2	9	2	4
rnd/3000	0	60.5	3	4	2	38	2	1	2	20.6	2	4	2	4

Data Set / n	1-nn		SVM		cart	
	TR	TE	TR	TE	TR	TE
rnd/20	0	264	3	139	3	64
rnd/100	0	387	0	245	3	53
rnd/200	0	457	0	318	5	78
rnd/3000	0	491	0	448	6	43

After training the classifiers the generalization error was estimated by applying the trained classifier to an independent test set with 500 positive and 500 negative samples which were chosen i.i.d. from C and C^c.

3.3 Application to Microarray Data

The described method was applied to a previously published gene expression dataset (see Buchholz et al. [16]) with $n = 169$ features and 62 samples divided into a training set of $m = 42$ ($m_p = 25$ and $m_n = 17$) and a test set of 12 positive and 8 negative samples. The best result on the test set was obtained with the following classifier (Algorithm SCM.1):

s	err_{TR}	err_{TE}	feature	ray
1	4	4	Annexin A2 (ANXA2)	$[0.5505, \infty)$
2	1	4	serine/threonine-protein kinase PLK1	$[0.1245, \infty)$
3	0	3	Asparagine synthetase (= ts11, a G1 prog protein)	$[0.982, \infty)$

For $s > 1$ every preceding feature $s' < s$ is included in the classifier (conjunction).

We have compared the ray learning algorithm with other standard methods on the PaCa data. PaCa training and test set, and 10× 5-fold cross-validation results are given in the table below. For the SCM simulations up to 50 solutions were allowed. Results are given as cases (mean ± stdev):

PaCa	TR	TE	features	solutions	CV
1-nn	0	2			14.1 ± 1.45
SVM	0	6			13.8 ± 2.82
CART	2	3			7.9 ± 1.60
SCM.1	0	5.46 ± 1.04	3 ± 0	37	19.62 ± 2.76
SCM.2	0	7 ± 0	3 ± 0	2	17.7 ± 2.46
SCM.3	1 ± 0	5.25 ± 0.96	3 ± 0	4	16.15 ± 2.13

4 Discussion and Conclusion

In contrast to the original SCM, which uses data dependent balls, the proposed conjunction of rays allows a direct correspondence to the original feature space leading to concise interpretable hypotheses, which in turn may trigger further biological investigations. It is easy to show that for high-dimensional spaces with a low sample size the probability to find a consistent hypothesis reaches one.

Malignant and inflammatory tumors of the pancreas could be separated with 3 genes (which would be improbable for random data) leading to a low error on the test set. The results on the artificial data indicate a very good performance of the SCM with data dependent rays when there are only a few informative features within a large set of features containing noise. This construction was chosen

to resemble the microarray data, as it is assumed that in standard microarray studies the number of genes which are regulated across the different conditions is low in comparison to the total number of genes investigated. The nearest neighbor and the support vector machine attained almost an error rate of 50% on the test sets. It seems that in these cases feature reduction is of greater importance than the complexity of the classifier. Even for the highly selected (for involvement in cancer) gene sets used in the PaCa data the performance is still comparable to SVMs, only CART gave better results here.

Extensions of the scheme could include the optimization of the rays using different utility functions and combination of sets of consistent hypotheses to possibly increase robustness or to consider margins of the rays to break ties of equivalent greedy solutions.

The main advantage of fusing decisions on singular features is the independence of a precise knowledge of their scale. For instance it is still unclear on what scale expression values in microarray experiments are. It is argued that only comparisons of expression values within a gene are reasonable. With our approach we only rely on an ordinal scale. Decision tree algorithms allowing only axis parallel splits behave similar in this aspect, but have an infinite VC dimension and thus possibly generate much more complex decision rules. In contrast our algorithm only generates rules of the form "IF expression for gene A is above 3.4 AND expression for gene B is below 2.5 AND ... THEN the risk of the subject having disease C is increased". These types of rules seem to be much more appropriate for performing diagnosis or differential diagnosis in a clinical setting.

Acknowledgments

This work is supported by the German Science Foundation, SFB 518, Project C5 (HAK and AM) and the Stifterverband für die Deutsche Wissenschaft (HAK).

References

1. Haussler, D.: Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. Artificial Intelligence **36** (1988) 177–221
2. Marchand, M., Shawe-Taylor, J.: The set covering machine. Journal of Machine Learning Research **3** (2002) 723–746
3. Littlestone, N., Warmuth, M.: Relating data compression and learnability. Technical report, University of California, Santa Cruz (1986)
4. Marchand, M., Shah, M., Shawe-Taylor, J., Sokolova, M.: The set covering machine with data-dependent half-spaces. In: Proceedings of the Twentieth International Conference on Machine Learning (ICML). (2003) 520–527
5. Marchand, M., Shah, M.: PAC-Bayes Learning of Conjunctions and Classification of Gene-Expression Data. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA (2005) 881–888

6. McAllester, D.: Some PAC-Bayesian theorems. Machine Learning **37** (1999) 355–363
7. Garey, M., Johnson, D.: Computers and Intractability – A Guide to the Theory of NP-Completeness. Freeman and Company (1979)
8. Kearns, M., Vazirani, U.: An Introduction to Computational Learning Theory. MIT Press (1994)
9. Vapnik, V.: Statistical Learning Theory. Wiley (1998)
10. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Learnability and the Vapnik-Chervonenkis dimension. Journal of the ACM **36** (1989) 929–965
11. Shawe-Taylor, J., Bartlett, P.L., Williamson, R.C., Anthony., M.: Structural risk minimization over data-dependent hierarchies. IEEE Trans. on Information Theory **44** (1998) 1926–1940
12. Floyd, S., Warmuth, M.: Sample compression, learnability, and the Vapnik-Chervonenkis dimension. Machine Learning **21** (1995) 269–304
13. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2006) ISBN 3-900051-07-0.
14. Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A.: e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. (2006) R package version 1.5-13.
15. Therneau, T.M., port by Brian Ripley <ripley@stats.ox.ac.uk>, B.A.R.: rpart: Recursive Partitioning. (2005) R package version 3.1-27.
16. Buchholz, M., Kestler, H.A., Bauer, A., Bock, W., Rau, B., Leder, G., Kratzer, W., Bommer, M., Scarpa, A., Schilling, M., Adler, G., Hoheisel, J., Gress, T.: Specialized DNA arrays for the differentiation of pancreatic tumors. Clin. Cancer Res. **11** (2005) 8048–54