Feature Selection with Conjunctions of Decision Stumps and Learning from Microarray Data

Mohak Shah, *Member*, *IEEE*, Mario Marchand, and Jacques Corbeil

Abstract—One of the objectives of designing feature selection learning algorithms is to obtain classifiers that depend on a small number of attributes *and* have verifiable future performance guarantees. There are few, if any, approaches that successfully address the two goals *simultaneously*. To the best of our knowledge, such algorithms that give theoretical bounds on the future performance have not been proposed so far in the context of the classification of gene expression data. In this work, we investigate the premise of learning a conjunction (or disjunction) of *decision stumps* in Occam's Razor, Sample Compression, and PAC-Bayes learning settings for identifying a small subset of attributes that can be used to perform reliable classification tasks. We apply the proposed approaches for gene identification from DNA microarray data and compare our results to those of the well-known successful approaches proposed for the task. We show that our algorithm not only finds hypotheses with a much smaller number of genes while giving competitive classification accuracy but also having tight risk guarantees on future performance, unlike other approaches. The proposed approaches are general and extensible in terms of both designing novel algorithms and application to other domains.

Index Terms—Microarray data classification, risk bounds, feature selection, gene identification.

1 Introduction

An important challenge in the problem of classification of high-dimensional data is to design a learning algorithm that can construct an accurate classifier that depends on the smallest possible number of attributes. Further, it is often desired that there should be realizable guarantees associated with the future performance of such feature selection approaches. With the recent explosion in various technologies generating huge amounts of measurements, the problem of obtaining learning algorithms with performance guarantees has acquired a renewed interest.

Consider the case of the biological domain, where the advent of microarray technologies [1], [2] has revolutionized the outlook on the investigation and analysis of genetic diseases. In parallel, on the classification front, many interesting results have appeared aiming to distinguish between two or more types of cells (e.g., diseased versus normal, or cells with different types of cancers), based on gene expression data in the case of DNA microarrays (see, for instance, [3] for results on Colon Cancer, [4] for Leukemia). Focusing on very few genes to give insight into the class association for a microarray sample is quite important due to a variety of reasons. For instance, a small subset of genes is easier to analyze as opposed to the set of genes output by the DNA microarray chips. It also makes it relatively easier to deduce biological relationships among them as well as to study their interactions. An approach able to identify a very few number of genes can facilitate customization of chips and validation experiments—making the utilization of microarray technology cheaper, affordable, and effective.

In the view of a diseased versus a normal sample, these genes can be considered as indicators of the disease's cause. Subsequent validation study focused on these genes, their behavior, and their interactions can lead to better understanding of the disease. Some attempts in this direction have yielded interesting results. See, for instance, a recent algorithm proposed by [5] involving the identification of a gene subset based on importance ranking and, subsequently, combinations of genes for classification. Another example is the approach of [6] based on nearest shrunken centroids. Some kernel-based approaches, such as the BAHSIC algorithm [7] and their extensions (e.g., [8] for short time-series domains), have also appeared.

The traditional methods used for classifying high-dimensional data are often characterized as either "filters" (e.g., [5], [9]) or "wrappers" (e.g., [10]), depending on whether the attribute selection is performed independent of, or in conjunction with, the base learning algorithm.

Despite the acceptable empirical results achieved by such approaches, there is no theoretical justification of their performance nor do they come with a guarantee on how well they will perform in the future. Note that some successful approaches have appeared in regression analysis, most notably LASSO [11], [12] and, recently, the forward-backward greedy approach of [13], but following a different line of analysis. In the present context of classification, what is really needed is a learning algorithm that has *provably good performance guarantees* in the presence of many irrelevant attributes. This is the focus of the work presented here.

- *M. Shah is with Accenture, 161 N. Clark St., Chicago, IL 60601. E-mail: mohak.shah@accenture.com.*
- *M. Marchand is with the Department of Computer Science and Software Engineering, Pav. Adrien Pouliot, Laval University, Sainte-Foy, QC G1V 0A6, Canada. E-mail: Mario.Marchand@ift.ulaval.ca.*
- *J. Corbeil is with the Infectious Diseases Research Centre, Faculty of Medicine, Laval University, 2705 Laurier Blvd., T3-67, QC G1V-4G2, Canada. E-mail: Jacques.Corbeil@crchul.ulaval.ca.*

Manuscript received 4 May 2010; revised 14 Dec. 2010; accepted 3 Mar. 2011; published online 26 Apr. 2011.
Recommended for acceptance by T. Jebara.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-05-0345.
Digital Object Identifier no. 10.1109/TPAMI.2011.82.

0162-8828/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

1.1 Contributions

The main contributions of this work come in the form of formulation of feature selection strategies within well-established learning settings, resulting in learning algorithms that combine the tasks of feature selection and discriminative learning. Consequently, we obtain feature selection algorithms for classification with tight realizable guarantees on their generalization error. The proposed approaches are a step toward more general learning strategies that combine feature selection with the classification algorithm *and* have tight realizable guarantees. We apply the approaches to the task of classifying microarray data, where the attributes of the data sample correspond to the expression level measurements of various genes. In fact, the choice of decision stumps as learning bias has in part been motivated by this application. The framework is general and extensible in a variety of ways. For instance, the learning strategies proposed in this work can readily be extended to other similar tasks that can benefit from this learning bias. An immediate example would be classifying data from other microarray technologies, such as in the case of Chromatin Immunoprecipitation experiments. Similarly, learning biases other than the conjunctions of decision stumps can also be explored in the same frameworks, leading to novel learning algorithms.

1.2 Motivation

For learning the class of conjunctions of features, we draw motivation from the guarantee that exists for this class in the following form: *If there exists a conjunction that depends on r out of the n input attributes and that correctly classifies a training set of m examples, then the greedy covering algorithm of [14] will find a conjunction of at most $r \ln m$ attributes that makes no training errors*. Note the absence of dependence on the number n of input attributes. The method is guaranteed to find at most $r \ln m$ attributes and, hence, depends on the number of available samples m but not on the number of attributes n to be analyzed.

We propose learning algorithms for building small conjunctions of *decision stumps*. We examine three approaches to obtain an optimal classifier based on this premise that mainly vary in the coding strategies for the threshold of each decision stump. The first two approaches attempt to do this by encoding the threshold either with message strings (Occam's Razor) or by using training examples (Sample Compression). The third strategy, Probably Approximately Correct (PAC) Bayes, attempts to examine if an optimal classifier can be obtained by trading off the sparsity[1] of the classifier with the magnitude of the separating margin of each decision stump. In each case, we derive an upper bound on the generalization error of the classifier and subsequently use it to guide the respective algorithm. Finally, we present empirical results on the microarray data classification tasks and compare our results to the state-of-the-art approaches proposed for the task including the Support Vector Machine (SVM) coupled with feature selectors, and Adaboost. The preliminary results of this work appeared in [15].

1. This refers to the number of decision stumps used.

1.3 Organization

Section 2 gives the basic definitions and notions of the learning setting that we utilize and also characterizes the hypothesis class of conjunctions of decision stumps. All subsequent learning algorithms are proposed to learn this hypothesis class. Section 3 proposes an Occam's Razor approach to learn conjunctions of decision stumps leading to an upper bound on the generalization error in this framework. Section 4 then proposes an alternate encoding strategy for the message strings using the Sample Compression framework and gives a corresponding risk bound. In Section 5, we propose a PAC-Bayes approach to learn conjunction of decision stumps that enables the learning algorithm to perform an explicit nontrivial margin-sparsity trade-off to obtain more general classifiers. Section 6 then proposes algorithms to learn in the three learning settings proposed in Sections 3, 4, and 5, along with a time complexity analysis. Note that the learning (optimization) strategies proposed in Section 6 do not affect the respective theoretical guarantees of the learning settings. The algorithms are evaluated empirically on real-world microarray data sets in Section 7. Section 8 presents a discussion on the results and also provides an analysis of the biological relevance of the selected genes in the case of each data set and their agreement with published findings. Finally, we conclude in Section 9.

2 Definitions

The input space $\mathcal{X}$ consists of all n-dimensional vectors $\mathbf{x} = (x_1, \ldots, x_n)$, where each real-valued component $x_i \in [A_i, B_i]$ for $i = 1, \ldots n$. Each attribute x_i, for instance, can refer to the expression level of gene i. Hence, A_i and B_i are, respectively, the a priori lower and upper bounds on values for x_i. The output space $\mathcal{Y}$ is the set of classification labels that can be assigned to any input vector $\mathbf{x} \in \mathcal{X}$. We focus here on binary classification problems. Thus, $\mathcal{Y} = \{0, 1\}$. Each example $\mathbf{z} = (\mathbf{x}, y)$ is an input vector $\mathbf{x} \in \mathcal{X}$ with its classification label $y \in \mathcal{Y}$ chosen *i.i.d.* from an unknown distribution D on $\mathcal{X} \times \mathcal{Y}$. The true risk $R(f)$ of any classifier f is defined as the probability that it misclassifies an example drawn according to D:

$$R(f) \stackrel{\text{def}}{=} \Pr_{(\mathbf{x},y)\sim D}(f(\mathbf{x}) \neq y) = \mathbf{E}_{(\mathbf{x},y)\sim D} I(f(\mathbf{x}) \neq y),$$

where $I(a) = 1$ if predicate a is true and 0 otherwise. Given a training set $S = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ of m examples, the *empirical risk* $R_S(f)$ on S, of any classifier f, is defined according to

$$R_S(f) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} I(f(\mathbf{x}_i) \neq y_i) \stackrel{\text{def}}{=} \mathbf{E}_{(\mathbf{x},y)\sim S} I(f(\mathbf{x}) \neq y).$$

The goal of any learning algorithm is to find the classifier with minimal true risk based on measuring empirical risk (and other properties) on the training sample S.

We focus on learning algorithms that construct a *conjunction of decision stumps* from a training set. Each *decision stump* is just a threshold classifier defined on a single attribute (component) x_k. More formally, a decision stump is identified by an *attribute index* $k \in \{1, \ldots, n\}$, a *threshold value* $t \in \mathbb{R}$, and a *direction* $d \in \{-1, +1\}$ (that

specifies whether class 1 is on the largest or smallest values of x_k). Given any input example $\mathbf{x}$, the output $r^k_{td}(\mathbf{x})$ of a decision stump is defined as

$$r^k_{td}(\mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if} \quad (x_k - t)d > 0, \\ 0, & \text{if} \quad (x_k - t)d \leq 0. \end{cases}$$

We use a vector $\mathbf{k} \stackrel{\text{def}}{=} (k_1, \ldots, k_{|\mathbf{k}|})$ of attribute indices $k_j \in \{1, \ldots, n\}$ such that $k_1 < k_2 < \cdots < k_{|\mathbf{k}|}$, where $|\mathbf{k}|$ is the number of indices present in $\mathbf{k}$ (and thus the number of decision stumps in the conjunction).[2] Furthermore, we use a vector $\mathbf{t} = (t_{k_1}, t_{k_2}, \ldots, t_{k_{|\mathbf{k}|}})$ of threshold values and a vector $\mathbf{d} = (d_{k_1}, d_{k_2}, \ldots, d_{k_{|\mathbf{k}|}})$ of directions where $k_j \in \{1, \ldots, n\}$ for $j \in \{1, \ldots, |\mathbf{k}|\}$. On any input example $\mathbf{x}$, the output $C^{\mathbf{k}}_{\mathbf{td}}(\mathbf{x})$ of a conjunction of decision stumps is given by

$$C^{\mathbf{k}}_{\mathbf{td}}(\mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if} \quad r^j_{t_j d_j}(\mathbf{x}) = 1 \quad \forall j \in \mathbf{k}, \\ 0, & \text{if} \quad \exists j \in \mathbf{k} : r^j_{t_j d_j}(\mathbf{x}) = 0. \end{cases}$$

Finally, any algorithm that builds a conjunction can be used to build a disjunction just by exchanging the role of the positive and negative labeled examples. In order to keep our description simple, we describe here only the case of a conjunction. However, the case of disjunction follows symmetrically.

3 An Occam's Razor Approach

Our first approach toward learning the conjunction (or disjunction) of *decision stumps* is the Occam's Razor approach. Basically, we wish to obtain a hypothesis that can be coded using the least number of bits. We first propose an Occam's Razor risk bound which will ultimately guide the learning algorithm.

In the case of zero-one loss, we can model the risk of the classifier as a binomial. Let $\text{Bin}(\kappa, m, r)$ be the the binomial tail associated with a classifier of (true) risk r. Then, $\text{Bin}(\kappa, m, r)$ is the probability that this classifier makes at most κ errors on a set of m examples:

$$\text{Bin}(\kappa, m, r) \stackrel{\text{def}}{=} \sum_{i=0}^{\kappa} \binom{m}{i} r^i (1-r)^{m-i}.$$

The *binomial tail inversion* $\overline{\text{Bin}}(\kappa, m, \delta)$ then gives the largest risk value that a classifier can have while still having a probability of at least δ of observing at most κ errors out of m examples [16], [17]:

$$\overline{\text{Bin}}(\kappa, m, \delta) \stackrel{\text{def}}{=} \sup\{r : \text{Bin}(\kappa, m, r) \geq \delta\}.$$

From this definition, it follows that $\overline{\text{Bin}}(mR_S(f), m, \delta)$ is the *smallest* upper bound which holds with probability at least $1 - \delta$ on the true risk of any classifier f with an observed empirical risk $R_S(f)$ on a test set of m examples:

$$\forall f : \quad \Pr_{S \sim D^m}\big(R(f) \leq \overline{\text{Bin}}\big(mR_S(f), m, \delta\big)\big) \geq 1 - \delta.$$

Our starting point is the Occam's razor bound of [16], which is a tighter version of the bound proposed by [18]. It is also more general in the sense that it applies to any prior distribution P over any countable class of classifiers.

Theorem 1 ([16]). *For any prior distribution P over any countable class $\mathcal{F}$ of classifiers, for any data-generating distribution D, and for any $\delta \in (0, 1]$, we have*

$$\Pr_{S \sim D^m}\{\forall f \in \mathcal{F} : R(f) \leq \overline{\text{Bin}}(mR_S(f), m, P(f)\delta)\} \geq 1 - \delta.$$

The proof (available in [16]) directly follows from a straightforward union bound argument and from the fact that $\sum_{f \in \mathcal{F}} P(f) = 1$. To apply this bound for conjunctions of decision stumps we thus need to choose a suitable prior P for this class. Moreover, Theorem 1 is valid when $\sum_{f \in \mathcal{F}} P(f) \leq 1$. Consequently, we will use a *subprior* P whose sum is ≤ 1.

In our case, decision-stumps' conjunctions are specified in terms of the discrete-valued vectors $\mathbf{k}$ and $\mathbf{d}$ and the continuous-valued vector $\mathbf{t}$. We will see below that we will use a finite-precision bit string σ to specify the set of threshold values $\mathbf{t}$. Let us denote by $P(\mathbf{k}, \mathbf{d}, \sigma)$ the prior probability assigned to the conjunction $C^{\mathbf{k}}_{\sigma\mathbf{d}}$ described by $(\mathbf{k}, \mathbf{d}, \sigma)$. We choose a prior of the following form:

$$P(\mathbf{k}, \mathbf{d}, \sigma) = \frac{1}{\binom{n}{|\mathbf{k}|}} p(|\mathbf{k}|) \frac{1}{2^{|\mathbf{k}|}} g_{\mathbf{k},\mathbf{d}}(\sigma),$$

where $g_{\mathbf{k},\mathbf{d}}(\sigma)$ is the prior probability assigned to string σ given that we have chosen $\mathbf{k}$ and $\mathbf{d}$. Let $\mathcal{M}(\mathbf{k}, \mathbf{d})$ be the set of all message strings that we can use given that we have chosen $\mathbf{k}$ and $\mathbf{d}$. If $\mathcal{I}$ denotes the set of all 2^n possible attribute index vectors and $\mathcal{D}_{\mathbf{k}}$ denotes the set of all $2^{|\mathbf{k}|}$ binary direction vectors $\mathbf{d}$ of dimension $|\mathbf{k}|$, we have that $\sum_{\mathbf{k} \in \mathcal{I}} \sum_{\mathbf{d} \in \mathcal{D}_{\mathbf{k}}} \sum_{\sigma \in \mathcal{M}(\mathbf{k},\mathbf{d})} P(\mathbf{k}, \mathbf{d}, \sigma) \leq 1$ whenever $\sum_{d=0}^{n} p(d) \leq 1$ and $\sum_{\sigma \in \mathcal{M}(\mathbf{k},\mathbf{d})} g_{\mathbf{k},\mathbf{d}}(\sigma) \leq 1 \ \forall \mathbf{k}, \mathbf{d}$.

The reasons motivating this choice for the prior are the following: The first two factors come from the belief that the final classifier, constructed from the group of attributes specified by $\mathbf{k}$, should depend only on the number $|\mathbf{k}|$ of attributes in this group. If we have complete ignorance about the number of decision stumps the final classifier is likely to have, we should choose $p(d) = 1/(n+1)$ for $d \in \{0, 1, \ldots, n\}$. However, we should choose a p that decreases as we increase d if we have reason to believe that the number of decision stumps of the final classifier will be much smaller than n. Since this is usually our case, we propose to use

$$p(|\mathbf{k}|) = \frac{6}{\pi^2} (|\mathbf{k}| + 1)^{-2}.$$

The third factor of $P(\mathbf{k}, \mathbf{d}, \sigma)$ gives equal prior probabilities for each of the two possible values of direction d_j.

To specify the distribution of strings $g_{\mathbf{k},\mathbf{d}}(\sigma)$, consider the problem of coding a threshold value $t \in [a, b] \subset [A, B]$ where $[A, B]$ is some predefined interval in which we are permitted to choose t and where $[a, b]$ is an interval of "equally good" threshold values.[3] We propose the following diadic coding scheme for the identification of a threshold value that

2. Although it is possible to use up to two decision stumps on any attribute, we limit ourselves here to the case where each attribute can be used for only one decision stump.

3. By a "good" threshold value, we mean a threshold value for a decision stump that would cover many negative examples and very few positive examples (see the learning algorithm).

belongs to that interval: Let l be the number of bits that we use for the code. Then, a code of l bits specifies one value among the set Λ_l of threshold values:

$$\Lambda_l \stackrel{\text{def}}{=} \left\{ \left[1 - \frac{2j-1}{2^{l+1}} \right] A + \frac{2j-1}{2^{l+1}} B \right\}_{j=1}^{2^l}.$$

We denote by A_i and B_i the respective a priori minimum and maximum values that the attribute i can take. These values are obtained from the definition of data. Hence, for an attribute $i \in \mathbf{k}$, given an interval $[a_i, b_i] \subset [A_i, B_i]$ of threshold values, we take the smallest number l_i of bits such that there exists a threshold value in Λ_{l_i} that falls in the interval $[a_i, b_i]$. In that way, we will need at most $\lfloor \log_2((B_i - A_i)/(b_i - a_i)) \rfloor$ bits to obtain a threshold value that falls in $[a_i, b_i]$. Note the relationship between the desired margin interval $[a_i, b_i]$ and the required number of bits to obtain a threshold in this interval. The finer the interval, the higher the number of bits required to obtain potential thresholds such that at least one such threshold falls in the desired interval. This can also be seen in the $\lfloor \log_2((B_i - A_i)/(b_i - a_i)) \rfloor$ required bits for threshold specification. The wider the interval $[a_i, b_i]$, the fewer the number of bits that would be needed so that a threshold falls in $[a_i, b_i]$. Hence, the coding scheme facilitates trading off the sparsity (as will be seen in terms of the learning bias in Section 6.1) in favor of more succinct represent (and hence a wider margin around the boundary of the threshold).

Hence, to specify the threshold for each decision stump $i \in \mathbf{k}$, we need to specify the number l_i of bits and an l_i-bit string s_i that identifies one of the threshold values in Λ_{l_i}. The risk bound does not depend on how we actually code σ (for some receiver). It only depends on the a priori probabilities we assign to each possible realization of σ. We choose the following distribution:

$$g_{\mathbf{k},\mathbf{d}}(\sigma) \stackrel{\text{def}}{=} g_{\mathbf{k},\mathbf{d}}(l_1, s_1, \ldots, l_{|\mathbf{k}|}, s_{|\mathbf{k}|}) \tag{1}$$

$$= \prod_{i \in \mathbf{k}} \zeta(l_i) \cdot 2^{-l_i}, \tag{2}$$

where

$$\zeta(a) \stackrel{\text{def}}{=} \frac{6}{\pi^2}(a+1)^{-2} \quad \forall a \in \mathbb{N}. \tag{3}$$

The overall sum of the possible realizations of σ gives 1 since $\sum_{i=1}^{\infty} i^{-2} = \pi^2/6$. Note that by giving equal a priori probability to each of the 2^{l_i} strings s_i of length l_i, we give no preference to any threshold value in Λ_{l_i}.

The distribution ζ that we have chosen for each string length l_i has the advantage of decreasing slowly so that the risk bound does not deteriorate too rapidly as l_i increases. Other choices are clearly possible. However, note that the dominant contribution comes from the 2^{-l_i} term, yielding a risk bound that depends linearly in l_i.

With this choice of prior, we have the following theorem:

Theorem 2. *Given all our previous definitions and for any $\delta \in (0, 1]$, we have, with probability at least $1 - \delta$ over random draws of $S \sim D^m$,*

$$\forall \mathbf{k}, \mathbf{d}, \sigma : R\big(C^{\mathbf{k}}_{\sigma\mathbf{d}}\big) \leq \overline{\mathrm{Bin}}\left(m R_S(C^{\mathbf{k}}_{\sigma\mathbf{d}}), m, \frac{p(|\mathbf{k}|) g_{\mathbf{k},\mathbf{d}}(\sigma)\delta}{\binom{n}{|\mathbf{k}|} 2^{|\mathbf{k}|}} \right).$$

Finally, we emphasize that the risk bound of Theorem 2, used in conjunction with the distribution of messages given by $g_{\mathbf{k},\mathbf{d}}(\sigma)$, provides a guide for choosing the optimal classifier. Note that the above risk bound suggests a nontrivial trade-off between the number of attributes and the length of the message string used to encode the classifier. Indeed, the risk bound may be smaller for a conjunction having a large number of attributes with small message strings (i.e., small l_is) than for a conjunction having a small number of attributes but with large message strings.

4 A Sample Compression Approach

The basic idea of the sample compression framework [19] is to obtain learning algorithms with the property that the generated classifier (with respect to some training data) can often be reconstructed with a very small subset of training examples. More formally, a learning algorithm A is said to be a *sample-compression algorithm* iff there exists a *compression function* $\mathcal{C}$ and a *reconstruction function* $\mathcal{R}$ such that for any training sample $S = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ (where $\mathbf{z}_i \stackrel{\text{def}}{=} (\mathbf{x}_i, y_i)$), the classifier $A(S)$ returned by A is given by $A(S) = \mathcal{R}(\mathcal{C}(S))\ \forall S \in (\mathcal{X} \times \mathcal{Y})^m$.

For a training set S, the compression function $\mathcal{C}$ of learning algorithm A outputs a subset $\mathbf{z_i}$ of S, called the *compression set*, and an *information message* σ, i.e., $(\mathbf{z_i}, \sigma) = \mathcal{C}(\mathbf{z}_1, \ldots, \mathbf{z}_m)$. The information message σ contains the additional information needed to reconstruct the classifier from the compression set $\mathbf{z_i}$. Given a training sample S, we define the compression set $\mathbf{z_i}$ by a vector of indices $\mathbf{i}$ such that $\mathbf{i} \stackrel{\text{def}}{=} (i_1, i_2, \ldots, i_{|\mathbf{i}|})$, with $i_j \in \{1, \ldots, m\}\ \forall j$ and $i_1 < i_2 < \cdots < i_{|\mathbf{i}|}$ and where $|\mathbf{i}|$ denotes the number of indices present in $\mathbf{i}$.

When given an arbitrary compression set $\mathbf{z_i}$ and an arbitrary information message σ, the *reconstruction function* $\mathcal{R}$ of a learning algorithm A must output a classifier. The information message σ is chosen from a set $\mathcal{M}(\mathbf{z_i})$ that consists of all the distinct messages that can be attached to the compression set $\mathbf{z_i}$. The existence of this reconstruction function $\mathcal{R}$ assures that the classifier returned by $A(S)$ is *always* identified by a compression set $\mathbf{z_i}$ and an information message σ.

In sample compression settings for learning decision stumps' conjunctions, the message string consists of the attributes and directions defined above. However, the thresholds are now specified by training examples. Hence, if we have $|\mathbf{k}|$ attributes, where $\mathbf{k}$ is the set of thresholds, the compression set consists of $|\mathbf{k}|$ training examples (one per threshold).

Our starting point is the following generic sample compression bound [20]

Theorem 3. *For any sample compression learning algorithm with a reconstruction function $\mathcal{R}$ that maps arbitrary subsets of a training set and information messages to classifiers, we have, with probability at least $1 - \delta$ over random draws of $S \sim D^m$:*

$$\forall \mathbf{i} \in \mathcal{I}, \sigma \in \mathcal{M}(\mathbf{Z_i}) : R(\mathcal{R}(\sigma, \mathbf{Z_i})) \leq \epsilon(\sigma, \mathbf{Z_i}, |\mathbf{j}|),$$

where

$$\epsilon(\sigma, \mathbf{z_i}, |\mathbf{j}|) = 1 - \exp\left(\frac{-1}{m - |\mathbf{i}| - |\mathbf{j}|}\left[\ln\binom{m}{|\mathbf{i}|} + \ln\binom{m - |\mathbf{i}|}{|\mathbf{j}|} + \ln\left(\frac{1}{P_{\mathcal{M}(\mathbf{Z_i})}(\sigma)}\right) + \ln\left(\frac{1}{\zeta(|\mathbf{i}|)\zeta(|\mathbf{j}|)\delta}\right)\right]\right),$$

and ζ is defined by (3).

Now, we need to specify the distribution of messages ($P_{\mathcal{M}(\mathbf{Z_i})}(\sigma)$) for the conjunction of decision stumps. Note that in order to specify a conjunction of decision stumps, the compression set consists of one example per decision stump. For each decision stump, we have one attribute and a corresponding threshold value determined by the numerical value that this attribute takes on the training example.

The learner chooses an attribute whose threshold is identified by the associated training example. The set of these training examples forms the compression set. Finally, the learner chooses a direction for each attribute.

The subset of attributes that specifies the decision stumps in our compression set $\mathbf{z_i}$ is given by the vector $\mathbf{k}$ defined in the previous section. Moreover, since there is one decision stump corresponding to each example in the compression set, we have $|\mathbf{i}| = |\mathbf{k}|$. Now, we assign equal probability to each possible set $|\mathbf{k}|$ of attributes (and hence thresholds) that can be selected from n attributes. Moreover, we assign equal probability over the direction that each decision stump can have $(+1, -1)$. Hence, we get the following distribution of messages:

$$P_{\mathcal{M}(\mathbf{z_i})}(\sigma) = \binom{n}{|\mathbf{k}|}^{-1} \cdot 2^{-|\mathbf{k}|} \quad \forall \sigma. \tag{4}$$

Equation (4), along with the *Sample Compression Theorem*, completes the bound for the conjunction of decision stumps.

5 A PAC-Bayes Approach

The Occam's Razor and sample compression in a sense aim at obtaining sparse classifiers with minimum number of stumps. This sparsity is enforced by selecting the classifiers with minimal encoding of the message strings and the compression set in respective cases.

We now examine if sacrificing this sparsity in terms of a larger separating margin around the decision boundary (yielding more confidence) can lead us to classifiers with smaller generalization error. The learning algorithm is based on the PAC-Bayes approach [21] that aims at providing **P**robably **A**pproximately **C**orrect guarantees to "Bayesian" learning algorithms specified in terms of a *prior distribution P* (before the observation of the data) and a data-dependent, *posterior distribution Q* over a space of classifiers.

We formulate a learning algorithm that outputs a stochastic classifier, called the *Gibbs Classifier* G_Q, defined by a data-dependent posterior Q. Our classifier will be partly stochastic in the sense that we will formulate a posterior over the threshold values utilized by the decision stumps while still retaining the deterministic nature for the selected attributes and directions for the decision stumps.

Given an input example $\mathbf{x}$, the Gibbs classifier first selects a classifier h according to the posterior distribution Q and then uses h to assign the label $h(\mathbf{x})$ to $\mathbf{x}$. The risk of G_Q is defined as the expected risk of classifiers drawn according to Q:

$$R(G_Q) \stackrel{\text{def}}{=} \mathbf{E}_{h\sim Q} R(h) = \mathbf{E}_{h\sim Q}\mathbf{E}_{(\mathbf{x},y)\sim D} I(h(\mathbf{x}) \neq y).$$

Our starting point is the PAC-Bayes theorem [16], [22], [23] that provides a bound on the risk of the Gibbs classifier:

Theorem 4. *Given any space $\mathcal{H}$ of classifiers. For any data-independent prior distribution P over $\mathcal{H}$, we have, with probability at least $1 - \delta$ over random draws of $S \sim D^m$,*

$$\forall Q : \mathrm{kl}(R_S(G_Q)\|R(G_Q)) \leq \frac{\mathrm{KL}(Q\|P) + \ln\frac{m+1}{\delta}}{m},$$

where $\mathrm{KL}(Q\|P)$ is the Kullback-Leibler divergence between distributions[4] Q and P:

$$\mathrm{KL}(Q\|P) \stackrel{\text{def}}{=} \mathbf{E}_{h\sim Q} \ln\frac{Q(h)}{P(h)},$$

and where $\mathrm{kl}(q\|p)$ is the Kullback-Leibler divergence between the Bernoulli distributions with probabilities of success q and p:

$$\mathrm{kl}(q\|p) \stackrel{\text{def}}{=} q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}.$$

This bound for the risk of Gibbs classifiers can easily be turned into a bound for the risk of Bayes classifiers B_Q over the posterior Q. B_Q basically performs a majority vote (under measure Q) of binary classifiers in $\mathcal{H}$. When B_Q misclassifies an example $\mathbf{x}$, at least half of the binary classifiers (under measure Q) misclassifies $\mathbf{x}$. It follows that the error rate of G_Q is at least half of the error rate of B_Q. Hence, $R(B_Q) \leq 2R(G_Q)$.

In our case, we have seen that decision stump conjunctions are specified in terms of a mixture of discrete parameters $\mathbf{k}$ and $\mathbf{d}$ and continuous parameters $\mathbf{t}$. If we denote by $P_{\mathbf{k},\mathbf{d}}(\mathbf{t})$ the probability density function associated with a prior P over the class of decision stump conjunctions, we consider here priors of the form

$$P_{\mathbf{k},\mathbf{d}}(\mathbf{t}) = \frac{1}{\binom{n}{|\mathbf{k}|}} p(|\mathbf{k}|) \frac{1}{2^{|\mathbf{k}|}} \prod_{j\in\mathbf{k}} \frac{I(t_j \in [A_j, B_j])}{B_j - A_j}.$$

As before, we have that

$$\sum_{\mathbf{k}\in\mathcal{I}} \sum_{\mathbf{d}\in\mathcal{D}_\mathbf{k}} \prod_{j\in\mathbf{k}} \int_{A_j}^{B_j} dt_j P_{\mathbf{k},\mathbf{d}}(\mathbf{t}) = 1,$$

whenever $\sum_{e=0}^{n} p(e) = 1$.

The factors relating to the discrete components $\mathbf{k}$ and $\mathbf{d}$ have the same rationale as in the case of the Occam's Razor approach. However, in the case of the threshold for each decision stumps, we now consider an explicitly continuous uniform prior. As in the Occam's Razor case, we assume

4. Here $Q(h)$ denotes the probability density function associated with Q, evaluated at h.

each attribute value x_k to be constrained, a priori, in $[A_k, B_k]$ such that A_k and B_k are obtained from the definition of the data. Hence, we have chosen a uniform prior probability density on $[A_k, B_k]$ for each t_k such that $k \in \mathbf{k}$. This explains the last factors of $P_{\mathbf{k},\mathbf{d}}(\mathbf{t})$.

Given a training set S, the learner will choose an attribute group $\mathbf{k}$ and a direction vector $\mathbf{d}$ deterministically. We pose the problem of choosing the threshold in a similar manner as in the case of Occam's Razor approach of Section 3 with the only difference that the learner identifies the interval and selects a threshold stochastically. For each attribute $x_k \in [A_k, B_k] : k \in \mathbf{k}$, a margin interval $[a_k, b_k] \subseteq [A_k, B_k]$ is chosen by the learner. A deterministic decision stump conjunction classifier is then specified by choosing the thresholds values $t_k \in [a_k, b_k]$ uniformly. It is tempting at this point to choose $t_k = (a_k + b_k)/2 \ \forall k \in \mathbf{k}$ (i.e., in the middle of each interval). However, the PAC-Bayes theorem offers a better guarantee for another type of deterministic classifier, as we see below.

Hence, the Gibbs classifier is defined with a posterior distribution Q having all its weight on the same $\mathbf{k}$ and $\mathbf{d}$ as chosen by the learner but where each t_k is uniformly chosen in $[a_k, b_k]$. The KL divergence between this posterior Q and the prior P is then given by

$$KL(Q\|P) = \ln\left(\binom{n}{|\mathbf{k}|} \cdot \frac{2^{|\mathbf{k}|}}{p(|\mathbf{k}|)}\right) + \sum_{k\in\mathbf{k}} \ln\left(\frac{B_k - A_k}{b_k - a_k}\right).$$

In the limit, when $[a_k, b_k] = [A_k, B_k] \ \forall k \in \mathbf{k}$, it can be seen that the KL divergence between the "continuous components" of Q and P vanishes. Furthermore, the KL divergence between the "discrete components" of Q and P is small for small values of $|\mathbf{k}|$ (whenever $p(|\mathbf{k}|)$ is not too small). *Hence, this KL divergence between our choices for Q and P exhibits a trade-off between margins ($b_k - a_k$) and sparsity (small value of $|\mathbf{k}|$) for Gibbs classifiers*. Theorem 4 suggests that the G_Q with the smallest guarantee of risk $R(G_Q)$ should minimize a nontrivial combination of $KL(Q\|P)$ and $R_S(G_Q)$.

The posterior Q is identified by an attribute group vector $\mathbf{k}$, a direction vector $\mathbf{d}$, and intervals $[a_k, b_k] \ \forall k \in \mathbf{k}$. We refine the notation for our Gibbs classifier G_Q to reflect this. Hence, we use $G^{\mathbf{kd}}_{\mathbf{ab}}$, where $\mathbf{a}$ and $\mathbf{b}$ are the vectors formed by the unions of a_ks and b_ks, respectively. We can obtain a closed-form expression for $R_S(G^{\mathbf{kd}}_{\mathbf{ab}})$ by first considering the risk $R_{(\mathbf{x},y)}(G^{\mathbf{kd}}_{\mathbf{ab}})$ on a single example $(\mathbf{x}, y)$ since $R_S(G^{\mathbf{kd}}_{\mathbf{ab}}) = \mathbf{E}_{(\mathbf{x},y)\sim S} R_{(\mathbf{x},y)}(G^{\mathbf{kd}}_{\mathbf{ab}})$. From our definition for Q, we find that

$$R_{(\mathbf{x},y)}\big(G^{\mathbf{kd}}_{\mathbf{ab}}\big) = (1-2y)\left[\prod_{k\in\mathbf{k}} \sigma^{d_k}_{a_k,b_k}(x_k) - y\right], \tag{5}$$

where

$$\sigma^d_{a,b}(x) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } (x < a \text{ and } d = +1) \text{ or } (b < x \text{ and } d = -1), \\ \dfrac{x-a}{b-a}, & \text{if } a \le x \le b \text{ and } d = +1, \\ \dfrac{b-x}{b-a}, & \text{if } a \le x \le b \text{ and } d = -1, \\ 1, & \text{if } (b < x \text{ and } d = +1) \text{ or } (x < a \text{ and } d = -1). \end{cases}$$

Note that the expression for $R_{(\mathbf{x},y)}(C^{\mathbf{k}}_{\mathbf{td}})$ is identical to the expression for $R_{(\mathbf{x},y)}(G^{\mathbf{kd}}_{\mathbf{ab}})$ except that the piece-wise linear functions $\sigma^{d_k}_{a_k,b_k}(x_k)$ are replaced by the indicator functions $I((x_k - t_k)d_k > 0)$.

The PAC-Bayes theorem provides a risk bound for the Gibbs classifier $G^{\mathbf{kd}}_{\mathbf{ab}}$. Since the Bayes classifier $B^{\mathbf{kd}}_{\mathbf{ab}}$ just performs a majority vote under the same posterior distribution as the one used by $G^{\mathbf{kd}}_{\mathbf{ab}}$, it follows that

$$B^{\mathbf{kd}}_{\mathbf{ab}}(\mathbf{x}) = \begin{cases} 1, & \text{if } \displaystyle\prod_{k\in\mathbf{k}} \sigma^{d_k}_{a_k,b_k}(x_k) > 1/2, \\ 0, & \text{if } \displaystyle\prod_{k\in\mathbf{k}} \sigma^{d_k}_{a_k,b_k}(x_k) \le 1/2. \end{cases} \tag{6}$$

Note that $B^{\mathbf{kd}}_{\mathbf{ab}}$ has a *hyperbolic* decision surface. Consequently, $B^{\mathbf{kd}}_{\mathbf{ab}}$ is not representable as a conjunction of decision stumps. There is, however, no computational difficulty in obtaining the output of $B^{\mathbf{kd}}_{\mathbf{ab}}(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{X}$. We now state our main theorem.

Theorem 5. *Given all our previous definitions, for any $\delta \in (0, 1]$ and for any p satisfying $\sum_{e=0}^{n} p(e) = 1$, we have, with probability at least $1 - \delta$ over random draws of $S \sim D^m$,*

$$\left(\forall \mathbf{k}, \mathbf{d}, \mathbf{a}, \mathbf{b} : R\big(G^{\mathbf{kd}}_{\mathbf{ab}}\big) \le \sup\left\{\epsilon : \mathrm{kl}\big(R_S(G^{\mathbf{kd}}_{\mathbf{ab}})\|\epsilon\big) \le \psi\right\}\right),$$

where

$$\psi = \frac{1}{m}\left[\ln\left(\binom{n}{|\mathbf{k}|} \cdot \frac{2^{|\mathbf{k}|}}{p(|\mathbf{k}|)} \cdot \frac{m+1}{\delta}\right) + \sum_{k\in\mathbf{k}} \ln\left(\frac{B_k - A_k}{b_k - a_k}\right)\right].$$

Furthermore, $R(B^{\mathbf{kd}}_{\mathbf{ab}}) \le 2R(G^{\mathbf{kd}}_{\mathbf{ab}}) \ \forall \mathbf{k}, \mathbf{d}, \mathbf{a}, \mathbf{b}$.

6 The Learning Algorithms

Having proposed the theoretical frameworks attempting to obtain the optimal classifiers based on various optimization criteria, we now detail the learning algorithms for these approaches. Ideally, we would like to find a conjunction of decision stumps that minimizes the respective risk bounds for each approach. Unfortunately, this cannot be done efficiently in all cases since this problem is at least as hard as the (NP-hard) minimum set cover problem as mentioned by [24]. Hence, we use a set covering greedy heuristic. It consists of choosing the decision stump i with the largest *utility* U_i^{SC} where

$$U_i^{SC} = |Q_i| - p|R_i|, \tag{7}$$

where Q_i is the set of negative examples covered (classified as 0) by feature i, R_i is the set of positive examples misclassified by this feature, and p is a learning parameter that gives a penalty p for each misclassified positive example. Once the feature with the largest U_i is found, we remove Q_i and R_i from the training set S and then repeat (on the remaining examples) until either no more negative examples are present or a maximum number s of features has been reached. This heuristic was also used by [24] in the context of a sample compression classifier called the set covering machine. For our sample compression approach (SC), we use the above utility function U_i^{SC}.

However, for the Occam's Razor and the PAC-Bayes approaches, we need utility functions that can incorporate the optimization aspects suggested by these approaches.

Hence, a utility function for the Occam's razor approach should be one that includes a trade-off of the classifier sparsity in favor of a more succinct representation of the classifier. Our utility function in the next section does precisely this. It considers a coding scheme that forms, together with the classifier sparsity and empirical risk, a basis for the learning bias of the algorithm. A classifier that is more general (that is, gives less empirical risk and potentially a better risk bound) and can be more succinctly described in terms of the bits utilized for representation (independent of the coding scheme) is favored more than one that focuses merely on the sparsity of the solution. This is in contrast to the sample compression approach (the utility function of (7)) where a sparser classifier giving acceptable empirical performance is preferred. Similarly, for the PAC-Bayes approach we aim to obtain a classifier that can trade off sparsity in favor of a more general classifier, as demonstrated by a wider margin around the decision threshold of the stumps. For both these cases, we propose utility functions that allow a user to tune the respective weights of sparsity and generality (expressed in terms of more succinct representation or wider margins) when performing the trade-off. The parameters are determined using cross validation. In the future, we plan to explore if the respective bounds can provide a successful model selection strategy. Let us now propose a utility function for the Occam's Razor approach.

6.1 The Occam's Razor Learning Algorithm

We propose the following learning strategy for Occam's Razor learning of conjunctions of decision stumps: For a fixed l_i and η, let N be the set of negative examples and P be the set of positive examples. We start with $N' = N$ and $P' = P$. Let Q_i be the subset of N' covered by decision stump i, let R_i be the subset of P' covered by decision stump i, and let l_i be the number of bits used to code the threshold of decision stump i. We choose the decision stump i that maximizes the *utility* U_i^{Occam} defined as

$$U_i^{Occam} \stackrel{\text{def}}{=} \frac{|Q_i|}{N'} - p\frac{|R_i|}{P} - \eta \cdot l_i,$$

where p is the *penalty* suffered by covering (and, hence, misclassifying) a positive example and η is the cost of using l_i bits for decision stump i. Once we have found a decision stump maximizing U_i, we update $N' = N' - Q_i$ and $P' = P' - R_i$ and repeat to find the next decision stump until either $N' = \emptyset$ or the maximum number v of decision stumps has been reached (early stopping the greedy). The best values for the learning parameters p, η, and v are determined by cross-validation.

6.2 The PAC-Bayes Learning Algorithm

Theorem 5 suggests that the learner should try to find the Bayes classifier $B_{\mathbf{ab}}^{\mathbf{kd}}$ that uses a small number of attributes (i.e., a small $|\mathbf{k}|$), each with a large separating margin $(b_k - a_k)$, while keeping the empirical Gibbs risk $R_S(G_{\mathbf{ab}}^{\mathbf{kd}})$ at a low value. As discussed earlier, we utilize the greedy set covering heuristic for learning.

In our case, however, we need to keep the Gibbs risk on S low instead of the risk of a deterministic classifier. Since the Gibbs risk is a "soft measure" that uses the piece-wise linear functions $\sigma_{a,b}^d$ instead of the "hard" indicator functions, we cannot make use of the hard utility function of (7). Instead, we need a "softer" version of this utility function to take into account covering (and erring on) partly an example. That is, a negative example that falls in the linear region of a $\sigma_{a,b}^d$ is in fact partly covered and vice versa for the positive example.

Following this observation, let $\mathbf{k}'$ be the vector of indices of the attributes that we have used so far for the construction of the classifier. Let us first define the *covering value* $\mathcal{C}(G_{\mathbf{ab}}^{\mathbf{k'd}})$ of $G_{\mathbf{ab}}^{\mathbf{k'd}}$ by the "amount" of negative examples assigned to class 0 by $G_{\mathbf{ab}}^{\mathbf{k'd}}$:

$$\mathcal{C}\big(G_{\mathbf{ab}}^{\mathbf{k'd}}\big) \stackrel{\text{def}}{=} \sum_{(\mathbf{x},y)\in S} (1-y)\left[1 - \prod_{j\in\mathbf{k}'} \sigma_{a_j,b_j}^{d_j}(x_j)\right].$$

We also define the *positive-side error* $\mathcal{E}(G_{\mathbf{ab}}^{\mathbf{k'd}})$ of $G_{\mathbf{ab}}^{\mathbf{k'd}}$ as the "amount" of positive examples assigned to class 0:

$$\mathcal{E}\big(G_{\mathbf{ab}}^{\mathbf{k'd}}\big) \stackrel{\text{def}}{=} \sum_{(\mathbf{x},y)\in S} y\left[1 - \prod_{j\in\mathbf{k}'} \sigma_{a_j,b_j}^{d_j}(x_j)\right].$$

We now want to add another decision stump on another attribute, call it i, to obtain a new vector $\mathbf{k}''$ containing this new attribute in addition to those present in $\mathbf{k}'$. Hence, we now introduce the *covering contribution* of decision stump i as

$$\begin{aligned}\mathcal{C}_{\mathbf{ab}}^{\mathbf{k'd}}(i) &\stackrel{\text{def}}{=} \mathcal{C}\big(G_{\mathbf{a'b'}}^{\mathbf{k''d'}}\big) - \mathcal{C}\big(G_{\mathbf{ab}}^{\mathbf{k'd}}\big)\\ &= \sum_{(\mathbf{x},y)\in S} (1-y)\big[1 - \sigma_{a_i,b_i}^{d_i}(x_i)\big] \prod_{j\in\mathbf{k}'} \sigma_{a_j,b_j}^{d_j}(x_j),\end{aligned}$$

and the *positive-side error contribution* of decision stump i as

$$\begin{aligned}\mathcal{E}_{\mathbf{ab}}^{\mathbf{k'd}}(i) &\stackrel{\text{def}}{=} \mathcal{E}\big(G_{\mathbf{a'b'}}^{\mathbf{k''d'}}\big) - \mathcal{E}\big(G_{\mathbf{ab}}^{\mathbf{k'd}}\big)\\ &= \sum_{(\mathbf{x},y)\in S} y\big[1 - \sigma_{a_i,b_i}^{d_i}(x_i)\big] \prod_{j\in\mathbf{k}'} \sigma_{a_j,b_j}^{d_j}(x_j).\end{aligned}$$

Typically, the covering contribution of decision stump i should increase its "utility" and its positive-side error should decrease it. Moreover, we want to decrease the "utility" of decision stump i by an amount which would become large whenever it has a small separating margin. Our expression for $KL(Q\|P)$ suggests that this amount should be proportional to $\ln((B_i - A_i)/(b_i - a_i))$. Furthermore, we should compare this margin term with the *fraction* of the remaining negative examples that decision stump i has covered (instead of the absolute amount of negative examples covered). Hence, the covering contribution $\mathcal{C}_{\mathbf{ab}}^{\mathbf{k'd}}(i)$ of decision stump i should be divided by the amount $\mathcal{N}_{\mathbf{ab}}^{\mathbf{k'd}}$ of negative examples that *remains* to be covered before considering decision stump i:

$$\mathcal{N}_{\mathbf{ab}}^{\mathbf{k'd}} \stackrel{\text{def}}{=} \sum_{(\mathbf{x},y)\in S} (1-y) \prod_{j\in\mathbf{k}'} \sigma_{a_j,b_j}^{d_j}(x_j),$$

which is simply the amount of negative examples that have been assigned to class 1 by $G_{\mathbf{ab}}^{\mathbf{k'd}}$. If P denotes the set of positive examples, we define the *utility* $U_{\mathbf{ab}}^{\mathbf{k'd}}(i)$ *of adding decision stump* i *to* $G_{\mathbf{ab}}^{\mathbf{k'd}}$ as

$$U_{\mathbf{ab}}^{\mathbf{k'd}}(i) \stackrel{\text{def}}{=} \frac{\mathcal{C}_{\mathbf{ab}}^{\mathbf{k'd}}(i)}{\mathcal{N}_{\mathbf{ab}}^{\mathbf{k'd}}} - p\frac{\mathcal{E}_{\mathbf{ab}}^{\mathbf{k'd}}(i)}{|P|} - \eta \ln \frac{B_i - A_i}{b_i - a_i},$$

where parameter p represents the *penalty* for misclassifying a positive example and η is another parameter that controls the importance of having a large margin. These learning parameters can be chosen by cross-validation. For fixed values of these parameters, the "soft greedy" algorithm simply consists of adding to the current Gibbs classifier a decision stump with maximum added utility until either the maximum number v of decision stumps has been reached or all the negative examples have been (totally) covered. It is understood that, during this soft greedy algorithm, we can remove an example $(\mathbf{x}, y)$ from S whenever it is totally covered. This occurs whenever $\prod_{j \in \mathbf{k}'} \sigma_{a_j,b_j}^{d_j}(x_j) = 0$.

Hence, we use the above utility function for the PAC-Bayes learning strategy. Note that, in the case of U_i^{PB} and U_i^{Occam}, we normalize the number of covered and erred examples so as to increase their sensitivity to the respective η terms.

6.2.1 Time Complexity Analysis

Let us analyze the time complexity of this algorithm for fixed p and η. For each attribute, we first sort the m examples with respect to their values for the attribute under consideration. This takes $O(m \log m)$ time. Then, we examine each potential a_i value (defined by the values of that attribute on the examples). Corresponding to each a_i, we examine all the potential b_i values (all the values greater than a_i). This gives us a time complexity of $O(m^2)$. Now, if k is the largest number of examples falling into $[a_i, b_i]$, calculating the covering and error contributions and then finding the best interval $[a_i, b_i]$ takes $O(km^2)$ time. Moreover, we allow $k \in O(m)$ giving us a time complexity of $O(m^3)$ for each attribute. Finally, we do this over all the attributes. Hence, the overall time complexity of the algorithm is $O(nm^3)$. Note, however, that for microarray data, we have $n \gg m$ (hence, we can consider m^3 to be a constant). Moreover, once the best stump is found, we remove the examples covered by this stump from the training set and repeat the algorithm. Now, we know that greedy algorithms of this kind have the following guarantee: If there exist r decision stumps that cover all the m examples, the greedy algorithm will find at most $r \ln(m)$ decision stumps. Since we almost always have $r \in O(1)$, the running time of the whole algorithm will almost always be $\in O(nm^3 \log(m))$. The good news is, since $n \gg m$, the time complexity of our algorithm is roughly linear in n.

6.2.2 Fixed-Margin Heuristic

In order to show why we prefer a uniformly distributed threshold as opposed to the one fixed at the middle of the interval $[a_i, b_i]$ for each stump i, we use an alternate algorithm that we call the fixed margin heuristic. The algorithm is similar to the one described above but with an additional parameter γ. This parameter decides a fixed margin boundary around the threshold, i.e., γ decides the length of the interval $[a_i, b_i]$. The algorithm still chooses the attribute vector $\mathbf{k}$, the direction vector $\mathbf{d}$, and the vectors $\mathbf{a}$ and $\mathbf{b}$. However, the a_is and b_is for each stump i are chosen such that $|b_i - a_i| = 2\gamma$. The threshold t_i is then fixed in the middle of this interval, that is, $t_i = \frac{(a_i+b_i)}{2}$. Hence, for each stump i, the interval $[a_i, b_i] = [t_i - \gamma, t_i + \gamma]$. For fixed p and γ, a similar analysis as in the previous section yields a time complexity of $O(nm^2 \log(m))$ for this algorithm.

7 Empirical Results

The focus of this work is to propose approaches for identification of most modulated genes in differential microarray experiments. Our specification of the decision stump feature based on simple threshold features was strongly motivated in the manner in which the genes undergo expression change under differential conditions. To study the effectiveness of the proposed approaches, these were tested on nine real-world binary microarray data sets. One limitation of the current microarray data sets is their limited sample size. The gene identification tasks are well known to be marred by the curse of dimensionality. Even though the proposed approaches, in particular the PAC-Bayes approach, perform competitively with other state-of-the-art approaches while at the same time identifying a minimal number of genes, as we show below, the theoretical strength of this approach remains underappreciated as a result of invariably loose bounds due to sample size limitations. In order to demonstrate this theoretical strength, as well as explore wider applicabililty of the decision stumps framework, we compare the PAC-Bayesian approach with other state-of-the-art approaches on a set of handwritten digit recognition tasks from the MNIST database [25] in Section 7.1, where we further discuss the theoretical considerations.

The nine real-world binary microarray data sets used were the *colon tumor* [3], the *Leukemia* [4], the *B_MD* and *C_MD* Medulloblastomas data [26], the *Lung* [27], the *BreastER* data [28], the Celliac disease data [29], the Colon epithelial biopsies data [30], and the multiple mylenoma and bone lesion data [31]. The *colon tumor* data set [3] provides the expression levels of 40 tumor and 22 normal colon tissues measured for 6,500 human genes. We use the set of 2,000 genes identified to have the highest minimal intensity across the 62 tissues. The *Leuk* data set [4] provides the expression levels of 7,129 human genes for 47 samples of patients with Acute Lymphoblastic Leukemia (ALL) and 25 samples of patients with Acute Myeloid Leukemia (AML). The *B_MD* and *C_MD* data sets [26] are microarray samples containing the expression levels of 7,129 human genes. Data set *B_MD* contains 25 classic and 9 desmoplastic medulloblastomas, whereas data set *C_MD* contains 39 medulloblastomas survivors and 21 treatment failures (nonsurvivors). The *Lung* data set consists of gene expression levels of 918 genes of 52 patients with 39 Adenocarcinoma and 13 Squamous Cell Cancer [27]. This data has some missing values which were replaced by zeros. Finally, the *BreastER* data set is the Breast Tumor data of [28] used with Estrogen Receptor status to label the various samples. The data consist of expression levels of 7,129 genes of 49 patients with 25 positive Estrogen Receptor samples and 24 negative Estrogen Receptor samples. The Celliac disease analysis of [29] studies the effects of genetic variations on the gene expression of 22,185 genes in primary leukocytes from an immune-mediated disease over 132 samples. The

TABLE 1
Results of SVM, SVM Coupled with Golub's Feature Selection Algorithm (Filter), SVM with Recursive Feature Elimination (Wrapper), Adaboost Algorithms, and L0 Feature Selection on Gene Expression Data Sets

Data Set			SVM	SVM+gs		SVM+rfe		Adaboost		L0	
Name	ex	Genes	Errs	Errs	S	Errs	S	Itrs	Errs	Errs	S
Colon	62	2000	12.8±1.4	14.4±3.5	256	15.4±4.8	128	20	15.2±2.1	22.2±-*	20*
B_MD	34	7129	13.2±1	7.2±2.6	32	10.4±2.4	64	20	9.8±1.1	8.74±-*	4-3500*
C_MD	60	7129	28.2±2.2	23.1±2.8	1024	28.2±2.2	7129	50	21.2±2.4	21±1.8	13
Leuk	72	7129	21.3±1.4	14±2.8	64	21±3.2	256	20	17.8±1.8	25.03±1.5	10
Lung	52	918	8.8±1.3	6.8±1.9	64	7.2±1.8	32	1	2.4±1.4	3.87±1.1	50
BreastER	49	7129	15.3±2.4	10.3±2.7	256	11.2±2.8	256	50	9.8±1.7	-*	-*
Celliac	132	22185	16.2±3.2	28.01±2.3	20	22.1±1.6	150	500	20.95±0.95	10.1±1.1	100
Colon-epi	202	44290	26±2.6	76.64±3.1	5	38.01±2.4	2769	500	32.8±4.14	30.02±2.5	100
Bone Lesion	173	12625	39±3.1	36.33±2.2	1000	36.3±2.3	150	500	38.2±2.1	36.33±-*	100*

An "" indicates that the resulting matrix, partially due to limited precision afforded by the implementation platform, tends to be singular. Consequently, the corresponding estimates are less reliable. (For instance, in some cases such as the Colon and Bone Lesion data sets, the same accuracy estimates were obtained for different sizes of gene subsets. We have chosen the smallest subset in these cases.) A "-" indicates that a solution was not found.*

TABLE 2
Results of the Proposed Occam's Razor, Sample Compression, and PAC-Bayes Learning Algorithms on Gene Expression Data Sets

Data Set			Occam		SC		PAC-Bayes		
Name	ex	Genes	Errs	S	Errs	S	S	G-errs	B-errs
Colon	62	2000	23.6±1.2	1.8±.6	18.2±1.8	1.2±.6	1.53±.28	14.68±1.8	14.65±1.8
B_MD	34	7129	17.2±1.8	1.2±.8	17.2±1.3	1.4±.8	1.2±.25	8.89±1.65	8.6±1.4
C_MD	60	7129	28.6±1.8	2.6±1.1	29.2±1.1	1.2±.6	3.4±1.8	23.8±1.7	22.9±1.65
Leuk	72	7129	27.8±1.7	2.2±.8	27.3±1.7	1.4±.7	3.2±1.4	24.4±1.5	23.6±1.6
Lung	52	918	21.7±1.1	1.8±1.2	18±1.3	1.2±.5	1.2±.3	4.4±.6	4.2±.8
BreastER	49	7129	25.4±1.2	3.2±.6	21.2±1.5	1.4±.5	2.6±1.1	12.8±.8	12.4±.78
Celliac	132	22185	46±2.1	3.1±1.1	42.1±1.4	2.1±.5	6.2±1.4	30±1.7	26±1.1
Colon-epi	202	44290	88.1±1.2	4.4±1.1	74.4±1.8	3.2±1.2	11±4.2	67±4.8	62±3.4
Bone Lesions	173	12625	44.6±1.4	6.6±1.2	42.3±1.8	5.2±1.1	10.2±2.2	34.4±2.2	32±1.4

Colon epithelial biopsies' data of [30] were obtained by analyzing inflamed and uninflamed colon epithelial biopsies over 202 samples (over 67 ulcerative colitis (UC) patients) from different anatomical locations of the gastrointestinal (GI) tract. The study analyzed a total of 44,290 genes per sample. Finally, the multiple mylenoma and bone lesion data [31] were obtained by comparing the expression of 12,625 genes in bone marrow plasma cells of 173 multiple myeloma patients with and without bone lesions. The number of examples and the number of genes in each data are given in the "ex" and "Genes" columns, respectively, under the "Data Set" tab in each table. The algorithms are referred to as "Occam" (Occam's Razor), "SC" (Sample Compression) and "PAC-Bayes" (PAC-Bayes) in the results tables. They utilize the respective theoretical frameworks discussed in Sections 3, 4, and 5, along with the respective learning strategies of Section 6.

We have compared our learning algorithm with a linear-kernel soft-margin SVM trained both on all the attributes (gene expressions) and on a subset of attributes chosen by the filter method of [4]. The filter method consists of ranking the attributes as a function of the difference between the positive-example mean and the negative-example mean and then using only the first ℓ attributes. The resulting learning algorithm, named *SVM+gs*, is the one used by [9] for the same task. Guyon et al. [10] claimed to obtain better results with the recursive feature elimination method, but, as pointed out by [32], their work contained a methodological flaw. We use the SVM recursive feature elimination algorithm with this bias removed and present these results as well for comparison (referred to as "SVM+rfe" in Table 1). Finally, we also compare our results with the state-of-the-art Adaboost algorithm. For this, we use the implementation in the Weka data mining software [33].

Each algorithm was tested over 20 random permutations of the data sets, with the 5-fold cross validation (CV) method. Each of the five training sets and testing sets was the same for all algorithms. The learning parameters of all algorithms and the gene subsets (for "SVM+gs" and "SVM+rfe") were chosen from the training sets *only*. This was done by performing a second (nested) 5-fold CV on each training set.

For the gene subset selection procedure of SVM+gs, we have considered the first $\ell = 2^i$ genes (for $i = 0, 1, \ldots, 12$) ranked according to the criterion of [4] and have chosen the i value that gave the smallest 5-fold CV error on the training set. The "Errs" column under each algorithm in Tables 1 and 2 refers to the average (nested) 5-fold cross-validation error of the respective algorithm with one standard deviation two-sided confidence interval. The "bits" column in Table 2 refers to the number of bits used for the Occam' Razor approach. The "G-errs" and the "B-errs" columns in Table 2 refer to the average nested 5-fold CV error of the optimal Gibbs classifier and the corresponding Bayes classifier with one standard deviation two-sided interval, respectively.

For Adaboost, 10, 20, 50, 100, 200, 500, 1,000, and 2,000 iterations for each data sets were tried and the reported results correspond to the best obtained 5-fold CV error. The size values reported here (the "S" columns for "SVM+gs"and "SVM+rfe," and "Itr" column for "AdaBoost" in Table 1) correspond to the number of attributes (genes) selected most frequently by the respective algorithms over all the permutation runs.[5] Choosing, by cross-validation, the number of the boosting iteration is somewhat inconsistent with Adaboost's goal of minimizing the empirical exponential risk. Indeed, to comply with Adaboost's goal, we should choose a large-enough number of boosting rounds that assures the convergence of the empirical exponential risk to its minimum value. However, as shown by [34], Boosting is known to overfit when the number of attributes exceeds the number of examples. This happens in the case of microarray experiments frequently where the number of genes far exceeds the number of samples, and is also the case in the data sets mentioned above. Early stopping is the recommended approach in such cases and hence we have followed the method described above to obtained the best number of boosting iterations.

7.1 Efficacy of the Risk Bound and the Fixed-Margin Heuristic

Table 3 gives the result for a single run of the deterministic algorithm using the fixed-margin heuristic described above as well as the results for the PAC-Bayes bound values for the results obtained for a single run of the PAC-Bayes algorithm on the respective microarray data sets. Recall that the PAC-Bayes bound provides a uniform upper bound on the risk of the Gibbs classifier. The column labels refer to the same quantities as above although the errors reported are over a single nested 5-fold CV run. The "Bound" columns of Table 3 refer to the average risk bound of Theorem 5 multiplied by the total number of examples in respective

5. There were no close ties with classifiers with fewer genes.

TABLE 3
An Illustration of the PAC-Bayes Risk Bound on a Sample Run of the PAC-Bayes Algorithm and Comparison with the Fixed-Margin Heuristic on Gene Expression Data Sets

Data Set			Stumps:PAC-Bayes(fixed margin)			Stumps:PAC-Bayes (var. margin)			
Name	ex	Genes	Size	Errors	Bound	Size	G-errs	B-errs	Bound
Colon	62	2000	1	14	34	1	12	11	33
B_MD	34	7129	1	7	20	1	7	7	20
C_MD	60	7129	3	28	48	5	21	20	45
Leuk	72	7129	2	21	46	3	22	21	48
Lung	52	918	2	9	29	1	3	3	18
BreastER	49	7129	3	11	31	2	11	11	29
Celliac	132	22185	3	29	49	3	26	24	59
Colon-epi	202	44290	7	69	94	9	63	59	113
Bone Lesions	173	12625	8	38	69	8	34	31	78

TABLE 4
Results of Various Feature Selection Algorithms and the Corresponding Risk Bound for the PAC-Bayes Approach on Different Binary Handwritten Digit Recognition Tasks from the MNIST Database of [25]

Data Set			SVM-rfe		L0		PAC-Bayes			
Name	ex	Attributes	Errs	S	Errs	S	S	G-errs	B-Errs	Bound
MNIST-0vs8	2416	784	0.033	50	0.03	40	5	0.031	0.031	0.06
MNIST-1vs7	2422	784	0.04	50	0.038	50	8	0.0342	0.034	0.07
MNIST-1vs8	2436	784	0.063	100	0.064	100	7	0.063	0.063	0.13
MNIST-2vs3	2405	784	0.09	100	0.096	100	4	0.13	0.12	0.17

The errors shown are as a fraction of training examples misclassified to demonstrate direct correlation with the calculated bound.

data sets. Note, again, that these results are on a single permutation of the data sets and are presented just to illustrate the practicality of the risk bound and the rationale of not choosing the fixed-margin heuristic over the current learning strategy.

Note that the risk bounds are quite effective and their relevance should not be misconstrued by observing the results in just the microarray data scenario. One of the most limiting factors in the current analysis is the unavailability of microarray data with larger number of examples. As the number of examples increases, the risk bound of Theorem 5 gives tighter guarantees. Consider, for instance, if the data sets for the Lung and Colon Cancer had 500 examples. A classifier with the same performance over 500 examples (i.e., with the same classification accuracy and number of features as currently) would have a bound of about 12 and 30 percent error instead of current 34.6 and 54.6 percent, respectively. This only illustrates how the bound can be more effective as a guarantee when used on data sets with more examples. Similarly, a data set of 1,000 examples for Breast Cancer with a similar performance can have a bound of about 30 percent instead of the current 63 percent. Hence, the current limitation in the practical application of the bound comes from limited data availability. As the number of examples increases, the bounds provides tighter guarantees and become more significant. The following application sheds more insight into the behavior of the risk bound.

7.1.1 Handwritten Digit Recognition Tasks

As we saw above, the limited sample size of various microarray data sets results in relatively loose risk bounds despite competitive performance and meaningful feature selection performed by different approaches. We have found that the PAC-Bayes approach, as also validated by the results above, yields a framework that can successfully perform a nontrivial margin-sparsity trade-off resulting in more general classifiers. In order to examine the wider utility of the proposed PAC-Bayes approach as well as better appreciate the risk bound that it yields, we have compared its performance with that of other competitive feature selection approaches on a set of handwritten digit recognition tasks from the MNIST database [25]. In accordance with [35] and [36], we have obtained four binary digit classification data sets, viz. classifying digit 0 from digit 8, digit 1 from digit 7, digit 1 from digit 8, and digit 2 from digit 3 since these represent the difficulties in digit classification by taking into account more closely resembling digit pairs. The results presented in Table 4 correspond to the nested 5-fold cross validation runs for each of the algorithms with the nested cross validation runs used to perform model selection. The columns labels denote the same quantities as in earlier tables.

The results of Table 4 clearly demonstrate the utility and the relevance of the risk bounds. The advantage of larger sample size in the handwritten digit recognition domain, even in high-dimensional settings, results not only in general classifiers with a significantly smaller number of features compared to the state-of-the-art approaches, but also yields tight generalization guarantees. One important point to note is that the set of features used here in the form of variable margin decision stumps are but one possibility in the current framework of learning conjunction or disjunctions of data-dependent features. These features were specifically chosen keeping in perspective the nature of gene expression changes in differential experiments. Other applications can certainly exploit more suitable features sensitive to the chosen context of domain. Nevertheless, this application demonstrates that conjunction of decision stumps in their current form can also be useful in other high-dimensional learning settings.

8 Analysis

The results clearly show that even though "Occam" and "SC" are able to find sparse classifiers (with very few genes), they are not able to obtain acceptable classification accuracies. One possible explanation is that these two approaches focus on the most succinct classifier with their respective criterion. The sample compression approach tries to minimize the number of genes used but does not take into account the magnitude of the separating margin and hence compromises accuracy. On the other hand, the Occam's Razor approach tries to find a classifier that depends on margin *only indirectly*. Approaches based on sample compression as well as minimum description length have shown encouraging results in various domains. An alternate set of features can possibly yield more competitive results with these approaches. This is a potential direction for future investigation. The PAC-Bayes approach seems to alleviate the problems of these approaches with the current set of features by performing a significant margin-sparsity trade-off. That is, the advantage of adding a new feature is seen in terms of a combination of the gain in both margin and the empirical risk. This can be compared to the strategy used by the regularization approaches. The classification accuracy of PAC-Bayes algorithm is competitive with the best performing classifier but has an added advantage, quite importantly, of using *very few genes*.

For the PAC-Bayes approach, we expect the Bayes classifier to generally perform better than the Gibbs

TABLE 5
Genes Identified by the *Final* PAC-Bayes Classifier on a Subset of Data Sets from Table 2

Dataset	Gene(s) identified by PAC-Bayes Classifier
Colon	1. Hsa627 M26383-Human monocyte-derived neutrophil-activating protein (MONAP) mRNA
B_MD	1. D79205_at-Ribosomal protein L39
C_MD	1. S71824_at-Neural Cell Adhesion Molecule, Phosphatidylinositol-Linked Isoform Precursor
	2. D83542_at-Cadherin-15
	3. U29195_at-NPTX2 Neuronal pentraxin II
	4. X73358_s_at-HAES-1 mRNA
	5. L36069_at-High conductance inward rectifier potassium channel alpha subunit mRNA
Leuk	1. M84526_at-DF D component of complement (adipsin)
	2. U34360_at-Lymphoid nuclear protein (LAF-4) mRNA
	3. M16937_at-Homeo box c1 protein, mRNA
Lung	1. GENE221X-IMAGE_841641-cyclin D1 (PRAD1-parathyroid adenomatosis 1) Hs82932 AA487486
BreastER	1. X03635_at,X03635- class C, 20 probes, 20 in all_X03635 5885 - 6402
	Human mRNA for oestrogen receptor
	2. L42611_f_at, L42611- class A, 20 probes, 20 in L42611 1374-1954,
	Homo sapiens keratin 6 isoform K6e $KRT6E$ mRNA, complete cds

classifier. This is reflected to some extent in the empirical results for the Colon, C_MD, and Leukemia data sets. However, there is no means to prove that this will always be the case. It should be noted that there exist several different utility functions that we can use for each of the proposed learning approaches. We have tried some of these and reported results only for the ones that were found to be the best (and discussed in the description of the corresponding learning algorithms).

A noteworthy observation with regard to Adaboost is that the gene subset identified by this algorithm almost always include the ones found by the proposed PAC-Bayes approach for decision stumps. Most notably, the *only* gene *Cyclin D1*, a well-known marker for Cancer, found for the lung cancer data set is the most discriminating factor and is commonly found by both approaches. In both cases, the size of the classifier is almost always restricted to 1. These observations not only give insights into the absolute peaks worth investigating but also experimentally validate the proposed approaches.

Finally, while the upper bound on the number of genes along with a tight risk bound makes a compelling case, this upper bound does not guarantee the discovery of desired features are discovered.[6] However, qualitative analysis of the discovered genes lends credibility to the proposed approach. Many of the genes identified by the *final*[7] PAC-Bayes classifier include some prominent markers for the corresponding diseases, as detailed below.

8.1 Biological Relevance of the Selected Features

In this section we investigate, for some representative microarray data sets, if the genes identified by the PAC-Bayes learner are indeed meaningful in biological terms. Table 5 details the genes identified by the *final* PAC-Bayes classifier learned over some data sets after the parameter selection phase. There are some prominent markers identified by the classifier. Some of the main genes identified by the PAC-Bayes approach are the ones identified by previous studies for each disease— giving confidence in the proposed approach. Some of the discovered genes in this case include *Human monocyte-derived neutrophil-activating protein (MONAP) mRNA* in the case of Colon Cancer data set and *estrogen receptor* in the case of Breast Cancer data, *D79205_at-Ribosomal protein L39*, *D83542_at-Cadherin-15* and *U29195_at-NPTX2 Neuronal pentraxin II* in the case of Medulloblastomas data sets B_MD and C_MD. Other genes identified have biological relevance, for instance, the identification of *Adipsin*, *LAF-4*, and *HOX1C* with regard to ALL/AML by our algorithm is in agreement with that of the findings of [39], [40], and [41], respectively, and the studies that followed.

Further, in the case of breast cancer, Estrogen receptors (ER) have been shown to interact with BRCA1 to regulate VEGF transcription and secretion in breast cancer cells [42]. These interactions are further investigated by [43]. Further studies for ER have also been done. For instance, [44] discovered three putative estrogen-response elements in Keratin6 (the second gene identified by the PAC-Bayes classifier in the case of BreastER data) in the context of E2-responsive genes identified by microarray analysis of MDA-MD-231 cells that reexpress ER_α. An important role played by cytokeratins in cancer development is also widely known (see for instance [45]).

Furthermore, the importance of *MONAP* in the case of colon cancer and *Adipsin* in the case of leukemia data has been further confirmed by various rank-based algorithms as detailed by [46] in the implementation of "RankGene," a program that analyzes and ranks genes for the gene expression data using eight ranking criteria including Information Gain (IG), Gini Index (GI), Max Minority (MM), Sum Minority (SM), Twoing Rule (TR), t-statistic (TT), Sum of variances (SV), and one-dimensional Support Vector Machine (1S). In the case of Colon Cancer data, *MONAP* is identified as the top ranked gene by four of the eight criteria (IG, SV, TR, GI), second by one (SM), eighth by one (MM), and in the top 50 by 1S. Similarly, in the case of the Leukemia data, *Adipsin* is top ranked by 1S, fifth by SM, seventh by IG, SV, TR, GI, and MM, and is in the top 50 by TT. These observations provides a strong validation for our approaches.

Cyclin as identified in the case of Lung Cancer data set is a well-known marker for cell division whose perturbations are considered to be one of the major factors causing cancer [47], [48].

Finally, the discovered genes in the case of Medulloblastomas are important with regard to the neuronal functioning (especially S71824, U29195, and L36039) and can have relevance for nervous system related tumors.

9 Conclusion

Learning from high-dimensional data such as that from DNA microarrays can be quite challenging, especially when the aim is to identify only a few attributes that characterize the differences between two classes of data. We investigated the premise of learning conjunctions of *decision stumps* and proposed three formulations based on different learning principles. We observed that the approaches that aim solely to optimize sparsity or the message code with regard to the classifier's empirical risk limit the algorithm in terms of its generalization performance, at least in the present case of small data set sizes. By trading off the sparsity of the classifier with the separating margin in addition to the empirical risk, the PAC-Bayes approach seems to alleviate this problem to a significant extent. This allows the PAC-Bayes algorithm to yield competitive classification performance while at the same time utilizing significantly fewer attributes.

6. Some such guarantees on model selection consistency in the context of ℓ_1 norm are available [37], [38]

7. This is the classifier learned after choosing the best parameters using nested 5-fold CV and trained on the full data set.

As opposed to the traditional feature selection methods, the proposed approaches are accompanied by a *theoretical justification of the performance*. Moreover, the proposed algorithms *embed the feature selection as a part of the learning process* itself.[8] Furthermore, the generalization error bounds are practical (especially as seen in the large sample handwritten digit recognition tasks) and can potentially guide the model (parameter) selection. When applied to classify DNA microarray data, the genes identified by the proposed approaches are found to be biologically significant as experimentally validated by various studies, an empirical justification that the approaches can successfully perform meaningful feature selection. Consequently, this represents a significant improvement in the direction of successful integration of machine learning approaches for use in high-throughput data to provide *meaningful, theoretically justifiable*, and *reliable* results. Such approaches that yield a compressed view in terms of a small number of biological markers can lead to a targeted and well-focused study of the issue of interest. For instance, the approach can be utilized in identifying gene subsets from the microarray experiments that should be further validated using focused RT-PCR techniques which are otherwise both costly and impractical to perform on the full set of genes.

Finally, as mentioned previously, the approaches presented in this work have a wider relevance and can have significant implications in the direction of designing theoretically justified feature selection algorithms. These are among the few approaches that combine the feature selection with the learning process *and* provide generalization guarantees over the resulting classifiers *simultaneously*. This property assumes even more significance in the wake of the limited size of microarray data sets since it limits the amount of empirical evaluation that can be reliably performed otherwise. Most natural extensions of the approaches and the learning bias proposed here would be in other similar domains, including other forms of microarray experiments such as Chromatin Immunoprecipitation promoter arrays (chIP-Chip) and from Protein arrays. Within the same learning settings, other learning biases can also be explored, such as classifiers represented by features or sets of features built on subsets of attributes. More generally, alternate sets of features can also yield general classifier in the proposed sample compression and the Occam's Razor settings than the decision stumps. In fact, just as the choice of variable margin decision stumps is motivated by the application domain of microarray data analysis in the present case, suitable domain-specific features can yield better performances. Another direction of future investigation would be to explore the utility of the risk bound in performing model selection.

ACKNOWLEDGMENTS

This work was supported by the National Science and Engineering Research Council (NSERC) of Canada (Discovery Grant No. 122405 to Mario Marchand), the Canadian Institutes of Health Research (operating grant to Jacques Corbell), and the Canada Research Chair in Medical Genomics to Jacques Corbell.

8. Note that [49] proposed one such approach. However, they need multiple SVM learning runs. Hence, their method basically works as a wrapper.

REFERENCES

[1] M. Eisen and P. Brown, "DNA Arrays for Analysis of Gene Expression," *Methods in Enzymology,* vol. 303, pp. 179-205, 1999.

[2] R. Lipshutz, S. Fodor, T. Gingeras, and D. Lockhart, "High Density Synthetic Oligonucleotide Arrays," *Nature Genetics,* vol. 21, no. 1 Suppl, pp. 20-24, 1999.

[3] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. Levine, "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays," *Proc. Nat'l Academy of Sciences USA,* vol. 96, no. 12, pp. 6745-6750, 1999.

[4] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science,* vol. 286, no. 5439, pp. 531-537, 1999.

[5] L. Wang, F. Chu, and W. Xie, "Accurate Cancer Classification Using Expressions of Very Few Genes," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 4, no. 1, pp. 40-53, Jan.-Mar. 2007.

[6] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Class Predicition by Nearest Shrunken Centroids with Applications to DNA Microarrays," *Statistical Science,* vol. 18, pp. 104-117, 2003.

[7] L. Song, J. Bedo, K.M. Borgwardt, A. Gretton, and A. Smola, "Gene Selection via the BAHSIC Family of Algorithms," *Bioinformatics,* vol. 23, no. 13, pp. 490-498, 2007.

[8] M. Shah and J. Corbeil, "A General Framework for Analyzing Data from Two Short Time-Series Microarray Experiments," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 8, no. 1, pp. 14-26, 2011.

[9] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, and D. Haussler, "Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data," *Bioinformatics,* vol. 16, pp. 906-914, 2000.

[10] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification Using Support Vector Machines," *Machine Learning,* vol. 46, pp. 389-422, 2002.

[11] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. Royal. Statistical Soc. B,* vol. 58, no. 1, pp. 267-288, 1996.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer, 2001.

[13] T. Zhang, "Adaptive Forward-Backward Greedy Algorithm for Sparse Learning with Linear Models," *Proc. Advances in Neural Information Processing Systems,* pp. 1921-1928, 2009.

[14] D. Haussler, "Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework," *Artificial Intelligence,* vol. 36, pp. 177-221, 1988.

[15] M. Marchand and M. Shah, "PAC-Bayes Learning of Conjunctions and Classification of Gene-Expression Data," *Proc. Advances in Neural Information Processing Systems,* pp. 881-888, 2005.

[16] J. Langford, "Tutorial on Practical Prediction Theory for Classification," *J. Machine Learning Research,* vol. 3, pp. 273-306, 2005.

[17] A. Blum and J. Langford, "PAC-MDL Bounds," *Proc. 16th Ann. Conf. Learning Theory,* pp. 344-357, 2003.

[18] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, "Occam's Razor," *Information Processing Letters,* vol. 24, pp. 377-380, 1987.

[19] D. Kuzmin and M.K. Warmuth, "Unlabeled Compression Schemes for Maximum Classes," *J. Machine Learning Research,* vol. 8, pp. 2047-2081, 2007.

[20] M. Marchand and M. Sokolova, "Learning with Decision Lists of Data-Dependent Features," *J. Machine Learning Reasearch,* vol. 6, pp. 427-451, 2005.

[21] D. McAllester, "Some PAC-Bayesian Theorems," *Machine Learning,* vol. 37, pp. 355-363, 1999.

[22] D. McAllester, "PAC-Bayesian Stochastic Model Selection," *Machine Learning,* vol. 51, pp. 5-21, 2003.

[23] M. Seeger, "PAC-Bayesian Generalization Bounds for Gaussian Processes," *J. Machine Learning Research,* vol. 3, pp. 233-269, 2002.

[24] M. Marchand and J. Shawe-Taylor, "The Set Covering Machine," *J. Machine Learning Reasearch,* vol. 3, pp. 723-746, 2002.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE,* vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[26] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y. 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,* vol. 415, no. 6870, pp. 436-442, 2002.

[27] M.E. Garber, O.G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. van de Rijn, G.D. Rosen, C.M. Perou, R.I. Whyte, R.B. Altman, P.O. Brown, D. Botstein, and I. Petersen, "Diversity of Gene Expression in Adenocarcinoma of the Lung," *Proc. Nat'l Academy of Sciences USA,* vol. 98, no. 24, pp. 13784-13789, 2001.

[28] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.A.O. Jr., J.R. Marks, and J.R. Nevins, "Predicting the Clinical Status of Human Breast Cancer by Using Gene Expression Profiles," *Proc. Nat'l Academy of Sciences USA,* vol. 98, no. 20, pp. 11462-11467, 2001.

[29] G.A. Heap et al., "Complex Nature of SNP Genotype Effects on Gene Expression in Primary Human Leucocytes," *BMC Medical Genomics,* vol. 2, no. 1, 2009.

[30] C.L. Noble et al., "Regional Variation in Gene Expression in the Healthy Colon Is Dysregulated in Ulcerative Colitis," *Gut.,* vol. 57, no. 10, pp. 1398-1405, 2008.

[31] E. Tian et al., "The Role of the Wnt-Signaling Antagonist dkk1 in the Development of Osteolytic Lesions in Multiple Myeloma," *New England J. Medicine,* vol. 349, no. 26, pp. 2483-2494, 2003.

[32] C. Ambroise and G.J. McLachlan, "Selection Bias in Gene Extraction on the Basis of Microarray Gene-Expression Data," *Proc. Nat'l Academy Sciences USA,* vol. 99, no. 10, pp. 6562-6566, 2002.

[33] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques,* second ed. Morgan Kaufmann, 2005.

[34] T. Zhang and B. Yu, "Boosting with Early Stopping: Convergence and Consistency," *The Annals of Statistics,* vol. 33, pp. 1538-1579, 2005.

[35] P. Germain, A. Lacasse, F. Laviolette, M. Marchand, and S. Shanian, "From Pac-Bayes Bounds to KL Regularization," *Proc. Advances in Neural Information Processing Systems,* pp. 603-610, 2009.

[36] P. Germain, A. Lacasse, F. Laviolette, and M. Marchand, "PAC-Bayesian Learning of Linear Classifiers," *Proc. 26th Ann. Int'l Conf. Machine Learning,* 2009.

[37] M. Wainwright, "Information-Theoretic Limitations on Sparsity Recovery in the High-Dimensional and Noisy Setting," *IEEE Trans. Information Theory,* vol. 55, no. 12, pp. 5728-5741, Dec. 2009.

[38] P. Ravikumar, G. Raskutti, M. Wainwright, and B. Yu, "Model Selection in Gaussian Graphical Models: High-Dimensional Consistency of ℓ_1-Regularized MLE," *Proc. Advances in Neural Information Processing Systems,* pp. 1329-1336, 2009.

[39] M.L. Chow, E.J. Moler, and I.S. Mian, "Identifying Marker Genes in Transcription Profiling Data Using a Mixture of Feature Relevance Experts," *Physiological Genomics,* vol. 5, no. 2, pp. 99-111, 2001.

[40] M. Hiwatari, T. Taki, T. Taketani, M. Taniwaki, K. Sugita, M. Okuya, M. Eguchi, K. Ida, and Y. Hayashi, "Fusion of an AF4-Related Gene, LAF4, to MLL in Childhood Acute Lymphoblastic Leukemia with t(2;11)(q11;q23)," *Oncogene,* vol. 22, no. 18, pp. 2851-2855, 2003.

[41] H.J. Lawrence and C. Largman, "Homeobox Genes in Normal Hematopoiesis and Leukemia," *Blood,* vol. 80, no. 10, pp. 2445-2453, 1992.

[42] H. Kawai, H. Li, P. Chun, S. Avraham, and H.K. Avraham, "Direct Interaction between BRCA1 and the Estrogen Receptor Regulates Vascular Endothelial Growth Factor (VEGF) Transcription and Secretion in Breast Cancer Cells," *Oncogene,* vol. 21, no. 50, pp. 7730-7739, 2002.

[43] Y.X. Ma, Y. Tomita, S. Fan, K. Wu, Y. Tong, Z. Zhao, L.N. Song, I.D. Goldberg, and E.M. Rosen, "Structural Determinants of the BRCA1 Estrogen Receptor Interaction," *Oncogene,* vol. 24, no. 11, pp. 1831-1846, 2005.

[44] J.G. Moggs, T.C. Murphy, F.L. Lim, D.J. Moore, R. Stuckey, K. Antrobus, I. Kimber, and G. Orphanides, "Anti-Proliferative Effect of Estrogen in Breast Cancer Cells that Re-Express ERalpha is Mediated by Aberrant Regulation of Cell Cycle Genes," *J. Molecular Endocrinology,* vol. 34, pp. 535-551, 2005.

[45] B.A. Gusterson, D.T. Ross, V.J. Heath, and T. Stein, "Basal Cytokeratins and Their Relationship to the Cellular Origin and Functional Classification of Breast Cancer," *Breast Cancer Research,* vol. 7, pp. 143-148, 2005.

[46] Y. Su, T. Murali, V. Pavlovic, M. Schaffer, and S. Kasif, "RankGene: Identification of Diagnostic Genes Based on Expression Data," *Bioinformatics,* vol. 19, no. 12, pp. 1578-1579, 2003.

[47] B. Driscoll, S. Buckley, L. Barsky, K. Weinberg, K.D. Anderson, and D. Warburton, "Abrogation of Cyclin D1 Expression Predisposes Lung Cancer Cells to Serum Deprivation-Induced Apoptosis," *Am. J. Physiology,* vol. 276, no. 4 Pt 1, pp. L679-L687, 1999.

[48] T. Masaki, Y. Shiratori, W. Rengifo, K. Igarashi, M. Yamagata, K. Kurokohchi, N. Uchida, Y. Miyauchi, H. Yoshiji, S. Watanabe, M. Omata, and S. Kuriyama, "Cyclins and Cyclin-Dependent Kinases: Comparative Study of Hepatocellular Carcinoma Versus Cirrhosis," *Hepatology,* vol. 37, no. 3, pp. 534-543, 2003.

[49] H. Huang and F. Chang, "ESVM: Evolutionary Support Vector Machine for Automatic Feature Selection and Classification of Microarray Data," *Biosystems,* vol. 90, no. 2, pp. 516-528, 2007.

Mohak Shah received the PhD degree in computer science from the University of Ottawa in 2006 and was a postdoctoral fellow at Laval and McGill Universities, respectively. Currently, he is working as a researcher at Accenture, Chicago. He is coauthor of *Evaluating Learning Algorithms: A Classification Perspective* (Cambridge University Press, 2011). His research interests span machine learning, statistical learning theory, and their overlap with fields such as bioinformatics, medical imaging, and natural language processing. He is a member of the IEEE and the IEEE Computer Society.

Mario Marchand received the PhD degree in physics from Sherbrooke University in 1987. He was as a professor in the University of Ottawa's Physics Department in 1988 and then moved to the University of Ottawa's Computer Science Department (now the School of Information Technology and Engineering). In 2003, he was appointed a professor in the Computer Science and Software Engineering Department at Laval University and he has been the head of the department since 2008. For more than 20 years, his research activities have focused on learning algorithm design and analysis. More specifically, he is interested in risk bounds, sample-compression, kernel methods and their applications in bioinformatics.

Jacques Corbeil is a professor of medicine in the Department of Molecular Medicine at Laval University. He holds the Canada Research Chair in Medical Genomics (tier 1). His research is centered on the investigation of host-pathogen relationships. He utilizes many approaches to carry out these investigations, applying novel state-of-the-art technologies coupled with the power of bioinformatics. The present focus is on HIV/AIDS and more recently Leishmania, Streptococcus pneumoniae, and respiratory viruses. He is also interested in the metagenomic aspects of microbes in diverse environments. He is part of both the Infectious Disease Research Center and the Computational Biology Center.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.