Fast and Scalable Feature Selection for Gene Expression Data Using Hilbert-Schmidt Independence Criterion

Mehrdad J. Gangeh, Hadi Zarkoob, and Ali Ghodsi

Abstract—*Goal*: In computational biology, selecting a small subset of informative genes from microarray data continues to be a challenge due to the presence of thousands of genes. This paper aims at quantifying the dependence between gene expression data and the response variables and to identifying a subset of the most informative genes using a fast and scalable multivariate algorithm. *Methods*: A novel algorithm for feature selection from gene expression data was developed. The algorithm was based on the Hilbert-Schmidt independence criterion (HSIC), and was partly motivated by singular value decomposition (SVD). *Results*: The algorithm is computationally fast and scalable to large datasets. Moreover, it can be applied to problems with any type of response variables including, biclass, multiclass, and continuous response variables. The performance of the proposed algorithm in terms of accuracy, stability of the selected genes, speed, and scalability was evaluated using both synthetic and real-world datasets. The simulation results demonstrated that the proposed algorithm effectively and efficiently extracted stable genes with high predictive capability, in particular for datasets with multiclass response variables. *Conclusion/Significance*: The proposed method does not require the whole microarray dataset to be stored in memory, and thus can easily be scaled to large datasets. This capability is an important attribute in big data analytics, where data can be large and massively distributed.

Index Terms—Big data, feature selection, gene expression, Hilbert-Schmidt independence criterion, kernel methods, scalability

1 Introduction

In the era of big data, developing algorithms that can handle large and massively distributed data is of crucial importance. A successful algorithm is therefore not only evaluated by its accurate performance to solve the problem at hand, but also on its scalability to large datasets.

In the context of computational biology and bioinformatics, DNA microarrays are capable of measuring the expression levels of thousands of genes, and even of a whole genome in a single experiment. Based on this capability, they have been widely used to extend biological studies to the genomic level. One main property of microarray datasets is that they usually include the expression levels of a large number of genes (*features*) from a limited number of samples. Most of these features, however, are irrelevant when a specific biological problem is being studied. Therefore, the task of feature selection plays an important role in extracting a small subset of genes, such that the output of the experiment mainly depends on this subset [1]. This process enhances both the efficiency and accuracy of the subsequent analysis of the data. In addition, the task of identifying the subset of relevant genes can shed light on the underlying biological process. Considering the constraints in big data analytics, the impetus of research in this area is to design a feature selection algorithm that can easily scale to datasets containing thousands of genes.

Several methods have been proposed for feature selection from DNA microarray data. They can generally be classified into three groups: *filter*, *wrapper*, and *embedded* methods [2], [3]. *Filter* methods use intrinsic properties of the genes to rank them. They can be divided into univariate and multivariate filters. Univariate filters visit genes one-by-one and assign them a degree of relevance. Multivariate filters, on the other hand, aim to select a subset of genes that, together, can describe the response variable. Filter methods based on signal-to-noise ratio, fold change (FC), and t-statistics are examples of univariate filters [4], [5]. The minimum-redundancy-maximum-relevance (mRMR) framework developed by Peng et al. [6] offers an example of multivariate filters. In this framework, the goal is to minimize the number of redundant genes while selecting a subset of the most relevant ones. A more recent multivariate filter approach specifically designed to handle high-dimensional data is Fisher-Markov selector (FMS) [7], which can handle datasets with multiclass response variables. The approach is based on the discriminating notion of linear discriminant analysis (LDA) and Markov random field (MRF) optimization techniques. Unlike filter methods, *wrapper* methods are classifier-dependent because they build and employ a classifier to evaluate the quality of a subset of

- *M.J. Gangeh is with the Department of Medical Biophysics, and Radiation Oncology, University of Toronto, Toronto, ON M5G 2M9, Canada, and the Department of Radiation Oncology, and Imaging Research-Physical Sciences, Sunnybrook Health Sciences Centre, Toronto, ON M4N 3M5, Canada. E-mail: mehrdad.gangeh@utoronto.ca.*
- *H. Zarkoob is with BaseHealth Inc., Sunnyvale, CA 94086. E-mail: hzarkoob@alumni.stanford.edu.*
- *A. Ghodsi is with the Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: aghodsib@uwaterloo.ca.*

Manuscript received 31 Mar. 2015; revised 13 Jan. 2016; accepted 18 Oct. 2016; date of current version 2 Feb. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2016.2631164

genes (see, e.g., [8], [9]). The *embedded* techniques rely on the properties of an underlying classifier to assign degrees of relevance to genes. Support vector machine (SVM) recursive feature elimination (SVM-RFE) [10] is a typical embedded method that is widely used for gene selection. Other examples of embedded methods have been reported, which are based on random forests and logistic regression [11], [12].

A challenge in designing gene selection algorithms is to quantify the dependence between the expression levels of genes and the response variables, which may take a complex form. Another issue in designing feature selection algorithms is the trade-off between accuracy and speed. Multivariate algorithms are in general more accurate, but slower than univariate methods. Univariate methods, on the other hand, are fast, but cannot take into account gene-to-gene interactions when extracting the important genes.

In this paper, we describe a fast and scalable feature selection method for DNA microarrays, based on the Hilbert-Schmidt independence criterion (HSIC) [13], [14], [15]. The HSIC provides a measure of dependence between two random variables. Once a set of observations (realizations) for the two random variables is given, the HSIC can be estimated based on these observations. Song et al. [16], [17] proposed a family of gene selection algorithms based on the HSIC called the backward elimination HSIC (BAHSIC). These algorithms encapsulate a number of well-known gene selection methods, such as fold change, signal-to-noise ratio, shrunken centroid, and ridge regression [17]. Given a set of genes (features), Song et al. used HSIC to evaluate the dependence between the data obtained from these features and the response variables. Having obtained this measure of dependence, they used backward elimination to extract a subset of the most relevant genes. To account for the dependency between features (genes), the BAHSIC requires nonlinear kernels. Otherwise, it reduces to a univariate method when using linear kernels on both the data and labels (in this case, the backward elimination is not needed) [17]. However, using nonlinear kernels, one major drawback of the BAHSIC is its reliance on backward elimination that makes it extremely slow, as shown in the scalability experiments (Section 3.2.4) of this paper.

In this work, a completely different approach is used to accomplish feature selection based on the HSIC. Here, the HSIC has been used in conjunction with a fast technique for sparse decomposition of matrices to identify a *sparse projection* of the DNA microarray features, representing the underlying response variable well. Only a small subset of genes will have non-zero weights in the extracted projection vector, and are thus identified as the relevant genes for the given response variable. Unlike the BAHSIC with nonlinear kernels, as well as most other multivariate methods in the literature, the proposed algorithm is computationally very fast and scalable, as it is not required to store the whole microarray dataset in memory. The algorithm can, therefore, easily be applied to real-world large datasets, which is an important attribute in big data analytics. This capability exist mainly because, unlike the BAHSIC, the proposed method is not reliant on the backward elimination.

The main attributes of the proposed feature selection algorithm are as follows:

1) The proposed method is a multivariate feature selection algorithm that takes into account the correlation among the genes and selects the subset of the most relevant ones by maximizing the dependency between the data and response variables, using the Hilbert-Schmidt independence criterion.
2) Unlike the BAHSIC, the method is not reliant on the backward elimination, which makes it very fast.
3) The computation of the method is based on inner products on matrices of sizes n (number of data samples) or c (number of classes), which is limited in gene expression data. Therefore, the proposed method is very fast.
4) In the proposed feature selection algorithm, one row of the data matrix has to be examined at a time, and therefore, the approach is scalable to large datasets. The whole dataset does not have to be stored in memory when the algorithm runs, an attractive attribute in big data analytics.
5) By transposing the input matrix, the proposed algorithm can comfortably be applied to the application of column subset selection (CSS) [18], [19] from big data with millions of data samples (more explanation is provided in Section 2.6).
6) In the case of categorical response variables, by introducing a kernel for the labels that combines the information in the response variable and the data, we alleviated the requirement for model selection on the kernels.
7) Although the proposed method is multivariate, and therefore, the correlation among the features is taken into consideration in the process of finding the most informative features, the speed and scalability of the proposed method is close to univariate feature selection methods, without compromising accuracy.
8) The method requires only a kernel of labels, and hence, can be applied to datasets with any type of response variables: biclass, multiclass, and continuous variables.

An extensive set of experiments on synthetic and real-world gene expression data with categorical (both biclass and multiclass) and continuous response variables using the proposed method demonstrates that the approach achieves the same level of accuracy and stability as multivariate features selection methods, while its speed and scalability are close to univariate methods.

The organization of the rest of the paper is as follows: we review the mathematical background and the formulation for the proposed approach in Section 2. The experimental setup and results on synthetic and real-world data are presented in Sections 3, followed by our conclusion in Section 4.

2 METHODS

2.1 Problem Statement

Suppose $\mathbf{X} \in \mathbb{R}^{m \times n}$ represents genomic microarray data with m genes and n samples and $\mathbf{y} \in \mathbb{R}^{n \times 1}$ is a discrete or continuous response variable. For example, $\mathbf{y}$ could either represent labels of patients with high-risk cancer (discrete) or their survival time (continuous).[1] The goal is to

1. Here we assume the response variable is univariate. However, the results can be directly used for the case of multivariate response variables. In such a case, the observations of the response variable is captured in matrix $\mathbf{Y}$ rather than by the vector $\mathbf{y}$.

select a small feature set that contains as much predictive information about the response $\mathbf{y}$ as possible. In other words, we aim to identify a subset of features, so that $\mathbf{y}$ depends mainly on this subset and not on the other features.

Suppose there are n samples and n response values $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ denotes the ith column of matrix $\mathbf{X}$, and y_i is the ith entry of $\mathbf{y}$. We are looking for a projection $\mathbf{s} = \mathbf{u}^\top \mathbf{X}$ such that $\mathbf{y}$ depends mainly on $\mathbf{s}$. Here, $\mathbf{u}^\top \mathbf{X}$ is a linear combination of all the features, where elements of $\mathbf{u}$ determine the importance, or the weight, of each feature. If $\mathbf{u}$ is a *sparse* vector, then the weight of some features is zero and the subset of features with nonzero weights is the subset with the desired maximum predictive information.

Therefore, the formulation of the proposed feature selection algorithm involves two main steps: finding a projection $\mathbf{s} = \mathbf{u}^\top \mathbf{X}$ whose dependency is maximized with the response variable $\mathbf{y}$, and ensuring that the projected space is sparse, such that only salient features in that space have nonzero representation. Here, the former step is achieved by using the Hilbert-Schmidt independence criterion, and the latter by using a modified Jordan algorithm for a singular value decomposition (SVD) of signals. To provide a formal formulation of our approach, we first provide an introduction to the HSIC and Jordan algorithm, followed by the details of the proposed method.

2.2 Background of the Hilbert-Schmidt Independence Criterion

The Hilbert-Schmidt norm of the cross-covariance operator has been proposed as an independence criterion in reproducing kernel Hilbert spaces (RKHS) [14], [15]. This measure is referred to as the Hilbert-Schmidt independence criterion, which has been used in various applications, including independent component analysis [20], sorting/matching [21], supervised dictionary learning [22], and multiview learning [23].

The HSIC uses the fact that two random variables $\mathbf{x}$ and $\mathbf{y}$ are independent if and only if any bounded continuous function of the two random variables is uncorrelated. Consider two multivariate random variables $\mathbf{x}$ and $\mathbf{y}$ with joint probability distribution $p_{\mathbf{xy}}$. Let $\mathcal{X}$ and $\mathcal{Y}$ be the support (the set of possible values) of the random variables $\mathbf{x}$ and $\mathbf{y}$, respectively. Let $\mathcal{F}$ be a separable RKHS of real-valued functions from $\mathcal{X}$ to $\mathbb{R}$ with universal[2] kernel $k(\cdot,\cdot)$. Similarly, define $\mathcal{G}$ to be a separable RKHS of real-valued functions from $\mathcal{Y}$ to $\mathbb{R}$ with universal kernel $b(\cdot,\cdot)$. We are interested in the cross-covariance between elements of $\mathcal{F}$ and $\mathcal{G}$

$$\operatorname{cov}(f(\mathbf{x}), g(\mathbf{y})) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[f(\mathbf{x})g(\mathbf{y})] - \mathbb{E}_{\mathbf{x}}[f(\mathbf{x})]\mathbb{E}_{\mathbf{y}}[g(\mathbf{y})],$$

where $f \in \mathcal{F}$, $g \in \mathcal{G}$, and $\mathbb{E}$ is the expectation function.

There exists a unique operator $C_{\mathbf{x},\mathbf{y}} : \mathcal{G} \to \mathcal{F}$ mapping elements of $\mathcal{G}$ to elements of $\mathcal{F}$ such that: $\langle f, C_{\mathbf{x},\mathbf{y}}(g)\rangle_{\mathcal{F}} = \operatorname{cov}(f, g)$ for all $f \in \mathcal{F}$ and $g \in \mathcal{G}$ [14]. This operator is called the cross-covariance operator.

2. By the Moore-Aronszajn theorem in RKHS [24], there is a unique correspondence between any kernel and the RKHS it reproduces. A kernel k is called *universal* if the corresponding RKHS $\mathcal{F}$, includes all continuous functions on the domain $\mathcal{X}$.

The measure of dependence between two random variables can be defined as the squared Hilbert-Schmidt norm of the cross-covariance operator

$$\operatorname{HSIC}(p_{\mathbf{xy}}, \mathcal{F}, \mathcal{G}) := \|C_{\mathbf{xy}}\|^2_{\mathrm{HS}}.$$

Note that if $\|C_{\mathbf{xy}}\|^2_{\mathrm{HS}}$ is zero, then the value of $\langle f, C_{\mathbf{x},\mathbf{y}}(g)\rangle$, i.e., $\operatorname{cov}(f, g)$, will always be zero for any $f \in \mathcal{F}$ and $g \in \mathcal{G}$. Thus, the random variables $\mathbf{x}$ and $\mathbf{y}$ are independent, since kernels k and b are assumed to be universal and the corresponding RKHSs $\mathcal{F}$ and $\mathcal{G}$ include all bounded continuous functions on $\mathcal{X}$ and $\mathcal{Y}$, respectively.

2.2.1 *Empirical HSIC*

For computational purposes, the HSIC has to be expressed in terms of kernel functions. This expression is achieved using the following identity [14]

$$\begin{aligned}\operatorname{HSIC}(p_{\mathbf{xy}}, \mathcal{F}, \mathcal{G}) = {} & \mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}[k(\mathbf{x},\mathbf{x}')b(\mathbf{y},\mathbf{y}')] \\ & + \mathbb{E}_{\mathbf{x},\mathbf{x}'}[k(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y},\mathbf{y}'}[b(\mathbf{y},\mathbf{y}')] \\ & - 2\,\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\mathbb{E}_{\mathbf{x}'}[k(\mathbf{x},\mathbf{x}')]\mathbb{E}_{\mathbf{y}'}[b(\mathbf{y},\mathbf{y}')]\right],\end{aligned}$$

where $\mathbb{E}_{\mathbf{x},\mathbf{x}',\mathbf{y},\mathbf{y}'}$ is expectation over $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x}',\mathbf{y}')$ with $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x}',\mathbf{y}')$ being random variables taken independently from $p_{\mathbf{xy}}$. Now let $\mathcal{Z} := \{(\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_n,\mathbf{y}_n)\} \subseteq \mathcal{X}\times\mathcal{Y}$ be a collection of n independent observations drawn from $p_{\mathbf{xy}}$. An estimator of HSIC is then given by [14]

$$\operatorname{HSIC}(\mathcal{Z}, \mathcal{F}, \mathcal{G}) := (n-1)^{-2}\operatorname{tr}(\mathbf{KHBH}), \quad (1)$$

where tr is the trace operator, $\mathbf{H}, \mathbf{K}, \mathbf{B} \in \mathbb{R}^{n\times n}$, $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{B}_{ij} = b(\mathbf{y}_i, \mathbf{y}_j)$, $\mathbf{H} = \mathbf{I}_{n\times n} - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$ ($\mathbf{H}$ is a centering matrix of dimension $n \times n$), k and b are positive semidefinite kernel functions, $\mathbf{I}_{n\times n}$ is the identity matrix of size $n \times n$, and $\mathbf{1}_n$ is a vector of n ones. Thus, to maximize the dependence between the two random variables $\mathbf{x}$ and $\mathbf{y}$, the value of the empirical estimate, i.e., $\operatorname{tr}(\mathbf{KHBH})$ has to be maximized.

2.3 Jordan Algorithm for SVD

The Jordan algorithm for the singular value decomposition has been classically used for computing singular values/vectors of matrices [25]. The sparsity part of the proposed feature selection algorithm is motivated by the Jordan algorithm. Therefore, we present this algorithm in this section, and use it later to show how it can be modified to accomplish a sparse SVD. Algorithm 1 describes how the Jordan algorithm can be used to compute the rank d SVD of a matrix $\mathbf{A}$. At each iteration, the first singular value and singular vectors of $\mathbf{A}$ are computed, and then the value of $\mathbf{A}$ is updated by reducing the optimal rank one matrix obtained in this step. The iteration continues until all d eigenvectors of $\mathbf{A}$ are computed. In Algorithm 1, $\mathbf{U}_{:,i}$ and $\mathbf{V}_{:,i}$ denote the ith column of the outputs $\mathbf{U}$ and $\mathbf{V}$, respectively.

2.4 Feature Selection via HSIC and Sparse SVD

2.4.1 *Projection to the Space of Maximum Dependency with Response Variable*

In order to formulate the proposed feature selection algorithm, as stated in the problem statement (Section 2.1), the

Algorithm 1. Jordan Algorithm $[\mathbf{U}, \Sigma, \mathbf{V}] = \texttt{JordanAlgorithm}(\mathbf{A})$

```
Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$.
Outputs: $\mathbf{U} \in \mathbb{R}^{m\times d}$, $\mathbf{V} \in \mathbb{R}^{n\times d}$, $\Sigma \in \mathbb{R}^{d\times d}$.
1: for $i = 1, \dots, d$ do
2:     Select a random nonzero $\bar{\mathbf{u}} \in \mathbb{R}^m$.
3:     $\sigma = \|\bar{\mathbf{u}}\|$.
4:     $\mathbf{u} = \bar{\mathbf{u}}/\sigma$.
5:     repeat
6:         $\bar{\mathbf{v}} = \mathbf{A}^\top \mathbf{u}$.
7:         $\mathbf{v} = \bar{\mathbf{v}}/\|\bar{\mathbf{v}}\|$.
8:         $\bar{\mathbf{u}} = \mathbf{A}\mathbf{v}$.
9:         $\sigma = \|\bar{\mathbf{u}}\|$.
10:        $\mathbf{u} = \bar{\mathbf{u}}/\sigma$.
11:    until change in $\mathbf{u}$ and $\mathbf{v}$ is smaller than $\epsilon$.
12:    $\mathbf{A} = \mathbf{A} - \mathbf{u}\sigma\mathbf{v}^\top$.
13:    $\mathbf{U}_{:,i} = \mathbf{u}$.
14:    $\mathbf{V}_{:,i} = \mathbf{v}$.
15:    $\Sigma_{i,i} = \sigma$.
16: end for
```

first step is to maximize the dependency between the projection $\mathbf{s} = \mathbf{u}^\top\mathbf{X}$ and the response variable $\mathbf{y}$. To measure the dependency between the two, we use the empirical estimation of HSIC given in (1). If $[\mathbf{x}_1, \dots, \mathbf{x}_n]$ are projected to $\mathbf{s} = [\mathbf{u}^\top\mathbf{x}_1, \dots, \mathbf{u}^\top\mathbf{x}_n] = \mathbf{u}^\top\mathbf{X}$, a linear kernel on subspace $\mathbf{s}$ can be computed as $\mathbf{K} = \mathbf{X}^\top\mathbf{u}\mathbf{u}^\top\mathbf{X}$. Considering $\mathbf{B}$ as a kernel of $\mathbf{y}$, the dependency between the projection $\mathbf{s}$ and the response variable $\mathbf{y}$ can be measured using the empirical HSIC as

$$\begin{aligned}\mathrm{tr}(\mathbf{HKHB}) &= \mathrm{tr}(\mathbf{HX}^\top\mathbf{uu}^\top\mathbf{XHB})\\ &= \mathrm{tr}(\mathbf{u}^\top\mathbf{XHBHX}^\top\mathbf{u})\\ &= \mathbf{u}^\top\mathbf{XHBHX}^\top\mathbf{u}.\end{aligned} \tag{2}$$

We can make the objective function (2) arbitrarily large by increasing the magnitude of $\mathbf{u}$. To ensure the problem is well-posed, we choose $\mathbf{u}$ to maximize (2) while constraining $\mathbf{u}$ to a unit length. To accomplish the feature selection task, $\mathbf{u}$ is required to be sparse. Based on these two constraints, the optimization problem becomes

$$\begin{aligned}\max_{\mathbf{u}} \quad & \mathbf{u}^\top\mathbf{XHBHX}^\top\mathbf{u}\\ \text{s.t.} \quad & \mathbf{u}^\top\mathbf{u} = 1 \text{ and } \mathbf{u} \text{ is sparse.}\end{aligned} \tag{3}$$

If we relax the sparsity constraint, this problem can be solved in closed-form. If the symmetric and real matrix $\mathbf{Q} = \mathbf{XHBHX}^\top$ has eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ and eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$, then the maximum value of the cost function is λ_n and the optimal solution is $\mathbf{u} = \mathbf{v}_n$ [26].

Note that both $\mathbf{B}$ and $\mathbf{Q}$ are positive semidefinite matrices, and thus we can define

$$\begin{aligned}\mathbf{B} &= \Delta^\top\Delta\\ \mathbf{Q} &= \mathbf{AA}^\top\\ \mathbf{A} &= \mathbf{XH}\Delta^\top.\end{aligned}$$

Clearly the solution for $\mathbf{u}$ can be expressed as the first singular vector of $\mathbf{A}$, because the singular vectors of $\mathbf{A}$ are the eigenvectors of $\mathbf{AA}^\top = \mathbf{XHBHX}^\top$. Therefore, $\mathbf{u}$ can be obtained by finding the optimal rank one approximation to $\mathbf{A}$ in the Frobenius norm. As explained in next lines, the same result is true even when $\mathbf{u}$ is constrained to be sparse. That is, the optimal value of $\mathbf{u}$ in (3) can be found by solving the following optimization problem:

$$\begin{aligned}\min_{\sigma,\mathbf{u},\mathbf{v}} \quad & \|\mathbf{A} - \sigma\mathbf{uv}^\top\|_{\mathrm{F}}^2,\\ \text{s.t.} \quad & \mathbf{u}^\top\mathbf{u} = 1,\ \mathbf{v}^\top\mathbf{v} = 1, \text{ and } \mathbf{u} \text{ is sparse,}\end{aligned} \tag{4}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix, σ is an arbitrary nonnegative real number, and $\mathbf{v}$ is an arbitrary orthogonal vector in $\mathbb{R}^n$. Note that there is no sparsity constraint on $\mathbf{v}$, and that the above result is valid no matter how the sparsity constraint for $\mathbf{u}$ is defined. To explain this, we note that, given the orthogonal vector $\mathbf{u}$, the optimal value of $\sigma\mathbf{v}$ is $\mathbf{A}^\top\mathbf{u}$, which is the solution to a simple least square problem. Substituting this value into the objective function in (4), and simplifying the result leads to the same optimization problem as the one presented in (3).

The advantage of working with $\mathbf{A}$ rather than $\mathbf{Q}$ is that $\mathbf{A}$ is much smaller in size than $\mathbf{Q}$. This advantage is because the number of samples in DNA microarray experiments is usually much less than the number of features, and thus $\mathbf{A}$ has significantly fewer columns than $\mathbf{Q}$.

2.4.2 Inducing Sparsity into the Projected Space

The second step for the proposed feature selection algorithm is to induce sparsity into the projected space such that only the most salient features (with nonzero coefficients), which are the most representative ones, are selected. A number of techniques have been proposed in the literature for obtaining sparse rank one approximations for a given matrix. One common approach has been to use a least squares error function with an ℓ_1 penalty term [27], [28]. Another approach aimed to reformulate the problem as a semidefinite program [29]. A simple approach is to set a threshold and keep only those elements in the first singular vector that exceed the threshold. This approach is commonly employed in text mining and has been used extensively in methods such as latent semantic indexing (LSI) [30]. However, as detailed in the following example, we believe that this method will not produce an appropriate solution in the case that we have described.

Consider the following matrix $\mathbf{A}$, which is the sum of a completely separable matrix $\mathbf{F}$ and a noise matrix $\mathbf{E}$

$$\begin{aligned}\mathbf{A} &= \mathbf{F} + \mathbf{E}\\ &= \begin{pmatrix} 1.01 & 1.01 & 0 & 0\\ 1.01 & 1.01 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & 1\end{pmatrix}\\ &\quad + \begin{pmatrix} -0.02 & -0.02 & 0.02 & 0.02\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{pmatrix}.\end{aligned}$$

In this example, clearly, there are two separate sets of columns in $\mathbf{A}$, as shown by the two diagonal blocks. A practical algorithm should be able to identify the two blocks and produce an answer close to $[1 \quad 1 \quad 0 \quad 0]^\top$. The dominant right singular vector of $\mathbf{A}$ is almost proportional to $[1 \quad 1 \quad 1 \quad 1]^\top$, which is different than expected. An explanation for this behavior is that the matrix $\mathbf{F}$ has two nearly equal singular values, and so its singular vectors are highly sensitive to small perturbations (such as matrix $\mathbf{E}$). This pitfall can be avoided by computing a sparse singular vector that is then used to factorize a submatrix of the original $\mathbf{A}$, instead of the whole matrix. We will use this idea based on the Jordan algorithm, as stated in Section 2.3, to propose a method for a sparse decomposition of matrix $\mathbf{A}$.

2.4.3 Sparse SVD and the Proposed Feature Selection

Motivated by the Jordan algorithm, we present here the sparse SVD for inducing sparsity into the projected space with the maximum dependency with the response variable. We return to the sparse SVD problem presented in (4). Let M be the set of indices of nonzero elements in $\mathbf{u}$ and let $\mathbf{A}_{M,:}$ be a submatrix of $\mathbf{A}$, consisting of all the rows indexed in the set M and in all the columns of $\mathbf{A}$. Similarly, $\mathbf{A}_{\bar{M},:}$ denotes the submatrix of $\mathbf{A}$ consisting of all the rows which do *not* exist in the set M and in all the columns of $\mathbf{A}$. Finally, $\mathbf{u}_M$ denotes a subvector of $\mathbf{u}$ consisting of the elements indexed in M. Using the above notation, the objective function presented in (4) can be expressed as

$$\|\mathbf{A} - \sigma\mathbf{u}\mathbf{v}^\top\|_F^2 = \|\mathbf{A}_{\bar{M},:}\|_F^2 + \|\mathbf{A}_{M,:} - \sigma\mathbf{u}_M\mathbf{v}^\top\|_F^2.$$

Using the equality $\|\mathbf{A}_{\bar{M},:}\|_F^2 = \|\mathbf{A}\|_F^2 - \|\mathbf{A}_{M,:}\|_F^2$, the above expression can be written as

$$\|\mathbf{A}\|_F^2 - \|\mathbf{A}_{M,:}\|_F^2 + \|\mathbf{A}_{M,:} - \sigma\mathbf{u}_M\mathbf{v}^\top\|_F^2.$$

Because here $\|\mathbf{A}\|_F^2$ is a constant term, the optimization problem in (4) can be represented as a maximization problem with the following objective function:

$$\|\mathbf{A}_{M,:}\|_F^2 - \|\mathbf{A}_{M,:} - \sigma\mathbf{u}_M\mathbf{v}^\top\|_F^2. \tag{5}$$

In this formulation, given the submatrix $\mathbf{A}_{M,:}$, the optimal values of σ, $\mathbf{u}_M$, and $\mathbf{v}$ are simply the singular values and singular vectors of the submatrix. Therefore, solving the above optimization problem with a sparsity constraint on $\mathbf{u}$ is equivalent to finding a submatrix of $\mathbf{A}$ for which the above objective function is maximized.

Note that the objective function in (5) consists of two opposing terms: the first expresses the objective function that the extracted submatrix should be large and the second penalizes deviations of the extracted submatrix from being rank one. This may occur at the expense of reducing the size of the extracted submatrix. To control the sparsity of the solution for the above optimization problem, different weights can be assigned to the two opposing terms. In particular, if we down-weight the first term (which favours large matrices) or up-weight the second term, the solutions should become sparse. This leads to the following modified objective function

$$f(M, \sigma, \mathbf{u}, \mathbf{v}) = \|\mathbf{A}_{M,:}\|_F^2 - \gamma\|\mathbf{A}_{M,:} - \sigma\mathbf{u}_M\mathbf{v}^\top\|_F^2, \tag{6}$$

where $\gamma > 1$ is a penalty parameter controlling sparsity of the solution. Our aim was to find an optimal submatrix $\mathbf{A}_{M,:}$ (or equivalently, an optimal set M) that maximizes the objective function in (6). This is an NP-hard optimization problem. Instead, we have developed a heuristic algorithm to find the optimal solution.

The proposed algorithm can find the optimal submatrix $\mathbf{A}_{M,:}$ and the corresponding parameters σ, $\mathbf{u}$, and $\mathbf{v}$ simultaneously. To accomplish this, the values of the above parameters are updated in a periodic manner. The vector $\mathbf{v}$ is first assumed to be given. The objective function in (6) is separable in terms of the rows of the matrix $\mathbf{A}$. In particular, the contribution of the ith row is given by

$$r_i = \|\mathbf{A}_{i,:}\|^2 - \gamma\|\mathbf{A}_{i,:} - \beta_i\mathbf{v}^\top\|^2, \tag{7}$$

where $\beta_i = \sigma u_i$ and u_i is the ith element of $\mathbf{u}$. Because of this separability property, one can consider the contribution of the rows separately, and choose only those rows that can make a positive contribution to the objective function. The value of the first term in (7) is fixed when a row is given. To obtain the optimal value of the second term, one can solve a simple least squares problem to show that this term is maximized (or equivalently, $\|\mathbf{A}_{i,:} - \beta_i\mathbf{v}^\top\|^2$ is minimized) when β_i is set to $\mathbf{A}_{i,:}\mathbf{v}$. Thus, the maximal contribution of the ith row would be:

$$\begin{aligned} r_i &= \|\mathbf{A}_{i,:}\|^2 - \gamma\|\mathbf{A}_{i,:} - \mathbf{A}_{i,:}\mathbf{v}\mathbf{v}^\top\|^2 \\ &= \mathbf{A}_{i,:}\mathbf{A}_{i,:}^\top - \gamma\left(\mathbf{A}_{i,:} - \mathbf{A}_{i,:}\mathbf{v}\mathbf{v}^\top\right)\left(\mathbf{A}_{i,:} - \mathbf{A}_{i,:}\mathbf{v}\mathbf{v}^\top\right)^\top \\ &= -(\gamma - 1)\mathbf{A}_{i,:}\mathbf{A}_{i,:}^\top + \gamma(\mathbf{A}_{i,:}\mathbf{v})^2. \end{aligned} \tag{8}$$

The above expression should be evaluated for all rows, and only those rows which make a positive contribution should be selected. By dividing both sides of (8) by $\gamma - 1$, and by defining $\bar{\gamma} = \frac{\gamma}{\gamma - 1}$, the following criterion for selecting the ith row is obtained

$$-\mathbf{A}_{i,:}\mathbf{A}_{i,:}^\top + \bar{\gamma}(\mathbf{A}_{i,:}\mathbf{v})^2 > 0. \tag{9}$$

Thus, the above procedure provides the set M and the corresponding coefficients β_i, $i \in M$. Subsequently, σ and $\mathbf{u}$ can be obtained by normalizing the vector consisting of the β_i's, $i \in M$. In the next step, given vector $\mathbf{u}$, the optimal value of $\mathbf{v}$ is obtained by $\mathbf{A}_{M,:}^\top\mathbf{u}_M / \|\mathbf{A}_{M,:}^\top\mathbf{u}_M\|$, which is the solution to a least squares problem. The iteration is continued until the values of M, σ, $\mathbf{u}$, and $\mathbf{v}$ stagnate.

The initial values for the parameters $M, \sigma, \mathbf{u}$, and $\mathbf{v}$ should properly be set such that the initial value of the objective function (6) is positive otherwise, it may happen that no rows are selected in the subsequent steps. We selected the row with the greatest norm as the initial value of $\mathbf{v}$, and started iterations to find M. Since the initial value in (6) is deterministic, it always leads to the same set of selected features.

The objective function presented in (6) can be modified by adding a term of the form $-\rho|M|$. This additional term penalizes rows with very low norms, which might be produced because of noise, and hence, prevents those rows from being selected in the submatrix $\mathbf{A}_{M,:}$. The term can also be used to control the sparsity of the solution, and consequently, the number of features selected. Because this new term is also separable in terms of the rows of $\mathbf{A}$, it can be easily inserted into the criterion given in (9) for selecting a row, resulting in a modified criterion as follows:

$$-\mathbf{A}_{i,:}\mathbf{A}_{i,:}^{\top} + \bar{\gamma}(\mathbf{A}_{i,:}\mathbf{v})^2 - \bar{\rho} > 0, \tag{10}$$

where $\bar{\rho} = \rho/(\gamma - 1)$. In practice, we found it useful to control the sparsity of the solution using both values of $\bar{\gamma}$ and $\bar{\rho}$. In particular, it is useful to fix the value of $\bar{\gamma}$ and use $\bar{\rho}$ to change sparsity of the solution. The aforementioned steps have been summarized in Algorithm 2—called SparseSVD in this paper—for the sparse decomposition of matrices.

```
Algorithm 2. Sparse SVD $[M, \sigma, \mathbf{u}, \mathbf{v}] = \texttt{SparseSVD}(\mathbf{A}, \bar{\gamma})$
Inputs: $A \in \mathbb{R}^{m\times n}$, parameter $\bar{\gamma}$.
Outputs: $M \subset \{1, \ldots, m\}$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{v} \in \mathbb{R}^n$, $\sigma \in \mathbb{R}$.
 1: Select $i_0 \in \{1, \ldots, m\}$ to maximize $\|\mathbf{A}_{i_0,:}\|$.
 2: $M = \{i_0\}$.
 3: $\sigma = \|\mathbf{A}_{i_0,:}\|$.
 4: $\mathbf{v} = \mathbf{A}_{i_0,:}^{\top}/\sigma$.
 5: repeat
 6:     Let $\bar{\mathbf{u}} = \mathbf{A}\mathbf{v}$.
 7:     $M = \{i : -\mathbf{A}_{i,:}\mathbf{A}_{i,:}^{\top} + \bar{\gamma}(\mathbf{A}_{i,:}\mathbf{v})^2 - \bar{\rho} > 0\}$.
 8:     $\mathbf{u}_M = \bar{\mathbf{u}}_M/\|\bar{\mathbf{u}}_M\|$.
 9:     Let $\bar{\mathbf{v}} = \mathbf{A}_{M,:}^{\top}\mathbf{u}_M$.
10:     $\sigma = \|\bar{\mathbf{v}}\|$.
11:     $\mathbf{v} = \bar{\mathbf{v}}/\sigma$.
12: until change in $\mathbf{v}$ is smaller than $\epsilon$ and no change in $|M|$.
```

The proposed feature selection algorithm, which is named here as sparse Hilbert-Schmidt independence criterion (SHS) based feature selection, is presented in Algorithm 3.

```
Algorithm 3. Feature Selection $M = \texttt{SHS}(\mathbf{X}, \mathbf{y}, \bar{\gamma}, \bar{\rho})$
Inputs: $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\mathbf{y} \in \mathbb{R}^n$, parameters $\bar{\gamma}$, and $\bar{\rho}$.
Output: $M \subset \{1, 2, \ldots, m\}$.
1: Compute $\mathbf{B} = b\{y_i, y_j\} = \Delta^{\top}\Delta$.
2: Compute $\mathbf{A} = \mathbf{X}\mathbf{H}\Delta^{\top}$.
3: $[M, \sigma, \mathbf{u}, \mathbf{v}] = \texttt{SparseSVD}(\mathbf{A}, \bar{\gamma})$.
4: return $M$.
```

2.5 Choosing the Kernel of Labels for the Categorical Response Variables

To perform gene selection, the proposed algorithm requires a kernel of labels. For the categorical response variables, however, common kernels (such as RBF, polynomial, and sigmoid) cannot be used in a straightforward way. Here, we outline a systematic approach for obtaining the kernel of labels for datasets with categorical response variables. We followed the proposal of Blaschko and Gretton [31], [32] and combined the information in the response variable and data to obtain a kernel for labels. If c denotes the number of classes in the categorical response variable under study, and $\Pi_{n\times c}$ the corresponding partition matrix, then $\Pi_{ij} = 1$ if ith data point belongs to the jth class; otherwise, it is equal to zero. The kernel of labels was then assumed to take the form $\mathbf{B} = \Pi\mathbf{W}\Pi^{\top}$, where the initially unknown matrix $\mathbf{W}_{c\times c}$ captures the similarity between classes. To determine the value of $\mathbf{W}$, Blaschko and Gretton [31], [32] suggested that HSIC can be used to maximize the dependence between this kernel of labels and a kernel of data, as follows:

$$\begin{aligned} \max_{\mathbf{W}} \quad & \operatorname{tr}(\mathbf{K}\mathbf{H}_n\Pi\mathbf{W}\Pi^{\top}\mathbf{H}_n), \\ \text{s.t.} \quad & \operatorname{tr}(\Pi\mathbf{W}\Pi^{\top}\mathbf{H}_n\Pi\mathbf{W}\Pi^{\top}\mathbf{H}_n) = 1. \end{aligned} \tag{11}$$

In the above formulation, $\mathbf{K}_{n\times n}$ is a kernel of data, and $\mathbf{H}_n = \mathbf{I}_{n\times n} - n^{-1}\mathbf{1}_n\mathbf{1}_n^{\top}$ is the centering matrix of dimension $n \times n$. The authors in [32] showed that the solution to the above optimization problem takes the form $\mathbf{H}_c\mathbf{W}^*\mathbf{H}_c$ up to a constant factor, where $\mathbf{H}_c = \mathbf{I}_{c\times c} - c^{-1}\mathbf{1}_c\mathbf{1}_c^{\top}$ is a centering matrix of dimension $c \times c$, and $\mathbf{W}^*_{c\times c}$ is given by

$$\mathbf{W}^*_{ij} = \frac{1}{N_iN_j}\sum_{l\in C_i}\sum_{k\in C_j}\tilde{\mathbf{K}}_{lk}, \tag{12}$$

where N_i and N_j are the number of elements in clusters i and j, respectively, C_i and C_j are the sets of indices of samples in clusters i and j, and $\tilde{\mathbf{K}} = \mathbf{H}_n\mathbf{K}\mathbf{H}_n$. Matrix $\mathbf{W}^*$ in (12) is a standard kernel as previously defined [33].

Note that, if we decompose the positive semidefinite matrix $\mathbf{W}$ as $\mathbf{W} = \Psi^{\top}\Psi$, then matrix $\Delta_{c\times n}$ in the proposed feature selection algorithm can be directly obtained as $\Delta = \Psi\Pi^{\top}$. To obtain $\Psi_{c\times c}$, one can use eigenvalue decomposition of $\mathbf{W}$ to write $\mathbf{W} = \mathbf{P}\Lambda\mathbf{P}^{\top}$, and set Ψ to $\Lambda^{\frac{1}{2}}\mathbf{P}^{\top}$. This technique was adopted in this paper.

2.6 Computational Complexity and the Convergence of the Algorithm

The separability characteristics of the objective function (6) in terms of rows not only provide an easy way to implement the algorithm proposed above, but also makes the algorithm scalable to large datasets. The whole microarray dataset does not need to be stored in the memory when the algorithm is run because only one row of the data is required at a time. All the intermediate matrices to be computed, including $\Delta_{c\times n}$, $\mathbf{W}_{c\times c}$, $\Psi_{c\times c}$, $\Pi_{n\times c}$, and others are of sizes n (number of data samples) or c (number of classes), which are limited in gene expression datasets to typically less than 400 and 20, respectively. Therefore, the inner products have the complexity of at most $\mathcal{O}(n)$, which is comfortably manageable on any single machine. It is worthwhile to highlight here that for the computation of each row of matrix $\mathbf{A}$, only one row of matrix $\mathbf{X}$ is needed, and therefore different genes (features or rows) of matrix $\mathbf{X}$ can reside on different machines in a distributed system without any problem. This feature is particularly important when large real-world datasets are used, and is a very attractive feature for an algorithm designed for big data analysis, where the data might be large and massively distributed.

On the other hand, the proposed method can also be employed in the application of column subset selection [18], [19], where a few representative data samples should be selected from a big data matrix, provided that the dimensionality of the data is not very high. This application corresponds with finding representative columns of a big data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, where m the dimensionality of the data is much less than the number of data samples n ($m \ll n$). In this case, the proposed (SHS) algorithm can be employed with the transposed data matrix $\mathbf{X}^{\top}$ as input. In

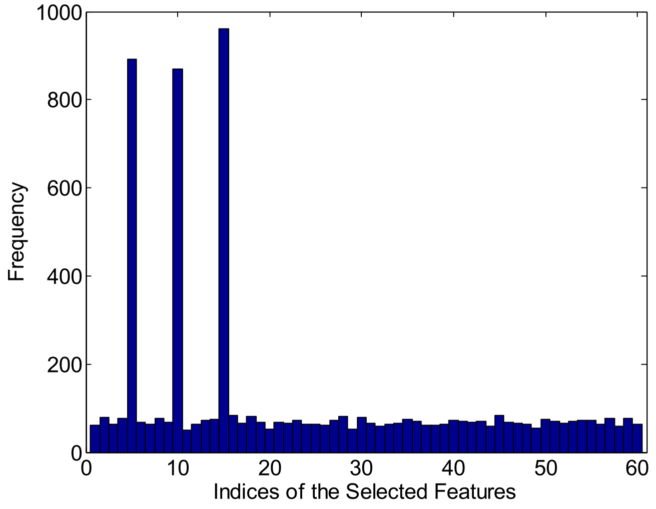


Fig. 1. Histogram of the selected features in the synthetic example.

this paper, however, the emphasis is given to the application of feature selection for gene expression data, rather than on column subset selection.

In Algorithm 2, the value of the objective function never decreases. This is because, given vector $\mathbf{v}$, all rows that can make a positive contribution to the objective function are selected. This implies that the values of M, σ, and $\mathbf{u}$ are globally optimized, given the value of $\mathbf{v}$. Also, given vector $\mathbf{u}$, the value of $\mathbf{v}$ is globally optimized by solving a least squares problem. The fact that the value of the objective function never decreases implies that the algorithm is guaranteed to converge. In practice, we found that the algorithm converged within a few iterations (typically in less than 5 iterations) which allowed the resulting feature selection method to be fast.

3 Results and Discussion

3.1 Synthetic Data

To evaluate the performance of the SHS algorithm, we first considered an experiment using synthetic data. We assumed a dataset of 50 data points $\{\mathbf{x}_i\}_{i=1}^{50}$ each with 60 features, stacked in a 60×50 matrix $\mathbf{X}$. A univariate response variable $\mathbf{y}$ which depends only on a specific subset of the features was constructed as follows:

$$\mathbf{y} = \operatorname{sgn}\left[\sin(\mathbf{X}_{5,:}) + \sin(\mathbf{X}_{10,:}) + \mathbf{X}_{15,:} \odot \mathbf{X}_{15,:} - 1.2 + \epsilon\right],$$

where $\epsilon \sim N(0, 0.01)$ is normally-distributed additive i.i.d. noise, $\mathbf{X}_{i,:}$ denotes the ith feature (the ith row of $\mathbf{X}$), $\operatorname{sgn}[\cdot]$ denotes the sign function, and $\odot$ stands for the elementwise product (also called Hadamard product) between two vectors. The relationship between the response variable $\mathbf{y}$ and the features of $\mathbf{X}$ is relatively complicated. Elements of the data points $\mathbf{X}$ were produced according to the unit uniform distribution. To use the SHS algorithm, we set the kernel of labels to the kernel introduced in Section 2.5. In the proposed method, we used $\bar{\rho} = 0.1$, as this value resulted in a reasonable number of selected features. The value of the parameter $\bar{\gamma}$ was set to a default value of 12 (corresponding to $\gamma \approx 1.1$), and was kept constant throughout. The process was repeated 1,000 times to explore different possibilities for the variables $\mathbf{X}$ and $\mathbf{y}$. We found that the size of the returned feature set varied between 1 to 16 over the 1,000 trials, with an average of 6.6 features per trial. A histogram of the indices of the selected features is presented in Fig. 1. As illustrated in Fig. 1, the true features, 5, 10, and 15 were selected in 89.1, 87.0, and 96.0 percent of the trials, respectively, while each of the other features occurred in, at most, 8.3 percent of the trials. These results demonstrated that the proposed method could successfully identify the most informative genes.

TABLE 1
Description of the DNA Microarray Datasets Used in the Study

Dataset	No. of Samples	No. of Genes	No. of Classes
Lymphoma	96	4,026	2
Leukemia	72	7,129	2
Brain Tumour	90	5,920	5
11-Tumours	174	12,533	11
SRBCT	83	2,308	4
Lung	203	12,600	5

3.2 Gene Expression Data-Categorical Response Variables

To demonstrate the performance of the proposed feature selection algorithm, we present the experimental results obtained using a number of real-world datasets. The datasets used in this study were divided into two groups depending on the type of their response variables. In this section, we have provided the results for the datasets with categorical response variables, where the response variables indicate different types of human cancers. In the next section, the results will be presented for a dataset with continuous response variables, where the response variables represent the survival times of patients.

3.2.1 Data

Six publicly available DNA microarray datasets with categorical response variables were used. The datasets included those with both biclass and multiclass response variables. The biclass datasets were lymphoma [34] and leukemia [4] and the multiclass datasets were brain tumour [35], lung [36], small round blue cell tumours (SRBCT) [37], and a dataset containing 11 different tumour types, which we refer to as the 11-tumours dataset [38].

Table 1 provides a summary of the datasets with categorical response variables used in this study, and the details of each dataset are provided in this paragraph. The lymphoma dataset [34] contains 96 samples, 62 neoplastic, and 34 normal samples. For each sample, the gene expression levels of 4,026 genes are given. The leukemia dataset [4] contains expression levels of 7,129 genes given for 72 samples, 47 *acute lymphoblastic leukemia* (ALL) and 25 *acute myeloid leukemia* (AML) samples. The brain tumour dataset [35] contains 90 samples in five classes: *medulloblastoma*, *malignant glioma*, *atypical teratoid/rhabdoid tumour* (AT/RT), *normal cerebellum*, and *primitive neuroectodermal tumours* (PNET), where there are 60, 10, 10, 4, and 6 samples in the classes, respectively, and each sample has 5,920 genes. The lung dataset [36] is based on 12,600 variables describing 203 samples. The samples are divided into five subclasses: *adenocarcinomas*, *squamous cell lung carcinomas*, *pulmonary carcinoids*, *small-cell lung carcinomas*, and normal lung, with 139, 21, 20, 6, and 17 samples in each subclass, respectively. The SRBCT dataset [37] contains a total of 83 samples in four subclasses: the *Ewing family of tumours* (EWS), *Burkitt lymphoma* (BL), *neuroblastoma* (NB), and *rhabdomyosarcoma* (RMS), which have 29, 11, 18, and 25 samples, respectively. Finally, the 11-tumours dataset [38] includes the gene expression levels of 12,533 genes for 174 samples. These samples belong to 11 different tumour types: ovary (27 samples), bladder/ureter (8 samples), breast (26 samples), colorectal (23 samples), gastroesophagus (12 samples), kidney (11 samples), liver (7 samples), prostate (26 samples), pancreas (6 samples), lung adeno (14 samples), and lung squamous (14 samples).

3.2.2 Implementation Details

To evaluate the quality of the selected subset of genes, we divided each dataset into training and test sets. We normalized the training data such that each row in the data (corresponding to a gene) has zero mean and unit variance. We then normalized the test data using the same shift and scaling parameters obtained from the training data. We trained a classifier based on the top ranked genes in the training set and reported the classification accuracy of this classifier on the test set. We used leave-one-out (LOO) to obtain training/test splits, and defined the classification accuracy as the percentage of the samples for which the label is correctly predicted. For the classification task, we used two classifiers: a linear support vector machine classifier and a k-nearest-neighbour (k-NN) in all the experiments. For the biclass problems, the built-in SVM classifier in Matlab with linear kernel was used. For the multiclass problems, we used the one-versus-one linear SVM classifier implemented in LIBSVM [39]. The trade-off parameter C of the SVM was set to 1 and the k value of the k-NN was set to 3. In the proposed SHS method, the kernel $\mathbf{B}$ was set to the kernel introduced in Section 2.5. Also, in the proposed method, the number of selected genes is not explicitly given as an input. Instead, the number of genes is controlled by the parameters $\bar{\gamma}$, and $\bar{\rho}$, which control the sparsity of the proposed algorithm as mentioned in Section 2.4.3. In this work, the parameter $\bar{\gamma}$ was kept constant at 12 in all the experiments. We manually adjusted the value of $\bar{\rho}$ so that the average number of selected genes over different partitions of training/test data was equal to the desired number of selected genes mentioned in the experiments.

3.2.3 Rival Methods

We compared the proposed SHS feature selection algorithm (provided in Algorithms 2 and 3) with six standard methods available in the literature. The six standard methods that we used were: mRMR [6]; SVM-RFE [10] and L0 [40]: two SVM-based embedding methods; BAHSIC [17]; FMS [7]; and BWSS (between group to within group sum of squares): a univariate filter that ranks genes based on the ratio of between-class to within-class sum of squares [41]. For the two biclass datasets, i.e., Lymphoma and Leukemia, we also compared our method with the Pearson's correlation (PC), and a filter that simply ranks

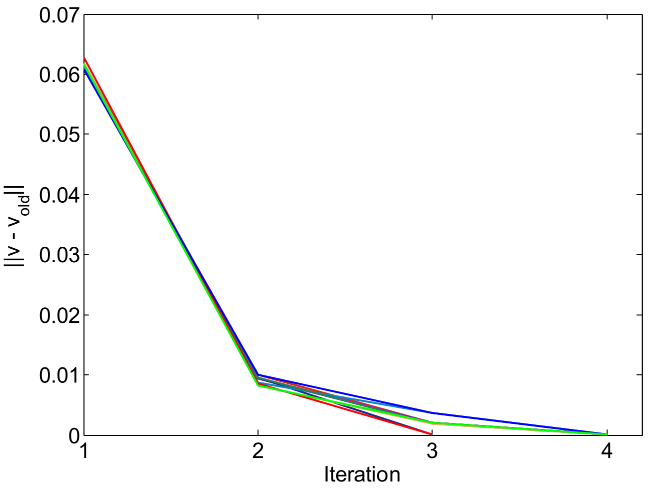


Fig. 2. The convergence of the proposed feature selection algorithm on the 11-tumours dataset. The convergence has been shown for 10 different leave-one-out subsamples of the dataset. A 100 genes were selected in the experiments.

genes based on their fold change [42], [43], denoted as FC in Table 2 (two upper sections). In this study, we used the codes for the SVM-RFE and L0 methods that have been implemented in the SPIDER toolbox [44]. However, the L0 code did not converge for the lung dataset. One-versus-one approach was used to handle multiclass cases in the SVM-RFE and L0 methods. For the mRMR method, the Matlab implementation available in [45] was employed. We used the mutual information difference (MID) variant of this method, and discretized the gene expression information into three levels as the preprocessing step [6]. For the FMS approach, the MATLAB code provided by the authors was used. BAHSIC was implemented using a linear kernel for the data that typically provides the best results for feature selection from microarray data [17]. With this kernel, algorithms in the BAHSIC family can be implemented as univariate filters [17].

3.2.4 Results

Below, the results are provided for the various performance evaluations of the proposed algorithm.

Convergence. Section 2.6 explained in theoretical terms that the proposed algorithm is convergent. In order to demonstrate this quality experimentally, the convergence of the SHS algorithm was shown on the 11-tumours dataset as an example. The results of convergence are shown in Fig. 2 on 10 arbitrary leave-one-out subsamples taken from the dataset when the requested number of features was 100. As can be seen from the figure, the convergence was very fast, occurring in only four iterations. Similar results were obtained on other subsamples and other datasets.

Classification Accuracies. In Table 2, the classification accuracies are presented for the number of selected genes equals to 50, 100, 200, 1,000 and the whole set of genes for the SVM and k-NN classifiers. The rightmost column in Table 2 lists the best classification accuracy obtained for each method and each classifier. The best classification accuracies in this column are highlighted for each classifier. In the cases where the best accuracy was achieved by several methods, the method that achieves the best accuracy with the least number of genes is highlighted. As shown in the two upper

TABLE 2
SVM and *k*-NN Classification Accuracies of the Different Methods for Real-World Datasets Using the Leave-One-Out Method

Dataset	Method	No. of Genes										Best Accuracy/No. of Genes	
		50		100		200		1,000		All			
		SVM	*k*-NN	SVM	*k*-NN	SVM	*k*-NN	SVM	*k*-NN	SVM	*k*-NN	SVM	*k*-NN
Lymphoma	PC[a]	92.71	87.50	94.79	93.75	94.79	94.79	95.83	90.63	95.83	87.50	95.83/1,000	94.79/200
	FC[a]	92.71	93.66	94.79	93.01	95.83	93.47	96.88	94.92	95.83	87.50	**96.88/1,000**	94.92/1,000
	BWSS	90.63	93.66	94.79	93.01	95.83	93.47	96.88	94.92	95.83	87.50	**96.88/1,000**	94.92/1,000
	BAHSIC	90.63	93.66	94.79	93.01	95.83	93.47	96.88	94.92	95.83	87.50	**96.88/1,000**	94.92/1,000
	SHS	90.63	97.84	95.83	97.63	95.83	97.39	96.88	96.87	95.83	87.50	**96.88/1,000**	**97.84/50**
	FMS	90.63	93.66	94.79	93.01	95.83	93.47	96.88	94.92	95.83	87.50	**96.88/1,000**	94.92/1,000
	L0	92.71	88.35	92.71	86.30	95.83	88.00	96.88	92.85	95.83	87.50	**96.88/1,000**	92.85/1,000
	SVM-RFE	84.38	88.04	87.50	92.43	88.54	90.95	95.83	95.15	95.83	87.50	95.83/1,000	95.15/1,000
	mRMR	94.79	87.06	94.79	89.06	95.83	88.01	93.75	91.76	95.83	87.50	95.83/200	91.76/1,000
Leukemia	PC[a]	95.83	93.06	98.61	93.06	97.22	93.06	95.83	93.06	95.83	87.50	**98.61/100**	93.06/50
	FC[a]	95.83	92.65	98.61	94.63	97.22	92.15	98.61	93.58	98.61	87.50	**98.61/100**	94.63/100
	BWSS	95.83	92.65	98.61	94.63	97.22	92.15	98.61	93.58	98.61	87.50	**98.61/100**	94.63/100
	BAHSIC	95.83	92.65	98.61	94.63	97.22	92.15	98.61	93.58	98.61	87.50	**98.61/100**	94.63/100
	SHS	97.22	96.98	97.22	97.68	97.22	96.29	98.61	96.24	98.61	87.50	98.61/1,000	**97.68/1,000**
	FMS	95.83	92.65	98.61	94.63	97.22	92.15	98.61	93.58	98.61	87.50	**98.61/100**	94.63/100
	L0	94.44	91.19	97.22	89.36	97.22	87.79	98.61	91.13	98.61	87.50	98.61/1,000	91.19/50
	SVM-RFE	90.28	86.39	90.28	88.17	90.28	88.16	97.22	90.41	98.61	87.50	98.61/All	90.41/1,000
	mRMR	95.83	86.41	97.22	86.71	97.22	86.09	98.61	88.15	98.61	87.50	98.61/1,000	88.15/1,000
Brain Tumour	BWSS	82.22	94.14	82.22	92.81	83.33	93.56	86.67	94.80	90.00	82.22	90.00/All	94.80/1,000
	BAHSIC	76.67	91.70	80.00	92.00	82.22	95.15	90.00	95.12	90.00	82.22	90.00/1,000	95.15/200
	SHS	80.00	94.22	86.67	94.81	88.89	96.18	91.11	96.19	90.00	82.22	**91.11/1,000**	**96.16/1,000**
	FMS	82.22	94.14	82.22	92.81	83.33	93.56	86.67	94.80	90.00	82.22	86.67/1,000	94.80/1,000
	L0	72.22	79.90	85.56	80.03	86.67	84.58	90.00	89.59	90.00	82.22	90.00/1,000	89.59/1,000
	SVM-RFE	78.89	23.95	78.89	26.96	82.22	31.79	88.89	68.39	90.00	82.22	90.00/All	82.22/All
	mRMR	85.56	85.78	86.67	86.69	88.89	87.78	90.00	90.15	90.00	82.22	90.00/1,000	90.15/1,000
11-Tumours	BWSS	77.59	97.80	85.06	97.35	86.78	97.55	93.10	97.72	90.23	75.86	93.10/1,000	97.80/50
	BAHSIC	63.22	98.43	72.99	97.01	85.63	96.25	91.95	97.11	90.23	75.86	91.95/1,000	98.43/50
	SHS	74.14	99.19	86.21	99.19	89.08	99.40	93.68	97.50	90.23	75.86	**93.68/1,000**	**99.40/200**
	FMS	77.59	97.80	85.06	97.35	86.78	97.55	93.10	97.72	90.23	75.86	93.10/1,000	97.80/50
	L0	74.14	77.55	84.48	75.02	86.21	73.61	91.95	71.34	90.23	75.86	91.95/1,000	77.55/50
	SVM-RFE	78.74	49.47	89.08	50.57	87.93	48.72	91.38	54.40	90.23	75.86	91.38/1,000	75.86/All
	mRMR	89.66	90.84	90.80	93.31	89.66	92.76	91.38	93.73	90.23	75.76	91.38/1,000	93.73/1,000
SRBCT	BWSS	100.00	97.21	100.00	97.26	98.80	96.98	100.00	95.16	100.00	80.72	**100.00/50**	97.26/100
	BAHSIC	100.00	94.01	100.00	94.85	100.00	94.58	100.00	93.87	100.00	80.72	**100.00/50**	94.85/100
	SHS	90.36	97.48	96.39	97.88	100.00	95.30	100.00	93.81	100.00	80.72	100.00/200	**97.88/100**
	FMS	100.00	97.21	100.00	97.26	100.00	96.98	100.00	95.16	100.00	80.72	**100.00/50**	97.26/100
	L0	90.36	81.47	95.18	83.64	100.00	83.66	100.00	83.06	100.00	80.72	100.00/200	83.66/200
	SVM-RFE	87.95	42.27	91.57	39.40	95.18	45.75	100.00	76.41	100.00	80.72	100.00/1,000	80.72/All
	mRMR	100.00	90.44	100.00	93.10	100.00	91.75	100.00	90.81	100.00	80.72	**100.00/50**	93.10/100
Lung	BWSS	90.64	98.86	91.13	98.19	95.07	98.53	95.07	98.60	94.58	91.13	95.07/200	**98.86/50**
	BAHSIC	90.15	96.56	90.15	97.61	93.10	97.48	95.57	97.58	94.58	91.13	95.57/1,000	97.61/100
	SHS	90.15	98.74	93.60	98.03	94.58	98.65	96.55	98.53	94.58	91.13	**96.55/1,000**	98.74/50
	FMS	90.64	98.86	91.13	98.19	95.07	98.53	95.07	98.60	94.58	91.13	95.07/200	**98.86/50**
	L0	NC	NC[b]	NC[b]	NC[b]	NC[b]	NC[b]	NC[b]	NC[b]	94.58	91.13	NC[b]	NC[b]
	SVM-RFE	93.60	63.06	94.09	74.39	92.61	81.44	95.07	93.94	94.58	91.13	95.07/1,000	93.94/1,000
	mRMR	92.61	91.63	95.07	93.68	94.09	94.36	95.07	96.22	94.58	91.13	95.07/100	96.22/1,000

The results are provided for the top 50, 100, 200, and 1,000 genes, as well as for the whole set of genes in each dataset.
[a] *The comparison could only be performed on biclass datasets.*
[b] *NC: Not Converged.*

sections of Table 2, mainly simple univariate filter methods (BAHSIC, BWSS, and FC) achieved the optimal classification accuracies in datasets with biclass response variables using the SVM classifier. However, when using the *k*-NN classifier, the proposed method outperformed the other feature selection methods in biclass response variable datasets. In the datasets with multiclass response variables, the proposed SHS method compared favourably with the other methods in terms of its classification accuracy for both SVM and *k*-NN classifiers.

Stability. In addition to the importance of accuracy in evaluating the merit of a feature selection algorithm, the set of features selected should not vary significantly when different subsamples of the same dataset are submitted to the algorithm. This is measured by an index called stability, and the study of its applicability on high dimensional data

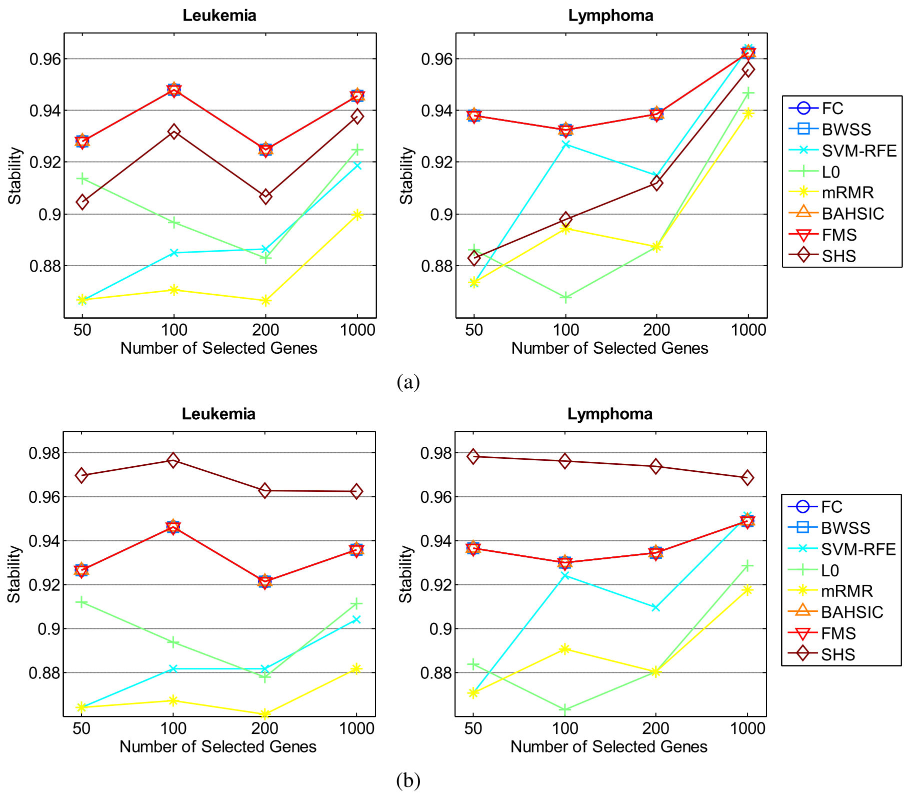


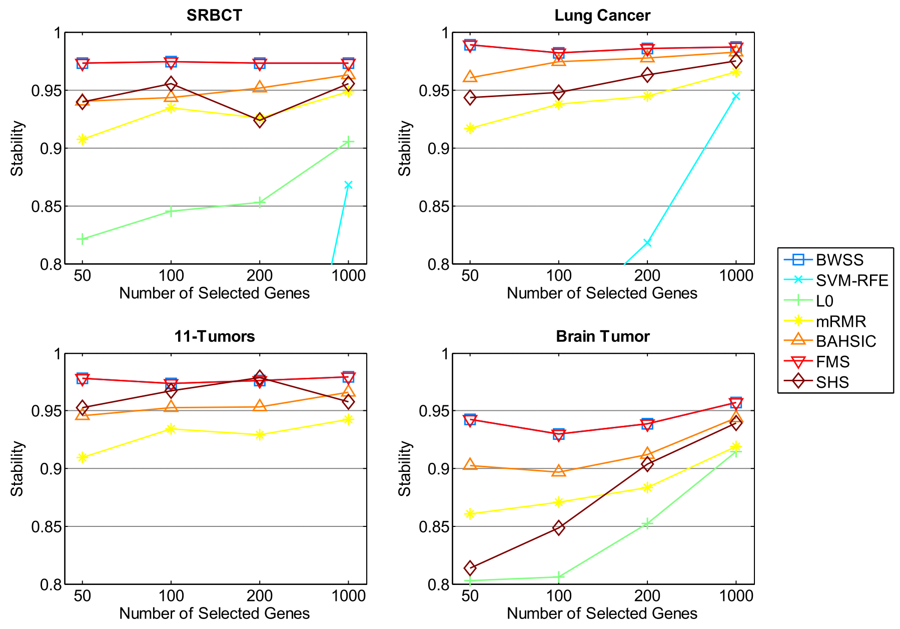
Fig. 3. The results of the stability tests using (a) Lausser et al. [46], and (b) Kuncheva [47] methods for biclass datasets. Leave one out was used for sampling the datasets.

has received a good amount of attention in gene selection literature [46], [47], [48], [49]. As part of our analysis, we studied how the SHS method compared to other feature selection algorithms in terms of stability of the selected features. We used two measures of stability proposed by Lausser et al. [46] and Kuncheva [47]. Lausser's method [46] for measuring the stability of a feature selection algorithm creates a histogram of selected features from different subsamples. A stability measure is then defined that favours sparse histograms over histograms, in which many features are touched. The stability measure proposed by Kuncheva [47] is based on the average distance between the set of features selected from different subsamples. The notion of set distance in Kuncheva's method is based on the difference between the actual overlap between two sets compared to the overlap that is expected due to a random chance. The results are presented in Figs. 3 and 4 for biclass and multiclass datasets, respectively. As demonstrated in these figures, SHS demonstrates a good level of stability based on the measures proposed by Lausser et al., and is in most cases among the top three methods. Also, based on the stability measure proposed by Kuncheva, SHS usually outperforms other methods.

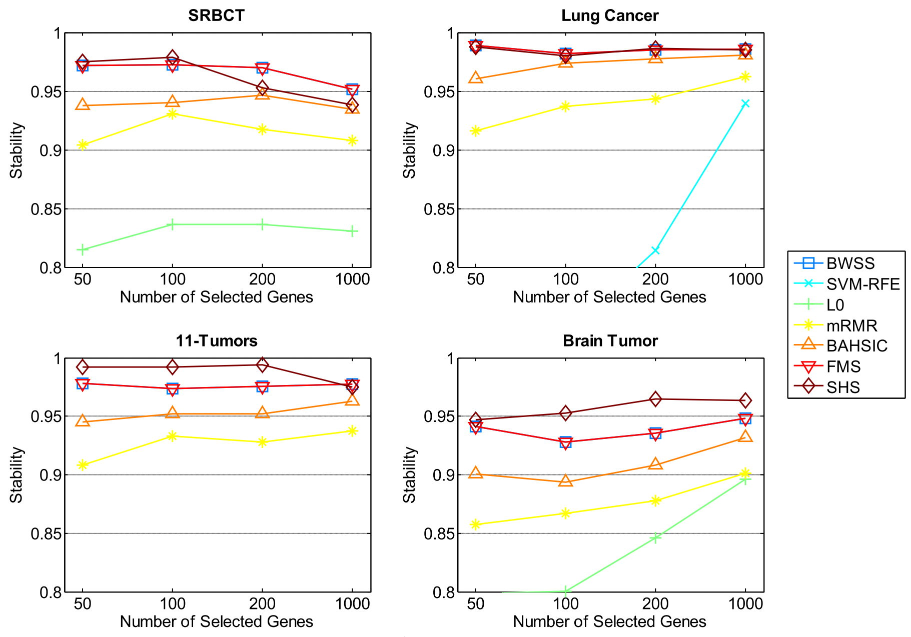
Furthermore, to investigate the effect of sampling method on the stability of selected features, we have also provided the results for the stability tests using the two aforementioned methods (Lausser's and Kuncheva's methods) for a different sampling method from LOO, i.e., using 10-fold cross validation. The results are provided in the supplementary Figs. S1 and S2, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2016.2631164. Similar trends of stability as those in the LOO sampling method can be observed from these figures for different feature selection approaches discussed in this paper.

Speed and Scalability. Two other desirable aspects of the proposed method are its speed and scalability. The run times of the studied algorithms in terms of the selected number of genes are presented in Table 3. We used the 11-tumours dataset to evaluate the run times, but also observed similar trends in other datasets. The run times in Table 3 represent the accumulative time required for each algorithm to perform the feature selection step in all folds of the LOO procedure described above. The run times do not include the time required for the subsequent classification tasks. As mentioned earlier, in the proposed method, the number of genes represents the average number of genes over different training/test partitions. As shown in Table 3, the speed of the proposed algorithm is more than other multi-gene algorithms, and in fact, is comparable to the univariate methods.

The scalability of the studied algorithms was compared on two datasets: the 11-tumours and a synthetic dataset with one million features. The results are shown in Figs. 5 and 6, respectively. As in the previous case, the 11-tumours dataset was used as a representative example among the gene expression datasets investigated in this study. Here, a comparison has been provided on the increase of the algorithms' run times with growing the number of genes in the input dataset. The size of the input dataset was varied by selecting only a subset of genes, specifically, the first 1,000, 2,000, 4,000, 8,000, and 10,000 genes, as well as including all (12,533) genes, and the effect of this variation was evaluated on the run times of the gene selection algorithms. In all cases, the number of selected genes was kept constant at



(a)



(b)

Fig. 4. The results of the stability test using (a) Lausser et al. [46], and (b) Kuncheva [47] methods for multiclass datasets. Leave one out was used for sampling the datasets.

TABLE 3
Comparison of the Speed of the Studied Feature Selection Algorithms

No. of Selected Genes	Univariate Methods		Multivariate Methods				
	BWSS	BAHSIC-Lin	SHS	FMS	L0	SVM-RFE	mRMR
50	33.7	3.8	35.2	179.4	416.9	890.4	737.7
100	24.3	3.5	38.8	177.5	415.8	820.7	1,365.5
200	33.4	3.5	38.1	176.6	416.8	756.8	2,579.4
1,000	33.2	3.7	44.4	168.7	424.1	662.7	7,523.6

Run times of the studied feature selection algorithms are presented in terms of the number of selected genes. The results are obtained using the 11-tumours dataset. Run times are given in seconds.

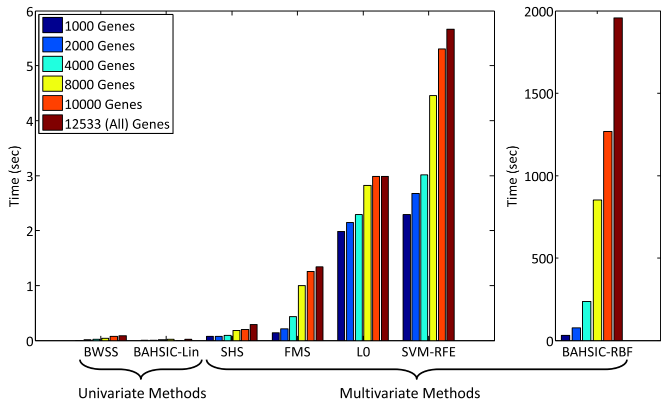


Fig. 5. Comparison of the scalability of the studied feature selection algorithms. Run times are presented in terms of the total number of input genes. The results are obtained using the 11-tumours dataset. In all experiments, the number of selected genes was kept constant at 100. The run time reported is the average time required for feature selection in each round of leave one out. The run time results are not shown for mRMR method, as they are significantly higher than other approaches (their inclusion would scale down other run times and make their comparison very difficult).

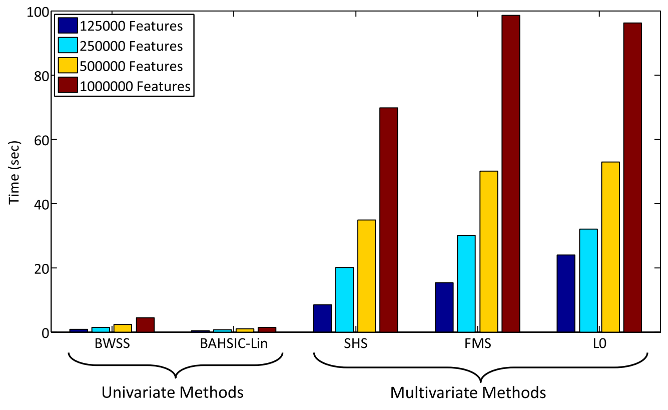


Fig. 6. Comparison of the scalability of the studied feature selection algorithms on a synthetic data with one million features and 200 data samples. Run times are presented in terms of the total number of input genes. In all experiments, the number of selected features was kept constant at 1,000. The run time reported is the average time required for feature selection in each round of leave one out. The run time results are not shown for SVM-RFE and mRMR methods, as they are significantly higher than other approaches (their inclusion would scale down other run times and make their comparison very difficult).

100. The run time reported (vertical axis) corresponds to the average time spent on feature selection in each round of leave one out. In Fig. 5, the results for the mRMR method was excluded because of its significantly higher run time compared to other methods. As illustrated in this figure, the run time of the proposed SHS method changed at a rate comparable to univariate filter methods. As mentioned previously, using the BAHSIC method with linear kernels (BAHSIC-Lin) reduces it to a univariate feature selection method. However, in order to account for the interaction and correlation among the genes in the BAHSIC, nonlinear kernels should be used. Using a nonlinear kernel, such as RBF kernel on the data in the BAHSIC, makes the process extremely slow due to its backward elimination nature, even if 10 percent of remaining features are removed in each elimination step as suggested in [17]. The rightmost panel in Fig. 5 depicts the scalability of the BAHSIC with an RBF kernel (BAHSIC-RBF) on the 11-tumours dataset. A different panel is used for BAHSIC-RBF in Fig. 5 to enable its comparison with other methods, as the vertical axis needs a different scaling in BAHSIC-RBF due to its significantly lower scalability.

In addition to the 11-tumours dataset, the scalability results are shown in Fig. 6 on a synthetic dataset of one million features and 200 data samples randomly generated with a uniform distribution. In all algorithms, the number of selected genes was kept unchanged at 1,000 and the number of input genes was changed across 125,000, 250,000, 500,000, and 1,000,000. Similar to the results on 11-tumours dataset, the run times reported (vertical axis) corresponded to the average time required for feature selection on each fold of the LOO approach. The results for SVM-RFE, mRMR and BAHSIC-RBF are not shown, as they took significantly more time compared to other approaches. As can be seen form this figure, the SHS outperforms other rival multivariate approaches.

The development took place on a 64-bit machine with the specifications: Intel (R) Xeon(R) CPU E5-2609 @ 2.40 GHz with 32 GB of memory.

3.3 Gene Expression Data-Continuous Response Variables

As mentioned earlier, the proposed method can be used with any type of response variable. Here, we demonstrate the performance of the SHS method on a dataset with continuous response variables. The dataset is from a study by Rosenwald et al. [50] and consists of 240 samples from patients with diffuse large B-cell lymphoma (DLBCL). For each sample, the gene expression levels of 7,399 genes are available. The response variable here is the survival time of patients, either observed or right censored. We used 160 samples in the training set and 80 samples in the test set. For the censored survival times, we used an estimate of the real survival time. Specifically, to estimate the true survival time for the survival time T censored at t_0, we evaluated $\mathbb{E}[T|T > t_0]$ based on the Kaplan-Meier [51] estimator.

In order to evaluate the quality of the selected genes, a linear regressor was fitted to the identified genes in the training set. Subsequently, the mean-square error (MSE) of the predicted survival times on the test set using the fitted model was used to quantify the quality of the selected subset of genes. For feature selection, the SHS was employed with two different types of kernels on the response variables: a linear kernel and an RBF kernel. We compared the SHS with two other feature selection methods that can be applied to continuous response variables, including the BAHSIC with an RBF kernel on the response variables and the Pearson correlation. Fig. 7 depicts the MSE of the predicted survival times on the test set when the number of selected genes was set from 1 to 10. As shown, the SHS with linear kernel outperformed the BAHSIC when the numbers of selected genes was between 7 and 10, and otherwise, performed equivalently. The Pearson correlation's performance fluctuated more over the range of selected genes, as its performance deteriorated significantly for the number of selected genes equal to 4, and from 8 to 10.

In order to investigate the effect of the kernel type on the SHS method, the performance of the SHS using linear and RBF kernels was compared on two synthetic data each with a

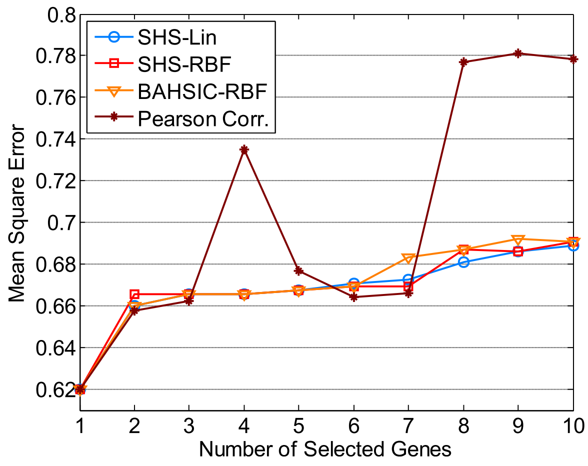


Fig. 7. Comparison of the feature selection algorithms on a dataset with continuous response variable. Feature selection algorithms were used to identify genes predictive of patients' survival times in the DLBCL dataset. A linear regressor was fitted to the data consisting of the test samples and genes identified in the training set. The mean-square error of the predicted survival times for the test set are presented. The number of selected genes were varied from 1 to 10.

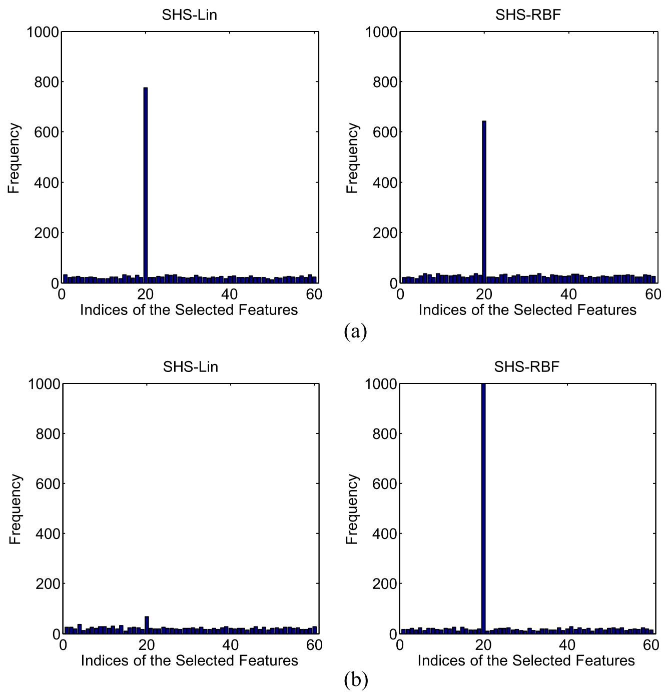


Fig. 8. The histograms of the selected features using the SHS with a linear or RBF kernel on two synthetic continuous response variable datasets: (a) response variable with additive noise given in (13) and (b) response variable with multiplicative noise given in (14). As can be observed, in case of (b), the SHS with the linear kernel has a significantly lower success rate (10.7 percent) to identify the correct feature compared with the SHS with the RBF kernel (100 percent).

univariate continuous response variable. Similar to the experiment on the synthetic data for the categorical response variables in Section 3.1, a dataset $\mathbf{X}$ of 50 data samples and 60 features was generated with a uniform distribution. Two univariate response variables $\mathbf{y}$ were generated, which were dependent on only one specific feature, as follows:

$$\mathbf{y} = \sin^2(\pi \mathbf{X}_{20,:}) + 0.5\epsilon, \tag{13}$$

and

$$\mathbf{y} = 0.5(\mathbf{X}_{20,:}) \odot \epsilon, \tag{14}$$

where $\epsilon \sim N(0,1)$ is a normally-distributed i.i.d. noise. Whereas both response variables are only dependent on feature 20, the noise in (13) is additive, while in (14), it is multiplicative. Just as in the experiment on the synthetic data for the categorical response variable, the parameter $\bar{\gamma}$ was kept constant at the value of 12, and $\bar{\rho}$ was set to select two features on average over 1,000 runs of the random generations of $\mathbf{X}$ and $\mathbf{y}$. The kernel width of the RBF kernel was set to the median of all the pairwise distances among the elements in $\mathbf{y}$. Fig. 8 displays the histograms of the indices of the selected features by the SHS with linear (left graphs) and RBF (right graphs) kernels. Whereas the SHS with the linear and RBF kernels could successfully identify the correct feature (feature 20) for the response variable given in (13), the success rate for identifying the correct feature by the SHS with the linear kernel is far below (only 10.7 percent of the trials) compared to the one with the RBF kernel (100 percent of the trials) for the response variable given in (14).

4 Conclusion

This paper has described a novel method for selecting a small subset of informative genes in a microarray experiment.[3] The method was based on the Hilbert-Schmidt independence criterion, a recently proposed kernel-based criterion for measuring the dependence between two random variables. The method requires only a kernel of labels and hence, it can be applied to datasets with any type of response variables, biclass, multiclass, and continuous variables. Moreover, since only one row of the data has to be examined at a time in the proposed feature selection algorithm (see Line 7 in Algorithm 2), the method can be applied to large and distributed datasets and is highly scalable, a very attractive attribute in large-scale data analytics over a distributed environment of commodity machines. Simulation results on real-world data have demonstrated that the method is effective and efficient in extracting stable genes with high predictive capability, in particular on datasets with multiclass response variables.

Although the proposed feature selection algorithm was described with the application to gene selection, it can be adapted in other applications where feature selection is needed. Moreover, the approach can easily be employed as a column subset selection algorithm in big data analytics—provided that the dimensionality of data features is not too high—by just transposing the input data before its submission to the proposed algorithm.

3. The code for the proposed SHS algorithm is now available and can be downloaded from: https://uwaterloo.ca/data-science/sites/ca.data-science/files/uploads/files/shs.zip

Acknowledgments

The first and second authors have equal contributions. This work was supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada under Postdoctoral Fellowship (PDF-454649-2014).

REFERENCES

[1] P. Maji, “f-Information measures for efficient selection of discriminative genes from microarray data,” *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1063–1069, Apr. 2009.

[2] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.

[3] A. J. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[4] T. R. Golub, et al., “Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring,” *Science*, vol. 286, no. 5439, pp. 531–537, Oct. 1999.

[5] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jun. 2003.

[6] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[7] Q. Cheng, H. Zhou, and J. Cheng, “The Fisher-Markov selector: Fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1217–1233, Jun. 2011.

[8] X. Zhou and K. Z. Mao, “LS bound based gene selection for DNA microarray data,” *Bioinf.*, vol. 21, no. 8, pp. 1559–1564, 2005.

[9] E. K. Tang, P. N. Suganthan, and X. Yao, “Gene selection algorithms for microarray data based on least squares support vector machine,” *BMC Bioinf.*, vol. 7, no. 1, 2006, Art. no. 95.

[10] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Mach. Learn.*, vol. 46, no. 1–3, pp. 389–422, 2002.

[11] R. Dìaz-Uriarte and S. Alvarez de Andrés, “Gene selection and classification of microarray data using random forest,” *BMC Bioinf.*, vol. 7, 2006, Art. no. 3.

[12] S. Ma and J. Huang, “Regularized ROC method for disease classification and biomarker selection with microarray data,” *Bioinf.*, vol. 21, no. 24, pp. 4356–4362, 2005.

[13] A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, and B. Schölkopf, “Kernel methods for measuring independence,” *J. Mach. Learn. Res.*, vol. 6, pp. 2075–2129, Dec. 2005.

[14] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, “Measuring statistical dependence with hilbert-schmidt norms,” in *Proc. 16th Int. Conf. Algorithmic Learn. Theory*, 2005, pp. 63–77.

[15] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola, “A kernel statistical test of independence,” in *Proc. Advances Neural Inf. Process. Syst.*, 2007, pp. 585–592.

[16] L. Song, A. Smola, A. Gretton, K. Borgwardt, and J. Bedo, “Supervised feature selection via dependence estimation,” in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 823–830.

[17] L. Song, J. Bedo, K. M. Borgwardt, A. Gretton, and A. J. Smola, “Gene selection via the BAHSIC family of algorithms,” *Bioinf.*, vol. 23, pp. i490–i498, Jul. 2007.

[18] E. Elhamifar, G. Sapiro, and R. Vidal, “See all by looking at a few: Sparse modeling for finding representative objects,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1600–1607.

[19] A. K. Farahat, A. Elgohary, A. Ghodsi, and M. S. Kamel, “Greedy column subset selection for large-scale data sets,” *Knowl. Inf. Syst.*, vol. 45, pp. 1–34, 2015.

[20] H. Shen, S. Jegelka, and A. Gretton, “Fast kernel-based independent component analysis,” *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3498–3511, Sep. 2009.

[21] N. Quadrianto, A. Smola, L. Song, and T. Tuytelaars, “Kernelized sorting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1809–1821, Oct. 2010.

[22] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, “Kernelized supervised dictionary learning,” *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4753–4767, Oct. 2013.

[23] M. J. Gangeh, P. Fewzee, A. Ghodsi, M. S. Kamel, and F. Karray, “Multiview supervised dictionary learning in speech emotion recognition,” *IEEE/ACM Trans. Audio Speech Language Process.*, vol. 22, no. 6, pp. 1056–1068, Jun. 2014.

[24] N. Aronszajn, “Theory of reproducing kernels,” *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[25] G. W. Stewart, “On the early history of the singular value decomposition,” *SIAM Rev.*, vol. 35, no. 4, pp. 551–566, 1993.

[26] H. Lütkepohl, *Handbook of Matrices*. New York, NY, USA: Wiley, 1997.

[27] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, “A modified principal component technique based on the LASSO,” *J. Comput. Graph. Statist.*, vol. 12, no. 3, pp. 531–547, Sep. 2003.

[28] D. M. Witten, R. Tibshirani, and T. Hastie, “A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis,” *Biostatistics*, vol. 10, no. 3, pp. 515–534, 2009.

[29] A. d’Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, “A direct formulation for sparse PCA using semidefinite programming,” *SIAM Rev.*, vol. 49, no. 3, pp. 434–448, Jul. 2007.

[30] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[31] M. B. Blaschko and A. Gretton, “Learning taxonomies by dependence maximization,” in *Proc. Advances Neural Inf. Process. Syst.*, 2009, pp. 153–160.

[32] M. B. Blaschko and A. Gretton, “Taxonomy inference using kernel dependence measures,” Max Planck Inst. Biol. Cybern., Tübingen, Germany, Tech. Rep. 181, Nov. 2008.

[33] D. Haussler, “Convolution kernels on discrete structures,” Univ. California Santa Cruz, Santa Cruz, CA, USA: Tech. Rep. UCSC-CRL-99-10, Jul. 1999.

[34] A. A. Alizadeh, et al., “Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling,” *Nature*, vol. 403, no. 6769, pp. 503–511, 2003.

[35] S. L. Pomeroy, et al., “Prediction of central nervous system embryonal tumour outcome based on gene expression,” *Nature*, vol. 415, no. 6870, pp. 436–442, Jan. 2002.

[36] A. Bhattacharjee, et al., “Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses,” *Proc. Nat. Academy Sci. United States America*, vol. 98, no. 24, pp. 13790–13795, Nov. 2001.

[37] J. Khan, et al., “Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks,” *Nature Med.*, vol. 7, no. 6, pp. 673–679, Jun. 2001.

[38] A. I. Su, et al., “Molecular classification of human carcinomas by use of gene expression signatures,” *Cancer Res.*, vol. 61, no. 20, pp. 7388–7393, Oct. 2001.

[39] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011. software available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[40] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, “Use of the zero norm with linear models and kernel methods,” *J. Mach. Learn. Res.*, vol. 3, pp. 1439–1461, 2003.

[41] S. Dudoit, J. Fridly, and T. P. Speed, “Comparison of discrimination methods for the classification of tumours using gene expression data,” *J. Amer. Statist. Assoc.*, vol. 97, no. 457, pp. 77–87, 2002.

[42] C. M. Yanofsky and D. R. Bickel, “Validation of differential gene expression algorithms: Application comparing fold-change estimation to hypothesis testing,” *BMC Bioinf.*, vol. 11, no. 1, 2010, Art. no. 63.

[43] D. Dembélé and P. Kastner, “Fold change rank ordering statistics: A new method for detecting differentially expressed genes,” *BMC Bioinf.*, vol. 15, no. 14, 2014, Art. no. 14.

[44] J. Weston, A. Elisseeff, G. BakIr, and F. Sinz, “The Spider machine learning toolbox,” 2005. [Online]. Available: http://www.kyb.tuebingen.mpg.de/bs/people/spider/

[45] (2005). [Online]. Available: http://penglab.janelia.org/proj/mRMR/

[46] L. Lausser, C. Müssel, M. Maucher, and H. A. Kestler, “Measuring and visualizing the stability of biomarker selection techniques,” *Comput. Statist.*, vol. 28, no. 1, pp. 51–65, 2013.

[47] L. Kuncheva, “A stability index for feature selection,” in *Proc. 25th Int. Multi-Conf. Artif. Intell. Appl.*, 2007, pp. 390–395.

[48] A. Kalousis, J. Prados, and M. Hilario, “Stability of feature selection algorithms: A study on high-dimensional spaces,” *Knowl. Inf. Syst.*, vol. 12, no. 1, pp. 95–116, 2007.

[49] B. Di Camillo, et al., “Effect of size and heterogeneity of samples on biomarker discovery: Synthetic and real data assessment,” *PLoS One*, vol. 7, no. 3, 2012, Art. no. e32200.

[50] A. Rosenwald, et al., “The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma,” *New England J. Med.*, vol. 346, no. 25, pp. 1937–1947, 2002.

[51] E. L. Kaplan and P. Meier, “Nonparametric estimation from incomplete observations,” *J. Amer. Statist. Assoc.*, vol. 53, no. 282, pp. 457–481, Jun. 1958.

Mehrdad J. Gangeh (M'05-SM'15) received the PhD degree in electrical and computer engineering from the University of Waterloo, Canada, in 2013. He is a research associate jointly in the Department of Medical Biophysics, University of Toronto, and the Department of Radiation Oncology, Sunnybrook Health Sciences Center, Canada. His current research is mainly focused on the design and development of computer-aided-theragnosis (CAT) systems using novel machine learning algorithms in conjunction with quantitative ultrasound methods in order to predict and monitor therapeutic cancer responses. His research interests are on multiview learning, dictionary learning and sparse representation, kernel methods, and deep belief networks with the applications to medical imaging, pattern recognition, big data analytics, computer vision, data mining, and bioinformatics. He is an associate editor of the *IEEE Transactions on Medical Imaging* and an adjunct professor in the Department of System Design Engineering, University of Waterloo. He is a senior member of the IEEE.

Hadi Zarkoob received the BSc and MSc degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2005 and 2008, respectively, and the master's degree in statistics from the University of Waterloo, ON, Canada, and the master's degree in management science and engineering from Stanford University, California. He is currently a research scientist with BaseHealth Inc., California.

Ali Ghodsi is an associate professor in the Department of Statistics, University of Waterloo. He is also cross-appointed in the School of Computer Science, a member of the Center for Computational Mathematics in Industry and Commerce and the Artificial Intelligence Research Group, University of Waterloo. His research involves applying statistical machine-learning methods to dimensionality reduction, pattern recognition, and bioinformatics problems. His research spans a variety of areas in computational statistics. He studies theoretical frameworks and develops new machine learning algorithms for analyzing large-scale datasets with applications to big data, bioinformatics, pattern recognition, and sequential decision-making.