Learning Proximity Relations for Feature Selection

Taiping Zhang, *Member, IEEE*, Pengfei Ren, Yao Ge, Yali Zheng, *Member, IEEE*, Yuan Yan Tang, *Fellow, IEEE*, and C.L. Philip Chen, *Fellow, IEEE*

Abstract—This work presents a feature selection method based on proximity relations learning. Each single feature is treated as a binary classifier that predicts for any three objects X, A, and B whether X is close to A or B. The performance of the classifier is a direct measure of feature quality. Any linear combination of feature-based binary classifiers naturally corresponds to feature selection. Thus, the feature selection problem is transformed into an ensemble learning problem of combining many weak classifiers into an optimized strong classifier. We provide a theoretical analysis of the generalization error of our proposed method which validates the effectiveness of our proposed method. Various experiments are conducted on synthetic data, four UCI data sets and 12 microarray data sets, and demonstrate the success of our approach applying to feature selection. A weakness of our algorithm is high time complexity.

Index Terms—Feature selection, feature evaluation, classification, microarray analysis, gene selection

1 Introduction

In many pattern recognition and machine learning applications, such as appearance-based image classification, document clustering, data mining, and information retrieval, we are involved to deal with high-dimensionality data which has thousands of features. Learning and classifying in such a high-dimensionality space is extremely difficult due to the *curse of dimensionality* [9], [62]. In fact, a small fraction among thousands of features is significant and relevant to their classes. The remaining is insignificant which only complicates data learning and modeling. Those insignificant features may seriously degrade the performance of machine learning algorithms. When involved with many insignificant features, even Support Vector Machine (SVM) [79], as one of the most successful classifiers, also works badly in that situation [38]. Thus, those insignificant features are in a way, irrelevant, redundant, and need to be removed.

1.1 Background and Main Applications

Feature selection is one of the most important branches of dimensionality reduction techniques [4], [27]. Unlike other dimensionality reduction techniques (feature transformation) to transform features into different feature spaces which cause the physical meaning loss, feature selection aims to select a feature subset which can highly preserve the original properties of samples for learning tasks. Beyond alleviating the effect of curse of dimensionality and speeding up the learning processing, there is a distinguishing advantage for feature selection. It's beneficial to discover the potential association among samples by visualizing the original attributes of points in low-dimension. A typical application for feature selection is Microarray Analysis [7], [11], [25], [32], [48], [49], [86]. Many researchers have explored the microarray technology to build cancer diagnosis, prognosis and prediction from gene expression data. However, the number of gene from microarray data is significantly large, and each gene carries independent genetic instructions for the development of the living organisms. Discovering the underlying associations from gene expressions to cancers needs feature selection techniques not only to reduce efficiently the high-dimensionality gene expression data, but also to preserve the physical integrity of gene for subsequent biological analysis.

1.2 Paper Contributions

In this work, we propose a new feature selection algorithm for high-dimensional data classification. The formulation of the proposed algorithm is based on proximity relations learning concept that feature selection problem is reduced to ensemble learning problem. Firstly, proximity relations is used to define a new quantitative criterion of features quality, that directly measures how well the features preserves the nearest neighbor structure. The key idea is that each feature defines a binary classifier (F) that predicts, for any three samples X, A, and B where X and A are from the same class, X and B are from the different class, whether X is closer to A. The error of F on a specific set of triples (X, A, B) can be used to measure how well the features preserve the nearest neighbor, which is a direct measure of the goodness of a feature. Secondly, since each feature is treated as a binary classifier, any linear combination of feature-based binary classifiers naturally corresponds to feature

- *T. Zhang is with the College of Computer Science, Institute of Computing and Data Sciences, Chongqing University, Chongqing 400030, P.R. China. E-mail: tpzhang@cqu.edu.cn.*
- *P. Ren and Y. Ge are with the College of Computer Science, Chongqing University, Chongqing 400030, P.R. China. E-mail: {pfren, geyao}@cqu.edu.cn.*
- *Y. Zheng is with School of Automation Engineering, University of Electronic Science and Technology of China, 2006 Xiyuan Avenue, Chengdu, China. E-mail: zhengyl@uestc.edu.cn.*
- *Y.Y. Tang and C.L.P. Chen are with the Faculty of Science and Technology, University of Macau, Avenida da Universidade, Taipa, Macau, China. E-mail: {yytang, philipchen}@umac.mo.*

Manuscript received 24 Feb. 2015; revised 7 Dec. 2015; accepted 9 Dec. 2015. Date of publication 7 Jan. 2016; date of current version 30 Mar. 2016. Recommended for acceptance by S. Yan. 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/TKDE.2016.2515588

selection. Thus, feature selection problem is reduced to an ensemble learning problem of combining many weak classifiers into an optimized strong classifier. The proposed proximity relations learning algorithm performs feature selection according to the proposed measure of feature quality using [34]. Finally, based on the proposed feature selection algorithm, we provide a theoretical analysis of the generalization error bound which is independent on the data dimensionality. This is a very encouraging result with respect to "curse of dimensionality" that also validates the effectiveness of our proposed method. We summarize the contributions of this paper as follows:

1) A quantitative criterion of feature quality is used to measure how well a feature preserve the nearest neighbor structure. Under this criterion, each feature is treated as a binary classifier. Therefore, the feature selection problem is transformed into classifier ensemble learning problem.
2) We provide a theoretical analysis show that the generalization error bounds of our proposed algorithm are not associated with the data dimensionality. The existing algorithm like Logo, the generalization error bounds depends logarithmically on the data dimensionality.
3) Our algorithm can easily deal with multi-class feature selection problem. Without any extension or approximation the multi-class problem is transformed into a binary classification problem under our algorithm framework. Most other competing algorithms are originally designed for binary classification which need to be extended approximately to multi-class settings.

The remainder of this paper is organized as follows. We briefly overview the related work in Section 2. Then we present the proposed feature evaluation criterion in Section 3. In Section 4, we describe how to select the optimal features from the candidate features. In Section 5, we analyze systematically the training error, generalization error and the time complexity. An set of extensive experiments are conducted to show the performance of the proposed methods on various benchmark data sets in Section 6. Finally, we conclude in Section 7.

2 Related Work

Many feature selection algorithms have been proposed for wide applications [36], [67], [88] in the past several years. The existing algorithms of feature selection can be categorized into three types generally: *Wrapper*, *Filter*, and *Embedded* approaches [10], [11], [16], [25], [65].

2.1 The Wrapper Methods

The wrapper methods first appeared in [24] and popularized in [64]. In *wrapper* methods, a learning algorithm as a black box is employed to measure the quality of each feature subset. When learning algorithms achieve the best results (e.g., classification accuracy) on a feature subset, and then the feature subset is selected. The typical learning algorithms are, for example, a nearest neighbor classifier [42], a decision tree [70], a Naive Bayes classifier [33]. However, these algorithms require an exhaustive search in the space of the feature subset, which is $NP-$hard, so they are computationally expensive. In practice, many applications have tens of thousands of features, such an exhaustive search is not feasible in this situation. Many efficient search strategies including forward selection, backward elimination, floating search, and genetic algorithm, were developed to avoid the exhaustive search. It can not guarantee to reach the optimal for selecting features even using these strategies [58], [59], [64], [89].

2.2 The Filter Methods

The *filter* methods evaluate the features without utilizing any learning or classification algorithms. A typical filter algorithm consists of two steps. In the first step, it ranks features based on certain criteria. In the second step, the features with highest rankings are chosen to induce classification models. RELIEF [39] defined a *consistency* measure to evaluate each feature in the k Nearest Neighbors of each sample, and consistency measures were extended in [6], [46]. Relevance measure [14] was used to evaluate the relevance between the features, the selected features can keep the top-relevance. Peng et al. used Relevance-redundancy [28] measure to select the relevant features which attempt to maximize the relevance between each feature and class label, and minimize the redundancy between two features, then the selected features addresses both relevance and redundancy. Ferreira and Figueiredo also proposed a relevance-redundancy feature selection (RRFS) method [20]. Maji and Pal used f-information [57] measure to evaluate features. *Dependence* measures [17] or *correlation* measures [30] were applied to evaluate the features. A fast correlation-based filter (FCBF) feature selection method was proposed in [44]. A square-correlation coefficient in [82] was employed to measure the similarity between features, and then an unsupervised forward orthogonal search algorithm is used to perform feature selection. Distance measures [80], [81] are common criteria for feature evaluation, including class separability, Fisher's criterion, or discrimination measure. The class separability measures the ratio of the between-class scattering to the within-class scattering of data. A feature subset with high class separability is regarded as a good one. Zhang et al. [15] used the pairwise constraints, constraint score to evaluate the features, which specified whether a pair of data samples belong to the same class or different classes. Brown et al. proposed a feature selection method by maximizing conditional likelihood [23]. The filter methods also incorporate combinatorial search through the space of possible features. The search strategies used in *wrapper* methods are also adopted in *filter* methods. Since the filter methods do not use any learning algorithms, they usually are not superior in accuracy when compared with *wrapper* methods. However, the wrapper methods take much longer than the filter methods, with their use being prohibitive on high-dimensionality data. Moreover, the wrapper methods may overfit the data for a given classifier, whereas the filter methods usually do not overfit.

2.3 The Embedded Methods

In *embedded* methods, the feature selection is incorporated into the classifier learning process [54], which have the advantages of *wrapper* methods—they include the interaction with the classification model and *filter* methods. The

TABLE 1
Notations

Notations	Description	Notations	Description
$\ell(x_i)$	Label function of x_i	$\mathbb{T}^j$	The triple set of the jth attribute
$S = \{x_i\}_{i=1^N}$	Training sample set	x_i^j	The jth attribute of data point x_i
J	The number of features	x^j	The jth attribute set
N	The number of samples	$d(x_i, x_k)$	The dissimilarity between points x_i and x_k
T_{max}	The number of selected features	$v(x^j)$	A threshold over x^j
$E(X)$	The expectation of X	$P(X)$	the probability of X

interested reader may refer to [25] for a detailed review. Embedded methods assign real-valued weights to different features instead of binary ones (a feature is either selected or not), to indicate their significance. Unlike *wrapper* and *filter* methods, embedded algorithms usually treat a feature selection problem as an optimization problem. Representative algorithm is to perform feature selection directly in the SVM classifier [5], [54], [55]. Feature selection problem in [53], [87] were solved as L_1 norm minimization problems. L_1 norm minimization leaded to a sparse solution [19], where relevant features received nonzero weights. Shah et al. in [49] proposed a feature selection algorithm by learning a conjunction (or disjunction) of decision stumps in Occam's Razor, Sample Compression, and PAC-Bayes learning settings. Zhao et al. proposed a conventional combinatorial optimization formulation for similarity preserving feature selection [90]. Song et al. proposed a feature selection algorithm based on dependence maximization between the selected features and the labels of an estimation problem [43]. Wu et al. proposed a new online streaming feature selection method to select strongly relevant and non-redundant features [85]. Song et al. proposed a fast clustering-based feature subset selection algorithm for high-dimensional data. Firstly, features are divided into clusters by using graph-theoretic clustering methods, Secondly, the most representative feature that is strongly related to target classes is selected from each cluster to form a subset of features [61]. Liu et al. proposed a global and local structure preservation framework for feature selection which integrates both global pairwise sample similarity and local geometric data structure to conduct feature selection [84]. A feature selection method was proposed based on a continuous ranking of the features measured by a least-squares optimization [83]. Recently, many feature selection methods based on sparse representation model in original and kernel spaces are proposed in [37], [50], [56], [60], [71], [86].

Though the wrapper and the filter are efficient in selecting relevant features. They may be unstable when performing on wide feature sets, respectively. Recently, A hybrid method which combines the filter and the wrapper methods is proposed to perform feature selection [16], [26], [45], [68], [91]. These methods first apply filter techniques to reduce the number of features, then, the wrapper techniques focuses on a smaller subset of features. It can obtain a reasonable computation complexity and accuracy.

3 Feature Evaluation

In this section we describe our theory of feature evaluation. We begin with some notations in Table 1 which are used in subsequent derivation, then several simple yet intuitively reasonable sufficient conditions for a feature to be a "good" feature are proposed in this section.

3.1 Fundamental Measure

Let $d(X, A)$ denote the euclidean distance between samples X and A, then if X, A, and B are three samples, one of the following relations among them must be true [78]:

- $d(X, A)$ is less than $d(X, B)$;
- $d(X, A)$ equals than $d(X, B)$;
- $d(X, A)$ is greater than $d(X, B)$.

We define a measure F as follows:

$$F(X, A, B) = d(X, B) - d(X, A). \tag{1}$$

Using the measure F, the dissimilarity relationships of triple (X, A, B) can be described as follows:

$$\text{sign}(F(X, A, B)) = \begin{cases} 1, & d(X, A) < d(X, B) \\ 0, & d(X, A) = d(X, B) \\ -1, & d(X, A) > d(X, B), \end{cases} \tag{2}$$

where sign(x) equals 1 for $x > 0$, 0 for $x = 0$, and -1 for $x < 0$.

3.2 Feature Quality Measure Over Triples

The objective of feature selection under supervised learning is to find the smallest feature subset which can achieve perfect classification accuracy. In order to achieve this objective, we firstly evaluate the goodness of a single feature. We define a set of triples $\mathbb{T}^j$ over a single feature set,

$$\begin{aligned} \mathbb{T}^j = \{(x_A^j, x_B^j, x_C^j) | x_A^j, x_B^j, x_C^j \in x^j, \\ \ell(x_B) = \ell(x_A) \neq \ell(x_C)\}, \end{aligned} \tag{3}$$

where $\ell(x_A)$, $\ell(x_B)$, and $\ell(x_C)$ are the labels of x_A, x_B, and x_C, respectively, and $j = 1, 2, \ldots, J$. In the ideal case, we expect that the following would hold true: $\forall x_A^j, x_B^j, x_C^j \in x^j$, $\ell(x_B) = \ell(x_A) \neq \ell(x_C), F(x_A, x_B, x_C) > 0$, which means the attributes between samples from the same class are closer than the ones from different classes. For x^j, we can classify the samples into correct groups using the dissimilarity measure F. However, in the actual cases, the measure F may not always indicate perfectly on $\mathbb{T}^j$, then the error rate of classification on $\mathbb{T}^j$ provides us with a quantitative measure of how well x^j classifying samples into correct classes. Lower error rate indicates that feature x^j is a good feature. In feature selection task, we wish to find the linear combination of features which achieves lowest error rate by using

the measure F. Next we will show how to describe the goodness of a feature.

We begin with a simplest notion of "good feature" motivated by [47] which is quite intuitive. This definition shows that x^j is a good feature if most samples are likely to be close to random samples from the same class than to random samples from different classes in term of this attribute. Assume the error parameter is denoted by ϵ which is used to measure the error rate, and the margin parameter is denoted by γ which is used to measure the distance from the decision boundary for each triple, we give the definition of being good for a feature.

Definition 1. *A feature x^j is said to be a $(\epsilon, \gamma)-$good feature for the data S, if at least with $1-\epsilon$ probability mass of samples for the attribute x_i^j satisfy:*

$$P\big(F(x_i^j, x_k^j, x_m^j) > 0 | \ell(x_i) = \ell(x_k) \neq \ell(x_m)\big) \geq 1/2 + \gamma/2. \tag{4}$$

Definition 1 captures an intuitive notion of a feature being good, and suggests a natural learning algorithm: draws pairs of samples with different labels, and predicts a new sample according to which class it is more likely to be close to. Following the definition $(\epsilon, \gamma)-$good feature, we can derive the following theorem:

Theorem 1. *With probability at least $1-\delta$ over the choice of $n = (4/\gamma^2)\ln(1/\delta)$ pairs of samples (z_{k1}^j, z_{k2}^j), where z_{k1}^j and z_{k2}^j are pair samples for the jth attribute $(z_{k1}^j, z_{k2}^j \in x^j)$ with labels $\ell(z_{k1})$ and $\ell(z_{k2})$ $(\ell(z_{k1}) \neq \ell(z_{k2}))$, $k = 1, 2, \ldots, n$, if x^j is a $(\epsilon, \gamma)-$good feature for the data S, the classifier:*

$$h(x) = \frac{1}{n}\sum_{k=1}^{n} \operatorname{sign}\Big(F(x, z_{k1}^j, z_{k2}^j)\Big) \tag{5}$$

has an error of at most $\epsilon + \delta$.

In many applications, the number of triple set is huge. Thus, there is high time complexity when Definition 1 is used to define a good feature in that situation. In order reduce the time complexity, we will consider another definition in the next section.

3.3 Feature Quality Measure over Pairs

Definition 1 requires that $x_i^j, (i = 1 \ldots N, j = 1 \ldots J$ is more likely to be close to random samples x_k^j with the same label than to x_m^j with the different label. Intuitively, we can think about a threshold which the distance between two random points with the same label is less than, and the distance between two random points with the different label is greater than. Formally:

Definition 2. *A feature x^j is said to be a $(\epsilon, \gamma)-$good feature for the data S, if there exists a threshold $v(x^j)$ such that at least $1-\epsilon$ probability mass of samples for the attribute x_i^j satisfies:*

$$P\Big(d(x_i^j, x_k^j) < v(x^j) \text{ and } d(x_i^j, x_m^j) > v(x^j) | \ell(x_k) = \ell(x_i) \neq \ell(x_m)\Big) \geq 1/2 + \gamma/2. \tag{6}$$

From the above definition, we can see that $v(x^j)$ only depends on the feature x^j. It should be pointed out that the single feature evaluation based Definition 2 has complexity $\mathcal{O}(N^2)$, this is much better than $\mathcal{O}(N^3)$ required by the algorithms based Definition 1. Actually, Definition 2 is more strict than Definition 1, if a feature x^j is a $(\epsilon, \gamma)-$good feature for the data S in the Definition 2 sense, it is also a $(\epsilon, \gamma)-$good feature in the Definition 1 sense. When a feature satisfies the condition of the $(\epsilon, \gamma)-$good feature in Definition 2, we have the following theorem:

Theorem 2. *With probability at least $1-\delta$ over the choice of $n = (4/\gamma^2)\ln(1/\delta)$ samples z_k^j with labels $\ell(z_k), z_k^j \in x^j$, if a feature x^j is a $(\epsilon, \gamma)-$good feature in the Definition 2 sense, the above algorithm produces a classifier as follows:*

$$h(x) = \frac{1}{n}\sum_{k=1}^{n} \operatorname{sign}\big(v(x^j) - d(x, z_k^j)\big) \tag{7}$$

has an error rate of at most $\epsilon + \delta$.

Definition 2 is defined on the pair sets. Therefore, It is lower time complexity than Definition 1.

4 Feature Selection

Based on the famous proposition by Thomas Cover—The two best features are not the best two [13]. It means the combined feature by the best two features is not a good feature. Therefore, feature selection is not a feature combination with the good features. In order to select the best combined features, then we must define some criterions so as to ensure the best combined features are selected.

In addition, each selected feature contributes classification power of the selected feature subset differently, Thus, each feature must weight according to their contribution. In our work, the task of feature selection is not only to label a feature either selected or not, but also to assign weight to the single feature to achieve the most discriminative power of the selected features. Based on the Definitions 1 and 2, the criterion of feature selection can be expressed as: the final feature subset constituted by the nonnegative weighted selected features is a $(\epsilon, \gamma)-$good feature for data S, where the error ϵ is expected to keep as small as possible and the margin γ is expected to keep as large as possible. Feature selection is to select the optimal feature subset such that the selected features can achieve the minimum classification error. The observation leads us to minimize the following objective function,

$$\begin{aligned}\operatorname{Err}(x, \alpha) = 1 - \frac{2}{n^2 - n}\sum_{i=1}^{n}\sum_{k=1(k\neq i)}^{n} I_{-1}[\ell(\bar{x}_i) = \ell(\bar{x}_k)] \\ \cdot \operatorname{sign}(v(\bar{x}) - d(\bar{x}_i, \bar{x}_k)),\end{aligned} \tag{8}$$

where $\bar{x}_i = \alpha \otimes x_i = (\alpha^1, \alpha^2, \ldots, \alpha^J) \otimes (x_i^1, x_i^2, \ldots, x_i^J) = (\alpha^1 x_i^1, \alpha^2 x_i^2, \ldots, \alpha^J x_i^J)$, here $\otimes$ is the Hadamard product operator, $\alpha \geq 0$ are the weighted coefficients of features.

Note that the minimization of the objective function (8) is a very difficult task, which involves an exhaustive search for all possible combinations of candidate features. Since each feature is a weak binary classifier, in order to make the optimization tractable, we employ boosting method [22], to learn a strong classifier as a linear combination of

these weak classifiers such that the strong classifier can achieve the smallest misclassification rate on training data. The final strong classifier is the result of feature selection. We choose Gentleboost algorithm [34] to build our feature selection algorithms due to the numerically robustness and easy implementation. Gentleboost algorithm attempts to optimize the following cost function

$$E\Big(e^{-\sum_j \sum_{i=1}^{n} \sum_{k=1(k\neq i)}^{n} I_{-1}[\ell(\bar{x}_i^j)=\ell(\bar{x}_k^j)]\text{sign}\left(v^j - d(\bar{x}_i^j, \bar{x}_k^j)\right)}\Big). \quad (9)$$

It trains a weak classifier by minimizing a weighted squared error at each round. That is, at t round, the strong classifier H_m will be updated as: $H_m = H_{m-1} + h_t$, where h_t is chosen by minimizing the following cost function:

$$\arg\min_{h_t} E\big(e^{-l(x)\cdot H_{m-1}(x)}(l(x) - h_t)^2\big), \quad (10)$$

which is a second order Taylor approximation of the cost function of AdaBoost. In general, Gentleboost use the regression error to measure the goodness of a single feature at each iteration. The goodness of the combination of features is measured by the classification error. Similarly, we attempt to optimize the following objective function at m round in our task

$$\begin{aligned}\arg\min_{x^m}\ E\Big(e^{-I_{-1}[\ell(x_i^m)=\ell(x_k^m)]H_{m-1}(x)}\Big(I_{-1}[\ell(x_i^m) = \ell(x_k^m)] \\ - \boldsymbol{\alpha}^m \text{sign}(v(x^m) - d(x_i^m, d_k^m))\Big)^2\Big).\end{aligned} \quad (11)$$

If we denote the distribution weights $w_{i,k}^m = e^{-I_{-1}[\ell(x_i^m)=\ell(x_k^m)]H_{m-1}(x)}$, the objective function (11) can be rewritten as:

$$\begin{aligned}\arg\min_{a^m} \sum_i \sum_{k,k\neq i} w_{i,k}\Big(I_{-1}[\ell(x_i^m) = \ell(x_k^m)] \\ - \boldsymbol{\alpha}^m \text{sign}(v(x^m) - d(x_i^m, d_k^m))\Big)^2.\end{aligned} \quad (12)$$

The weak learners h_t can be written as

$$\begin{aligned}H_t(v(x^m)) = a^m \delta\big(d(x_i^m, d_k^m) < v(x^m)\big) \\ - a^m \delta\big(d(x_i^m, d_k^m) > v(x^m)\big),\end{aligned} \quad (13)$$

where δ is the indicator function. We can learn the best weak learner: We search over all possible features x^m to split on, and for each one, we search over all possible thresholds v^m induced by sorting the pair distances, given x^m and v^m, we can obtain the optimal a^m by weighted least squares,

$$\begin{aligned}a^m = \frac{\sum_i \sum_k (k \neq i) w_{i,k} I_{-1}[\ell(x_i^m) = \ell(x_k^m)]}{\sum_i \sum_k (k \neq i) w_{i,k}} \\ \cdot \big(\delta\big(d(x_i^m, d_k^m) < v(x^m)\big) - \delta\big(d(x_i^m, d_k^m) > v(x^m)\big)\big).\end{aligned} \quad (14)$$

For each feature x^m, we choose the thresholds v^m corresponding a with the lowest cost in (12) and add this weak learner (feature) to the selected features, and then the algorithm goes to next round until stop condition is satisfied. The detail implementation is summarized in Algorithm 1.

When we obtain a good feature subset $(x^{j1}, x^{j2}, \ldots)$ and the weight sets $(\alpha_{j1}, \alpha_{j2}, \ldots)$ by run Algorithm 1. The weighted vector $(\alpha_{j1}x^{j1}, \alpha_{j2}x^{j2}, \ldots)$ as a good feature vector is used in subsequent applications. It should be pointed out that our proposed algorithm is an embedded method. In addition, for unsupervised learning problem, we label the samples based on k-nearest neighbor(k-NN) relationship, that is, if x_j lie in the k-NN set of x_i , then we set $\ell(x_i) = \ell(x_j)$, otherwise, $\ell(x_i) \neq \ell(x_j)$. Therefore, the proposed algorithm can be work for unsupervised learning problem.

Algorithm 1. The Feature Selection Algorithm

1: **Input:** candidate feature $x^j, j = 1, 2, \ldots, J$, the label $\ell(x_i)$, $i = 1, 2, \ldots, N$
2: Initialize the selected feature subset $\mathbf{F}_0 = [\,]$, and the weight subset $\boldsymbol{\alpha}_0 = [\,]$, and the distribution $w_{i,k}^1 = \frac{2}{n^2-n}$
3: Compute the pairwise dissimilarities $d(x_i^j, x_k^j)$ for all candidate features
4: t = 0
5: **while** the stop condition is not satisfied, where the stopping condition is t is equals to T_x or $Err(\mathbf{F}_t, \boldsymbol{\alpha}_t) > Err(\mathbf{F}_{t-1}, \boldsymbol{\alpha}_{t-1})$ **do**
6: **for** j, all feature indices without being selected **do**
7: Build the binary classifier for each feature.

$$h_t^j = \text{sign}(v(x^j) - d(x_i^j, x_k^j)).$$

8: Evaluate the error rate of binary classifier corresponding each feature

$$error(j) = \sum_i \sum_{k,k\neq i} w_{i,k}^t \Big(I_{-1}[\ell(x_i)^j = \ell(x_k^j)] - \boldsymbol{\alpha}^j \cdot h_t^j\Big)^2.$$

9: Find the best binary classifier

$$j^* = \arg\min_j error(j),$$

and obtain the weight α^{j^*}, and threshold $v(x^{j^*})$
10: **end for**
11: Update the selected feature subset and weight subset

$$\mathbf{F}_t = [\mathbf{F}_{t-1}, x^{j^*}], \boldsymbol{\alpha}_t = [\boldsymbol{\alpha}_{t-1}, \alpha^{j^*}].$$

12: Compute the error of the strong classifier $Err(\mathbf{F}_t, \boldsymbol{\alpha}_t)$.
13: Update the distribution

$$\begin{aligned}w_{i,k}^{t+1} = w_{i,k}^t \exp\Big(-I_{-1}[\ell(x_i) = \ell(x_k)]\alpha^{j^*} \\ \cdot \text{sign}(v(x^{j^*}) - d(x_i^{j^*}, x_k^{j^*}))\Big), \\ w_{i,k}^{t+1} = w_{i,k}^{t+1} / \sum_i \sum_{k,k\neq i} w_{i,k}^{t+1},\end{aligned}$$

14: $t = t + 1$.
15: **end while**
16: **Output:** $\mathbf{F}_t$ and $\boldsymbol{\alpha}_t$.

5 Error and Complexity Analysis

In this section, we provide a theoretical analysis on the generalization error of our feature selection algorithm, which can effectively explain why our algorithm can perform well for high-dimensional data. Let us start from the training error analysis. The training error is also called the "empirical error".

Definition 3 (Training error). *Given a training set S which independently draws from the unknown distribution D, the*

training error Err_S of the classifier H is the observed number of errors :

$$\begin{aligned} Err_S(H) &= Pr_{(X,Y)\sim S}(H(x_i,x_k) \neq I_{-1}[\ell(x_i)=\ell(x_k)]) \\ &= \sum_i \sum_{k,k\neq i} I_0[H(x_i,x_k) \neq I_{-1}[\ell(x_i)=\ell(x_k)]]. \end{aligned} \tag{15}$$

Theorem 3. *Let X^m, α^m be the selected features and corresponding feature weights that are computed by running the Algorithm 1, $m = 1, 2, \ldots, T_{max}$. Then, the following bound holds on the training error of $H = \text{sign}(\sum_{m=1}^{T_{max}} \alpha^m h_m)$*

$$\begin{aligned} &\frac{2}{n^2-n}\left\{ i,k : \text{sign}\left(\sum_m \text{sign}(v^m - d(x_i^m, x_k^m))\right) \right. \\ &\quad \left. \neq I_{-1}[\ell(x_i)=\ell(x_k)] \right\} \\ &\leq \frac{2}{n^2-n}\prod_{m=1}^{T_{max}} \exp(1-2e_m)e_m + \exp(2e_m-1)(1-e_m), \end{aligned} \tag{16}$$

where $x_i = (x_i^1, x_i^2, \ldots, x_i^J)$, $x_k = (x_k^1, x_k^2, \ldots, x_k^J)$, and e_m is the normalized weighted error defined as

$$\begin{aligned} e_m = \sum_i \sum_{k,k\neq i} w_{i,k}^m I_0 \left[\sum_m (v^m - d(x_i^m, x_k^m)) \right. \\ \left. \neq I_{-1}[\ell(x_i)=\ell(x_k)] \right] \Big/ \sum_i \sum_{k,k\neq i} w_{i,k}^m. \end{aligned} \tag{17}$$

The indicator $I_0[x]$ outputs 1 when the Boolean variable x is true, and 0 otherwise.

The generalization error is called the "true error". We define the following quantities which are used to define the generalization error.

Definition 4 (Rademacher complexity [40]). *Given a set X with a fixed distribution D, let $S = \{x_1, x_2, \ldots, x_n\}$ be a set of samples of X drawn i.i.d from D. Let $\mathcal{F}$ be a class of function $f : X \to \mathbb{R}$. Define the random variable*

$$\hat{R}_n(\mathcal{F}) = E_\sigma \left[sup_{f\in\mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^n \sigma_i f(x_i) \right) \right], \tag{18}$$

where $\sigma_1, \sigma_2, \ldots, \sigma_n$ are independent random variables uniformly chosen from $\{\pm 1\}$, and $E(\cdot)$ is the expectation. Then Rademacher complexity *of $\mathcal{F}$ is defined as*

$$R_n(\mathcal{F}) = E(\hat{R}_n(\mathcal{F})).$$

From a regression point of view, the quantity $\hat{R}_n(\mathcal{F})$ measures the ability of function from $\mathcal{F}$ to fit random noise on a fixed set S under correlation measure, while $R_n(\mathcal{F})$ is a measure of the expected noise-fitting-ability of $\mathcal{F}$ over all data sets $S \in X$ that could be drawn according to the distribution D.

Definition 5 (Generalization error). *The generalization error Err_D of the classifier H (a linear combination of binary classifiers that corresponds to the selected features) is defined as the probability of classification error over samples drawn from the distribution D:*

$$Err_D(H) = P_{(X,Y)\sim D}(H(x_i,x_k) \neq I_{-1}[\ell(x_i)=\ell(x_k)]). \tag{19}$$

In real application, the generalization error is not an observable quantity because the distribution D is unknown. However, one has the access to the training error which is measurable. A good learning algorithm trained on a limited number of training samples can perform well on out-of-samples. It is well known in the study of machine learning that directly minimizing the training error tends to overfit the training data [48]. We use the exponential loss function (9) instead of the misclassification loss (8) for quantizing the generalization error.

Given a data set X, let H be the collection of the binary classifier h^m defined by Definition 2 corresponding to each selected feature x^m, z be a variable defined over a pair of samples, and $Y = \{\pm 1\}$ is the label set which indicates the sample pair belonging to the same class or not, $Z = X \times X \times Y$, $z \in Z$. Let $\mathcal{F}$ be a class of measurable function as:

$$\mathcal{F} = \left\{ f : z \mapsto \sum_{h^t\in\mathcal{H}} a_t h_t, z \in Z \right\}. \tag{20}$$

We have the following theorem regarding of the generalization error of our algorithm:

Theorem 4. *Let D be a probability distribution on Z. If $\mathcal{F}$ be a class of functions: $f : Z \mapsto (R)$, and $S = \{X_i, X_k, Y_{ik}\}$ be a set of examples drawn i.i.d. from D, then with probability at least $1-\delta$ over the draw of S, for every function $f \in \mathcal{F}$,*

$$Err_D(f) \leq= Err_S(f) + R_n(\mathcal{F}) + \sqrt{\frac{ln(1/\delta)}{n}}. \tag{21}$$

Unlike the other feature selection algorithms (e.g., Logo), its generalization error bounds depend logarithmically on the data dimensionality. As can be seen from (21), the generalization error bounds of our algorithm are not associated with the data dimensionality.

5.1 Complexity Analysis

Before we start the feature selection algorithm, we need to compute pairwise distances for all candidate features, which needs $O(N^2J)$ operations. To decide thresholds v, it take $O(N \log(N))$ to sort the error, and find the optimal v for each single feature. During feature selection, as each round, we evaluate J weak classifiers by measuring their performance on $N(N-1)/2$ pairwise distances which takes $O(N(N-1)J)$. Therefore, our algorithm will take $O(N \log N \cdot J + (T_{max}+1)N^2J)$ totally for selecting T_{max} features, which has a linear complexity with respect to the number of features, and a quadratic complexity with respect to the number of samples. The existing algorithms I-RELIEF [73], Simba [63], RRFS [20], FCBF [44], and Logo [87] also have low complexity with respect to the dimensionality of the data. In feature selection applications (e.g., gene classification), the number of feature is usually excessively large. Our algorithm is better than the popular greedy search methods (e.g., forward search) in computational complexity which has a quadratic complexity with respect to the number of features.

TABLE 2
Summary of the UCI and Artificial Sets Used in the Experiments

Dataset	Samples	Features	Positive	Negative	Training	Testing	No. of Classes
Spiral	460	$2+(10^2, 10^4)$	230	230	/	/	2
COIL2000	4,000	85	3,762	238	800	3,200	2
Musk	476	166	207	269	85	391	2
DBworld	64	4,702	29	35	13	51	2
SECOM	1,567	590	104	1,463	312	1,255	2

6 Experiments

In this section, various experiments are conducted on 18 synthetic and real-world data sets to demonstrate the effectiveness of the proposed algorithm by comparing with the results of several existing feature selection methods, including Logo [87], I-RELIEF [73], Simba [63], mRMR [28], SVM-RFE [32], Simple Forward Search (SFS) [59], Lasso [29], InfoGain [2], and Fisher Score (FScore) [8]. The L^1 distance is used as the dissimilarity measure. The data sets are summarized in Tables 2 and 5.

6.1 Experiments on Synthetic Data

This experiment is conducted on the well-known Fermat's spiral data. This data includes two classes, and each class contains 230 samples distributed in 2D space, as illustrated in Fig. 1. We design an intuitive experiment to show the benefits of our method which is able to find the good features among many irrelevant features. We randomly sample 1,000 and 10,000 irrelevant features from Gaussian distribution with zero mean and unit variance, and add to each samples respectively. The first two features are good, so an effective feature selection algorithm must be able to identify the first two relevant features.

The results of our algorithm on the spiral data with additional 1,000 and 10,000 irrelevant features are illustrated in Fig. 1. Our algorithm effectively finds out the good features (the first two features as good ones) among all irrelevant ones. It should be pointed out the feature weights are learned by our algorithm from the spiral data set with additional 1,000 and 10,000 irrelevant features are identical. This is consistent with our feature selection algorithm presented in Section 4, which means that the feature weights are only dependent on the classification ability of corresponding feature, and are independent on the dimensionality of data.

6.2 Experiments on UCI Data

In this section, we test our algorithm experiments on the UCI data sets [21], including Insurance Company Benchmark (COIL2000), Musk, DBWorld e-mails (DBWorld), and SECOM. Some data is imbalance data. The detail of these data sets are summarized in Table 2. It should be pointed out that the COIL2000 and SECOM are imbalanced data.

In this experiment, KNN, Naive Bayes, and SVM classifiers are used to estimate the classification accuracy for its simplicity. For all algorithms, the number of selected feature is set as 5 and 15. In order to obtain stable results, each algorithm runs 10 times for each data set. In each test, a data set is randomly partitioned into the training and testing set. The averaged classification accuracy are reported in Tables 3 and 4. For a rigorous comparison, the McNemar [18] statistical test with the significance level of 0.05 is performed. According to the test result, the sign "win" indicates the other competing algorithms are significantly better than our algorithm, the sign "loss" indicates the competing algorithms are significantly worse than our algorithm, and the sign "tie" indicates there is no statistical difference between the competing algorithms and our algorithm. We observe that our algorithm performs the best in nearly all data sets. The result of statistical test show our algorithm is significantly better than other competing algorithm in most situations. The results show that our algorithm can effectively discover the best relevant features, while the other competing algorithms fail to find the best relevant features in most situations.

6.3 Experiments on Microarray Data

In this section, we apply the feature selection algorithm to gene expression data (microarray data). In this experiment, we demonstrate the effectiveness of our algorithm using 12 microarray data sets, including Prostate cancer [3], Central

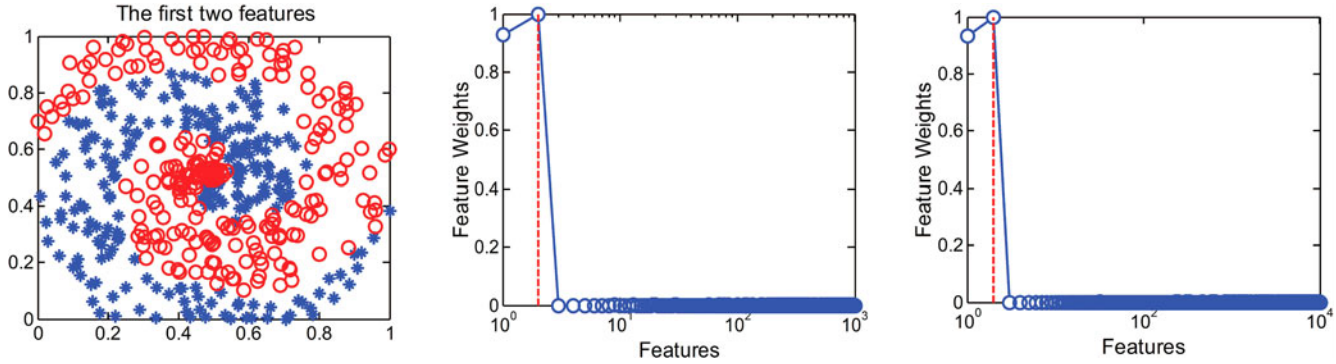


Fig. 1. The spiral data and feature weights are learned on the spiral data with 1,000 and 10,000 irrelevant features. Note the first two features of this data are only relevant features. The larger feature weight, the better classification ability of the feature is. Zero-valued feature weights indicate the irrelevant.

TABLE 3
The Classification Accuracy (Percent) on COIL2000 and DBworld Data Sets

Dataset	COIL2000						DBworld					
Classifier	kNN (k = 1)		Naive Bayes		SVM		kNN (k = 1)		Naive Bayes		SVM	
features	5	15	5	15	5	15	5	15	5	15	5	15
SVM-RFE	89.7	89.0	93.9	91.6	87.3	86.5	70.7	73.7	56.1	55.3	**77.0**	73.1
SFS	81.5	89.0	93.5	93.6	83.9	86.9	67.0	67.2	55.2	56.2	76.6	**79.9**
I-RELIEF	89.1	89.3	88.1	81.9	86.6	86.7	74.8	78.7	54.6	55.9	65.6	67.2
Simba	89.2	89.1	93.3	87.2	86.6	86.3	73.3	74.6	56.2	55.3	72.3	72.3
mRMR	89.4	89.1	93.6	**94.0**	85.8	86.6	56.5	63.6	51.3	56.1	52.1	61.5
FScore	89.7	89.2	91.4	89.4	86.4	86.9	75.8	80.1	52.6	56.2	74.6	75.4
InfoGain	89.2	89.2	90.0	89.7	86.6	86.9	75.0	78.3	46.8	55.9	70.4	77.6
Lasso	93.5	89.2	93.2	**94.0**	85.1	86.7	66.0	66.0	55.8	56.0	66.8	68.4
Logo	89.9	89.7	91.7	90.2	86.1	86.8	70.7	70.7	56.2	56.0	72.5	77.8
ours	**93.7**	**92.9**	**94.0**	**94.0**	**92.3**	**87.1**	**77.5**	**81.1**	**56.8**	**56.8**	**77.0**	79.1
Significance Test (win/tie/loss) with respect to our algorithm using McNemar Test ($\alpha = 0.05$)												
SVM-RFE	2/0/8	1/2/7	0/10/0	0/10/0	0/0/10	0/10/0	0/2/8	1/2/7	0/10/0	0/10/0	0/10/0	0/7/3
SFS	0/0/10	0/0/10	0/6/4	0/8/2	0/0/10	0/10/0	0/3/7	0/2/8	0/10/0	0/10/0	0/9/1	1/9/0
I-RELIEF	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	0/10/0	2/3/5	1/2/7	0/10/0	0/10/0	0/0/10	0/1/9
Simba	0/0/10	0/0/10	0/2/8	0/0/10	0/0/10	0/10/0	0/2/8	0/1/9	0/10/0	0/10/0	0/1/9	0/1/9
mRMR	0/0/10	0/0/10	0/8/2	0/10/0	0/0/10	0/10/0	0/2/8	0/2/8	0/8/2	0/10/0	0/0/10	0/10/0
FScore	0/1/9	0/0/10	0/0/10	0/0/10	0/0/10	0/10/0	0/2/8	0/2/8	0/10/0	0/10/0	0/6/4	0/9/1
InfoGain	0/0/10	0/0/10	0/2/8	0/3/7	0/0/10	0/10/0	0/2/8	0/1/9	0/10/0	0/10/0	0/0/10	0/10/0
Lasso	0/0/10	1/2/7	2/2/6	0/10/0	0/0/10	0/10/0	0/3/7	0/2/8	0/10/0	0/10/0	0/3/7	0/6/4
Logo	0/0/10	2/0/8	2/1/7	2/0/8	0/0/10	0/10/0	0/2/8	0/1/9	0/10/0	0/10/0	0/5/5	0/10/0

TABLE 4
The Classification Accuracy (Percent) on Musk and SECOM Data Sets

Dataset	Musk						SECOM					
Classifier	kNN (k = 1)		Naive Bayes		SVM		kNN (k = 1)		Naive Bayes		SVM	
features	5	15	5	15	5	15	5	15	5	15	5	15
SVM-RFE	65.8	67.9	61.7	64.0	63.7	65.9	86.4	88.2	91.2	93.1	**93.3**	92.1
SFS	56.3	69.1	57.3	62.9	56.7	63.8	87.4	86.8	92.7	90.8	84.3	87.3
I-RELIEF	65.4	67.8	62.3	60.6	63.7	66.7	88.6	88.6	90.2	88.3	92.1	86.6
Simba	65.8	70.6	63.8	66.2	63.3	67.9	88.4	88.5	92.3	66.4	85.7	86.4
mRMR	65.3	73.2	59.4	61.5	59.9	61.9	87.7	87.7	**93.3**	93.1	90.3	90.7
FScore	65.7	67.9	64.2	66.4	65.2	68.8	88.3	88.0	90.8	90.4	91.2	87.0
InfoGain	67.0	69.8	62.6	65.7	64.1	68.4	87.3	87.7	77.8	75.1	86.5	87.7
Lasso	**69.0**	73.9	59.8	62.4	59.0	65.9	87.6	88.2	92.0	87.8	86.4	88.8
Logo	63.5	63.5	65.6	62.3	66.3	67.7	89.1	89.2	91.1	89.8	86.1	88.9
ours	68.9	**73.5**	64.4	**66.5**	**66.6**	**69.9**	**93.2**	**93.4**	93.0	**93.3**	**93.3**	**93.1**
Significance Test (win/tie/loss) with respect to our algorithm using McNemar Test ($\alpha = 0.05$)												
SVM-RFE	0/3/7	0/3/7	0/2/8	0/3/7	0/1/9	0/3/7	0/2/8	0/2/8	0/8/2	0/10/0	0/10/0	0/7/3
SFS	0/4/6	0/4/6	0/8/2	0/8/2	0/3/7	0/6/4	0/0/10	0/0/10	0/6/4	0/8/2	0/1/9	0/1/9
I-RELIEF	0/4/6	0/5/5	0/7/3	0/6/4	0/6/4	0/4/6	0/0/10	0/0/10	0/3/7	0/2/8	0/9/1	0/1/9
Simba	0/6/4	0/4/6	0/5/5	0/5/5	0/4/6	0/5/5	0/0/10	0/0/10	0/8/2	0/0/10	0/1/9	0/1/9
mRMR	0/3/7	0/2/8	0/5/5	0/5/5	0/4/6	0/3/7	0/0/10	0/0/10	2/6/2	2/6/2	0/4/6	0/5/5
FScore	0/6/4	0/5/5	0/8/2	1/5/4	0/6/4	0/6/4	0/1/9	0/0/10	0/3/7	0/4/6	0/6/4	0/9/1
InfoGain	0/4/6	0/5/5	0/7/3	0/7/3	0/6/4	0/5/5	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	0/10/0
Lasso	3/4/3	3/5/2	0/8/2	0/4/6	0/5/5	0/4/6	0/0/10	0/1/9	0/4/6	0/3/7	0/0/10	0/1/9
Logo	0/5/5	0/6/4	3/5/2	0/7/3	0/4/6	0/4/6	0/0/10	0/2/8	0/3/7	0/2/8	0/5/5	0/10/0

Nervous System(CNS) [69], Colon tumor [76], diffuse large B-cell lymphoma (DLBCL) [51], ALL-AML [75], Breast cancer [41], SRBCT [35], MLL [66], Lymphoma [1], ALL-AML-3 [75], ALL-AML-4[1] [75], Multi-CNS[2] [69]. A overview of these 12 data sets is summarized in Table 5. Note that two types of experiments are performed on microarry data sets, one is binary classification, another involves multi-class classification. Due to the small number of samples and imbalance samples in microarray data sets, cross-validation error estimation would be considered highly unreliable [7], [77]. Braga-Neto and Dougherty in [77] suggest .632 boostrap may be more

1. For a detailed description see [74].
2. Named "Dataset A" in online page, available at http://www-genome.wi.mit.edu/mpr/CNS/.

TABLE 5
Summary of the Microarry Sets Used in the Experiments

Notation	Dataset	No. of Features	No. of Instances	Train	Test	No. of Classes
PRO	Prostate cancer	12,600	102	/	/	2
CNS	CNS	7,129	60	/	/	2
COL	Colon tumor	2,000	60	/	/	2
DLBCL	DLBCL	7,129	77	/	/	2
AA	ALL-AML	7,129	72	/	/	2
BRE	Breast cancer	24,481	97	/	/	2
SRBCT	SRBCT	2,308	83	/	/	4
MLL	MLL	12,582	72	/	/	3
LYM	Lymphoma	4,026	62	/	/	3
AA-3	ALL-AML-3	7,129	72	/	/	3
AA-4	ALL-AML-4	7,129	72	/	/	4
MCNS	Multi-CNS	7,129	42	/	/	5

TABLE 6
Classification Accuracy (Percent) on Microarray Data Sets (the Number of Genes Is 30)

Datasets		PRO	COL	DLBCL	AA	BRE	CNS	SRBCT	MLL	LYM	AA-3	AA-4	MCNS
kNN Classifier (K = 3)													
Wrappers	SVM-RFE	90.8	85.9	91.5	94.3	74.2	68.7	96.5	93.3	95.1	89.6	87.5	78.8
	SFS	87.7	84.2	89.6	92.5	72.4	67.6	95.6	89.3	95	89.2	88.6	79.2
	I-RELIEF	88.3	83.2	86.5	86.2	71.8	66.5	96.7	93.5	95.2	90.4	88.5	80.1
	Simba	84.1	83.3	86.1	85.8	67.5	65.8	97.6	93.1	96.1	90.8	89.1	80.5
Filters	mRMR	79.5	76.7	80.5	82.3	62.2	66.1	95.2	84	94.1	79.6	74.1	65.7
	FScore	89.2	82.7	90.9	93.5	72.3	69.2	94.7	88.2	94.5	86.4	85.6	76.5
	InfoGain	80.2	75.8	82.1	85.5	65.1	67.3	94.5	87.8	94.1	80.5	74	65.5
	Lasso	89.2	85.3	90.7	95.2	74.2	68.1	/	/	/	/	/	/
Embeddeds	Logo	91.1	85.9	89.8	94.9	**75.4**	69.5	97.5	93.5	96.3	90.6	88.9	80.1
	ours	**92.4**	**86.1**	**92.5**	**95.5**	74.5	**70.8**	**98.5**	**96.5**	**97.3**	**96.6**	**92.6**	**82.3**
Naive Bayes Classifier													
Wrappers	SVM-RFE	90.5	85.5	91.1	93.8	73.8	68.5	95.2	91.8	93.9	89.0	86.4	78.0
	SFS	87.6	83.9	89.5	92.1	72.1	67.3	94.2	89.0	93.5	88.2	88.1	78.5
	I-RELIEF	88.1	82.9	86.0	85.8	71.6	66.3	96.5	92.0	94.2	90.1	87.0	79.1
	Simba	83.9	83.1	85.7	85.6	67.3	65.3	96.2	91.6	94.2	89.7	89.0	79.4
Filters	mRMR	79.0	76.3	80.2	81.8	61.9	65.8	94.2	83.2	92.8	79.5	73.4	64.5
	FScore	89.1	82.6	90.6	93.2	72.1	69.1	94.5	86.9	93.0	85.9	85.0	76.0
	InfoGain	80.1	75.4	81.8	85.3	64.6	66.8	94.0	87.5	93.0	80.4	72.8	64.4
	Lasso	89.1	85.2	90.5	94.7	74.1	67.6	/	/	/	/	/	/
Embeddeds	Logo	90.7	85.7	89.5	94.4	**75.2**	69.2	96.6	92.8	95.1	90.4	87.7	79.1
	ours	**92.0**	**85.8**	**92.2**	**95.2**	74.4	**70.7**	**97.0**	**95.1**	**96.1**	**95.3**	**92.3**	**82.0**
SVM Classifier with Linear Kernel													
Wrappers	SVM-RFE	90.7	85.8	91.1	94.2	73.8	68.5	96.2	91.8	93.3	87.8	87.3	78.0
	SFS	87.6	84.2	89.5	92.9	72.4	67.3	94.5	87.4	94.4	87.9	88.4	78.8
	I-RELIEF	88.3	82.5	86.8	85.7	71.5	66.6	94.6	91.4	93.2	89.2	87.3	78.4
	Simba	84.0	83.2	86.1	86.2	67.5	65.8	96.8	91.9	95.3	88.8	87.4	79.8
Filters	mRMR	79.5	76.9	80.5	82.3	62.3	66.0	93.9	83.7	93.6	78.9	72.1	64.5
	FScore	89.2	82.4	90.7	93.7	72.3	69.3	94.2	87.8	93.9	84.8	85.3	76.1
	InfoGain	79.9	75.9	82.0	85.1	65.3	67.4	92.9	87.2	92.8	78.9	72.8	64.2
	Lasso	89.0	85.3	90.7	95.0	74.4	68.1	/	/	/	/	/	/
Embeddeds	Logo	90.9	85.8	89.8	94.8	**75.2**	69.7	96.9	91.7	95.3	89.8	87.9	79.5
	ours	**92.2**	**86.1**	**92.4**	**95.5**	74.5	**70.9**	**97.4**	**95.9**	**96.5**	**95.4**	**92.5**	**80.9**

appropriate than other error estimators, like k-fold cross-validation, leave-one-out, and re-substitution estimator. Therefore, in this test, a balanced external .632 bootstrap error estimator [12] is applied to evaluate the performances of the feature selection algorithms, where each sample is made to appear C times in the resampling. The .632 bootstrap estimator samples a training set with replacement from original data set. The testing set is formed by the other samples omitted from the training set. The bootstrap error $\varepsilon_{.632}$ is defined as

$$\varepsilon_{.632} = 0.368\beta_1 + 0.632\beta_2, \tag{22}$$

where β_1 and β_2 are the training error[3] and test error, respectively. To eliminate statistical variations, the

3. Leave-one-out estimator is used to evaluate training error.

TABLE 7
CPU Time per Run (in Seconds) of All Algorithms Performed on All Data Sets

Data sets	SVM-RFE	SFS	I-RELIEF	Simba	mRMR	FScore	InfoGain	Lasso	Logo	ours
COIL2000	120.1	163.7	4.7	13.5	0.03	**0.01**	**0.01**	6.5	14.1	0.4
Dbworld	55.3	69.4	0.0	0.3	0.32	0.71	***0.2***	2.8	1.8	5.1
LSVTVoice	4.2	5.1	0.0	0.1	0.09	0.04	**0.01**	1.1	0.0	0.4
Musk	6.2	7.1	0.1	0.8	0.04	0.02	**0.01**	24.7	0.6	0.3
SECOM	111.2	125.0	7.6	10.2	0.17	0.09	**0.02**	77.1	28.5	1.6
Prostate cancer	42.5	47.3	12.9	5.5	2.10	**1.77**	1.91	71.1	34.5	55.6
Colon tumor	9.7	10.8	2.2	1.6	0.80	**0.39**	0.51	16.7	4.8	7.0
DLBCL	27.8	30.8	7.1	3.1	1.80	**1.57**	1.61	46.5	18.6	29.8
ALL-AML	20.1	22.9	6.8	2.9	1.70	**1.48**	1.52	35.4	18.8	25.6
Breast cancer	109.5	120.4	32.3	12.6	3.60	**3.18**	3.32	181.6	79.7	120.2
CNS	32.0	35.6	6.9	2.8	1.50	1.45	**1.42**	53.5	17.9	37.2
SRBCT	14.0	15.5	95.3	1.8	0.90	0.83	**0.82**	24.0	16.2	10.5
MLL	56.0	62.5	14.7	6.1	1.80	**1.71**	1.75	94.1	42.6	64.2
Lymphoma	19.0	21.6	20.8	14.9	2.40	**2.20**	2.35	32.8	7.2	20.6
ALL-AML-3	50.0	55.5	9.8	3.2	1.80	**1.38**	1.55	83.7	63.6	28.5
ALL-AML-4	34.0	37.8	7.9	3.3	1.70	**1.29**	1.45	57.7	37.6	22.9
Multi-CNS	19.0	20.9	4.9	2.2	1.60	**1.56**	1.58	31.8	14.4	24.7

bootstrap sampling is repeated P times, finally the averaged error is recorded. In our experiments, both P and C are set as 50. KNN, Naive Bayes, and SVM are chosen as the classifier. For all methods, we always select 30 genes for classification. The first experiment test the binary classification. The best classification accuracy of each algorithm are presented in Table 6. We observe that our algorithm and Logo perform well on all data sets. Our algorithm outperforms other competing algorithms in terms of classification accuracy on four data sets. Logo performs the best on one data set. We observe that SVM-RFE does not perform well on data with a very high dimensionality. This may be SVM can not effectively discover the best relevant features when the data involves in many irrelevant features. It should be pointed out that statistical test is not perform in this experiment since the number of testing sample is small.

We also consider feature selection for multi-class problem. The best classification accuracy of each algorithm are presented in Table 6. We note that our algorithm outperforms other competing algorithms on all multi-class microarray data sets. It is because of that our proposed method can effectively deal with multi-class data without any extension or approximation. Feature selection algorithms like Logo, I-RELIEF, and Simba are originally designed for a binary classification problem, using some approximate techniques, they can be used for multi-class settings.

6.4 Computational Complexity

In this subsection, we study the efficiency of the proposed feature selection algorithm by measuring the CPU time, and compare it with the other competing algorithms. The experiment is performed on a computer with Pentium 4 3.10 GHz and 4 GB RAM. The results are summarized in Table 7. The computational expense of our algorithm is high on most of data sets. The mRMR, FScore, and InforGain algorithms are to evaluate each feature, then the top features are selected. Therefore, the time complexities of mRMR, FScore, and InforGain are lower than the other competing algorithms. The time complexity of our algorithm is comparable with Logo, Lasso, SVM-RFE, and SFS. In real applications, we can speed up our algorithm by parallel implementation of step 7 in Algorithm 1.

7 Conclusion

In this paper, a newly algorithm of feature evaluation is developed to measure the quality of feature, and applied to as a feature selection criterion. A feature subset that gives rise to higher classification ability is considered to be more important. With this criterion, the feature selection task is transformed into an optimization problem. The optimization problem is efficiently solved by following the principle of the AdaBoost-based search method, rather than the exhaustive search. In addition, we also analyze the generalization error bounds of our feature selection algorithm. Various experiments have been conducted on four UCI and 12 microarray data sets to demonstrate the effectiveness of our algorithm, and verify the theoretical results established in this paper.

Appendix

In this section, we prove the theoretical results in the paper for our proposed Theorem in Sections 3 and 5. Firstly, we define a notation "Cond" which denotes $\{\ell(x_i) = \ell(z_{k1}) \neq \ell(z_{k2})\}$. The notation is used in the following proof.

Proof of Theorem 1. Let M be the set of samples for the jth attribute satisfying Definition 1. x^j is a $(\epsilon, \gamma)-$good feature, for any $x_i^j \in M$, we have

$$\begin{aligned} &P\big(F(x_i^j, z_{k1}^j, z_{k2}^j) > 0|\text{Cond}\big) \\ &= E\big(F(x_i^j, z_{k1}^j, z_{k2}^j) > 0|\text{Cond}\big) \\ &= \frac{1}{2}E\big(\text{sign}\big(F(x_i^j, z_{k1}^j, z_{k2}^j)\big)|\text{Cond}\big) + \frac{1}{2} \\ &\geq \frac{1}{2} + \frac{\gamma}{2}. \end{aligned} \tag{23}$$

Thus, we obtain

$$E\big(\text{sign}\big(F(x_i^j, z_{k1}^j, z_{k2}^j)\big)|\text{Cond}\big) \geq \gamma. \tag{24}$$

By Hoeffding bounds [31], we have

$$P\left(\frac{1}{n}\sum_{k=1}^{n} F(x_i^j, z_{k1}^j, z_{k2}^j) \le 0|\text{Cond}\right) \le e^{-n\gamma^2/2}, \quad (25)$$

which means there is at most $e^{-n\gamma^2/2}$ probability of error over the choice of n pairs of training samples for a given x_A^j, so the expectation of error is at most $e^{-n\gamma^2/2}$, mathematically,

$$E\left[P\left(\frac{1}{n}\sum_{k=1}^{n} F(x_i^j, z_{k1}^j, z_{k2}^j) \le 0|\text{Cond}\right)\right] \le e^{-n\gamma^2/2}. \quad (26)$$

Using the Markov inequality [72], there is at most a η probability such that the error is more than η:

$$P\left[P\left(\frac{1}{n}\sum_{k=1}^{n} F(x_i^j, z_{k1}^j, z_{k2}^j) \le 0|\text{Cond}\right) \ge \eta\right] \le \frac{e^{-n\gamma^2/2}}{\eta}. \quad (27)$$

The proof of the theorem can be obtained by setting $\delta = \frac{e^{-n\gamma^2/2}}{\eta}$ and adding the ϵ probability of samples for the jth attribute x_i^j not in M. □

Proof of Theorem 2. It equals to prove that

$$P(I_{-1}[\ell(x) = \ell(x_k^j)]H(x) \le 0) \le \epsilon + \delta, \quad (28)$$

where $H(x) = \frac{1}{n}\sum_{k=1}^{n} \text{sign}\big(v(x^j) - d(x, z_k^j)\big)$, the indicator $I_{-1}[x]$ outputs 1 when the Boolean variable x is true and -1 otherwise. We can prove it by following the proof of the Theorem 1. □

Proof of Theorem 3.

$$\begin{aligned}
&\frac{2}{n^2-n}\left\{i,k : \text{sign}\left(\sum_m \text{sign}(v^m - d(x_i^m, x_k^m))\right)\right.\\
&\left.\neq I_{-1}[\ell(x_i) = \ell(x_k)]\right\}\\
&\le \frac{2}{n^2-n}\sum_{m=1}^{T_{max}} \exp\Bigg(- I_{-1}[\ell(x_i) = \ell(x_k)] \quad (29)\\
&\qquad \cdot \text{sign}\left(\sum_m \text{sign}(v^m - d(x_i^m, x_k^m))\right)\Bigg)\\
&= \frac{2}{n^2-n}\prod_{m=1}^{T_{max}} (\exp(\alpha_m)e_m + \exp(-\alpha_m)(1-e_m)).
\end{aligned}$$

Based on Boosting theory, α^m can be rewritten with respect to e_m as:

$$\alpha^m = 1 - 2e_m.$$

Substituting a^m back into (29), the result of the training error in step $T_{max}+1$ is

$$\frac{2}{n^2-n}\prod_{m=1}^{T_{max}} \exp(1-2e_m)e_m + \exp(2e_m - 1)(1-e_m).$$

□

Proof of Theorem 4. Firstly, we introduce an auxiliary function $\mathcal{L}(Y, f(z)) = \frac{1-yf(z)}{2}$. For all $f \in \mathcal{F}$, we have

$$\begin{aligned}
Err_D[\mathcal{L}(Y, f(z))] &\le Err_S[\mathcal{L}(Y, f(z))]\\
&+ \sup_{g\in\mathcal{L}\circ\mathcal{F}} (Err_D[g(z)] - Err_S[g(z)]),
\end{aligned} \quad (30)$$

where $\mathcal{L}\circ\mathcal{F} = \{(z,y) \mapsto \mathcal{L}(y, f(z)) : f \in \mathcal{F}\}$. Note that the supremum term is a random variable that depends on the draw of S, the supremum term changes no more than $1/n$ when (Z,Y) changes, so McDiarmid's inequality [52] implies that with probability at least $1-\delta, \forall f \in \mathcal{F}$ satisfies

$$\begin{aligned}
&Err_D[\mathcal{L}(Y, f(z))] \le Err_S[\mathcal{L}(Y, f(z))]\\
&+ E[\sup_{g\in\mathcal{L}\circ\mathcal{F}} (Err_D[g(z)] - Err_S[g(z)])] + \sqrt{\frac{ln(1/\delta)}{n}}.
\end{aligned} \quad (31)$$

For the supremum term, we have

$$\begin{aligned}
&E\left[\sup_{g\in\mathcal{L}\circ\mathcal{F}} \Big(Err_D[g(z)] - Err_S[g(z)]\Big)\right]\\
&\le E\left[\sup_{g\in\mathcal{L}\circ\mathcal{F}} \frac{2}{n}\sigma_i g(y_i, z_i)\right]\\
&= E\left[\sup_{f\in\mathcal{F}} \frac{2}{n}\sigma_i(1 - y_i f(z_i))/2\right] \quad (32)\\
&= E\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1^n} \sigma_i f(z_i)\right]\\
&= R_n(\mathcal{F}),
\end{aligned}$$

where $\sigma_1, \sigma_2, \ldots, \sigma_n$ are independent random variables uniformly chosen from $\{\pm 1\}$. So we have

$$Err_D(f) \le Err_S(f) + R_n(\mathcal{F}) + \sqrt{\frac{ln(1/\delta)}{n}}. \quad (33)$$

□

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part HongKong, Macao and Taiwan Science & Technology Cooperation Program of China (L2015TGA9004), and in part by the Program for New Century Excellent Talents of Educational Ministry of China (NCET-13-0639), and in part by the National Natural Science Foundations of China (61572087, 61173030, and 61003120), and in part by the Science and Technology Development Fund (FDCT) of Macau 100-2012-A3, 026-2013-A, 019/2015/A, 164/2014/SB/4052, and Macau-China join Project 008-2014-AMJ.

REFERENCES

[1] A. A. Alizadeh, et al., "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 286, no. 5439, pp. 531–537, 1999.

[2] A. Arauzo-Azofra, J. Aznarte, and J. Benítez, "Empirical study of feature selection methods based on individual feature evaluation for classification problems," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8170–8177, 2011.

[3] A. J. Stephenson, A. Smith, M. W. Kattan, J. Satagopan, V. E. Reuter, P. T. Scardino, and W. L. Gerald, "Integration of gene expression profiling and clinical variables to predict prostate carcinoma recurrence after radical prostatectomy," *Cancer*, vol. 104, no. 2, pp. 290–298, 2005.

[4] A. K. 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.

[5] A. Tayal, T. Coleman, and Y. Li, "Primal explicit max margin feature selection for nonlinear support vector machines," *Pattern Recognit.*, vol. 47, no. 6, pp. 2153–2164, 2014.

[6] H. Almuallim and T. Dietterich, "Learning boolean concepts in the presence of many irrelevant features," *Artif. Intell.*, vol. 69, pp. 279–305, 1994.

[7] C. Ambroise and G. J McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 10, pp. 6562–6566, 2002.

[8] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[9] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[10] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 483–519, 2013.

[11] C. Lazar, J. Taminau, et al., "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 9, no. 4, pp. 1106–1119, Jul. 2012.

[12] M. R. Chernick, *Bootstrap Methods: A Practitioner's Guide*, Wiley Series in Probability and Statistics. Hoboken, NJ, USA: Wiley, 1999.

[13] T. Cover, "The best two independent measurements are not the two best," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, no. 1, pp. 116–117, Jan. 1974.

[14] D. Belland and H. Wang, "A formalism for relevance and its application in feature subset selection," *Mach. Learn.*, vol. 41, no. 2, pp. 175–195, 2000.

[15] D. Zhang, S. Chen, and Z.-H. Zhou, "Constraint score: A new filter method for feature selection with pairwise constraints," *Pattern Recognit.*, vol. 41, no. 5, pp. 1440–1451, 2008.

[16] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 74–81.

[17] S. K. Das, "Feature selection with a linear dependence measure," *IEEE Trans. Comput.*, vol. C-20, no. 9, pp. 1106–1109, Sep. 1971.

[18] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, 1998.

[19] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 5, pp. 2197–2202, 2003.

[20] A. Ferreira and M. Figueiredo, "Efficient feature selection filters for high-dimensional data," *Pattern Recognit. Lett.*, vol. 33, no. 13, pp. 1794–1804, 2012.

[21] A. Frank and A. Asuncion. (2010). {UCI} machine learning repository [Online]. Available: http://archive.ics.uci.edu/ml

[22] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, pp. 119–139, 1997.

[23] G. Brown, A. Pocock, M. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 27–66, 2012.

[24] G. John, K. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proc. 11th Int. Mach. Learn.*, 1994, pp. 121–129.

[25] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[26] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8144–8150, 2011.

[27] H.-L. Wei and S.A. Billings, "Feature subset selection and ranking for data dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 162–166, Jan. 2007.

[28] 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.

[29] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statistical Soc.: Series B (Statistical Methodol.)*, vol. 67, no. 2, pp. 301–320, 2005.

[30] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 359–366.

[31] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statistical Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.

[32] 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.

[33] J . Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5432–5435, 2009.

[34] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[35] 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, 2001.

[36] J. Tang and H. Liu, "An unsupervised feature selection framework for social media data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2914–2927, Dec. 1, 2014.

[37] J. Wang, H. Bensmail, and X. Gao, "Feature selection and multikernel learning for sparse representation on a manifold," *Neural Netw.*, vol. 51, pp. 9–16, 2014.

[38] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," in *Proc. Adv. Neural Inf. Process. Syst. 13*, 2000, pp. 668–674.

[39] K. Kira and L. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proc. 10th Nat. Conf. Artif. Intell.*, 1992, pp. 129–134.

[40] V. Koltchinskii and D. Panchenko, "Empirical margin distributions and bounding the generalization error of combined classifiers," *Ann. Statist.*, vol. 30, no. 1, pp. 1–50, 2000.

[41] L. J. van't Veer, et al., "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, no. 415, pp. 530–536, 2002.

[42] L. Kuncheva and C. L. Jain, "Nearest neighbor classifier: Simultaneous editing and feature selection," *Pattern Recognit. Lett.*, vol. 20, no. 11, pp. 1149–1156, 1999.

[43] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, "Feature selection via dependence maximization," *J. Mach. Learn. Res.*, vol. 13, pp. 1393–1434, May 2012.

[44] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. Int. Conf. Mach. Learn.*, 2003, vol. 3, pp. 856–863.

[45] M.-C. Lee, "Using support vector machine with a hybrid feature selection method to the stock trend prediction," *Expert Syst. Appl.*, vol. 36, no. 8, pp. 10896–10904, 2009.

[46] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artif. Intell.*, vol. 151, nos. 1/2, pp. 155–176, 2003.

[47] M.-F. Balcan, A. Blum, and N. Srebro, "A theory of learning with similarity functions," *Mach. Learn.*, vol. 72, pp. 89–112, 2008.

[48] M. Marchand and M. Shah, "PAC-Bayes learning of conjunctions and classification of gene-expression data," presented at the Adv. Neural Inf. Process. Syst. 17, Dec. 13–18, 2004, Vancouver, BC, Canada, 2004.

[49] M. Shah, M. Marchand, and J. Corbeil, "Feature selection with conjunctions of decision stumps and learning from microarray data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 174–186, Jan. 2012.

[50] M. Yamada, W. Jitkrittum, L. Sigal, E. Xing, and M. Sugiyama, "High-dimensional feature selection by feature-wise kernelized lasso," *Neural Comput.*, vol. 26, no. 1, pp. 185–207, 2014.

[51] M. A. Shipp, et al., "Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Med.*, vol. 8, no. 1, pp. 68–74, 2002.

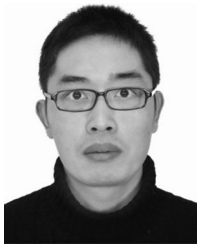
[52] C. McDiarmid, "On the method of bounded differences," *Surveys Combinatorics*, vol. 141, no. 1, pp. 148–188, 1989.

[53] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 78–86.

[54] M. H. Nguyen and F. de la Torre, "Optimal feature selection for support vector machines," *Pattern Recognit.*, vol. 43, no. 3, pp. 584–591, 2010.

[55] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, pp. 131–159, Mar. 2002.

[56] P. Jawanpuria, M. Varma, and S. Nath, "On p-norm path following in multiple kernel learning for non-linear feature selection," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 118–126.
[57] P. Maji and S. K. Pal, "Feature selection using f-information measures in fuzzy approximation spaces," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 6, pp. 854–867, Jun. 2010.
[58] P. Pudil and J. Novovicova, "Novel methods for subset selection with respect to problem knowledge," *IEEE Intell. Syst. Appl.*, vol. 13, no. 2, pp. 66–74, Mar./Apr. 1998.
[59] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, pp. 1119–1125, 1994.
[60] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognit.*, vol. 48, no. 2, pp. 438–446, 2015.
[61] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 1–14, Jan. 2013.
[62] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2001.
[63] R. Gilad-Bachrach, and A. Navot, and N. Tishby, "Margin based feature selection - Theory and algorithms," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 43–50.
[64] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.
[65] R. Shyamsundar, Y. Kim, et al., "A DNA microarray survey of gene expression in normal human tissues," *Genome Biol.*, vol. 6, no. 3, p. R22, 2005.
[66] S. A. Armstrong, et al., "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia," *Nature Genetics*, vol. 30, pp. 41–47, 2002.
[67] S. Baccianella, A. Esuli, and F. Sebastiani, "Feature selection for ordinal text classification," *Neural Comput.*, vol. 26, no. 3, pp. 557–591, 2014.
[68] S. Kannan and N. RamarajN, "A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm," *Knowl.-Based Syst.*, vol. 23, no. 6, pp. 580–585, 2010.
[69] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, and L. M. Sturla, et al., "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, no. 6870, pp. 436–442, 2002.
[70] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May/Jun. 1991.
[71] S. Xiang, T. Yang, and J. Ye, "Simultaneous feature and feature group selection through hard thresholding," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 532–541.
[72] E. Stein and R. Shakarchi, *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[73] Y. Sun, "Iterative RELIEF for feature weighting: Algorithms, theories, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1035–1051, Jun. 2007.
[74] T. Li, C. Zhang, M. Ogihara, "A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression," *Bioinformatics*, vol. 20, no. 15, pp. 2429–2437, 2004.
[75] 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, 1999.
[76] U. Alon, N. Barka, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 12, pp. 6745–6750, 1999.
[77] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.
[78] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "BoostMap: An embedding method for efficient nearest neighbor retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 89–104, Jan. 2008.
[79] V. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley-Interscience, 1998.
[80] L. Wang, "Feature selection with kernel class separability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1534–1546, Sep. 2008.
[81] A. R. Webb, *Statistical Pattern Recognition*, 2 ed. Hoboken, NJ, USA: Wiley, 2002.
[82] H.-L. Wei and S. Billings, "Feature subset selection and ranking for data dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 162 –166, Jan. 2007.
[83] L. Wolf and A. Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *J. Mach. Learn. Res.*, vol. 6, pp. 1855–1887, Dec. 2005.
[84] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1083–1095, Jun. 2014.
[85] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1178–1192, May 2013.
[86] X. Zhu, H. Suk, and D. Shen, "Matrix-similarity based loss function and feature selection for Alzheimer's disease diagnosis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3089–3096.
[87] Y. Sun, S. Todorovic, and S. Goodison, "Local-learning-based feature selection for high-dimensional data analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1610–1626, Sep. 2010.
[88] Y. Yang, Z. Ma, A. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.
[89] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intell. Syst.*, vol. 13, no. 2, pp. 44–49, Mar. 1998.
[90] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.
[91] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 70–76, Feb. 2007.



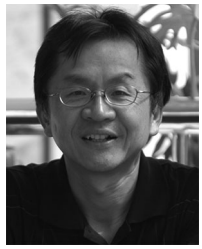
Taiping Zhang (M'10) received the BS and MS degrees in computational mathematics, and the PhD degree in computer science from Chongqing University, Chongqing, China, in 1999, 2001, and 2010, respectively. He is currently an associate professor in the Department of Computer Science, Chongqing University. His research interests include pattern recognition, image processing, machine learning, and computational mathematics. He has published extensively in the *IEEE Transactions on Image Processing*, the *IEEE Transactions on Systems, Man, and Cybernetics, Part B (TSMC-B)*, the *IEEE Transactions on Knowledge and Data Engineering*, *Pattern Recognition*, *Neurocomputing*, etc. He is a member of the IEEE.

Pengfei Ren received the BS degrees in computer science from Chongqing University, Chongqing, China, in 2013. He is currently working toward the MS degree at the Chongqing University. His research focus is concerned with the topics of pattern recognition, machine learning, and data mining.

Yao Ge received the PhD degree in the School of Computer Science, Chongqing University, in 2013. He is currently a lecturer in the Department of Computer Science, Chongqing University. His research interests include pattern recognition, image processing, machine learning, and data mining.

Yali Zheng received the PhD degree in the School of Computer Science, Chongqing University, in 2012. She was visiting Carnegie Mellon University and Univeristy of Pittsburgh (October 2008-September 2011). She is currently a lecturer in Pattern Recognition and Machine Intelligence Lab (PRMIL), which is affiliated with School of Automation Engineering, UESTC, Chengdu, China. Her research focus is concerned with the topics of dynamic 3D reconstruction by geometry and optimization methodology and image modeling. She is a member of the IEEE.

Yuan Yan Tang (F'04) received the BS degree in electrical and computer engineering from Chongqing University, Chongqing, China, the MEng degree in electrical engineering from the Graduate School of Post and Telecommunications, Beijing, China, and the PhD degree in computer science from Concordia University, Montreal, Canada. He is currently a professor in the Department of Computer Science, Chongqing University and a chair professor in the Department of Computer Science, Hong Kong Baptist University and an adjunct professor in computer science at Concordia University. He is an honorary lecturer at the University of Hong Kong, and an advisory professor at many institutes in China. His current interests include wavelet theory and applications, pattern recognition, image processing, document processing, artificial intelligence, parallel processing, Chinese computing, and VLSI architecture. He has published more than 250 technical papers and is the author/coauthor of 21 books/book chapters on subjects ranging from electrical engineering to computer science. He has served as the general chair, program chair, and committee member for many international conferences. He will be the general chair in the 19th International Conference on Pattern Recognition (ICPR'06). He is the founder and editor-in-chief of the *International Journal on Wavelets, Multiresolution, and Information Processing (IJWMIP)* and associate editor of several international journals related to *Pattern Recognition and Artificial Intelligence*. He is a fellow of the IAPR and IEEE.



C.L. Philip Chen received the MS degree from the University of Michigan, Ann Arbor, in 1985, and the PhD degree from Purdue University, West Lafayette, Indiana, in 1988. He has been the chair in the Department of Electrical and Computer Engineering, and an associate dean for Research and Graduate Studies of the College of Engineering, University of Texas at San Antonio, San Antonio. He is currently a chair professor and dean in the Faculty of Science and Technology, University of Macau. He was a visiting research scientist at the Materials Directorate, U.S. Air Force Wright Laboratory, Ohio. He was also a senior research fellow sponsored by the U.S. National Research Council and a Research Faculty Fellow at the National Aeronautics and Space Administration (NASA) Glenn Research Center for several years. His current research interests include theoretic development in computational intelligence, information security, intelligent systems, robotics and manufacturing automation, networking, diagnosis and prognosis, and life prediction and life-extending control. He has been involved in IEEE professional service for 20 years. He is the president-elect, vice president on Conferences and Meetings of the IEEE Systems, Man and Cybernetics Society (SMCS), and has been the vice president for Tech Activities on Systems Science and Engineering, the treasurer, a member of SMCS Conference and Management Committee; a founding co-chair of three IEEE SMCS Technical Committees; an associate editor of the *IEEE Transactions on SMC-C* and the *IEEE Systems Journal*; the general chair of the IEEE International Conference on SMC 2009; the general co-chair of the IEEE 2007 Secure System Integration and Reliability (SSIRI) conference, the Program co-chair of the International Conference on Machine Learning and Cybernetics (ICMLC) 2008; the program chair of IEEE SMC-SoSE 2006; and the Conference co-chairs of the International Conference on Artificial Neural Networks in Engineering (ANNIE) 1995 and 1996. He received an Outstanding Contribution Award from IEEE SMCS in 2008 and 2010. He is a member of the Tau Beta Pi, Eta Kappa Nu honorary societies, and an elected IEEE fellow and AAAS fellow. He is the founding faculty advisor of an IEEE Computer Society Student Chapter and the faculty advisor of the Tau Beta Pi engineering honor society in his university.