2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

A Binary Feature Selection Framework in Kernel Spaces

Chengzhang Zhu*, Xinwang Liu*, Sihang Zhou*, Qiang Liu* and Jianping Yin†
*College of Computer
National University of Defense Technology, Changsha, Hunan Province, China 410073
Email: kevin.zhu.china@gmail.com, 1022xinwang.liu@gmail.com, 306114653@qq.com, qiangliu_nudt@163.com
†State Key Laboratory of High Performance Computing
National University of Defense Technology, Changsha, Hunan Province, China 410073
Email: JPYin@nudt.edu.cn

***Abstract*—In this paper, we propose a binary feature selection framework in kernel spaces, where each feature is projected into kernel spaces and a binary classification task is constructed in this space. Subsequently, the features are selected according to the normal vector of the learned classifier, which reflects the importance of each feature. To achieve the effect of feature selection, an ℓ_1-norm regularization is imposed on the normal vector to enforce its sparsity. Also, our framework can be naturally extended to the semi-supervised feature selection scenario via the well-known manifold regularization technique. Furthermore, the issue of eliminating the potential redundancy among the selected features is well discussed. Finally, we provide some theoretical results which guarantee the feasibility of the proposed framework. Comprehensive experiments have been conducted on six benchmark data sets and the results demonstrate the performance of our framework.**

I. Introduction

Data with high dimensional features is very common in many practical applications such as pattern recognition [1], bioinformatics [2], anomaly detection [3], to name just a few. Some of these features may be irrelevant and redundant for a learning task. More importantly, training a model with these high dimensional data directly would increase the risk of over-fitting, incur more computational cost, deterioration prediction accuracy and reduce the result comprehensibility. To address these issues, many feature selection algorithms have been proposed to select a subset of features from these high dimensional data during the past several years [4]–[8]. In some areas, researchers make better performance due to using of the feature selection method [9]–[11]. Existing feature selection algorithms can be roughly classified into two categories, i.e., filter and wrapper. The main idea of filter method is to score features according to some statistical criterion and filter out features by the score [12]–[16]. Obviously, filter method has the advantage of fast selection but lacking in robustness, and its capacity is insufficient for redundancy and related features processing. More importantly, this method needs to make a good trade-off for how many features to extract. Wrapper method chooses features embedded into learning algorithms [17], [18]. It can achieve high accuracy but always lack of speed [19]. Some approaches try to combine the filter and wrapper to achieve better results [20], [21].

Recently, some excellent feature selection methods have been proposed in both supervised cases [22]–[29] and semi-supervised cases [29]–[33]. In the supervised case, the work in [25] selects features based on the criterion of minimizing redundancy between feature and target while maximizing relevance among features. Differently, an $\ell_{2,1}$-norm joint in both loss function and regularization is adopted in [26] to induce an efficient and robust method. The approach in [27] attempts to select features by decomposing complex nonlinear problem into local linear one that uses local learning method to maximize the local margin. Both global and local structures are maintained in [29] to conduct feature selection. In semi-supervised case, the work in [30] discusses how to use multi-objective optimization to select features. the spectral graph theory is firstly introduced in [31] to solve semi-supervised feature tasks. The approach in [32] shows a novel method based on manifold learning. Also, a semi-supervised feature selection algorithm based on manifold regularization is proposed in [33].

In this paper, we propose a novel feature selection method termed as binary feature selection (BFS), which presents a new perspective of feature selection. It offers a new framework for feature selection, in which all existing binary classification methods can be applied to select features. This implies that, for different types of data, one can achieve a better result by adopting a more appropriate classifier. In order to maximize the difference between different classes, BFS first projects features to a kernel space, in which each base kernel is induced by one dimension of features in the original space. After that, any binary classification methods can be used to find a "good" kernel combination weight. In addition, an ℓ_1-norm regularization is incorporated in BFS to enforce the sparsity of the learned kernel combination weights. Finally, we theoretically show that "good" features can be selected based on the "good" base kernel combination weights. To deal with the presence of a large number of unlabeled samples in real feature selection problems, we further propose semi-supervised binary feature selection (semi-BFS) method by applying the well-known manifold regularization technique [34]. Comprehensive results on different real world data sets demonstrate the superior performance of the proposed framework.

The rest of paper is organized as follows. We present the BSF and semi-BSF framework in section II. The theoretical analysis is provided in section III. Section IV reports the experimental results and section V concludes the paper.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

II. Binary Feature Selection

Our approach considers the impacts of individual features and combining features on the prediction/classification results at the same time. For the classification problem, it initially projects features into a special kernel space, called K-space, which will be introduced in the following part, to reflect the discrimination of individual feature to different classes. And then it seeks a "good" combination of features through a transformed binary classification problem. In this section, we first introduce how to construct K-space from the original feature space. After that we discuss how to select features in both supervised and semi-supervised situations.

A. K-space Construction

First we assume there are n samples $(\mathbf{x}_i, y_i)$ from a same distribution $\mathcal{P}$, where $\mathbf{x}_i$ has m-dimensional features and can be presented as $(f_{i1}, f_{i2}, \cdots, f_{im})$ and y_i is the corresponding label. We use each dimension to reconstruct $n \times m$ new samples, i.e. $((\mathbf{f}_1, \mathbf{y}), (\mathbf{f}_2, \mathbf{y}), \cdots, (\mathbf{f}_m, \mathbf{y}))$, in which $\mathbf{f}_i$ means the i-th dimensional feature and $\mathbf{y}$ means the corresponding n labels, from the n samples. Then the impact on classification of each dimensional feature can be measured by using a kernel function $k(f, f')$ in each $\mathbf{f}_i$. In this way, we can get m kernel matrix $(\mathbf{K}(\mathbf{f}_1, \mathbf{f}_1^{'}), \mathbf{K}(\mathbf{f}_2, \mathbf{f}_2^{'}), \cdots, \mathbf{K}(\mathbf{f}_m, \mathbf{f}_m^{'}))$. To learn a "good" combination of features, we need to learn a "good" combination of these kernels since it reflects every features. Similar as [35], we define a new instance space, which called K-space, to achieve kernel selection through a binary classification. In order to keep the nature of features, we just use the linear kernel $k(x, x^{'}) = x \times x^{'}$ to project feature space to K-space. However, other kernels can be used in this approach either. We construct K-space as $\{(\mathbf{z}_{\mathbf{x},\mathbf{x}'}, t_{y,y'})|(x, y), (x^{'}, y^{'}) \sim \mathcal{P} \times \mathcal{P}\} \subset \mathbb{R}^m \times \{\pm 1\}$ where

$$\begin{aligned} \mathbf{z}_{\mathbf{x},\mathbf{x}'} &= (\mathbf{K}(f_{\mathbf{x}1}, f_{\mathbf{x}1}^{'}), \mathbf{K}(f_{\mathbf{x}2}, f_{\mathbf{x}2}^{'}), \cdots, \mathbf{K}(f_{\mathbf{x}m}, f_{\mathbf{x}m}^{'})) \\ t_{yy'} &= 2 \cdot \mathbf{1}\{y = y^{'}\} - 1. \end{aligned} \tag{1}$$

Our approach, which is based on the K-space, will be discussed in the following part.

B. Supervised Binary Feature Selection

For m-dimensional features $(f_1, f_2, \cdots, f_m)$, the feature selection problem can be seen as finding a $m \times 1$ vector $\boldsymbol{\mu}$, which represents the importance of each feature dimension, and select k most significant features based on $\boldsymbol{\mu}$. In particular, for the classification problem, feature selection needs to find the $\boldsymbol{\mu}$ that can select the features combination, which have the most discrimination for different class. Therefore, if we project features from the original feature space into the K-space, in which $\mathbf{z}_{\mathbf{x},\mathbf{x}'}$ represent feature combination and $t_{yy'}$ represent discrimination of different class, we can find this $\boldsymbol{\mu}$ by ensuring $\boldsymbol{\mu} \cdot \mathbf{z}_{\mathbf{x},\mathbf{x}'}$ can derive most correct $t_{yy'}$. In this end, feature selection problem has transformed to a binary classification problem and $\boldsymbol{\mu}$ is equal to the parameters of linear classifier. What's more [35] had improved the feasibility using binary classification method finds "good" kernel combination. So one can introduce any binary classification method to select feature. There are at least two advantages: 1) it is robust to different data; 2) it is easily expanded to semi-supervised feature selection. For the first one, our approach will be robust because there is no assumption in data structure. Meanwhile, in the worst case, one can use different binary classification method to overcome the impaction of different data structure. For the second one, because our approach has transformed feature selection to binary classification and there are many existing methods that can expand supervised classification problem to semi-supervised classification straightforwardly, such as [34], so we can develop it to semi-supervised feature selection conveniently.

In this paper, we just consider least squares classifier. However, various classifiers can be used as well, such as SVM [36], ELM [37] and so on. Moreover, we take into account the nature of feature selection, which is to select lesser features with better performance. Therefore, we add a ℓ_1-norm regularization to the objective function in order to ensure the sparsity of $\boldsymbol{\mu}$ [38]. Finally the $\boldsymbol{\mu}$ needs to be increased the limit of non-negative since we only need the positive impact of features.

To train this classifier, we put all $\mathbf{z}_{\mathbf{x},\mathbf{x}'}$ and the corresponding $t_{y,y'}$ in K-space as training samples. If we have n samples in the original feature space, then we can get N ($\mathbf{z}_{\mathbf{x},\mathbf{x}'}$,$t_{y,y'}$) pairs in K-space as training samples ($\mathbf{Z}$,$\mathbf{T}$). Here, N equal to $\frac{n(n-1)}{2}$ is the number of projected samples in K-space. Our objective function can be written as follow:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \quad & \|\mathbf{Z} \cdot \boldsymbol{\mu} - \mathbf{T}\|_2^2 + \alpha_1 \|\boldsymbol{\mu}\|_1 \\ s.t. \quad & \boldsymbol{\mu} \geq 0 \end{aligned} \tag{2}$$

where α_1 is a constraint parameter that controls the sparse degree of $\boldsymbol{\mu}$. We solve this least squares problem using an interior-point method for large-scale ℓ_1-regularized least squares proposed in [39]. The largest k values in $\boldsymbol{\mu}$ represent the most k important kernel matrix. Since each kernel matrix is projected from a corresponding feature, the important kernel matrixes correspond to the important features. Thus, we can select features according to the k largest values in $\boldsymbol{\mu}$. We summarize our approach in Algorithm 1.

Algorithm 1 Our proposed BFS

1: **Input**: training data $\mathbf{X} \in \mathbb{R}^{n \times m}$ and $\mathbf{Y} \in \mathbb{R}^{n \times 1}$, the number of selected features k.
2: **Output**: the selected features $\mathbf{F}_{set}$.
3: Construct K-space using Eq. 1.
4: Calculate $\boldsymbol{\mu}$ in K-space for Eq. 2
5: Select features from $\mathbf{X}$ according to the largest k values in $\boldsymbol{\mu}$

C. Semi-supervised Binary Feature Selection

Consider there are n samples in the original feature space, in which n_l are labeled samples while $n - n_l$ are unlabeled samples. After projected to K-space, the new samples have label only when both corresponding samples have label. So we can get N new samples with l labeled and u unlabeled. Here, l equal to $\frac{n_l^2(n_l^2-1)}{2}$ and u equal to $N - l$. We denote the set of new samples in K-space as $\{(\mathbf{z}_i, t_i)_{i=1}^N\}$.

We use the manifold regularization term to extend BF-S method to semi-BFS method, which can adapt to semi-supervised feature selection. This is a very natural but extremely significant extension, because in most reality cases people

cannot supervised label all samples. A basic assumption of manifold regularization is that points, which are close in the manifold, are more likely to have the same nature. In other word, if two points $x_1, x_2 \in X$ are close in the manifold, then the conditional distribution $\mathcal{P}(y|x_1)$ and $\mathcal{P}(y|x_2)$ are similar. So manifold regularization term is essentially a smoothness penalty to the probability distribution. It can mine the information of unlabeled data using labeled data based on their position on the manifold. If we note l labeled samples $\{(\mathbf{z}_i, t_i)_{i=1}^{l}\}$ and u unlabeled samples $\{\mathbf{z}_j\}_{j=l+1}^{j=l+u}$, the manifold regularization term can be written as $\sum_{i,j=1}^{l+u}(f(\mathbf{z}_i) - f(\mathbf{z}_j))^2 W_{ij}$, where W_{ij} represents the distance between $\mathbf{z}_i$ and $\mathbf{z}_j$. However, manifold regularization assume a low-dimensional manifold is embedded into a high-dimensional space. Therefore, the Euclidean distance in the high-dimensional space can approximate distance in low-dimensional manifold only when points are very close in a local space. Thus we only consider the h-nearest neighbor of a point in the manifold regularization term. We define

$$S_{ij} = \begin{cases} \exp(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{\sigma^2}) & \mathbf{z}_i \in \mathcal{N}_h(\mathbf{z}_j) \text{ or } \mathbf{z}_j \in \mathcal{N}_h(\mathbf{z}_i) \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathcal{N}_h(\mathbf{z})$ is the set of h-nearest neighbor of $\mathbf{z}$. Then the manifold regularization is minimization the following:

$$\min_f \sum_{i,j=1}^{l+u}(f(\mathbf{z}_i) - f(\mathbf{z}_j))^2 S_{ij} = \min_{\mathbf{f}} \mathbf{f}^\top \mathbf{L}\mathbf{f}, \tag{4}$$

where $\mathbf{f} = [f(\mathbf{z}_1), f(\mathbf{z}_1), \cdots, f(\mathbf{z}_{l+u})]$, and $\mathbf{L}$ given by $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{S})\mathbf{D}^{-\frac{1}{2}}$ is the normalized graph Laplacian matrix. Here $\mathbf{D}$ is a diagonal matrix with

$$D_{ii} = \sum_{j=1}^{N} S_{ij}. \tag{5}$$

In K-space, $f(\mathbf{z})$ is equal to $\mathbf{z} \cdot \boldsymbol{\mu}$. The objective function of semi-BFS can be defined as:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \quad & \|\mathbf{Z}_l \cdot \boldsymbol{\mu} - \mathbf{T}_l\|_2^2 + \alpha_1 \|\boldsymbol{\mu}\|_1 + \alpha_2 \boldsymbol{\mu}^\top \mathbf{Z}^\top \mathbf{L}\mathbf{Z}\boldsymbol{\mu} \\ s.t. \quad & \boldsymbol{\mu} \geq 0 \end{aligned} \tag{6}$$

where $\mathbf{Z}_l$ is the $l \times m$ matrix composed of labeled $\mathbf{z}$, $\mathbf{T}$ is the $l \times 1$ label vector correspond to $\mathbf{Z}$, and α_2 is a parameter that controls the importance of the manifold regularization term. Since μ is limited greater than 0, the ℓ_1-norm term $\|\boldsymbol{\mu}\|_1$ is equal to $\mathbf{1}^\top \boldsymbol{\mu}$, where $\mathbf{1}$ is a $m \times 1$ vector with all elements is 1 i.e. $(1, 1, \cdots, 1)_m^\top$. This objective function can be rewrite as a typically quadratic programming problem. Obviously, it can be efficiently solved by existing approaches. The semi-BFS approach is summarized as Algorithm 2.

D. Discussion

In this section, we will discuss several of the problems mentioned above and give a supplementary description of our approach.

Algorithm 2 Our proposed semi-BFS

1: **Input**: training data $\mathbf{X} \in \mathbb{R}^{n \times m}$ and $\mathbf{Y} \in \mathbb{R}^{n_l \times 1}$ with n_l labeled samples and $n - n_l$ unlabeld samples, the number of selected features k.
2: **Output**: the selected features $\mathbf{F}_{set}$.
3: Construct K-space as Eq. 1 where $t_{yy'}$ is meaningful only when $\mathbf{x}$ and $\mathbf{x}^{'}$ are labeled .
4: Calculate Laplacian matrix $\mathbf{L}$.
5: Calculate $\boldsymbol{\mu}$ in K-space for Eq. 6
6: Select features from $\mathbf{X}$ according to the largest k values in $\boldsymbol{\mu}$

1) Kernel choice: As mentioned above, we only use linear kernel to project feature space to K-space in this paper. The reason is that linear kernel can keep the absolute position information between the original features, but other kernels like Gaussian kernel only retain the relative position information. Since we want to use the new samples, which are projected from the original space, to train a classifier in K-space, samples only contain relative position information that will even reduce the information in the original space. For example, consider we have three samples $\{x_1 = (1, 2, 3); y_1 = 1\}$, $\{x_2 = (2, 3, 4); y_2 = 1\}$, $\{x_3 = (3, 4, 5); y_3 = 2\}$, after projected to K-space using Gaussian kernel (with $\sigma = 1$) we will get new samples as $\{\mathbf{z}_{x_1,x_2} = (\frac{1}{e}, \frac{1}{e}, \frac{1}{e}); t_{y_1,y_2} = 1\}$, $\{\mathbf{z}_{x_2,x_3} = (\frac{1}{e}, \frac{1}{e}, \frac{1}{e}); t_{y_2,y_3} = -1\}$ etc. As can be seen from the above, we got two new samples that have the same position in the K-space but with different labels, which were due to the only containing of original feature's relative position information. Quite clearly, such samples would bring great classification error, which may produce a "bad" classifier. So kernels that only contain relative information cannot be used in our approach. However, there are more kernels not only contain it, e.g. Polynomial kernel. The best way choice kernel to project original features to K-space is introducing kernel learning method.

2) Feature Redundancy: Reduce redundancy in the selected feature can effectively improve the classification accuracy. Thus, some existing methods, such as [25], are carried out based on this. Since our approach projects feature into kernel space and each dimensional of feature corresponds to a kernel matrix, so reducing the redundancy of selected features is equivalent to reducing the kernel matrix correlation. We can reduce kernel matrix correlation by minimizing alignment between kernels. Following [40] alignment between kernels $\mathbf{K}, \mathbf{K}^{'} \in \mathbb{R}^{m \times m}$ can be defined as:

$$\hat{\rho}(\mathbf{K}, \mathbf{K}^{'}) = \frac{\langle \mathbf{K}_c, \mathbf{K}_c^{'} \rangle_F}{\|\mathbf{K}_c\|_F \|\mathbf{K}_c^{'}\|_F} \tag{7}$$

where $\mathbf{K}_c$ is the centered kernel matrix defined as:

$$\mathbf{K}_c = [\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}]\mathbf{K}[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m}] \tag{8}$$

where $\mathbf{1} \in \mathbb{R}^{m \times 1}$ denote the vector with all entries equal to one, and $\mathbf{I}$ is the identity matrix. The alignment matrix in K-

space can be rewritten as:

$$\boldsymbol{\rho} = \begin{bmatrix} \hat{\rho}(\mathbf{K}_1, \mathbf{K}_1) & \cdots & \hat{\rho}(\mathbf{K}_1, \mathbf{K}_m) \\ \vdots & \ddots & \vdots \\ \hat{\rho}(\mathbf{K}_m, \mathbf{K}_1) & \cdots & \hat{\rho}\mathbf{K}_m, \mathbf{K}_m \end{bmatrix} \tag{9}$$

where $\mathbf{K}_i = \mathbf{K}(f_i, f_i)$ is the kernel matrix correspond to the i-th dimensional feature. To minimize the alignment, our objective function can be rewrite as:

$$\begin{aligned} \min_{\boldsymbol{\mu}} \quad & \|\mathbf{Z}_l \cdot \boldsymbol{\mu} - \mathbf{T}_l\|_2^2 + \alpha_1 \|\boldsymbol{\mu}\|_1 \\ & + \alpha_2 \boldsymbol{\mu}^T \mathbf{Z}^\top \mathbf{L} \mathbf{Z} \boldsymbol{\mu} + \alpha_3 \boldsymbol{\mu}^T \boldsymbol{\rho} \boldsymbol{\mu} \\ s.t. \quad & \boldsymbol{\mu} \geq 0 \end{aligned} \tag{10}$$

where α_3 controls penalties for kernel matrix relevance and α_2 will be set as 0 in supervised feature selection. In the experiment, we found that if using a linear kernel to project feature, the selected features would have low redundancy. But using other kernels does not have this nature. We will show these results in IV-D4. So one can ignore the redundancy regularization when use linear kernel in our approach. And all of results in section IV not consider this regularization.

III. Theoretical Results

In this section, we present our theoretical results to analyze the generalization error of the proposed method. We show that a "good" classifier in the K-space will induce a "good" subset of features.

Theorem 1: Let $\mathcal{P}$ be a distribution on $\mathcal{X} \times \mathcal{Y}$, $\mathbf{z}_{\mathbf{x},\mathbf{x}'}$ and $t_{y,y'}$ be as in Eq.(1), $\boldsymbol{\mu}$ is the parameters of the linear classifier in K-space, $\boldsymbol{\mu}_{set}$ is the vector which only retains the k largest values in $\boldsymbol{\mu}$ while others are set to be 0, and R is a constant $s.t.$ $\mathbf{Z} \cdot \boldsymbol{\mu}_{set} \leq R^2 \ \forall x \in \mathcal{X}$. Let $L_{\boldsymbol{\mu}}$ be the expected K-space loss of the K-classifer with the parameters $\boldsymbol{\mu}$. Then, with probability $1 - \delta$, a classifier $\hat{f}$ with generalization error

$$P_{(x,y)}(y\hat{f} \leq 0) \leq L_{\boldsymbol{\mu}_{set}} + \mathcal{O}(\sqrt{\frac{R^4 \ln(1/\delta)}{\gamma^2 n}}),$$

where γ is the margin of the $\hat{f}$, can be learned efficiently from a training samples of n instances drawn i.i.d. from $\mathcal{P}$.

The $L_{\boldsymbol{\mu}}$ in the above theorem will be different based on different classifiers in the K-space. If SVM is used as the classifier, $L_{\boldsymbol{\mu}}$ can be written as:

$$L_{\boldsymbol{\mu}} = E_{((\mathbf{x},y),(\mathbf{x}',y')) \in \mathcal{P} \times \mathcal{P}}([1 - \frac{t_{y,y'} \mathbf{z}_{\mathbf{x},\mathbf{x}'} \cdot \boldsymbol{\mu}}{\gamma}]_+), \tag{11}$$

where $[1-s]_+ = \max\{0, 1-s\}$ is the hinge loss. In this case, Theorem 1 can be proved by Theorem 3.1 in [35]. When a least squares classifier is adopted, $L_{\boldsymbol{\mu}}$ can be written as:

$$L_{\boldsymbol{\mu}} = E_{((\mathbf{x},y),(\mathbf{x}',y')) \in \mathcal{P} \times \mathcal{P}}((t_{y,y'} - \mathbf{z}_{\mathbf{x},\mathbf{x}'} \cdot \boldsymbol{\mu})^2). \tag{12}$$

In this case, Theorem 1 can be proven according to the following definition and lemmas.

Definition 1 formally defines what is a "good" set of features.

Definition 1: Considering n samples $(\mathbf{x}, y)$ drawn i.i.d. from $\mathcal{X} \times \mathcal{Y}$, $\mathbf{F}$ is the completed set of sample features. A set of features $\boldsymbol{F}_{set} \subseteq \mathbf{F}$ is an ϵ-good set of features if there exist a classifier $\boldsymbol{\mu}$ s.t.

$$E_{(\mathbf{x}_{set},y)}((\mathbf{T} - \mathbf{Z}_{set} \cdot \boldsymbol{\mu})^2) \leq \epsilon$$

where $\mathbf{T}$ and $\mathbf{Z}_{set}$ are new samples matrix in K-space correspond to n samples $(\mathbf{x}_{set}, y)$ only contain $\mathbf{F}_{set}$ features in original space by projecting each features using same kernel.

We first use Lemma 1 to show that a good set of features can be induced by a K-classifier, which has low expected loss in K-space. Then Lemma 2 is used to show that we can effectively obtained a good set of features from a finite training sample, which directly follows from Theorem 21 in [41].

Lemma 1: Let $\mathcal{P}$, $\boldsymbol{\mu}_{set}$, $L_{\boldsymbol{\mu}_{set}}$, R be as in Theorem 1. Then the $\hat{\mathbf{F}}$ is a $L_{\boldsymbol{\mu}}$-good set of features with respect to $\mathcal{P}$.

Lemma 2: Let $\mathbf{F}_{set}$ be an ϵ-good set of features, $\mathbf{Z}_{set}$ and $\mathbf{T}$ are represented as in Definition 1 correspond to n original samples, and $\hat{f}(x) = \mathbf{Z} \cdot \boldsymbol{\mu}_{set}$. Then, with probability at least $1 - \delta$, the generalization error is:

$$P_{(x,y)}(y\hat{f} \leq 0) \leq \epsilon + \mathcal{O}(\sqrt{\frac{R^2 \ln(1/\delta)}{\gamma^2 n}})$$

IV. Performance Evaluation

A. Experimental Setting

In this section, we evaluate our proposed BFS and semi-BFS methods with respect to many current supervised and semi-supervised methods. We aim to show that for supervised feature selection our method can get excellent and robust results while for semi-supervised feature selection our method can outperform state-of-arts.

For supervised feature selection, some methods such as SPFS [24], LLFS [27], L21RFS [26], mRMR [25] are adopted to serve as comparative methods with our BFS method. For semi-supervised feature selection, another group of methods, including LSDF [32], FS-Manifold [33] and semiMRSF [29], are selected as the comparative methods. All these methods setting follow the authors's suggestions.

B. Data Sets

Following [29], six real-world data sets are used in our experiments. The data sets downloaded from [42] that involves image and microarray applications. We show detailed information on the data sets in Table I. In supervised cases, training data are randomly selected form 50% samples and test data is selected as the rest samples for each data set. In semi-supervised case, we randomly choose 20% and 40% samples as labeled and unlabeled data respectively, and the rest 40% samples are used as test data. In order to eliminate the effects caused by randomly chosen, we repeat this process 20 times and obtain 20 partitions of original data. We evaluate above feature selection methods on each partition and report averaged results.

TABLE I. SUMMARY OF THE DATA SETS

Data Set	# Features	# Instances	# Classes
AR10P	2400	130	10
CLL-SUB-111	11340	111	3
ORL10P	10304	100	10
PIX10P	10000	100	10
PIE10P	2420	210	10
TOX-171	5748	171	4

C. Evaluation Criteria

We use classification accuracy, which is obtained by linear SVM using selected feature, to evaluate the feature selection methods for both supervised and semi-supervised cases. And we use *paired student's t-test* to evaluate the statistical significance of the improvement. The p-value of the *t-test* represents tdhe probability whether two sets of compared results come from distributions with an equal mean. One can consider two sets have statistically significant if p-value is smaller than 0.05.

D. Experimental Results

1) Results of the supervised case: We show the averaged SVM classification accuracy on data set CLL-SUB-111 in Fig. 1. The result of BFS has no significant difference with state-of-the-arts. And the "aggregated" SVM classification accuracy of different methods on each data set is showed in table II following [29]. The "aggregated" SVM classification accuracy is defined as the averaging the averaged SVM classification accuracy when $10, 20, \cdots, 200$ features are selected. Table II has the following meanings, the first part is the mean $\pm$ standard deviation and the second part is the p-value obtained by the *paired student's t-test*. The bold values in each cell of II represent the highest accuracy and those having no significant difference from the highest one.

As can be seen from the above results, the biggest advantage of BFS is that it has strong robustness. In the experiment, BFS has reached the highest accuracy or has no significant difference from the highest one for all data sets. But other methods may reach high accuracy in some data sets and cannot reach high accuracy in the others. What's more, since one can use different classifiers to implement our approach for different data structures, there is possibility for better classification accuracy. Due to reasons of time, we have not used other classifiers in this paper.

2) Results of the semi-supervised case: We show the averaged SVM classification accuracy on data set TOX-171 in Fig. 2. As can be seen, semi-BFS have the top classification performance. As previously, "aggregated" SVM classification accuracy is reported in table III. These results show our semi-BFS can be the highest accuracy except in PIX10P.

3) Discuss the importance of manifold regularization: In the semi-supervised feature selection, we use the manifold regularization to extent BFS to semi-BFS. As we mentioned above, α_3 controls the importance of manifold regularization, which reflect the local information of both labeled and unlabeled samples. Here we range α_3 from e^1 to e^5. The result in benchmark PIE10P shows in Fig. 3. We can see when α_2 is set as e^4 the classification accuracy is the highest. In this case, the value of manifold regularization term and the loss term is in one order of magnitude.

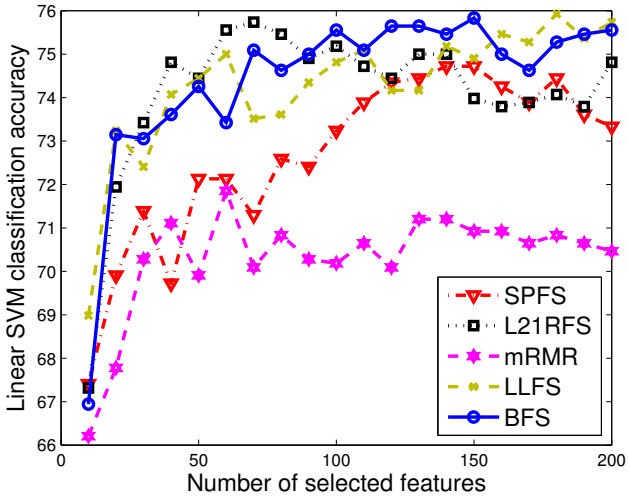

Fig. 1. **Supervised case:** Comparison of linear SVM classification accuracy of different supervised feature selection algorithms on the CLL-SUB-111 data set.

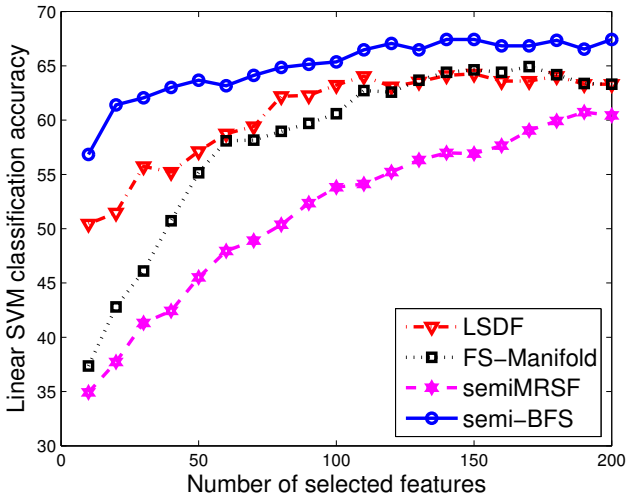

Fig. 2. **Semi-supervised case:** Comparison of linear SVM classification accuracy of different semi-supervised feature selection algorithms on the TOX-171 data set.

4) Feature Redundancy: As we mentioned above, a good set of features always has low redundancy. Here we compared the redundancy of selected features by BFS with linear kernel and Gaussian kernel. Results show in Fig. 4. From the result, using linear kernel can obtain least redundancy. We also test the redundancy of selected features by BFS with redundancy regularization. We also test the redundancy of selected features using linear kernel with and without redundancy regularization. It gets the same result. Finally, in Fig. 5 we compared the redundancy of selected features by SPFS [24], LLFS [27], L21RFS [26] , and mRMR [25] in the same data set. Result shows the lower redundancy the higher classification accuracy it will have.

TABLE II. **SUPERVISED CASE**: AGGREGATED LINEAR SVM CLASSIFICATION ACCURACY. BOLDFACE MEANS NO STATISTICAL DIFFERENCE FROM THE BEST ONE (P-VAL ≥ 0.05).

Data	BFS	SPFS [24]	L21RFS [26]	mRMR [25]	LLFS [27]
TOX-171	$\mathbf{77.65 \pm 3.86\,(1.00)}$	$\mathbf{77.10 \pm 4.60\,(0.08)}$	$\mathbf{77.55 \pm 5.60\,(0.56)}$	$75.72 \pm 3.60\,(0.00)$	$\mathbf{77.43 \pm 4.59\,(0.63)}$
PIX10P	$\mathbf{95.91 \pm 1.79\,(0.51)}$	$\mathbf{95.91 \pm 2.55\,(0.05)}$	$95.28 \pm 2.55\,(0.00)$	$95.39 \pm 2.42\,(0.02)$	$\mathbf{97.19 \pm 2.15\,(1.00)}$
ORL10P	$\mathbf{93.13 \pm 2.55\,(0.37)}$	$91.63 \pm 3.92\,(0.01)$	$91.44 \pm 3.62\,(0.01)$	$\mathbf{93.76 \pm 2.58\,(1.00)}$	$\mathbf{92.74 \pm 2.37\,(0.09)}$
AR10P	$\mathbf{86.81 \pm 4.23\,(1.00)}$	$\mathbf{85.34 \pm 5.88\,(0.23)}$	$\mathbf{86.11 \pm 4.66\,(0.44)}$	$\mathbf{85.80 \pm 4.56\,(0.19)}$	$78.93 \pm 6.69\,(0.00)$
CLL-SUB-111	$\mathbf{74.42 \pm 4.56\,(1.00)}$	$\mathbf{72.69 \pm 6.56\,(0.12)}$	$\mathbf{74.12 \pm 4.69\,(0.66)}$	$70.31 \pm 4.80\,(0.00)$	$\mathbf{74.29 \pm 4.24\,(0.87)}$
PIE10P	$\mathbf{97.85 \pm 1.26\,(0.06)}$	$\mathbf{98.25 \pm 2.04\,(0.37)}$	$\mathbf{98.55 \pm 0.78\,(1.00)}$	$96.18 \pm 1.57\,(0.00)$	$93.22 \pm 4.44\,(0.00)$
AVG	**87.63**	86.82	87.18	86.14	85.63

TABLE III. **SEMI-SUPERVISED CASE**: AGGREGATED LINEAR SVM CLASSIFICATION ACCURACY. BOLDFACE MEANS NO STATISTICAL DIFFERENCE FROM THE BEST ONE (P-VAL ≥ 0.05).

Data	semi-BFS	LSDF [32]	FS-Manifold [33]	semiMRSF [29]
TOX-171	$\mathbf{64.97 \pm 4.22\,(1.00)}$	$60.64 \pm 5.42\,(0.00)$	$58.29 \pm 3.74\,(0.00)$	$51.63 \pm 6.18\,(0.00)$
PIX10P	$90.82 \pm 3.30\,(0.03)$	$\mathbf{93.63 \pm 3.68\,(1.00)}$	$84.57 \pm 11.22\,(0.00)$	$\mathbf{92.59 \pm 3.18\,(0.20)}$
ORL10P	$\mathbf{86.06 \pm 4.35\,(0.48)}$	$81.73 \pm 7.76\,(0.01)$	$75.74 \pm 5.82\,(0.00)$	$\mathbf{86.96 \pm 3.85\,(1.00)}$
AR10P	$\mathbf{69.81 \pm 6.25\,(0.07)}$	$\mathbf{73.54 \pm 5.14\,(0.69)}$	$70.49 \pm 7.78\,(0.04)$	$\mathbf{74.23 \pm 8.90\,(1.00)}$
CLL-SUB-111	$\mathbf{62.67 \pm 9.17\,(1.00)}$	$53.55 \pm 6.06\,(0.00)$	$55.63 \pm 1.41\,(0.00)$	$57.85 \pm 0.68\,(0.00)$
PIE10P	$\mathbf{87.75 \pm 3.76\,(1.00)}$	$82.53 \pm 3.70\,(0.00)$	$82.72 \pm 6.32\,(0.00)$	$80.37 \pm 5.61\,(0.00)$
AVG	**77.01**	74.27	71.24	73.94

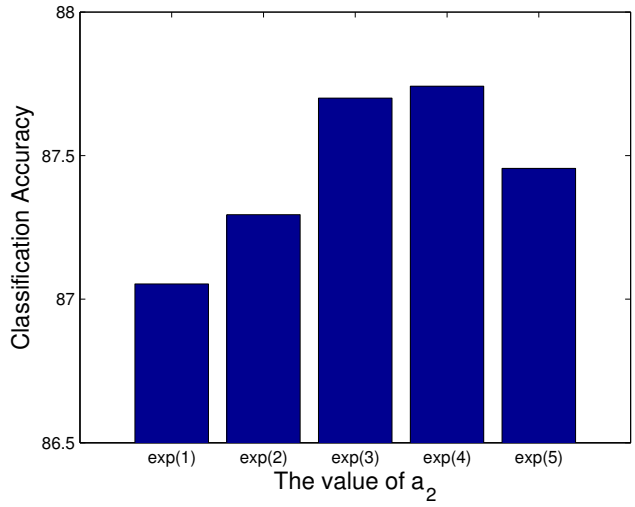

Fig. 3. Importance of manifold regularization

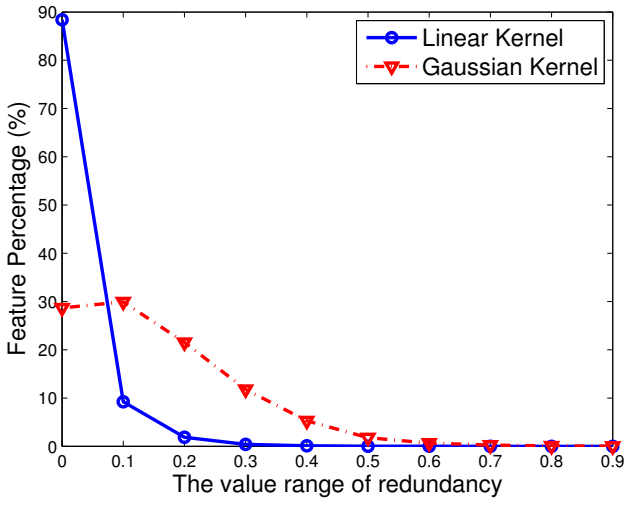

Fig. 4. Redundancy of selected features by different kernels, the value range of redundancy from 0 to 0.9 each represent a redundancy interval in step of 0.1.

V. CONCLUSIONS

In this paper, we have proposed a novel method for both supervised and semi-supervised feature selection using binary classification in kernel space. This is a new view of feature selection, which makes it possible to solve the feature selection problem through any state-of-arts binary classification methods. Theoretical and experimental results show that the proposed method has good performance in both supervised and semi-supervised feature selection. What's more, the method is robust due to the fact that it has no assumption towards data structure. In future work, how to effectively learn the kernel, which projects features to K-space, will be considered. And the speed of this method needs to be improved as well.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Project no. 60970034, 61170287, 61232016) and the Hunan Provincial Science and Technology Planning Project of China (Project no. 2012FJ4269).

REFERENCES

[1] E. Zhu, J. Yin, and G. Zhang, "Fingerprint matching based on global alignment of multiple reference minutiae," *Pattern Recognition*, vol. 38, no. 10, pp. 1685–1694, 2005.

[2] F. Liu, L. Zhou, C. Shen, and J. Yin, "Multiple kernel learning in the primal for multi-modal alzheimer's disease classification," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, no. 99, pp. 1–1, 2013. [Online]. Available: http://dx.doi.org/10.1109/JBHI.2013.2285378

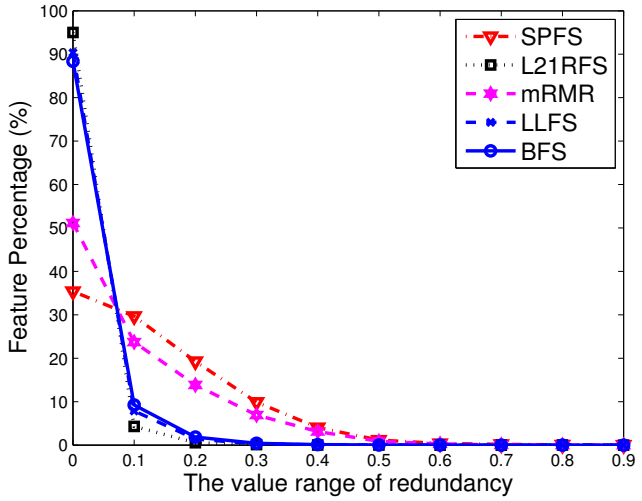

Fig. 5. Redundancy of selected features by different methods, the value range of redundancy from 0 to 0.9 each represent a redundancy interval in step of 0.1.

[3] Q. Liu, J. Yin, V. Leung, J.-H. Zhai, Z. Cai, and J. Lin, "Applying a new localized generalization error model to design neural networks trained with extreme learning machine," *Neural Computing and Applications*, pp. 1–8, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00521-014-1549-5

[4] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, no. 1, pp. 175–186, 2014.

[5] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray, "Multi-objective feature selection with NSGA II," *Adaptive and Natural Computing Algorithms*, pp. 240–247, 2007.

[6] S. Zaman and F. Karray, "Features selection for intrusion detection systems based on support vector machines," in *IEEE Consumer Communications and Networking Conference*, 2009, pp. 1–8.

[7] T. Kubo, M. Yoshida, T. Hattori, and K. Ikeda, "Feature selection for vowel recognition based on surface electromyography derived with multichannel electrode grid," *Intelligent Science and Intelligent Data Engineering*, pp. 242–249, 2012.

[8] K. Zhang, Y. Li, P. Scarf, and A. Ball, "Feature selection for high-dimensional machinery fault diagnosis data using multiple models and radial basis function networks," *Neurocomputing*, vol. 74, no. 17, pp. 2941–2952, 2011.

[9] C. Alippi, G. Baroni, A. Bersani, and M. Roveri, "Unsupervised feature selection algorithms for Wireless Sensor Networks," in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2009, pp. 32–37.

[10] W. H. Abdulla and N. Kasabov, "Reduced feature-set based parallel CHMM speech recognition systems," *Information Sciences*, vol. 156, no. 1-2, pp. 21–38, 2003.

[11] L. Goh, Q. Song, and N. Kasabov, "A novel feature selection method to improve classification of gene expression data," *Proceedings of the second conference on Asia-Pacific bioinformatics*, vol. 29, pp. 161–166, 2004.

[12] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of 20th International Conference on Machine Learning*, vol. 3, 2003, pp. 856–863.

[13] P. A. Estévez, M. Tesmer, C. a. Perez, and J. M. Zurada, "Normalized mutual information feature selection." *IEEE transactions on neural networks*, vol. 20, no. 2, pp. 189–201, 2009.

[14] J. Xu, G. Yang, H. Man, and H. He, "L_1 graph based on sparse coding for feature selection," *Advances in Neural Networks ISNN 2013*, pp. 594–601, 2013.

[15] J. Xu, Y. Yin, H. Man, and H. He, "Feature selection based on sparse imputation," in *International Joint Conference on Neural Networks*, 2012, pp. 1–7.

[16] M. Tesmer and P. A. Estévez, "Amifs: adaptive feature selection by using mutual information," in *International Joint Conference on Neural Networks*, 2004.

[17] H. H. Inbarani, A. T. Azar, and G. Jothi, "Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis," *Computer methods and programs in biomedicine*, vol. 113, no. 1, pp. 175–185, 2014.

[18] L. Liang, V. Cherkassky, and D. A. Rottenberg, "Spatial SVM for feature selection and fMRI activation detection," in *International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 1463–1469.

[19] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.

[20] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper–filter feature selection algorithm using a memetic framework," *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 37, no. 1, pp. 70–76, 2007.

[21] S. F. Crone and N. Kourentzes, "Feature selection for time series prediction A combined filter and wrapper approach for neural networks," *Neurocomputing*, vol. 73, no. 10-12, pp. 1923–1936, 2010.

[22] Z. Xu, R. Jin, J. Ye, M. Lyu, and I. King, "Non-Monotonic Feature Selection," in *Proceedings of 26th International Conference on Machine Learning*, no. 1, 2009, pp. 1145–1152.

[23] Y. Li and B.-L. Lu, "Feature selection based on loss-margin of nearest neighbor classification," *Pattern Recognition*, vol. 42, no. 9, pp. 1914–1921, 2009.

[24] Z. Zhao, L. Wang, and H. Liu, "Efficient Spectral Feature Selection with Minimum Redundancy," *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[25] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[26] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," *Advancess in Neural Information Processing Systems*, pp. 1813–1821, 2010.

[27] Y. Sun, S. Todorovic, and S. Goodison, "Local-learning-based feature selection for high-dimensional data analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1610–1626, Sep. 2010.

[28] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On Similarity Preserving Feature Selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 619–632, Mar. 2013.

[29] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–1, 2013. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2013.2287275

[30] H. Julia and K. Joshua, "Semi-supervised feature selection via multiobjective optimization," in *International Joint Conference on Neural Networks*, 2006, pp. 3319–3326.

[31] Z. Zheng and L. Huan, "Spectral Feature Selection for Supervised and Unsupervised Learning," in *Proceedings of 24th International Conference on Machine Learning*, 2007.

[32] J. Zhao, K. Lu, and X. He, "Locality sensitive semi-supervised feature selection," *Neurocomputing*, vol. 71, no. 10, pp. 1842–1849, 2008.

[33] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization." *IEEE transactions on neural networks*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.

[34] B. Mikhail, N. Partha, and S. Vikas, "Manifold Regularization A Geometric Framework for Learning from Labeled and Unlabeled Examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[35] A. Kumar, A. Niculescu-Mizil, K. Kavukcoglu, and H. Daume III, "A Binary Classification Framework for Two-Stage Multiple Kernel Learning Abhishek," in *Proceedings of 29th International Conference on Machine Learning*, 2012.

[36] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[37] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[38] T. Robert, "Regression Shrinkage and Selection via the Lasso," *Journal of Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, 1996.

[39] K. Seung-Jean, K. Kwangmoo, L. Michael, B. Stephen, and G. Dimitry, "An Interior-Point Method for Large-Scale L1-Regularized Least Squares," *IEEE Journal on Selected Topics in Signal Processing in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[40] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *The Journal of Machine Learning Research*, vol. 13, pp. 795–828, 2012.

[41] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *The Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2003.

[42] [Online]. Available: http://featureselection.asu.edu/