Feature Selection in a Kernel Space

Bin Cao CAOBIN@PKU.EDU.CN
Peking University, Beijing, China

Dou Shen DSHEN@CSE.UST.HK
Hong Kong University of Science and Technology, Hong Kong

Jian-Tao Sun JTSUN@MICROSOFT.COM
Microsoft Research Asia, 49 Zhichun Road, Beijing, China

Qiang Yang QYANG@CSE.UST.HK
Hong Kong University of Science and Technology, Hong Kong

Zheng Chen ZHENGC@MICROSOFT.COM
Microsoft Research Asia, 49 Zhichun Road, Beijing, China

Abstract

We address the problem of feature selection in a kernel space to select the most discriminative and informative features for classification and data analysis. This is a difficult problem because the dimension of a kernel space may be infinite. In the past, little work has been done on feature selection in a kernel space. To solve this problem, we derive a basis set in the kernel space as a first step for feature selection. Using the basis set, we then extend the margin-based feature selection algorithms that are proven effective even when many features are dependent. The selected features form a subspace of the kernel space, in which different state-of-the-art classification algorithms can be applied for classification. We conduct extensive experiments over real and simulated data to compare our proposed method with four baseline algorithms. Both theoretical analysis and experimental results validate the effectiveness of our proposed method.

1. Introduction

Finding a proper representation of data from their original features is a fundamental problem in machine learning. Generally speaking, not all original features are beneficial for classification tasks. Some of the features, which can even be noise, will hurt the classification performance. Finding a proper representation is essential for removing the noisy features or for deriving some new features from the original space, so that we only need to keep the necessary information for classification and data analysis.

Solutions for feature selection problems can be broadly classified into two groups: feature extraction (such as PCA(I.T.Jolliffe., 2002), LDA(Fukunaga, 1990)) and feature selection (such as IG, MI(Guyon & Elisseeff, 2003), *Relief*(Kira & Rendell, 1992)). Traditional linear feature selection and extraction operations are conducted in the original input space, and thus cannot handle nonlinear relationships in the data well. For example, the principal components of features may be nonlinearly related to the input variables and the data of different categories are not separable by a hyperplane. To tackle this problem, kernel methods are introduced by mapping the data from an original space (or an input space) to a kernel space using a specially designed mapping function. Among these methods, Kernel Principle Component Analysis (KPCA) (Scholkopf & A.J.Smola., 2002) was proposed to find nonlinear principal components. However, as an unsupervised learning method, KPCA does not take the

Appearing in *Proceedings of the 24^{th} International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

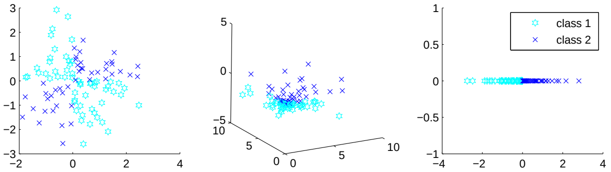

Figure 1: Illustrating the XOR data distribution. The left figure shows the data distribution in an original space, which is non-linearly separable. The middle figure shows the data distribution in the kernel space, which is linearly separable. The rightmost sub-figure shows the data distribution in a kernel subspace, where the data are linearly separable in a space with low dimensions

category information into consideration. Kernel Fisher Discriminant Analysis (KFD) (Mika et al., 1999a) and Generalized Discriminant Analysis (GDA) (Baudat & Anouar, 2000) can make use of category information. However, the number of extracted features by these methods is limited by the number of categories which is at most $n-1$ if the category count is n. In practice, it is not uncommon that there are more than $n-1$ features that are useful for classification. In addition, these kernelized methods has high computational complexity, since they rely on the computation of SVD (Horn & Johnson, 1985), which requires at least $O(N^2k)$ time, to our best knowledge, where N is the number data points to be considered and k is the number of eigenvectors.

Intuitively, we can adapt information gain (IG) or mutual information (MI) methods in the kernel space. However, the mapping functions from the original space to the kernel space are usually *implicit.* Even if the function is explicit, the dimension of kernel space may be infinite. Furthermore, feature dependency often prevents us from considering the features in a kernel space by IG or MI.

In this paper, we present a novel method for feature selection *directly* in the kernel space, without moving back to the original feature space as in (Bradley & Mangasarian, 1998; Liang & Zhao, 2006). To do this, we first construct a basis set in the kernel space. We then extend the margin-based feature-selection methods to select the best bases for classification. Finally, we classify the test data in the kernel space with the selected basis set directly. Compared to methods such as KPCA and GDA, the computational complexity of our method is reduced significantly to $O(N^2)$. We conduct extensive experiments over the IDA data sets to compare our method with four state-of-the-art baselines. Theoretical analysis and experimental results validate the effectiveness of our proposed method.

2. Related Works

Traditional feature selection methods such as Information Gain and χ^2-Test consider the contribution of each feature to the classification task independently. They have been proven to be effective in many applications (Guyon & Elisseeff, 2003; Yang & Pedersen, 1997). However, they do not work well when features are not independent from each other as in the *XOR* problem on the leftmost sub-figure of Figure 1.

Recently, margin based feature selection or feature weighting methods are proposed including *ReliefF* (Kononenko, 1994), *Simba*(Gilad-Bachrach et al., 2004), *I-Relief*(Sun & Li, 2006). They are derived from the basic *Relief* algorithm(Kira & Rendell, 1992) which weights each feature to get maximal margin. The margin of a data point x used in these methods is defined as the distance between the the nearest same-labelled data (nearhit) to x and the nearest different-labeled data (nearmiss) to x, respectively. The advantage of margin based feature selection methods is optimal weights of features can be estimated by considering the global information. Thus these methods work well even for data with dependent features (Gilad-Bachrach et al., 2004). The concept of margins has been widely used in machine learning in the past decades to measure the models' generalization ability.

Our work is also related to kernel methods for feature selection (Bradley & Mangasarian, 1998; Liang & Zhao, 2006). However, most of the previous work in this area uses kernels to help select features in the *original* space. In contrast, our goal is to select features in the kernel space, because we cannot change non-separability nature of the data through feature selection in the original space. However, feature selection in a kernel space can handle this problem, as illustrated in the right sub-figure of Figure 1.

Kernel based feature-extraction methods have also been designed to address the nonlinear projection problem. Kernel Principle Component Analysis (Scholkopf & A.J.Smola., 2002) was proposed to find principal components in high-dimensional feature spaces. KPCA is shown to be effective in many tasks such as image de-noising (Mika et al., 1999b). However, KPCA does not consider the category information. As a result, features extracted by KPCA may be irrelevant to a classification task. To tackle this problem, Kernel based LDA methods such as FKD (Mika et al., 1999a) , GDA (Baudat & Anouar, 2000) are proposed. GDA extracts the features that are discriminative and beneficial for classification in a kernel space. However, the number of extracted features by GDA is limited by the number of categories.

(Weston et al., 2003) discussed the problem of kernel feature selection when the mapping function between the orignal space and the kernel space can be explicitly expressed. This work is limited to only certain types of kernels, such as the polynomial kernels. However, our method does not have this restriction. We are able to deal with any kernels as a result. (Baudat & Anouar, 2003) proposed a method for feature-vector selection using data selection. However, similar to KPCA, the method does not find discriminative feature vectors. In (Niijima & Kuhara, 2006), Niijima proposed a wrapper method that utilizes the kernel-based classifiers for gene-subset selection. However, the wrapper method faces the problem of high computational complexity.

In (Wu et al., 2005), the author proposed the Sparse Large Margin Classifiers. They find that sparse SVM classifiers in a kernel space usually have better generalization ability. However, their work is restricted to SVM classifiers, whereas our method can be applied to different classifiers.

3. Feature Selection in a Kernel Space

Let $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N} \in \mathbb{R}^I \times \{\pm 1\}$ denote a training dataset, where I is the data dimensionality. $\mathbf{K}(\mathbf{x}, \mathbf{x}')$ is the kernel function with an implicit mapping function $\phi(\mathbf{x}) = z$ where $\mathbf{x} \in \mathbb{R}^I, z \in \mathcal{F}$, where $\mathcal{F}$ is a high-dimensional space. Generally speaking, the features in the kernel space are *not* assumed to be independent. Therefore, feature selection methods that consider each feature individually are unlikely to work well in a kernel space. However, a margin-based feature-selection method can handle the feature-dependency problem successfully, as explored in (Gilad-Bachrach et al., 2004). In the following, we give a brief introduction to the basic margin-based feature-selection method *Relief* (Kononenko, 1994; Sun & Li, 2006), and then present our algorithm to extend *Relief* in a kernel space.

3.1. Margin Based Feature Selection (Weighting) Methods

Margin-based feature-selection methods such as *ReliefF*, *Simba*, *I-Relief* have been developed based on the idea of *Relief*, which uses a weighting-based method for feature selection. The weight of each feature reflects its ability to discriminate different categories. *Relief* tries to find weights by maximizing the margins between data of different categories. The margin of a data point $\mathbf{x}$ used in *Relief* is defined by the distance between the nearhit $NH(\mathbf{x})$ and nearmiss $NM(\mathbf{x})$ of the target data point, which are the nearest point with the same and different label, respectively. The margin of the whole training dataset is the sum of the margin of all data points, as shown in the following equation.

$$\max_{w} \sum_{n=1}^{|\mathcal{D}|} (D_w(\mathbf{x}^{(n)}, NM(\mathbf{x}^{(n)})) - D_w(\mathbf{x}^{(n)}, NH(\mathbf{x}^{(n)}))) \quad s.t. \quad ||w||_2^2 = 1, w \geq 0$$

where $D_w(\mathbf{x}, \mathbf{x}') = \sum_{i=1} w_i |\mathbf{x}_i - \mathbf{x}'_i|$ is a projection of the difference-vector onto a dimension, and w_i is the weight of the i^{th} feature. Thus, the feature weighting problem is converted to an optimization problem, for maximizing the above function.

The solution to the above problem can be found by solving for the weight vector $\mathbf{w} = (\mathbf{m})^+/||(\mathbf{m})^+||_2$, where $\mathbf{m} = \sum_{n=1}^{|\mathcal{D}|} (\mathbf{x}^{(n)} - NM(\mathbf{x}^{(n)})) - (\mathbf{x}^{(n)} - NH(\mathbf{x}^{(n)}))$ which is the margin vector and $(m_i)^+ = max(m_i, 0)$. We refer the readers to (Kononenko, 1994; Sun & Li, 2006) for details of *Relief*.

3.2. Margin Based Feature Selection in Kernel Space

Now we discuss how to extend *Relief* in a kernel space. Note that *Relief* calculates the weight of each feature by $w_i = \mathbf{e}^{(i)} \cdot (\mathbf{m})^+/||(\mathbf{m})^+||_2$, where $\mathbf{e}^{(i)}$ is the i^{th} basis in a Euclidean space and $\mathbf{m}$ is a margin vector. Similarly, we can derive the weight of a feature in the kernel space by $w'_i = \mathbf{e}'^{(i)} \cdot (\mathbf{m}')^+/||(\mathbf{m}')^+||_2$, where $\mathbf{e}'^{(i)}$ is the basis of the kernel space and $\mathbf{m}'$ is a margin vector in the kernel space.

Recall that the margin defined in *Relief* is determined by the distance defined in a space. We can extend *Relief* to a kernel space by using a distance function induced by the corresponding kernel function, as shown in Equation 1. The nearhit and nearmiss points are found in the kernel space based on the distance.

$$\begin{aligned} \mathbf{D_K}(\mathbf{x}, \mathbf{x}') &= \langle \phi(\mathbf{x}) - \phi(\mathbf{x}'), \phi(\mathbf{x}) - \phi(\mathbf{x}') \rangle \\ &= \mathbf{K}(\mathbf{x}, \mathbf{x}) + \mathbf{K}(\mathbf{x}', \mathbf{x}') - 2\mathbf{K}(\mathbf{x}, \mathbf{x}') \end{aligned} \qquad (1)$$

where $\phi(\mathbf{x})$ is the mapping function corresponding to the kernel $K(\mathbf{x}, \mathbf{x}')$.

When we want to calculate the distance in the weighted kernel space, we need to use the weighted kernel $\mathbf{K}_w$ as follows

$$\mathbf{K}_w(\mathbf{x}, \mathbf{x}') = \sum_i w_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}')$$

where $\phi_i(\mathbf{x})$ is the i^{th} feature of $\phi(\mathbf{x})$. In the kernel space, the index of i is allowed to range to infinity.

Since we do not have $\phi_i(\mathbf{x})$ explicitly, we cannot calculate the above equation directly. However, if we have an orthogonal basis set $\{\eta^{(i)}\}$ in the kernel space, then the function $\mathbf{K}_w(\mathbf{x}, \mathbf{x}')$ can be calculated as

$$\mathbf{K}_w(\mathbf{x}, \mathbf{x}') = \sum_i w_i \langle \phi(\mathbf{x}), \eta^{(i)} \rangle \langle \phi(\mathbf{x}'), \eta^{(i)} \rangle$$

Given a data set $\mathcal{D}$, we let the bases of a kernel space be expressed by combinations of the data in high dimensional spaces. That is: $\eta^{(i)} = \sum_j \alpha_{ij} \phi(\mathbf{x}^{(j)})$. We have the following equation:

$$\begin{aligned} \mathbf{K}\ {}_w(\mathbf{x}, \mathbf{x}') &= \sum_i^{|\mathcal{D}|} w_i (\phi(\mathbf{x}) \cdot \eta^{(i)})(\phi(\mathbf{x}') \cdot \eta^{(i)}) \\ &= \sum_i^{|\mathcal{D}|} w_i (\sum_j \alpha_{ij} \mathbf{K}(\mathbf{x}, \mathbf{x}^{(j)}))(\sum_j \alpha_{ij} \mathbf{K}(\mathbf{x}', \mathbf{x}^{(j)})) \end{aligned}$$

We should point out that when the data are not sufficient or the dimension of the kernel space is infinite, not all the bases of kernel space can be expressed by combinations of the data. Therefore this method can only get an approximation $\mathbf{K}'_w(\mathbf{x}, \mathbf{x}')$ of $\mathbf{K}_w(\mathbf{x}, \mathbf{x}')$. However, the following theorem shows the approximation is reasonable.

Theorem 1. *Let $\mathcal{S}$ be the space spanned by the available high dimensional data. If $\phi(\boldsymbol{x}') \in \mathcal{S}$, then we have $\boldsymbol{K}_w(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{K}'_w(\boldsymbol{x}, \boldsymbol{x}')$*

Proof. $\mathbf{K}_w(\mathbf{x}, \mathbf{x}') = \mathbf{K}'_w(\mathbf{x}, \mathbf{x}') + \sum_{i=|\mathcal{D}|+1}^{\infty} w_i (\phi(\mathbf{x}) \cdot \eta^{(i)})(\phi(\mathbf{x}') \cdot \eta^{(i)})$, where $\eta^{(i)}, i = |\mathcal{D}| + 1 \cdots \infty$ are bases of orthogonal complement subspace of $\mathcal{S}$. Since $\phi(\mathbf{x})$ is contained in $\mathcal{S}$, therefore $\eta^{(i)} \cdot \phi(\mathbf{x}) = 0, i = |\mathcal{D}| + 1 \cdots \infty$. Hence we have $\mathbf{K}'_w(\mathbf{x}, \mathbf{x}') = \mathbf{K}_w(\mathbf{x}, \mathbf{x}')$. □

If $\phi(\mathbf{x}') \notin \mathcal{S}$, the difference between $\mathbf{K}'_w(\mathbf{x}, \mathbf{x}')$ and $\mathbf{K}_w(\mathbf{x}, \mathbf{x}')$ is controlled by $||\phi(\mathbf{x}') - \phi^*(\mathbf{x}')||$, where $\phi^*(\mathbf{x}')$ is the projection vector of $\phi(\mathbf{x}')$ on $\mathcal{S}$. If we assume the training and test data are sampled from the same distribution, then this term can be regarded as extremely small and the approximation would be reasonable.

3.3. Finding the Basis Set

From Section 3.2 we can see that, to extend *Relief* in a kernel space, a critical issue is to find a basis set in the kernel space. In this section, we propose to use the Kernel Gram-Schmidt Process to find an orthogonal basis set.

Algorithm 1 Kernel Gram-Schmidt Process

```
  Input: data $\mathbf{x}^{(i)}$ $(i = 1..N)$
  Output: an orthogonal set of basis vectors
  for $i = 1$ to $N$ do
    $\mathbf{v}^{(i)} = \phi(\mathbf{x}^{(i)})$
    for $j = 1$ to $i - 1$ do
      $\mathbf{v}^{(i)} \leftarrow \mathbf{v}^{(i)} - \langle \phi(\mathbf{x}^{(i)}), \mathbf{v}^{(j)} \rangle \mathbf{v}^{(j)}$
    end for
    Normalize: $\mathbf{v}^{(i)} \leftarrow \frac{\mathbf{v}^{(i)}}{||\mathbf{v}^{(i)}||}$
  end for
  Output: basis set $\{\mathbf{v}^{(i)}\}$
```

Given a data set in an original space $\mathcal{D}$ and a kernel function $\mathbf{K}(\mathbf{x}, \mathbf{x}')$, as well as an implicit mapping function $\phi(\mathbf{x})$, the mapped data in the kernel space are $\mathscr{D} = \{\phi(\mathbf{x}^{(i)}) | i = 1...|\mathcal{D}|\}$. Let $\mu = (\phi(\mathbf{x}^{(1)}), \phi(\mathbf{x}^{(2)}), ..., \phi(\mathbf{x}^{(|\mathcal{D}|)}))^T$. Assume that the rank of $\mathscr{D}$ is $|\mathcal{D}|$. The Kernel Gram-Schmidt Process can find a subset of a *orthogonal basis set* in a kernel space. This is done by Algorithm (1), where as output, each basis is a linear combination of the data in high dimensions. That is $\mathbf{v}^{(i)} = \sum_j \alpha_{ij} \phi(\mathbf{x}^{(j)})$. Thus, we wish to find the mixture weights calculated as follows:

$$\alpha^{(i)} = \frac{\mathbf{e}^{(i)} - \sum_{k=1}^{i-1} \alpha^{(k)} \sum_j \alpha_{kj} K(\mathbf{x}^{(k)}, \mathbf{x}^{(j)})}{||(\mathbf{e}^{(i)} - \sum_{k=1}^{i-1} \alpha^{(k)} \sum_j \alpha_{kj} K(\mathbf{x}^{(k)}, \mathbf{x}^{(j)})) \cdot \mu||}$$

where $\mathbf{e}^{(i)}$ is a vector in which the i^{th} element equals to one and all other elements are equal to zero. $\alpha^{(i)} = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{i|\mathcal{D}|})$.

When the input data are mapped into a kernel space, the dimensionality of the mapped data are usually very high, and may even reach infinity. We do not expect to find all the basis in the kernel space. In fact, the number of orthogonal basis we get by Gram-Schmidt Process is the *rank of the mapped data set* $\{\phi(\mathbf{x})\}$. This rank also equals to the numerical rank of the kernel matrix.

Besides the Kernel Gram-Schmidt Process, other methods such as Kernel PCA can also be used to find a basis set in a kernel space. Since the orthogonal basis sets found by different methods are all orthogonal in the kernel subspace $\mathcal{S}$, and since the number of the basis are equal to the rank of the high-dimensional data set $\{\phi(\mathbf{x})\}$, the basis set found by the Gram-Schmidt Process or by other methods are equivalent under an orthogonal transformation.

Now we show that the selection of a basis set *does not* affect the distance measure learned in the kernel space.

Theorem 2. *$\{\boldsymbol{e}^{(i)}\}$ and $\{\boldsymbol{e}'^{(i)}\}$ are two different orthogonal basis sets in a kernel space, which are equivalent under an orthogonal transformation. Then, a weighting transformation under $\{\boldsymbol{e}^{(i)}\}$ is also a weighting transformation under $\{\boldsymbol{e}'^{(i)}\}$.*

Proof. Suppose $\mathbf{e}'^{(i)} = \sum_j \xi_{ij}\mathbf{e}^{(j)}$, then

$$\begin{aligned}\phi_w(\mathbf{x}) &= \sum_i w_i\langle\phi(\mathbf{x}), \mathbf{e}'^{(i)}\rangle \\ &= \sum_i w_i\langle\phi(\mathbf{x}), \sum_j \xi_{ij}\mathbf{e}^{(j)}\rangle \\ &= \sum_j w'_j\langle\phi(\mathbf{x}), \mathbf{e}^{(j)}\rangle\end{aligned} \tag{3}$$

where $w'_j = \sum_i w_i\xi_{ij}$ □

Theorem 3. *If the orthogonal basis sets of a kerne space are equivalent under an orthogonal transformation, then the optimal $\boldsymbol{K}_w(\boldsymbol{x}, \boldsymbol{x}')$ learned under different basis sets are equivalent.*

Proof. Suppose $\{\mathbf{e}^{(i)}\}$ and $\{\mathbf{e}'^{(i)}\}$ are two different orthogonal basis sets of kernel space which are equivalen under an orthogonal transformation. Based on Theorem 2, the weighting problem under $\{\mathbf{e}^{(i)}\}$ can be converted to the weighting problem under $\{\mathbf{e}'^{(i)}\}$. Therefore there exist $\mathbf{K}'_{w'}(\mathbf{x}, \mathbf{x}')$ under $\{\mathbf{e}'^{(i)}\}$ equivalent to $\mathbf{K}_w(\mathbf{x}, \mathbf{x}')$ under $\{\mathbf{e}^{(i)}\}$. Since the optimal $\mathbf{K}_w(\mathbf{x}, \mathbf{x}')$ is unique, the above conclusion can be draw. □

From Theorem 3, we can see that no matter what particular method is used to find the basis set in a kernel space, the optimal distance $\mathbf{K}_w(\mathbf{x}, \mathbf{x}')$ remains the same.

However, although the weighted feature-space kernels $\mathbf{K}_w(\mathbf{x}, \mathbf{x}')$ are all equivalent, when we discretize the above-computed weights to binary values (zero or one) for feature selection, the results may still be different. We plan to investigate this difference in our future work. In this paper, we just use the Kernel Gram-Schmidt Process and Kernel PCA to find the orthogonal basis set in the kernel space.

3.4. Feature-Selection Algorithms

Our feature-selection framework is summarized in Algorithm (2), where by using different basis-computing method, we can get different algorithms. When Kernel PCA is applied for finding the kernel basis set, we call the algorithm FSKSPCA. When Kernel Gram-Schmidt Process is used, we call the algorithm FSKGP.

3.5. Discussion

Feature selection in the kernel space has very close relationship with SVM regularization methods such as (Wu et al., 2005). We share the interest to find a discriminating subspace of the feature space $\mathcal{F}$. The difference is that our method finds an explicit expression of the subspace and removes the dependence on certain classification algorithms.

Algorithm 2 FSKS

```
Input: training data $\mathbf{x}_i$, label $y_i$
Output: selected features in the kernel space
step 1: Constructing a basis set by either Kernel
        GP or Kernel PCA
step 2: Calculating $w_i$ by Kernel Relief
step 3: Ranking implicit features by $w_i$, select
        features based on the rank
step 4: Projecting the data into the learned
        subspace
```

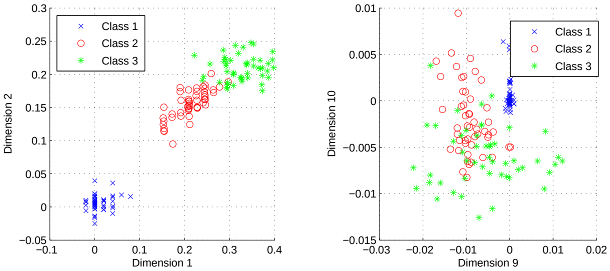

Figure 2: Iris data in different kernel subspaces

4. Experiments

In this section, we conduct extensive experiments to compare our proposed feature-selection method with some state-of-the-art baselines. The experimental results validate the effectiveness of our method.

4.1. Case Study: Discriminative Power

The iris-flower data set (Fisher, 1936) is widely used to test feature selection and feature extraction methods (Baudat & Anouar, 2000; Yan et al., 2005). The data set contains 150 examples in 3 categories. The original feature space of the data has four dimensions, which are *sepal length*, *sepal width*, *petal length* and *petal width*. We use this dataset to illustrate that our method, by extending *Relief* in kernel spaces, can effectively find discriminative features for classification.

The result of applying our method to the iris data is shown in Figure 2. In this experiment, an RBF kernel with $\sigma = 1$ is used. We construct 150 implicit features in the kernel space where they are ranked according to their weights. Features that are top-ranked are expected to contain more discriminative information. This is validated by the two sub-figures in Figure 2. The left sub-figure shows the data projected on the subspace constructed by the first and second top-ranked features, whereas the right sub-figure shows the

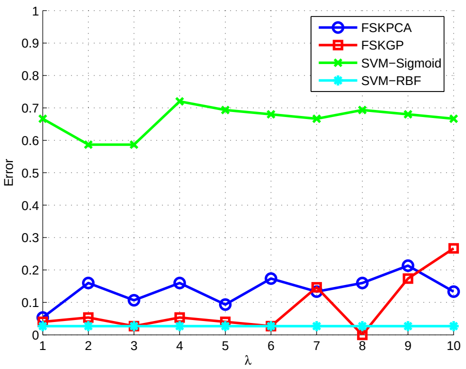

Figure 3: Plotting error rates against λ

data projected on the subspace of the ninth and tenth features. From figure 2, it is clear that the first two features are much more discriminative for classification than the ninth and tenth features.

4.2. Case Study: Robustness

Kernel selection has always been regarded as a critical issue in machine learning. For a given kernel function, if this function is not appropriately selected, one would generally obtain bad performance. However, our feature-selection method has a natural way of automatically correcting this improper selection: even when a kernel function is not appropriately selected, or the kernel parameters are not properly chosen, our method can still perform well. This idea is validated in the following experiment.

In this experiment, we still use the iris dataset. We use a Sigmoid kernel as defined by $K(\mathbf{x}, \mathbf{x}') = \tanh(\lambda \mathbf{x} \cdot \mathbf{x}' + 1)$. We vary the parameter λ so that it ranges between 1 to 10. From figure 3, we can see that sigmoid kernel is not appropriate for classifying this dataset, since the performance of SVM is very poor as compared to the classifier using the RBF kernel function. However, our method can still achieve good results by selecting 10 features, where the classification result using the Sigmoid kernel is comparable to the performance achieved by the SVM using RBF kernels.

4.3. Tests on Classification Performance

4.3.1. Data Sets and Baselines

Table 1: IDA data sets used in the experiments

Data set	#Train	#Test	#Features	#Basis	#Best
GERMAN	700	300	20	700	40-50
DIABETIS	468	300	8	468	10-15
HEART	170	100	13	170	20-30
THYROID	140	75	5	140	20-25

We next perform a series of classification experiments over two categories of datasets. The datasets that we use include a simulated dataset and a set of benchmark datasets from the IDA Benchmark reposiory[1]. The data are given in 100 predefined splits into training and test samples. We use four datasets, which are german, diabetis, heart and thyroid, in our experiments. The information of datasets is shown in Table 1 in which #Best is the number of features when best performance achieved. The results reported in this section are all average values over different training and test data splits.

In order to verify the effectiveness of our methods, we selected several state-of-the-art baselines in our comparison: 1. Generalized Discriminant Analysis(GDA) (Baudat & Anouar, 2000); 2. Kernel Fisher Discriminant Analysis(KFD) (Mika et al., 1999a); 3. Support Vector Machine (SVM); 4. Kernel K-Nearest Neighbor (KKNN). GDA and KFD can generate at most $n-1$ features if the total number of categories is n. In the following experiments, we use $n-1$ features for GDA and KFD uniformly. KNN is the K-Nearest Neighbor algorithm based on a distance function induced by the kernel function. In the experiments, we let $K = 1$.

4.3.2. Experiment on Simulated Data

We first generated some simulated data as shown in Figure 4. In this experiment, we use the RBF kernel with $\sigma = 1$. The rank of mapped data in the kernel space is 50, which corresponds to the number of implicit features in the kernel space. Given the number of the selected features, we map the data into the subspace in the kernel space and then calculate a *distance ratio* of this data. The *distance ratio* is defined as:

$$r = \frac{\sum_{i,j|y^{(i)}=y^{(j)}} Distance(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})}{\sum_{i,j} Distance(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})}$$

The numerator is the sum of the distance between the inner-class data pair, while the denominator is the sum of the distance between all data pairs. The smaller the r is, the easier it is for the data to be separated.

The relationship between the distance ratio and the number of implicit features is shown in Figure 5. The dashed line shows the ratio in the kernel space, where we consider all the features. The solid line shows the ratio in the subspace of the kernel space. From figure 5, we can see that the first few implicit features reduce the distance ratio. This means that the first few

[1] http://ida.first.fhg.de/projects/bench/benchmarks.htm

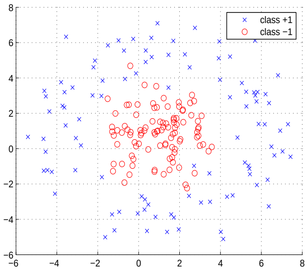

Figure 4: Data in original space

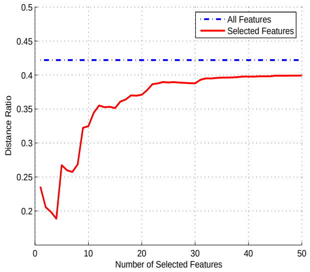

Figure 5: Distance Ratio against number of features

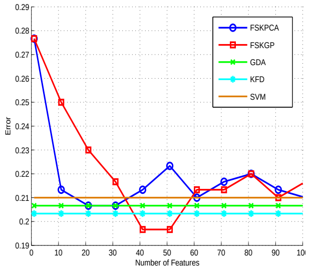

Figure 6: Classification result of german by SVM

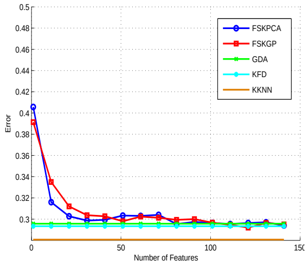

Figure 7: Classification result of german by KKNN

features contain much discriminative information for classification. However, when more implicit features are added, the distance ratio increases, which indicates that the newly added features are noisy and more or less irrelevant for our classification task. When all implicit features are considered, we return to the kernel space, where the distance ratio is the worst. This experiment also indicates that even when the data are located in a relative low dimension subspace in the kernel space, feature selection is still needed to reduce the noisy features.

4.3.3. Experiments on IDA Data

We test our feature selection algorithm on four IDA datasets using two classifiers: SVM and KNN. The results of using SVM are shown in Figures 6, 8, 10, 12 and the results of using the KNN classifier are shown in Figures 7, 9, 11, 13. In these figures, the X-axis is the number of selected features, and the Y-axis is the predictive error of the classifiers.

From the experiment results as shown in Figures 6 to 13, we can see that our method can achieve comparable performance, if not better, to other state-of-the-art algorithms. Often, our method achieves the best result when we use 10-20 features. When using SVM as the classifier, our methods can reduce the error by more than 10% and 5% on the thyroid data set and the heart data set as compared to the other baselines,

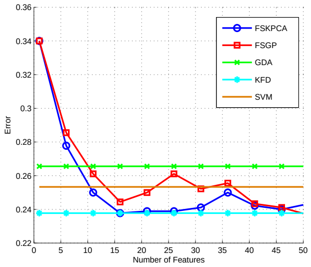

Figure 8: Classification result of diabetis by SVM

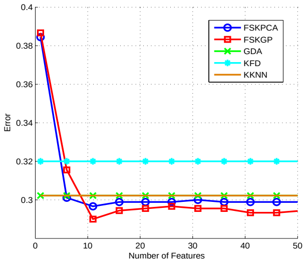

Figure 9: Classification result of diabetis by KKNN

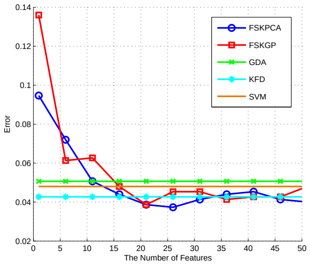

Figure 10: Classification result of thyroid by SVM

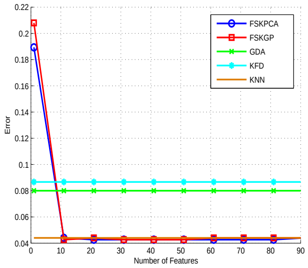

Figure 11: Classification result of thyroid by KKNN

respectively. When using KNN as the classifier, the improvement is over 5% on average.

Considering that the computational complexity of our method FSKPCA is $O(N^2)$ while those of GDA and KFD are $O(N^2k)$, the performance of our method is encouraging.

Another observation from the experimental results is that in most cases, the best performance of FSKPCA and FSKGP are similar. The first few features found by FSKPCA are better than FSKGP. However, FSKPCA's computational complexity is much higher than FSKGP because of the SVD computation.

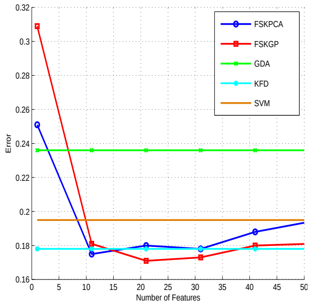

Figure 12: Classification result of heart by SVM

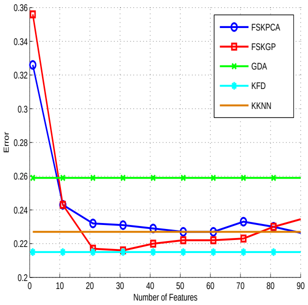

Figure 13: Classification result of heart by KKNN

5. Conclusions and Future Work

In this paper, we have presented a novel method for feature selection in a kernel space. By utilizing the Kernel Gram-Schmidt Process algorithm, we construct an orthogonal basis set in the kernel space. We then extend the margin-based feature-selection methods in the kernel space, so that we can select a subspace of the kernel space which contains the most discriminative information for classification. We have illustrated the advantages of our method over simulated and real data, where we compared our methods against some state-of-the-art baselines. The experimental results show that our methods can achieve comparable or better performance than the best baseline while having a lower computational complexity.

In the future, we plan to further explore the relationship between our method and SVM regularization methods. We also want to theoretically estimate the error bound when we apply feature selection in a kernel space.

Acknowledgments

We would like to thank Jianmin Wu for the valuable discussions. Dou Shen and Qiang Yang are supported by a grant from Hong Kong RGC 621606.

References

Aha, D. W. (1990). *A study of instance-based algorithms for supervised learning tasks: Mathematical, empirical, and psychological evaluations*. Doctoral dissertation, Department of Information & Computer Science, University of California, Irvine.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, *12*, 2385–2404.

Baudat, G., & Anouar, F. (2003). Feature vector selection and projection using kernels. *Neurocomputing*, *55*, 21–38.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. *ICML '98* (pp. 82–90).

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals Eugen.*, *7*, 179–188.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc.

Gilad-Bachrach, R., Navot, A., & Tishby, N. (2004). Margin based feature selection - theory and algorithms. *ICML '04* (p. 43).

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, *3*, 1157–1182.

Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge University Press.

I.T.Jolliffe. (2002). *Principal components analysis*. Springer Verlag.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. *ML92* (pp. 249–256).

Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. *ECML-94: Proceedings of the European conference on machine learning on Machine Learning* (pp. 171–182).

Liang, Z., & Zhao, T. (2006). Feature selection for linear support vector machines. *ICPR '06* (pp. 606–609).

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., & Muller, K. (1999",a). Fisher discriminant analysis with kernels.

Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M., & Rätsch, G. (1999b). Kernel PCA and de–noising in feature spaces. *Advances in Neural Information Processing Systems 11*. MIT Press.

Niijima, S., & Kuhara, S. (2006). Gene subset selection in kernel-induced feature space. *Pattern Recognition Letters*, *27*, 1884–1892.

Scholkopf, B., & A.J.Smola. (2002). *Learnin with kernels*. Cambridge, MA,: The MIT Press.

Sun, Y., & Li, J. (2006). Iterative relief for feature weighting. *ICML '06* (pp. 913–920).

Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M. (2003). Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.*, *3*, 1439–1461.

Wu, M., Schölkopf, B., & Bakir, G. (2005). Building sparse large margin classifiers. *ICML '05* (pp. 996–1003).

Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., & Ma, W.-Y. (2005). Ocfs: optimal orthogonal centroid feature selection for text categorization. *SIGIR '05* (pp. 122–129).

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *ICML '97* (pp. 412–420).