Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Margin-Based Feature Selection in Incomplete Data

Qiang Lou **Zoran Obradovic***
Computer and Information Science Department, Temple University, USA
qianglou@temple.edu zoran.obradovic@temple.edu

Abstract

This study considers the problem of feature selection in in complete data. The intuitive approach is to first impute the missing values, and then apply a standard feature selection method to select relevant features. In this study, we show how to perform feature selection directly, without imputing missing values. We define the objective function of the un certainty margin based feature selection method to maxim ize each instance's uncertainty margin in its own relevant subspace. In optimization, we take into account the uncer tainty of each instance due to the missing values. The exper imental results on synthetic and 6 benchmark data sets with few missing values (less than 25%) provide evidence that our method can select the same accurate features as the al ternative methods which apply an imputation method first. However, when there is a large fraction of missing values (more than 25%) in data, our feature selection method out performs the alternatives, which impute missing values first.

Introduction

Selecting appropriate features is an important step in the data mining process, whose objectives include providing a better understanding of data distribution as well as more accurate and efficient prediction (Guyon and Elisseeff, 2000; Koller and Sahami, 1996). Existing feature selection methods assume that the data is complete or almost complete. However, this is not the case in many real-life applications, such as bioinformatics (Liew, et al. 2000) and remote sensing networks (Radosavljevic et al. 2010). Applying existing feature selection methods to applications with incomplete data would require an imputation method (Lou and Obradovic, 2011) to estimate the missing values first, and then apply the feature selection procedure. This study proposed a method to perform feature selection directly from the incomplete data, without pre-applying any imputation method to estimate the missing values. To the best of our knowledge, this method is the first one to perform feature selection in incomplete data without pre-estimating the missing values.

*Corresponding author. 215 204 6265

We focus on margin-based feature selection, which assigns a weight for each feature, and then selects a set of features including a maximum margin. A margin is a measure for evaluating the quality of a classifier with respect to its decision (Schapire et al. 1998). In order to handle the incomplete data, we define an uncertainty margin for each instance in the presence of missing values. Uncertainty margins ensure that the distance between an instance and other instances is measured in a subspace where all features are observed instead of the whole feature space. Also, to measure the uncertainty margin, we used distance in weighted space rather than in original space. The weighted distance ensures that the feature weights are considered while computing the uncertainty margin.

We define the objective function of feature selection by embedding the uncertainty margin of the whole data set. The feature selection method is then converted to an optimization problem that learns optimal weights for features that maximize the uncertainty margin for the entire data. The new optimization problem is no longer convex, unlike the traditional margin-based feature selection methods (Gilad-Bachrach et al. 2004), since the uncertainty margin is a function of feature weights. To solve the optimization problem including the uncertainty margin, an EM algorithm is proposed to learn the feature weights and uncertainty margin interactively. The experimental results show that our method outperforms the method requiring data imputation in advance.

Related Work

Feature selection methods can be broadly categorized into filtering models (Yu and Liu, 2003) and wrapper models (Kohave and John, 1997). Filtering methods separate the feature selection from the learning process, whereas wrapper methods combine them. The main drawback of wrapper methods is their computational inefficiency.

There are three kinds of popular filtering methods. In (Sun and Li, 2006) a margin-based method is proposed as a feature-weighting algorithm that is a new interpretation of a RELIEF-based method (Gilad-Bachrach et al. 2004). The

method is an online algorithm that solves a convex optimization problem with a margin-based objective function. Markov Blanket-based methods perform feature selection by searching an optimal set of features using Markov Blanket approximation. The method proposed at (Lou and Obradovic, 2010) removed the feature whose Markov Blanket can be found in the rest of features. Dependence estimation-based methods use the Hilbert-Schmide Independence Criterion as a measure of dependence between the features and the labels (Song et al. 2007). The key idea in this method is that good features should maximize such dependence. However, all these methods assume that the data is complete without missing values.

Several classification methods have been proposed recently to handle the missing values directly, without imputing missing values in advance. A method was presented for incorporating second order uncertainties about the samples while keeping the classification problem convex in the presence of missing values (Pannagadatta et al. 2006). A method is presented to handle incomplete data where the missing features are structurally absent for some of the instances (Chechik et al. 2008). Instances are considered as sets of (feature value) pairs that naturally handle the missing value case (Grangier and Melvin, 2010). However, all of these are classification methods rather than feature selection methods and they are not applicable to high dimensional data with a large number of irrelevant features, since they are classifying on whole dimensional data instead of informative low dimensional data. In contrast, our method can handle high dimensional incomplete data by selecting informative features directly, without estimating the missing values in a pre-processing stage.

The Proposed Method

Let $D = \{(x_n, y_n \mid n = 1,..., N)\} \subset \Re^M \times \pm 1$ be the data set with N instances and M features. For a given instance $\mathbf{x}_n$, let $\mathbf{I}_n$ be the index function indicating whether features in $\mathbf{x}_n$ are missing or not. Specifically, $\mathbf{I}_n$ is defined as

$$I_n(j) = \begin{cases} 0 & x_n(j) \text{ is missing} \\ 1 & \text{otherwise} \end{cases} \quad where\ j = 1,2,...M \tag{1}$$

We will first define the uncertainty margin for each instance $\mathbf{x}_n$, and then present the uncertainty margin-based objective function as well as the algorithm for solving the corresponding optimization problem.

Uncertainty Margin

Given an instance, the margin of a hypothesis is the distance between the hypothesis and the closest hypothesis that assigns an alternative label. For a given instance $\mathbf{x}_n$, we find two nearest neighbors for $\mathbf{x}_n$, one with the same class label (called ***nearhit***), and the other with different class label (called ***nearmiss***). The hypothesis-margin of a given instance $\mathbf{x}_n$ in data set D is defined as:

$$L_D(\mathbf{x}_n) = \frac{1}{2}(\| \mathbf{x}_n - \text{nearmiss}(\mathbf{x}_n) \| - \| \mathbf{x}_n - \text{nearhit}(\mathbf{x}_n) \|) \tag{2}$$

In margin-based feature selection, we scale the feature by assigning a non-negative weight vector ***w***, and then choose the features with large weights that maximize the margin. One idea is to then calculate the margin in weighted feature space rather than the original feature space, since the nearest neighbor in the original feature space can be completely different from the one in the weighted feature space. Therefore, we define the instance margin for each instance $\mathbf{x}_n$ from D in a weighted feature space as:

$$\begin{aligned} \rho_D(\mathbf{x}_n \mid \mathbf{w}) = & d(\mathbf{x}_n, \text{nearmiss}(\mathbf{x}_n) \mid \mathbf{w}) \\ & - d(\mathbf{x}_n, \text{nearhit}(\mathbf{x}_n) \mid \mathbf{w}) \end{aligned} \tag{3}$$

where d(.) is a distance function. Although one can apply any kind of distance function, for the purpose of our study, we apply the Manhattan distance. Therefore, the above definition can be written as $\rho_D(\mathbf{x}_n \mid \mathbf{w}) = \mathbf{w}^T \beta_n$,
where $\beta_n = \mid \mathbf{x}_n - \text{nearmiss}(\mathbf{x}_n) \mid - \mid \mathbf{x}_n - \text{nearhit}(\mathbf{x}_n) \mid$, and $\mid \cdot \mid$ is the element-wise absolute operator.

In an incomplete data set, we cannot apply a uniform weight ***w*** to each instance to get the margin since each $\mathbf{x}_n$ has different missing values. We need to maintain a weight vector $\mathbf{w}_n$ for each instance $\mathbf{x}_n$, which is defined as $\mathbf{w}_n = \mathbf{w} \circ \mathbf{I}_n$, where $\mathbf{I}_n$ is the pre-defined indicative index for each instance $\mathbf{x}_n$ and $\circ$ is the element-wise product.

In order to take into account the uncertainty due to different values in each instance, for each $\mathbf{x}_n$, we define a scaling coefficient $\mathbf{s}_n = \| \mathbf{w}_n \|_1 / \| \mathbf{w} \|_1$. Therefore, the instance-based margin can be written as:

$$\begin{aligned} \rho_D(\mathbf{x}_n \mid \mathbf{w}_n, \mathbf{s}_n) &= d(\mathbf{x}_n, \text{nearmiss}(\mathbf{x}_n) \mid \mathbf{w}_n, \mathbf{s}_n) \\ &\quad - d(\mathbf{x}_n, \text{nearhit}(\mathbf{x}_n) \mid \mathbf{w}_n, \mathbf{s}_n) \\ &= \mathbf{s}_n {\mathbf{w}_n}^T \beta_n \end{aligned} \tag{4}$$

After applying the scaling coefficient $\mathbf{s}_n$, we decrease the instance margin for $\mathbf{x}_n$, which has a huge number of missing values. Another important aspect affected by missing values is the calculation of nearest neighbors for each $\mathbf{x}_n$. Due to the missing values, we cannot tell exactly which one is the nearest neighbor for $\mathbf{x}_n$. Therefore, extending the definition in (Sun et al. 2009) by taking into account the affects of missing values, we calculate the uncertainty of each instance being the nearest neighbor of $\mathbf{x}_n$. The uncertainty is evaluated by standard Gaussian kernel estimation with kernel width of σ. Specifically, we define the uncertainty that an instance $\mathbf{x}_i$ with the same class label as $\mathbf{x}_n$ can be the nearest hit neighbor of $\mathbf{x}_n$ as:

$$\begin{aligned} U_{\text{nearhit}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}_n, \mathbf{I}_n) &= \frac{\exp(d(\mathbf{x}_n, \mathbf{x}_i \mid \mathbf{w}_n, \mathbf{I}_n)/\sigma)}{\sum_j \exp(d(\mathbf{x}_n, \mathbf{x}_j \mid \mathbf{w}_n, \mathbf{I}_n)/\sigma)} \\ &\text{where } 1 \le i \le N, i \ne n, y_i = y_n \\ &\text{and } 1 \le j \le N, y_j = y_n \end{aligned} \tag{5}$$

Similarly, the uncertainty that an instance x_i with a different class label from $\mathbf{x}_n$ can be the nearest miss neighbor of $\mathbf{x}_n$ is defined as:

$$U_{\text{nearmiss}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}_n, \mathbf{I}_n) = \frac{\exp(d(\mathbf{x}_n, \mathbf{x}_i \mid \mathbf{w}_n, \mathbf{I}_n)/\sigma)}{\sum_j \exp(d(\mathbf{x}_n, \mathbf{x}_j \mid \mathbf{w}_n, \mathbf{I}_n)/\sigma)} \quad \text{where } 1 \le i \le N, y_i \ne y_n \text{ and } 1 \le j \le N, y_j \ne y_n \tag{6}$$

Please note that $\| \mathbf{x}_n - \mathbf{x}_i \|_{\mathbf{w}_n}$ in equations (5) and (6) denotes the distance between $\mathbf{x}_n$ and $\mathbf{x}_i$ in weighted space determined by $\mathbf{x}_n$'s weight vector $\mathbf{w}_n$. Finally, by checking the uncertainty of each instance to be the nearest neighbor of $\mathbf{x}_n$, we define our **uncertainty margin** as the expectation of the instance margin of $\mathbf{x}_n$, which can be written as:

$$E_{\rho_n}(\mathbf{x}_n \mid \mathbf{w}_n, \mathbf{s}_n) = \mathbf{s}_n \mathbf{w}_n{}^T \mathbf{E}_{\beta_n} \quad \text{where } \mathbf{E}_{\beta_n} = \sum_{i, when\ y_i \ne y_n} U_{\text{nearmiss}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}_n) \cdot |\mathbf{x}_n - \mathbf{x}_i| - \sum_{i, when\ y_i = y_n} U_{\text{nearhit}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}_n) \cdot |\mathbf{x}_n - \mathbf{x}_i| \tag{7}$$

As we mentioned before, our uncertainty margin incorporates the uncertainty due to the missing values in each instance ($\mathbf{s}_n$), and the uncertainty in calculating two nearest neighbors ($E_{\beta n}$). We maintain a weight vector $\mathbf{w}_n$ for each instance $\mathbf{x}_n$ such that our defined uncertainty margin can handle incomplete data directly.

Optimization Based on Uncertainty Margin

We define the uncertainty margin of the entire data D as the sum of instance margins, which can be written as:

$$E_{\rho_D} = \sum_{n=1}^{N} E_{\rho_n}(\mathbf{x}_n \mid \mathbf{w}_n, \mathbf{s}_n) \tag{8}$$

The feature weights can be learned by solving an optimization problem that maximizes the uncertainty margin of data D. This optimization problem can be represented as:

$$\max_{\mathbf{w}} \sum_{n=1}^{N} E_{\rho_n}(\mathbf{x}_n \mid \mathbf{w}_n, \mathbf{s}_n) \quad \text{subject to } \mathbf{w} \ge 0 \tag{9}$$

We followed logistic regression formulation framework. In order to avoid huge values in weight vector ***w***, we add a normalization condition $\| \mathbf{w} \|_1 \le \theta$. Given this condition, for each instance $\mathbf{x}_n$ with missing values, the weight vector $\mathbf{w}_n$ satisfies $\| \mathbf{w}_n \|_1 \le \| \mathbf{w} \|_1, \ \forall n = 1, 2, ...N$. Therefore, we can rewrite the optimization problem as:

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \log(1 + \exp(-E_{\rho_n}(\mathbf{x}_n \mid \mathbf{w}_n, \mathbf{s}_n)) \ \text{ subject to } \ \mathbf{w} \ge 0, \| \mathbf{w} \|_1 \le \theta \tag{10}$$

The above formulation is an optimization problem with respect to $\mathbf{w}_n$. It cannot be solved since there is a different $\mathbf{w}_n$ for each instance $\mathbf{x}_n$. Using pre-defined $\mathbf{w}_n$, we can rewrite the formulation with respect to ***w***. The optimization formulation (10) can also be written as:

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \log(1 + \exp(-\mathbf{s}_n \mathbf{w} \mathbf{E}_{\beta_n} \circ \mathbf{I}_n) \quad \text{subject to } \mathbf{w} \ge 0, \| \mathbf{w} \|_1 \le \theta \tag{11}$$

The above formulation is called nonnegative garrote. We can rewrite the formulation (11) as:

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \log(1 + \exp(-\mathbf{s}_n \mathbf{w} \mathbf{E}_{\beta_n} \circ \mathbf{I}_n) + \lambda \| \mathbf{w} \|_1 \quad \text{subject to } \mathbf{w} \ge 0 \tag{12}$$

For each solution to (12), there is a parameter θ, corresponding to the obtained λ in (12), which gives the same solution in (14). Formulation (12) is actually the optimization problem with ℓ_1 regularization. The benefits of adding the ℓ_1 penalty have been well studied (Rosset, 2005) and it is shown that the ℓ_1 penalty can effectively handle sparse data and huge amounts of irrelevant features.

Algorithm for Learning Feature Weights

In this section we will introduce our feature selection method which solves the optimization problem introduced in Section 3.2. As we can see from (12), the optimization problem is convex if $E_{\beta n}$ is fixed. Fox a fixed $E_{\beta n}$, (12) is a constrained convex optimization problem. However, it cannot be directly solved by gradient descent because of the nonnegative constraints on ***w***. To handle this problem, we introduce a mapping function:

$$f: \mathbf{w} \to \mathbf{u}, \text{ where } \mathbf{w}(i) = \mathbf{u}(i)^2, \ \forall i = 1, 2, ...M \tag{13}$$

Therefore, the formulation (12) can be rewritten as:

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \log(1 + \exp(-\mathbf{s}_n \mathbf{w} \mathbf{E}_{\beta_n} \circ \mathbf{I}_n) + \lambda \| \mathbf{u} \|_2{}^2 \tag{14}$$

By taking the derivative with respect to ***u***, we obtain the following updated rule for ***u***:

$$\mathbf{u}^{(new)} = \mathbf{u}^{(old)} - \alpha(\lambda - \frac{\sum_{n=1}^{N} \exp(-\mathbf{s}_n \sum_{j=1}^{M} \mathbf{u}_j{}^2 \mathbf{E}_{\beta_n} \circ \mathbf{I}_n(j))}{1 + \sum_{n=1}^{N} \exp(-\mathbf{s}_n \sum_{j=1}^{M} \mathbf{u}_j{}^2 \mathbf{E}_{\beta_n} \circ \mathbf{I}_n(j))}) \otimes \mathbf{u} \tag{15}$$

where α is learning rate and $\otimes$ is the Hadamard product.

However, $E_{\beta n}$ is determined by ***w*** so that (14) is not a convex problem. We use a fixed-point EM algorithm to find the optimal ***w***. The proposed algorithm for Margin-based Feature **Selection** in **Incomplete data** (we call it **SID**) is shown in Table 1.

The SID algorithm starts with initializing the values of w to be 1. With such initialization we can estimate the $\mathbf{s}_n$ and $E_{\beta n}$ for each instance $\mathbf{x}_n$. Then, in each iteration, the weights vector w is updated by solving the optimization problem (14) with estimated values of $\mathbf{s}_n$ and $E_{\beta n}$ in the previous iteration. We repeat the iteration until convergence. The SID algorithm requires pre-defined kernel width σ and a regularization parameter λ. We applied cross validation to select the values of parameters.

TABLE I. SID FEATURE SELECTION METHOD

Input:	data set D $\{(x_n, y_n)\}$ Indicate index I_n for each x_n kernel width σ regularization parameter λ
Output:	feature weights *w*
Initialization:	set $w^{(0)}$ 1, t 1
Do	
	Calculate scaling coefficient $s_n^{(t)}$ $\lvert w_n^{(t-1)}\rvert/\lvert w^{(t-1)}\rvert$
	Calculate $E_{\beta n}^{(t)}$ using $w^{(t-1)}$ and equation (7)
	Update $\boldsymbol{u}^{(t)}$ using updated rule in equation (15)
	Update $w^{(t)}$ using $\boldsymbol{u}^{(t)}$ using equation (13)
	t t + 1
Until convergence	

To prove convergence of SID algorithm we will use the following theorem.

Theorem 1 (Contraction Mapping Theorem). *Let T: X→X be a contraction mapping on a complete metric space X. The sequence generated by $x_n=T(x_{n-1})$ for n = 1, 2,3,…converges to unique limit x*, where x* is the fixed point of T (T(x*)=x*). In other words, there is a nonnega tive real number r<1 such that*

$$d(x^*, x_{n+1}) \le \frac{r^n}{1-r} d(x_1, x_0).$$

Proof: See (Kirk and Sims, 2001).

Based on this theorem we prove the following:

Theorem 2. *There exists σ_0 such that for any $\sigma > \sigma_0$ the SID algorithm converges to a fixed unique solution $\boldsymbol{w}^*$ when initial feature weights $\boldsymbol{w}^{(0)}$ are nonnegative.*

Proof sketch: Let **U** and **W** be sets of all possible uncertainty values defined in (8) and (9), and all possible feature weights values defined in (13), respectively. Specifically, we defined $\mathbf{U} = \{u \mid u = (u_{\text{nearhit}}(\mathbf{x}_i \mid \mathbf{x}_n), u_{\text{nearmiss}}(\mathbf{x}_i \mid \mathbf{x}_n))\}$ and $\mathbf{W} = \{\mathbf{w} \mid \mathbf{w} \in \Re^M, \| \mathbf{w} \| \le \theta, \mathbf{w} \ge 0\}$. Obviously, $\Re^M$ is a finite dimensional Banach space (complete normed vector space), and **W** is a closed subset of $\Re^M$. Therefore, **W** is a complete metric space (Kress, 1998, Sun et al. 2009).

The first step of SID algorithm calculates the uncertainty based on current feature weights and missing index, which can be represented by function F_1: **W**→**U**, where $F_1(\mathbf{w}^{(t-1)} \mid \{\mathbf{I}_n, \mathbf{s}_n\}_{n\ 1,2,\ldots,N}) = u^{(t)}$. The second step updates feature weights using current uncertainty, and can be represented by function F_2: **U**→**W**, where $F_2(u^{(t)} \mid \{\mathbf{I}_n, \mathbf{s}_n\}_{n\ 1,2,\ldots,N}) = \mathbf{w}^{(t)}$. Therefore, our SID algorithm can be represented as $\mathbf{w}^{(t)} = F_2(F_1(\mathbf{w}^{(t-1)})) = T(\mathbf{w}^{(t-1)})$, where T is the composition of function F_2 and F_1. Note that T is a function mapping a complete metric space **W** to itself. When $\sigma \to +\infty$, we have $\lim_{\sigma \to +\infty} \| T(\mathbf{w}_1, \sigma) - T(\mathbf{w}_2, \sigma) \| = 0$ for each $\mathbf{w}_1$ and $\mathbf{w}_2$. We can rewrite it for each $\mathbf{w}_1$ and $\mathbf{w}_2$ as: $\lim_{\sigma \to +\infty} \| T(\mathbf{w}_1, \sigma) - T(\mathbf{w}_2, \sigma) \| \le \frac{r}{1-r} \| T(\mathbf{w}_1, \sigma) - T(\mathbf{w}_0, \sigma) \|$ *where* $r = 0$.

Therefore, T is a contraction mapping in the limit of σ. We regard r as a function of σ and have $r(\sigma)=0$ when $\sigma \to +\infty$. So, for each $\xi>0$, there exists a σ_0 such that $r(\sigma) < \xi \ \forall\ \sigma > \sigma_0$. Therefore, for $\xi<1$, T is a contraction mapping. Consequently, based on Theorem 1 it follows that T converges to a unique fixed point. □

The complexity of the SID algorithm is $O(TN^2M)$ where T is the total number of iterations, *N* is the number of instances, and *M* is the number of features. Our experimental results show that the algorithm converges in a small number of iterations (less than 25). Therefore, the complexity of SID algorithm in real application is about $O(N^2M)$. Note that the SID algorithm is linear to the number of features, such that the proposed method can handle a huge number of features.

Experiments

To characterize the proposed algorithm, we conducted large-scale experiments on both synthetic and UCI benchmark data sets. All experiments of this study were performed on a PC with 3 GB of memory. We compared our proposed SID algorithm in incomplete data with three traditional margin-based feature selection methods (the method proposed in (Sun et al. 2009) that we call LBFS, Simba (Gilad-Bachrach et al. 2004) and Relief (Kira and Rendell, 1992)) based on applying the following three popular imputation methods (Chechik et al. 2008) in a pre-processing stage of three alternatives to estimate the missing values:

Mean. Missing values are estimated as the average value of the feature over all data (training + testing sets).

kNN. Missing values are estimated as the mean values obtained from *K* nearest neighbors. The number of neighbors is varied from *K*=1,5,10 and the best result is shown.

EM. A Gaussian mixture model is learned by iterating between learning the model with imputed data and re-imputing missing values with the model learned in the previous iteration. We apply the algorithm proposed in (Ghahramani and Jordan, 1994).

Results on Synthetic Data

Synthetic data experiments were designed to evaluate the ability of our SID algorithm to select relevant features in incomplete data in the presence of a large number of irrelevant variables. For this, 500 instances in 100 dimensional space were generated where two features define a *xor* function while the remaining 98 features were irrelevant sampled independently from a zero mean and one standard deviation normal distribution.

For simplicity, in experiments on synthetic data we compare only with LBFS (Sun et al. 2009). The number of irrelevant features selected together with both relevant features is compared when using SID and three alternatives methods. The methods are compared when 5% to 65% of data were missing randomly in each feature. In feature selection experiments with 5% of missing values SID and

feature selection based on EM and mean imputation worked equally well, selecting only two relevant features (see results at Fig. 1). However, the kNN based method had problems in computing nearest neighbors even with such a small number of missing values in the presence of a huge number of irrelevant features. When a large fraction of the data was missing, SID clearly outperformed the alternatives. In particular, in the presence of 35% of missing values in two relevant variables SID was still selecting only two relevant variables, while to capture these two variables alternative methods were also selecting 2 to 12 irrelevant variables on average. All methods performed badly when extremely large fractions of data were missing (>50%), but SID was still a better choice than the alternatives. The square mark on each line in Fig. 1 indicates the position from which the result of each method becomes unstable resulting in a large variance and high chance of selecting random features. As shown at Fig. 1, the SID method becomes unstable much later than the alternatives.

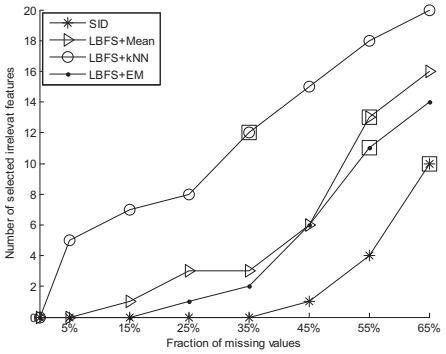

Figure 1: The number of irrelevant features selected together with two relevant features. Our method (SID) vs. LBFS applying three different estimation methods (Square on each line indicates the fraction of missing values where a particular feature selection algorithm has large variance in the number of features selected, and becomes unstable.)

Results on Benchmark Data

In this section we present the results on 6 benchmark data sets called *Wpbc*, *Splice*, *USPS*, *MNIST*, *DLBCL*, and *Arecene*. The properties of these data sets are summarized in Table 2. We perform binary classification on all data sets. For the multi-class data sets (*USPS*, *MNIST*), we converted the original 10-class problem to binary by setting digits 3, 6, 8, 9, 0 (round digits) as one class, and digits 1, 2, 4, 5, 7 (non-round digits) as the other class. For data with a small number of features (*Wpbc* and *Splice*), we added 2000 irrelevant features independently sampled from a Gaussian distribution with 0-mean and 1-variance.

Unlike the synthetic data from the previous section, in these experiments we didn't know the optimal features for all benchmark data, as there might be some irrelevant and weakly relevant features in the data. To evaluate the quality of selected features selected by different methods, we trained a SVM on selected features and tested the classification error on the selected feature space. We trained the same SVM with a Gaussian kernel on the features selected by different methods. The kernel width of SVM Gaussian was set to be the median distance between points in the sample. We applied 5-cross validations on data sets with more than 500 instances, and leave-one-out procedure on data sets with less than 500 instances.

TABLE II. SUMMARY OF BENCHMARK DATA SETS.

Dataset	Feature	Instance	Class
Wpbc	33+2000	194	2
Splice	60+2000	1655	2
USPS	256	7291	10
MNIST	484	5000	10
DLBCL	5469	77	2
Arecene	10000	100	2

The classification errors of SID are compared to those of LFSB, SIMBA and Relief with respect to their accuracy for different fractions of missing values on benchmark data. These results for the Mean-based imputations in LFSB, SIMBA and Relief are reported at Fig. 2 where the three alternatives are labeled as LFSB-mean, SIMBA-mean and Relief-mean. In all comparisons, parameters in the SID method were fixed to kernel width $\sigma = 1$ and regularization parameter $\lambda = 1$. Similarly, in Fig. 3 and Fig. 4 the results of SID are compared to three alternatives based on kNN and EM imputation.

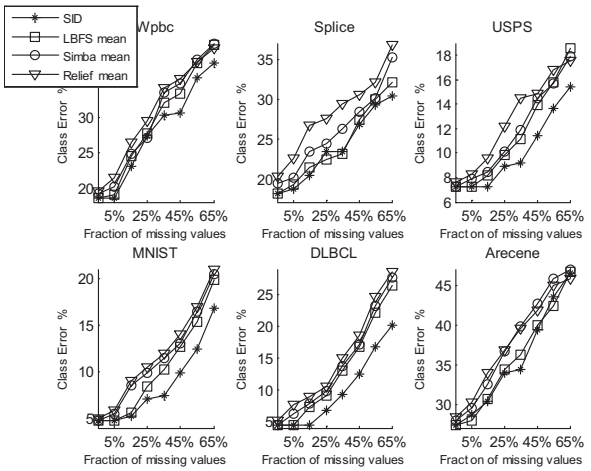

Figure 2. Classification error with respect to fraction of missing values by SID compared to three alternative feature selection methods that used **Mean** to perform data imputation

The results summarized at Fig. 2 and Fig. 3 provide evidence that **kNN** and **EM** methods for data imputation didn't work well on *Wpbc* and *Splice* for feature selection even when the data had a small fraction of missing values. The

reason is that 2000 completely irrelevant features were added to these two data sets. In a feature space with so many irrelevant features, nearest neighbors can be completely different from the nearest neighbors in the original feature space. **EM** estimated the missing values by exploiting the correlation among instances. However, instances with high correlation in the original feature space can be almost independent, as evident from Fig. 4 in the experiments where 2000 completely independent irrelevant features were present.

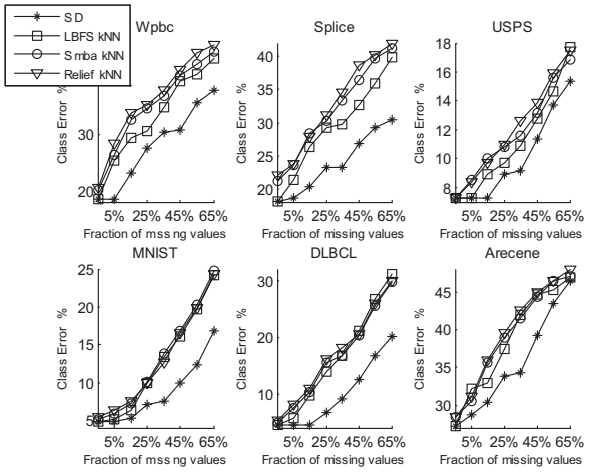

Figure 3 Classification error with respect to fraction of missing values by SID compared to three alternative feature selection methods that used **KNN** to perform data imputation

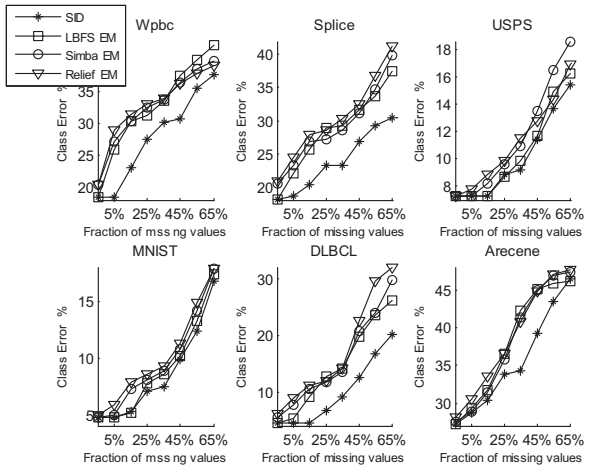

Figure 4. Classification error with respect to fraction of missing values by SID compared to three alternative feature selection methods that used **EM** to perform data imputation

Feature selections based on **kNN** and **EM** imputation were good on *USPS* and *WNIST* data, which have a small number of irrelevant features (see Fig. 2 and Fig. 3). However, these methods failed on *DLBCL* and *Arecene,* as most features in *these datasets* are irrelevant.

Fig. 2 shows that, similar to **Mean,** our **SID** was not sensitive to the number of irrelevant features. The **Mean** method estimated missing values for each feature by the observed values in the same values, so that irrelevant features did not affect estimation of the missing values. Therefore, the feature selection based on the **Mean** method is not sensitive to irrelevant features. Our proposed **SID** measured the distance in weighted feature space together by taking into account the uncertainty due to the missing values. It can correctly capture the nearest neighbors even in highly irrelevant feature space. The results shown at Fig. 2, Fig. 3 and Fig. 4 also provide evidence that **SID** method outperformed alternatives in all data sets for different fractions of missing values.

Number of selected features. Our **SID** method can automatically select optimal feature set by eliminating features with weight zero. **SID** selected 18 out of 2033 features on *Wpbc*, 32 out of 2060 features on *Splice*, 13 out of 256 features on *USPS*, 28 out 484 features on *MNIST*, 35 out of 5469 features on *DLBCL*, and 59 out of 10000 features on *Arecene*. However, **LBFS**, **Simba** and **Relief** cannot select optimal feature set automatically, since they are all feature ranking method. In all experiments, we let three alternatives select the same number as **SID** selected on each data.

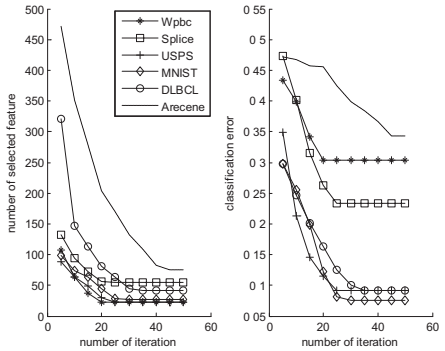

Figure 5. Convergence analysis. The number of selected features and classification error with respect to the number of iterations are shown at the left and the right panel, respectively.

Analysis of Convergence

To simplify convergence experiments we fixed the rate of missing values in each data set at 35%. For each data set, the number of features selected by **SID** at every 5 iterations is shown at the left side of Fig. 5. We can see that **SID** converged quickly on each data set (**SID** converged in 45 iterations on *Arecene* data, and in about 30 iterations on other data). The obtained results provide evidence that our method is applicable to large-scale data.

The classification error of **SID** on each data set at every 5 iterations until convergence is shown at the right side of

Fig. 5. Our method converged on all data sets in a small number of iterations (45 iterations on *Arecene* data and about 30 iterations on other data).

Conclusion

The proposed SID method performs feature selection directly from incomplete data, without applying an imputation method to estimate the missing values in advance. In SID, the objective function is formulated by taking into account the uncertainty of the instance due to the missing values. The weight for each feature is obtained by solving the revised optimization problem using an EM algorithm. Experimental results provide evidence that our method outperforms the alternative feature selection methods that require a data imputation step in a data pre-processing stage.

Acknowledgments

This work was supported in part by the National Science Foundation under Grant IIS-1117433 and also by DARPA grant DARPA-N66001-11-1-4183 negotiated by SSC Pacific (to ZO).

References

Chechik, G., Heitz, G., Elidan, G., Abbeel, P. and Koller, D. 2008. Max margin Classification of Data with Absent Features. *Journal of Machine Learning Research*. vol 9, pp. 1 21.

Ghahramani, Z. and Jordan, M. I. 1994. Supervised Learning From Incomplete Data via an EM Approach. *Advances in Neural Information Processing Systems*, vol 6, pp. 120 127.

Gilad Bachrach, R., Freund, Y., Bartlett, P. L. and Lee, W. S. 2004. Margin Based Feature Selecgtion theory and algorithms. In 21st *International Conference on Machine Learning*, pp. 43 50.

Grangier, D. and Melvin, I. 2010. Feature Set Embedding for In complete Data. In 24th *Annual Conference on Neural Information Processing Systems*.

Guyon, I. and Elisseeff A., 2000. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, vol 3, pp. 1157 1182.

Kira, K., and Rendell, L. 1992. A Practical Approach to Feature Selection. In *9th International Wrokshop on Machine Learning*, pp. 249 256.

Kirk, William A. and Sims, Brailey. 2001. Handbook of Metric Fixed Point Theroy. *Kluwer Academic, London.*

Kohave, R. and John, G. 1997. Wrappers for Feature Subset Se lection. *Artificial Intelligence*, vol 1 2, pp. 273 324.

Koller, D. and Sahami, M. 1996. Toward Optimal Feature Selec tion. In *International Conference on Machine Learning*, pp. 284 292 .

Kress, R. 1998. Numerical Analysis. *New York:springger Verlag.*

Liew, A., Law, N. and Yan, H. 2010. Missing Value Imputation for Gene Expression data: Computational Techniques to Recover Missing Data From Available Information. *Briefings in Bioinformatics.*

Lou, Q. and Obradovic, Z. 2011. Modeling Multivariate Spatio Temporal Remote Sensing Data with Large Gaps. In 22nd *International Joint Conference on Artificial Intelligence.*

Lou, Q. and Obradovic, Z. 2010. Feature Selection by Approxi mating the Markov Blanket in a Kernel Induced Space. *European Conference on Artificial Intelligence*.

Pannagadatta, S. K. Bhattacharyya, C. and Smola, A. J. 2006. Se cond Order Cone Programming Approaches for Handling Miss ing and Uncertain Data. *Journal of Machine Learning Research*, vol 7, pp. 1283 1314.

Radosavljevic, V., Vucetic, S. and Obradovic, Z. 2010. A Data Mining Technique for Aerosol Retrieval Across Multiple Accura cy Measures. *IEEE Geo science and Remote Sensing Lettres*, vol 7, pp. 411 415.

Rosset, S. 2005. Following Curved Regularized Optimization Solution Paths. In 17th *Advanced Neural Information Processing Systems*, pp. 1153 1160.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. 1998. Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *Annals of Statistics*.

Song, L., Smola, A., Gretton, A., Borgwardt, K. M., and Bedo, J. 2007. Supervised Feature Selection via Dependence Estimation. *International Conference on Machine Learning*. pp. 856 863.

Sun, J. and Li, J. 2006. Iterative RELIEF for Feature Weighting. In 23rd *International Conference on Machine Learning, Pittsburgh*.

Sun, Y., Todorovic, S. and Goodison, S. 2009. Local Learning Based Feature Selection for High Dimensional Data Analysis. *IEEE Transactions on Pattern Analysis and Machine Learning.*

Yu, L., and Liu, H. 2003. Feature Selection for High dimensional Data, A Fast Correlation based Filter Solution. In 20th *International Conference on Machine Learning*, pp. 856 863.