LEARNING FROM INCOMPLETE HIGH-DIMENSIONAL DATA

A Dissertation
Submitted
to the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

By
Qiang Lou
January, 2013

Examining Committee Members:

Dr Zoran Obradovic, Advisory Chair, Computer and Information Science
Dr Slobodan Vucetic, Computer and Information Science
Dr Longin Jan Latecki, Computer and Information Science
Dr Adam Davey, External Member, Department of Public Health

ABSTRACT

Data sets with irrelevant and redundant features and large fraction of missing values are common in the real life application. Learning such data usually requires some preprocess such as selecting informative features and imputing missing values based on observed data. These processes can provide more accurate and more efficient prediction as well as better understanding of the data distribution. In my dissertation I will describe my work in both of these aspects and also my following up work on feature selection in incomplete dataset without imputing missing values. In the last part of my dissertation, I will present my current work on more challenging situation where high-dimensional data is time-involving.

The first two parts of my dissertation consist of my methods that focus on handling such data in a straightforward way: imputing missing values first, and then applying traditional feature selection method to select informative features. We proposed two novel methods, one for imputing missing values and the other one for selecting informative features. We proposed a new method that imputes the missing attributes by exploiting temporal correlation of attributes, correlations among multiple attributes collected at the same time and space, and spatial correlations among attributes from multiple sources. The proposed feature selection method aims to find a minimum subset of the most informative variables for classification/regression by efficiently approximating the Markov Blanket which is a set of variables that can shield a certain variable from the target.

I present, in the third part, how to perform feature selection in incomplete high-dimensional data without imputation, since imputation methods only work well when data is missing completely at random, when fraction of missing values is small, or when there is prior knowledge about the data distribution. We define the objective function of

the uncertainty margin-based feature selection method to maximize each instance's uncertainty margin in its own relevant subspace. In optimization, we take into account the uncertainty of each instance due to the missing values. The experimental 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 alternative 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 outperforms the alternatives, which impute missing values first.

In the fourth part, I introduce my method handling more challenging situation where the high-dimensional data varies in time. Existing way to handle such data is to flatten temporal data into single static data matrix, and then applying traditional feature selection method. In order to keep the dynamics in the time series data, our method avoid flattening the data in advance. We propose a way to measure the distance between multivariate temporal data from two instances. Based on this distance, we define the new objective function based on the temporal margin of each data instance. A fixed-point gradient descent method is proposed to solve the formulated objective function to learn the optimal feature weights. The experimental results on real temporal microarray data provide evidence that the proposed method can identify more informative features than the alternatives that flatten the temporal data in advance.

ACKNOWLEDGMENTS

Foremost, I would like to thank my advisor Dr. Zoran Obradovic for his continuous support of my Ph.D study and research, for his patience, motivation, and enthusiasm. His guidance helped me in all the time of my Ph.D study and research.

My sincere thanks also goes to the rest of my thesis committee: Dr. Adam Davey, Dr. Longin Jan Latecki, and Dr. Slobodan Vucetic, for their insightful comments, encouragement, and hard questions.

I thank my fellow labmates in our research lab, for all the fun we had in the past five years.

I thank my parents Xiaohua Liu and Yougen Lou, my brother Gang Lou, for their support throughout my life.

Last but not the least, I would like to thank my wife Qiong, for her support, encouragement, understanding, patience and unwavering love in the past eight years. I also would like to thank my son Eric. The little one is the motivation and the source of power of all my hard work.

I lovingly dedicate this thesis to my wife, Qiong Wei, who supported and encouraged me each step of the way.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

FEATURE SELECTION

Selecting appropriate features is an important step in the data mining process whose objectives include providing more accurate and more efficient prediction as well as better understanding of data distribution. Feature selection approaches can be broadly categorized into a wrapper model [2,1,3] and a filter model [5,4]. The wrapper model combines the learning method and feature selection method, which is computationally expensive and is often impractical for datasets with a large number of features. The filter model separates the feature selection from learning process such that the results of the feature selection step are independent of the learning algorithm and are used for model learning in a follow up step.

The proposed feature selection method aims to find a minimum subset of the most informative variables for classification/regression by efficiently approximating the Markov Blanket which is a set of variables that can shield a certain variable from the target. Instead of relying on the conditional independence test or network structure learning, the new method uses Hilbert-Schmidt Independence criterion as a measure of dependence among variables in a kernel-induced space. This allows effective approximation of the Markov Blanket that consists of multiple dependent features rather than being limited to a single feature. In addition, the new method can remove both irrelevant and redundant features at the same time. This method for discovering the Markov Blanket is applicable to both discrete and continuous variables, whereas previous methods cannot be used directly for continuous features and therefore are not applicable to regression problems. Experimental evaluations on synthetic and benchmark classification and regression datasets provide evidence that the new feature selection method can remove useless variables in low and in high dimensional problems more accurately than existing Markov Blanket based alternatives.

1.1 Introduction of Feature Selection

Existing test-based Markov Blanket feature selection methods [8,9] are all using a 'Growing - Shrinking' (GS) [7] approach for discovering the Markov Blanket. In the growing phase of this approach, all features belonging to the Markov Blanket and possibly some false features enter the Markov Blanket. Then, in the shrinking phase, all features in the current Markov Blanket are checked again to remove the false features introduced at the growing phase. In both phases, conditional independence testing is used to judge if a feature belongs to the Markov Blanket or not. However, such conditional independence test-based method requires that the sample has a large number of instances to ensure the reliability of the independence test. Another limitation of test-based feature selection algorithms is that they are usually too aggressive in removing features [10].

In a structure learning-based method, a heuristic Bayesian network structure learning is performed and then the Markov Blanket is discovered corresponding to the learned structure [10]. In such an approach, to restrict search space, two heuristics (called 'sparse candidate' and 'screen-based') are proposed for selecting the promising candidates.

However, the Bayesian network structure is learned using heuristic methods as the optimization here is a very hard problem. These heuristics combine locally optimal structures, which results in the learned structure that is not a global optimal solution. An additional limitation of such approaches is that learning network structure could be computationally prohibitively expensive in the presence of a larger number of features.

In an approximate Markov Blanket method called FCBF, the redundant features were eliminated in a potentially relevant subset obtained by excluding the irrelevant features based on the correlation to the target variable [11]. In this approach, symmetrical uncertainty is used to measure the relation between variables. For a pair of features, FCBF measures their symmetrical uncertainty and also the symmetrical uncertainty between either of them and target variable. If the measured value between these two variables is bigger than the measured value between one of them and the target variable, the variable with larger symmetrical uncertainty to the target is regarded as the Markov Blanket of the other variable which is removed. FCBF assumes the Markov Blanket of a feature has only one feature, since it is based on pairwise comparison. Such an approach is often too restrictive in practical situations as illustrated in the results section of this article. In addition we found that FCBF is too aggressive in eliminating features, since it gives too much priority to dominant features.

1.2. Identification of Markov Blanket Candidates

Let F be the whole set of features. The Markov Blanket MB_i of feature F_i ($MB_i \subset F$ ($F_i \notin MB_i$)) is the set of features with a property that F_i is conditionally independent of the remaining features U and the target C. More formally, set MB_i is called the Markov Blanket of F_i iff:

$$P(U, C | F_i, MB_i) = P(U, C | MB_i)$$
$$\text{where} : U = F - \{F_i\} - MB_i$$

If MB_i is the Markov Blanket of F_i, then the prediction model learned without considering F_i is as accurate as the model learned using all features F. It is often difficult to find the exact Markov Blanket for a given feature. To address this problem we propose a novel method of finding an approximate Markov Blanket for F_i. We then use this method to develop a feature selection algorithm based on the discovered approximate Markov Blanket.

Given a set of features we can easily check if it is the Markov Blanket MB_i of feature F_i. However, evaluating all subsets of F for this property is prohibitively costly. Instead of searching for the exact Markov Blanket for a feature, in practice it is appropriate to determine an approximate Markov Blanket that can be used for removing this feature with little useful information lost. Intuitively, if MB_i is the Markov Blanket for feature F_i, the features in MB_i are more dependent to F_i than those features which are not in MB_i [12]. Therefore, we can choose a subset of *k* features which are strongly dependent to F_i as the candidate Markov Blanket of F_i. Then, for each feature, we only need to evaluate its candidate Markov Blanket rather than all possible subsets in the remaining features to see if such a candidate Markov Blanket is sufficiently accurate to be regarded as the Markov Blanket.

The problem with this reasoning is how to find efficiently the Markov Blanket candidate for each feature. The naive method [6] to find the set of *k* features which are most dependent to F_i requires computing the dependence HSIC(F_i ,F_j) = $(m - 1)^{-2}$ *tr* HK_i HK_j for all pairs of features F_i and F_j (here, K_i and K_j are the kernel matrices of feature F_i and F_j respectively). This is clearly computationally too expensive for applications in high dimensional datasets. Instead, for F_i we will compute an approximate Markov Blanket candidate MB_i whose each feature F_{MBi} satisfies:

$$F_{MB_i} = \arg\max_{F_B} HSIC(K_{F_B}, K_i),$$
$$\text{where} : F_B \in B_i - MB_i \cup \{F_{MB_i}\}$$

Here, B_i is a set of features which are more dependent to the target variable C than F_i, and K_{FB} and K_i are kernel matrices of feature F_B and F_i respectively.

Observe that we tend to find the approximate Markov Blanket for F_i in the features which are more dependent with the target variable C than F_i is. To find the Markov Blanket Candidate for F_i, we measure dependence of each feature in F to target C and, for each feature F_i, consider only a subset B_i of features that are more dependent to C than F_i is. Here, dependence of feature F_i to C is measured as HSIC(F_i ,C) = $(m - 1)^{-2}$ *tr* HK_i HL_c, where K_i and L_c are the kernel matrices of feature F_i and target variable C. We approximate the Markov Blanket Candidate of F_i as the set of *k* features from set B_i that are most dependent to F_i. For the features whose corresponding set B_i has less than *k* features, we choose all features in B_i as the Markov Blanket candidate of F_i.

We emphasize that quality of the Markov Blanket Candidate obtained by the proposed method depends on choice of *k* which should not be too large or too small (too large *k* could include features that are irrelevant while too small *k* could result in an incomplete Markov Blanket). However, our experiments reported in results section provide evidence that this is not a serious limitation in practice since the proposed method was quite robust over a large range of choices of *k*.

1.3. Screening Markov Blanket Candidates

Let MB_i be the Markov Blanket candidate of feature F_i, found as explain in section 1.2. We say that MB_i passes the dependence-based screening test and is regarded as an actual approximation of the Markov Blanket if it satisfies the following two conditions:

$$\begin{aligned}&1. HSIC(MB_i, C) > HSIC(MB_i \cup F_i, C)\\&2. HSIC(MB_i, C) > HSIC(F_i, C), \text{and}\\&\ \ HSIC(MB_i, F_i) > HSIC(F_i, C)\end{aligned}$$

where C is the target variable and HSIC(X, Y) is defined as the dependence measure between two variable sets X and Y. We remove the feature whose Markov Blanket candidate passes this screening test. In contrast to a previous work [15], we remove both irrelevant and redundant features at the same time rather than separating into two steps. This is appropriate since an independent irrelevant feature F_i always satisfies the first test condition while a dependent irrelevant F_i will satisfy the second condition as in such a case F_i is irrelevant to C resulting in $HSIC(F_i ,C)$ smaller than $HSIC(MB_i, F_i)$. Similar, condition 2 ensures that the redundant feature is removed, since the corresponding MB_i of such F_i can subsume the information this feature have about the target variable. $HSIC(MB_i ,F_i) > HSIC(F_i, C)$ implies F_i is more dependent to MB_i than to C; $HSIC(MB_i ,C) > HSIC(F_i ,C)$ means MB_i is more dependent to C than F_i and ensures MB_i has more deterministic information to the target variable C than F_i does.

1.4. Feature Selection Algorithm

The optimal feature selection using the Markov Blanket is based on removing a feature for which we can find the Markov Blanket in the remaining features. Instead, in our computationally efficient method, we remove the feature for which we find an approximate Markov Blanket. According to the approximate Markov Blanket construction described in Section 1.3, we propose the following independence-based feature selection algorithm that will be called Hilbert-Schmidt Markov Blanket method (HSMB).

For each F_i in the whole feature set F, this algorithm computes HSIC(F_i ,C) which is the dependence between F_i and target variable C and then sorts features into a list S in descending order based on the measured dependence. Then, for each feature F_i, the algorithm constructs set B_i consisting of the features which are located before F_i in the list S. Then, HSMB finds the Markov Blanket candidate MB_{can} of F_i which is exactly the *k* features in the set B_i that are most dependent to F_i. If MB_{can} passes the screen test, it is regarded as the Markov Blanket of feature F_i. Therefore, the algorithm will remove such feature F_i from the sorted list S. In this way in a single pass through the list of features, HSMB removes all features for which the algorithm finds the approximate Markov Blanket in the remaining set of features. No multi-iteration is needed in HSMB algorithm.

The main cost of HSMB algorithm is in computing HSIC values which has complexity of $O(M^2)$ in terms of the number of instances M. This is better than the cost of other kernel-based methods which usually have $O(M^3)$ complexity. Cost of HSMB in terms of the number of features N is smaller. However, for each feature we have to find the candidate Markov Blanket in which there are *k* features. This requires searching for *k* most dependent features in the set B_i (features before F_i in the list S) to find the Markov Blanket Candidate. Hence, to select the optimal subsets MB, the algorithm takes O(p*N) steps, where p is the number of features before a certain feature F_i in the list S. In the worst case, p becomes N, and then the cost of the algorithm is $O(N^2)$. However, p is a small constant when enough features are removed resulting in O(N) best case complexity. This best case scenario corresponds to many high dimensional datasets where we are likely to remove most features.

1.5. Conclusion and Preliminary Results

The results of 12 benchmark classification problems are summarized in Table 1.1 and Table 1.2. Each result listed in these two tables is an average of 5 repeated experiments. Table 1 shows the predictive accuracy. Here, the leave one out method is applied to the datasets with less than 300 instances, in order to get stable results of SVM predictors. Other results reported in these tables are obtained by average results of 5 repeated experiments. The obtained results (Table 1) provide evidence that HSMB outperforms alternative methods in accuracy over a variety of benchmark datasets. On Lung-Cancer, Arcene, Promoters and wpbc evaluations, the GS algorithm was less accurate as these datasets have a fairly small number of instances which makes a conditional dependence test unreliable.

The number of features selected by each feature selection method is reported in Table 1.2. All methods reduced the number of features significantly. However, the GS and FCBF algorithms were too aggressive in reducing features. In GS, this problem is due to unreliability of the conditional dependence test for a small sample. The reason FCBF tends to remove too many features is that it gives too much priority to features that are highly correlated with the target.

Table 1.1. Classification accuracy on benchmark datasets

Datasets	All	GS	FCBF	BAHSIC	HSMB
BC-Wisconsin	**96.8**	**96.8**	95.6	**96.8**	**96.8**
Hepatitis	82.6	83.0	86.9	78.3	**89.8**
German	**100**	**100**	**100**	**100**	**100**
Wdbc	96.7	97.0	96.2	95.0	**97.3**
wpbc	80.0	72.0	75.0	78.0	**81.5**
Lung-Cancer	67.9	62.0	65.0	**70.4**	**70.4**
COIL2000	93.7	94.0	**94.4**	90.2	**94.4**
High Dimensional	Data				
Musk2	86.0	88.0	89.2	89.1	**92.4**
Promoters	86.4	88.5	97.6	94.7	**98.6**
Madelon	56.0	63.3	61.5	62.0	**70.1**
Isolet	75.1	78.4	78.5	78.0	**81.3**
Arcene	59.4	60.2	62.4	64.2	**70.6**

Table 1.2. Number of selected features

Datasets	GS	FCBF	HSMB
BC-wisconsin	8	8	8
Hepatitis	4	3	10
German	20	21	22
Wdbc	6	3	15
wpbc	3	2	6
Lung-Cancer	3	5	3
COIL2000	5	6	8
High	Dimensional	Data	
Musk2	4	2	6
Promoters	15	13	17
Madelon	4	2	38
Isolet	4	4	10
Arcene	15	13	113

Our results show that the proposed algorithm works well on both low and high dimensional data sets. In the high dimensional data set with a large number of irrelevant features, HSMB was effective in selecting the dominant features from which accurate predictors were built. In contrast, BAHSIC had high error in high dimensional evaluations.

Six benchmark datasets were used for the evaluation of the new algorithm for regression problems. Available benchmark datasets for regression were low dimensional. Therefore, for high dimensional evaluation we have created three datasets based on the Housing benchmark dataset. Housing-100 includes the original 13 features of Housing dataset with 48 additional redundant features and 52 irrelevant features. Among the 48 redundant features, for each feature in the optimal set selected by HSMB from Housing, there are 8 relevant features which match the selected feature 9/16, 10/16, ... , 16/16 of time. Housing-500 includes the features used in Housing-100 and additional 400 irrelevant features. Similar, Housing-1000 includes all features of Housing-100 and 900 additional irrelevant features.

Table 1.3. Regression accuracy (R-square) on benchmark data

Data Set		All Features		BAHSIC	HSMB	
Name	# instances	R-square	# Features	R-square	R-square	Selected Features
Cpu-performance	209	**0.575**	6	**0.575**	**0.575**	5
Auto-mpg	392	**0.745**	7	0.70	0.742	5
Concrete	1030	**0.880**	8	0.79	0.86	5
Housing	506	**0.634**	13	0.534	0.613	6
Aerosol	2000	**0.756**	14	0.61	0.714	6
Auto-mobile	159	**0.492**	18	0.391	0.398	5
High Dimensional	Data					
Wpbc	194	0.989	33	0.982	**0.997**	11
Housing-100	506	0.601	113	0.520	**0.610**	9
Housing-500	506	0.453	513	0.531	**0.593**	23
Housing-1000	506	0.364	1013	0.482	**0.574**	82

For each regression benchmark dataset, we performed the experiments following the same procedure as that used in classification. Our method is compared only to BAHSIC as other two methods are specific for classification. Again, in BAHSIC the same number of features is used as returned by our method. The R-square accuracies of an SVM on the subsets selected by each of two feature selection methods on benchmark datasets are reported in Table 1.3. The obtained results were consistent results with classification results. In general, HSMB outperformed BAHSIC in these regression problems. In low dimensional experiments, the predictor using full features was the most accurate since there were almost no irrelevant features in these benchmark datasets. However, in high dimensional experiments with many irrelevant features (Housing-100, Housing-500 and Housing-1000), the dominant features selected by the proposed method were a much better choice.

CHAPTER 2

IMPUTING MISSING VALUES

Prediction models for multivariate spatio-temporal functions in geosciences are typically developed using supervised learning from attributes collected by remote sensing instruments collocated with the outcome variable provided at sparsely located sites. In such collocated data there are often large temporal gaps due to missing attribute values at sites where outcome labels are available. Our objective is to develop more accurate spatio-temporal predictors by using enlarged collocated data obtained by imputing missing attributes at time and locations where outcome labels are available. The proposed method for large gaps estimation in space and time (called LarGEST) exploits temporal correlation of attributes, correlations among multiple attributes collected at the same time and space, and spatial correlations among attributes from multiple sites.

LarGEST outperformed alternative methods in imputing up to 80% of randomly missing observations at a synthetic spatio-temporal signal and at a model of fluoride content in a water distribution system. LarGEST was also applied for imputing 80% of nonrandom missing values in data from one of the most challenging Earth science problems related to aerosol properties. Using such enlarged data a predictor of aerosol optical depth is developed that was much more accurate than predictors based on alternative imputation methods when tested rigorously over entire continental US in year 2005.

2.1. Introduction of Imputation Method

Many of the existing spatio-temporal data analysis methods assume that data is complete or almost complete. This assumption is violated in many spatio-temporal applications.

For estimation of missing values in multiple time series, a dynamic Bayesian model called DynaMMo was recently developed to simultaneously exploit temporal smoothness of each series and their correlations [24]. Although very effective for estimating missing values in coevolving time series, DynaMMo is less applicable to remote sensing where data is collected at multiple spatially correlated locations where multiple correlated time series are observed at each location and individual series have temporal continuity.

For imputing incomplete spatial data, one of the most successful practical methods is to use multivariate interpolation by empirical orthogonal functions [18]. In this singular value decomposition (SVD) based data imputation approach that we will simply call EOF, missing values are initially replaced by an unbiased guess and the missing values were interpolated incrementally by using truncated orthogonal functions of SVD decomposition for reconstruction and repeating the process while increasing the number of component functions. A limitation of EOF based data imputation is that it exploits only spatial correlations in data which is a problem when long continuous gaps are present in spatio-temporal data. An application of EOF to a transposed matrix that we will call T-EOF is proposed as a practical way to address such larger gaps in data [23]. The same technique of using a transposed data to catch a different aspect of correlations in data (spatial instead of temporal) can also be applied to linear interpolation and to DynaMMo imputations. Such versions we will call T-Linear and T-DynaMMo methods.

Previous work [25] in developing data-driven AOD retrieval methods using spatio-temporally collocated satellite and ground based observations (as shown at Figure 1) was simply removing the missing observations. The hypothesis explored in this study is that the accuracy of previously developed AOD predictors can be improved significantly by estimating the missing attributes and then train predictors on the data set consisting of both observed and imputed attributes.

2.2. Modeling correlations in a single sequence

Given a multivariate spatio-temporal data (a multi-dimensional sequence and bunch of neighboring multi-dimensional sequences), there are three types of correlations: temporal correlation of each dimension, correlations among multiple dimensions collected at the same time and space, and spatial correlations from neighboring sites. Our goal is to exploit all three kinds of correlations to estimate the missing values, and then build an accurate model on such enlarged data with imputed values.

Assume that an m-dimensional sequence X = $\{x_1, x_2, \ldots, x_N\}$ of length N is given, where vector x_n observed at the nth time tick of sequence (n = 1,…,N) is a m-dimensional multivariate Gaussian. For each m-dimensional observation of vector x_n we introduce a Gaussian latent variable z_n such that there is a linear dependence with a Gaussian noise between each x_n and z_n defined as $\mathbf{x_n} = \mathbf{C}\mathbf{z_n} + \mathbf{v_n}$, where C is the parameter matrix and $\mathbf{v} \sim N(\mathbf{v} \mid 0, \mathbf{\Sigma})$ is the Gaussian noise with mean of zero and variance of $\mathbf{\Sigma}$.

We also define a linear dependence with Gaussian noise between two adjacent latent variables z_{n-1} and z_n corresponding to two successive observation x_{n-1} and x_n as $\mathbf{z_n} = \mathbf{A}\mathbf{z_{n-1}} + \mathbf{w_n}$, where A is the parameter and $\mathbf{w} \sim N(\mathbf{w} \mid 0, \mathbf{\Gamma})$ is the Gaussian noise with mean of zero and variance of $\mathbf{\Gamma}$.

Therefore, the emission and transition distribution can be written as:

$$p(\mathbf{x_n} \mid \mathbf{z_n}) = N(\mathbf{x_n} \mid \mathbf{C}\mathbf{z_n}, \mathbf{\Sigma}) \tag{1}$$

$$p(\mathbf{z_n} \mid \mathbf{z_{n-1}}) = N(\mathbf{z_n} \mid \mathbf{A}\mathbf{z_{n-1}}, \mathbf{\Gamma}) \tag{2}$$

The initial latent variable z_1 also has a Gaussian distribution which can be written as

$$p(\mathbf{z_1}) = N(\mathbf{z_1} \mid \boldsymbol{\mu}_\mathbf{0}, \mathbf{V_0}) \tag{3}$$

where μ_0 is the initial state of z1 and $\mathbf{V_0}$ is the variance.

Let $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{\Gamma}, \mathbf{C}, \mathbf{\Sigma}, \boldsymbol{\mu}_\mathbf{0}, \mathbf{V_0}\}$ be the parameters of the model. Therefore, the joint probability given $\boldsymbol{\theta}$ is

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) = {} & p(\mathbf{z_1} \mid \boldsymbol{\mu}_\mathbf{0}, \mathbf{V_0}) \cdot \\ & \prod_{n=2}^{N} p(\mathbf{z_n} \mid \mathbf{z_{n-1}}, \mathbf{A}, \mathbf{\Gamma}) \cdot \prod_{n=1}^{N} p(\mathbf{x_n} \mid \mathbf{z_n}, \mathbf{C}, \mathbf{\Sigma}) \end{aligned} \tag{4}$$

Our model is different to the traditional Kalman Filter -based model, since we allow missing values to exist in the observation X. We define a Missing Index Matrix $\mathbf{I}$ to indicate the missing values. Each entry of $\mathbf{I}$ is defined as

$$\mathbf{I}_{pq} = \begin{cases} 0 & \text{when } \mathbf{X}_{pq} \text{ is missing} \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

For learning the model, we define the expectation of the complete-data log likelihood as

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{old}) = E_{\mathbf{Z}|\boldsymbol{\theta}_{old},\mathbf{I}}[\ln p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}, \mathbf{I})] \tag{6}$$

First, we initialize each missing value X_{pq} in data sequence (where $\mathbf{I}_{pq} = 0$) using simple linear interpolation from the values where $\mathbf{I}_{pq} \neq 0$ at the same time. Then, we apply the EM algorithm to maximize the equation (6). By extending the equation (6) by substituting $p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$ using the corresponding part from equation (4) and (1)-(3), we get

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{old}) &= -\frac{1}{2}\ln|\mathbf{V_0}| - E_{\mathbf{Z}|\boldsymbol{\theta}_{old}}[\frac{1}{2}(\mathbf{z_1} - \boldsymbol{\mu}_\mathbf{0})^T \mathbf{V_0}^{-1}(\mathbf{z_1} - \boldsymbol{\mu}_\mathbf{0})] \\ &-\frac{\mathrm{N}-1}{2}\ln|\boldsymbol{\Gamma}| - E_{\mathbf{Z}|\boldsymbol{\theta}_{old}}[\frac{1}{2}\sum_{n=2}^{N}(\mathbf{z_n} - \mathbf{A}\mathbf{z_{n-1}})^T \boldsymbol{\Gamma}^{-1}(\mathbf{z_n} - \mathbf{A}\mathbf{z_{n-1}})] \\ &-\frac{\mathrm{N}}{2}\ln|\boldsymbol{\Sigma}| - E_{\mathbf{Z}|\boldsymbol{\theta}_{old}}[\frac{1}{2}\sum_{n=1}^{N}(\mathbf{x_n} - \mathbf{C}\mathbf{x_n})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x_n} - \mathbf{C}\mathbf{x_n})] + const \end{aligned} \tag{7}$$

where *const* is the term which is not dependent on any part of parameter $\boldsymbol{\theta}$.We take the derivatives of equation (7) with respect to each part of parameter $\boldsymbol{\theta}$ and then set them to zero. We get the parameters updates as follows:

$$\boldsymbol{\mu}_\mathbf{0}^{new} = E[\mathbf{z_1}] \tag{8}$$

$$\mathbf{V_0}^{new} = E[\mathbf{z_1}\mathbf{z_1}^T] - E[\mathbf{z_1}]E[\mathbf{z_1}^T] \tag{9}$$

$$\mathbf{A}^{new} = (\sum_{n=2}^{N} E[\mathbf{z_n}\mathbf{z_{n-1}}^T]) \cdot (\sum_{n=2}^{N} E[\mathbf{z_n}\mathbf{z_{n-1}}^T])^{-1} \tag{10}$$

$$\begin{aligned} \boldsymbol{\Gamma}^{new} &= \frac{1}{N-1}\sum_{n=2}^{N}\{E[\mathbf{z_n}\mathbf{z_n}^T] - \mathbf{A}^{new}E[\mathbf{z_{n-1}}\mathbf{z_n}^T] \\ &\quad - E[\mathrm{B}^T]\mathrm{B}^{new} + \mathrm{B}^{new}E[\mathbf{z_{n-1}}\mathbf{z_{n-1}}^T](\mathbf{A}^{new})^T\} \end{aligned} \tag{11}$$

$$\mathbf{C}^{new} = (\sum_{n=1}^{N}\mathbf{x_n}E[\mathbf{z_n}^T]) \cdot (\sum_{n=1}^{N} E[\mathbf{z_n}\mathbf{z_n}^T])^{-1} \tag{12}$$

$$\begin{aligned} \boldsymbol{\Sigma}^{new} &= \frac{1}{N}\sum_{n=1}^{N}\{\mathbf{x_n}\mathbf{x_n} - \mathbf{C}^{new}E[\mathbf{z_n}]\mathbf{x_n}^T \\ &\quad - \mathbf{x_n}E[\mathbf{z_n}^T]\mathbf{C}^{new} + \mathbf{C}^{new}E[\mathbf{z_n}\mathbf{z_n}^T](\mathbf{C}^{new})^T\} \end{aligned} \tag{13}$$

At the end of each M step, we update the missing value X_{pq} (when $\mathbf{I}_{pq} = 0$) using

$$E[\mathbf{X}_{pq} \mid \mathbf{Z},\boldsymbol{\theta},\mathbf{I}] = \mathbf{C}^{new} \cdot E[\mathbf{Z}_{\{p,q\}}] \; (when \; I_{pq} = 0) \tag{14}$$

Calculating the updated parameters requires the inference in E step of the marginal distribution $p(\mathbf{Z} \mid \mathbf{X},\boldsymbol{\theta})$ for hidden latent variables given the data. The inference is similarly to the one in Kalman Filter-based model, since the missing values are updated at the end of each M step. We apply forward-backward message passing to calculate the expectation of posterior distribution of latent variables. The details of inference in the Kalman Filter-based model are omitted for lack of space (for more details see [19]).

Then, we use the updated X to recalculate the new parameters in the next EM iteration. We repeat this procedure until convergence. The estimation for missing values can be automatically obtained once the model is learned.

2.3. Modeling spatial correlations from neighbors

In this section, we describe how to explore spatial correlations among neighboring sites. We build a probabilistic model among multivariate sequence X and its neighboring observations to estimate the missing values in X conditioned on the observed values in neighboring sites.

Given a multivariate sequence X, let $L_i = \{l_{i1}, l_{i2},\ldots,l_{iN}\}$ (i = 1,…m) be the ith dimension of X, where l_{in} (n = 1,…,N) is a single value of the observation in the i^{th} dimension at the n^{th} time step. Assuming that X has observations at k neighboring locations, we define $\mathbf{L}_i^{(j)} = \{l_{i1}^{(j)}, l_{i2}^{(j)},\ldots,l_{iN}^{(j)}\}$ as the i^{th} dimension of X's j^{th} (j = 1,…,k) neighboring locations. Our objective is to impute the missing value in each L_i by exploiting the spatial correlation among L_i and $\{L_i^{(j)} \mid j = 1,\ldots,k\}$. In order to learn such spatial correlation, for each dimension L_i of X and corresponding $L_i^{(j)} = \{l_{i1}^{(j)}, l_{i2}^{(j)}, ..., l_{iN}^{(j)}\}$ form X's neighbors, we define $\mathbf{O}^{(i)} = \{\mathbf{o}_1^{(i)}, \mathbf{o}_2^{(i)}, ..., \mathbf{o}_N^{(i)}\}$ as the union of L_i and $\{\mathbf{L}_i^{(j)} \mid j = 1,...,k\}$, where $\mathbf{o}_n^{(i)} = \{l_{in}, l_{in}^{(1)}, l_{in}^{(2)}, ... l_{in}^{(k)}\}$ (n = 1,…,N) are the values of the ith dimension of X and its neighboring locations at the nth time step. For each observation $\mathbf{o}_n^{(i)}$, we define a Gaussian latent variable $\mathbf{y_n} \sim N(0, \mathbf{w})$ (n = 1,…,N). Each pair of nodes $\{y_n, \mathbf{o_n}^{(i)}\}$ represents a linear-Gaussian latent variable model for the particular multivariate observation. However, the latent variables $\{y_n\}$ are treated as independent to each other. Hence, the emission distribution is

$$p(\mathbf{o_n}^{(i)} \mid \mathbf{y_n}) = N(\mathbf{o_n}^{(i)} \mid \mathbf{D} \cdot \mathbf{y_n}, \phi) \tag{15}$$

Then, we can build a probabilistic graphical model for each dimension of X to exploit the spatial correlation between each L_i and its corresponding $\{\mathbf{L}_i^{(j)} \mid j = 1,...,k\}$ from neighboring locations. Let $\boldsymbol{\gamma} = \{\mathbf{w}, \mathbf{D}, \phi\}$ be the parameters of the model. Then, the joint distribution can be written as:

$$p(\mathbf{O}^{(i)}, \mathbf{Y} \mid \boldsymbol{\gamma}) = \prod_{n=1}^{N} p(\mathbf{y_n}) \cdot \prod_{n=1}^{N} p(\mathbf{o_n}^{(i)} \mid \mathbf{y_n}, \boldsymbol{\gamma}) \tag{16}$$

Therefore, maximizing the complete data log likelihood is equivalent to maximizing:

$$\begin{aligned} \ln\Big(\prod_{n=1}^{N} p(\mathbf{y_n}) \prod_{n=1}^{N} p(\mathbf{o_n}^{(i)} \mid \mathbf{y_n}, \boldsymbol{\gamma})\Big) &= -\frac{N}{2} \ln|\mathbf{w}| - \sum_{n=1}^{N} \frac{1}{2} \mathbf{y_n}^T \mathbf{w}^{-1} \mathbf{y_n} \\ &- \frac{N}{2} \ln|\phi| - \sum_{n=1}^{N} \frac{1}{2} (\mathbf{o_n}^{(i)} - \mathbf{D} \cdot \mathbf{y_n})^T \phi^{-1} (\mathbf{o_n}^{(i)} - \mathbf{D} \cdot \mathbf{y_n}) \end{aligned} \tag{17}$$

We take the derivatives of equation (17) with respect to $\mathbf{w}$, D and ϕ respectively, and set them to zero. The updated parameters are computed as

$$\mathbf{w}^{new} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{E}(\mathbf{y_n} \mid \mathbf{o_n}^{(i)})\mathrm{E}(\mathbf{y_n}^T \mid \mathbf{o_n}^{(i)}) \tag{18}$$

$$\mathbf{D}^{\mathrm{new}} = \sum_{n=1}^{N} \mathbf{o_n}^{(i)} E(\mathbf{y_n}^T \mid \mathbf{o_n}^{(i)}) \cdot (\sum_{n=1}^{N} E(\mathbf{y_n} \mid \mathbf{o_n}^{(i)}) E(\mathbf{y_n}^T \mid \mathbf{o_n}^{(i)}))^{-1} \tag{19}$$

$$\phi^{new} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{o_n}^{(i)} - \mathbf{D}^{new} \cdot E(\mathbf{y_n} \mid \mathbf{o_n}^{(i)})) \cdot (\mathbf{o_n} - \mathbf{D}^{new} \cdot E(\mathbf{y_n} \mid \mathbf{o_n}^{(i)}))^T \tag{20}$$

In order to get the optimal parameters which maximize equation (17) in the presence of missing observations, for each dimension of X we maintain another Missing Index Matrix $\mathbf{I}^{(i)}$ where $\mathbf{I}^{(i)}{}_{pq} = 0$ indicates a missing value of $\mathbf{O}^{(i)}{}_{pq}$. We initialize the each missing value using linear interpolation from values where $\mathbf{I}^{(i)}{}_{pq} \neq 0$ in the neighboring observations. Then, we calculate new parameters using equation (18)-(20) and use them to estimate the missing values as

$$E[\mathbf{O}_{pq(new)}{}^{(i)} \mid \mathbf{Y}, \gamma, \mathbf{I}^{(i)}] = \mathbf{D}^{new} \cdot E[\mathbf{Y}_{\{p,q\}} \mid \mathbf{O}_{pq}{}^{(i)}] \ (when\ \mathbf{I}^{(i)}{}_{pq} = 0) \tag{21}$$

After imputing the missing values using eq. (21), we use the new data to estimate new parameters in the next iteration. By repeating this process of estimating parameters and missing values until converging, we can get the optimal parameters of the model and the final estimation of missing values. After updating the missing values in $\mathbf{O}^{(i)}{}_{pq}$ for each dimension of X, we can get the estimated X.

2.4. Learning Algorithm

In order to estimate the missing values by exploring all three types of correlation, we propose the LarGEST algorithm which simultaneously learns two models described in Sections 2.2 and 2.3.

First, we initialize all model parameters and fill all the missing values by linear interpolation from the values of spatial neighbors. Then, we apply an Extended Expectation Maximization algorithm which works as follows.

In the E-step, we estimate the posterior distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$ of Model 1 which will be used when we maximize the expectation of log likelihood in M-step (using equations (8) -- (13)). After getting the updated parameters of Model 1, we can estimate the missing values using Model 1. The data with updated missing values from Model 1 is used to estimate the parameters of Model 2. We can re-estimate the missing values after learning the parameters of Model 2. The updated data with updated missing values estimated by Model 2 will be the input data of E-step of next iteration to calculate the posterior distribution of Model 1. We repeat this procedure of training two models interactively until convergence.

Our combining algorithm has a wide generality. Basically, we can replace Model 1 and Mode2 with any kind of appropriate training methods which can estimate missing values. This can help greatly if changing the proposed models to some more appropriate models when some domain knowledge is known about the data.

After imputing the missing values in multivariate spatio-temporal data, we then build a predictor on enlarged collocated spatio-temporal data. In the next section we compare results of predictors trained on enlarged dataset generated by LarGEST and by alternative methods.

2.5. Conclusion and Preliminary Results

The synthetic two dimensional temporal data (X, Y) for a certain location is generated as $X = \sin(t)*5$ and $Y = \sin(t + \pi / 2)*5$. At four neighboring locations two dimensional data is generated by shifting the first site data by 0.5 0.3, -0.3and -0.5 respectively.

The mean square error (MSE) of imputation by LarGEST and six alternatives was compared when 5% to 90% of data were missing randomly (see Figure 2.1). In data imputation experiments with small fraction (less than 25%) of missing values inserted completely at random, LarGEST and all alternative methods estimated missing values fairly well. However, LarGEST was clearly the best choice as Linear interpolation had problems when the missing values were located near the top or the bottom of the signal, while EOF had problems where four neighbors were missing and DynaMMo had some errors at high curvature sections.

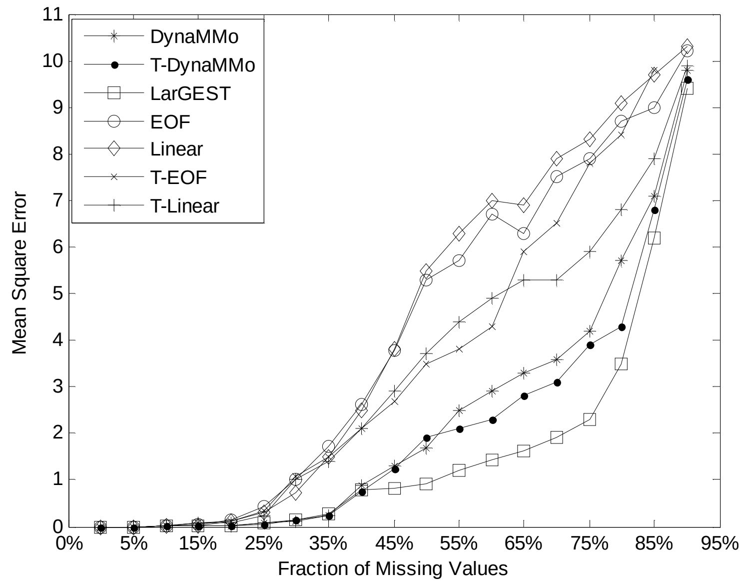

Figure 2.1: Mean predictor errors

When large fraction of data was missing LarGEST exploited well all three kinds of correlations in data simultaneously while alternative imputation methods resulted in much larger error. When extremely large fraction (85% and 90%) of data was missing, all methods performed badly. For a large fraction of missing values accuracy was improved when the number of neighbors was increased. Results with a larger number of neighbors are omitted on synthetic data for lack of space, but this effect will be shown in Section 4.3 on real life remote sensing Aerosol data.

Another experiment is conducted on ground based Aerosol data. Radiances from the MODIS instrument (19 attributes) over the entire continental US in year 2005 obtained at 4km*4km grid following a procedure from a previous study [22] were spatio-temporally integrated with ground-based AOD data at 33 AERONET sites [20] based at the nearest 4km*4km region and within ± 30 minutes of satellite passing the corresponding AERONET site. Data containing both attributes and AOD values consisted of 805 examples. AOD values observed from ground at 33 AERONET sites were also available at an additional 3,307 events, but in these cases all 19 sattelite-based attributes were missing. Missing attributes at these 80% cases were imputed by LarGEST relaying on spatial correlations with satellite observations at up to 80 neighbors at 4kmx4km grid as well as on temporal correlations among 330 daily observations. Six alternative methods used in Section 4.1 were also applied for attributes imputation at 3,307 events.

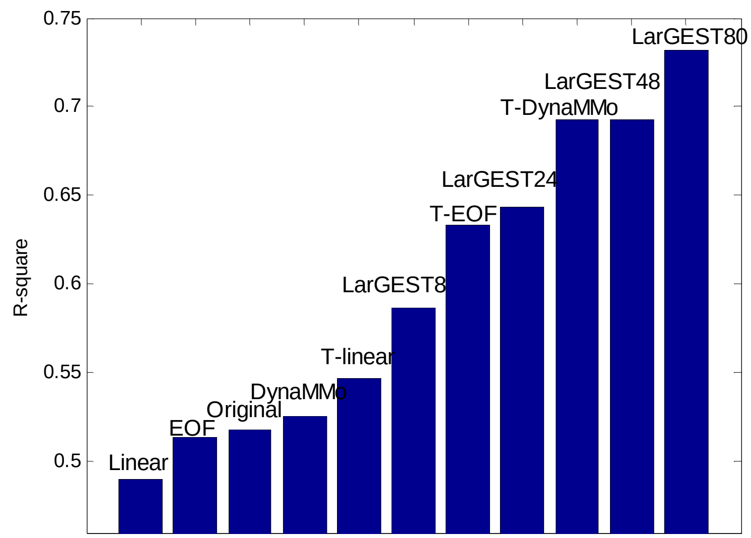

Figure 2.2. R-square accuracy on AOD data of US in 2005 with imputed data

A feed-forward neural network model with a single hidden layer of 10 nodes was trained on enlarged data consisting of examples with actual and those with imputed attributes. This choice was based on the best predictors from previous studies [25]. Experiments were performed by partitioning 805 examples from 33 sites where both attributes and AOD values are available in 33 disjoint subsets based on sites and using 32 subsets together with 3,307 additional examples whose attributes are imputed for training a neural network model which is tested on the remaining site's data. This is repeated in 33-cross validation experiments always keeping a different site for testing. The quality of the obtained predictors was compared using two measures following a protocol practiced by geoscience community [25]. To evaluate impact of spatial neighborhood size on imputation quality, in LarGEST imputation we considering nearest 8, 24, 48 and 80 neighboring observations in 4kmx4km grid shown at Figure 2.1. These methods we will call LarGEST8, LarGEST24, LarGEST48 and LarGEST80, respectively.

The quality measure we used is R-square. When comparing several predictors on a fixed test set larger R-square scores correspond to more accurate predictors. The results obtained by a predictor trained on data imputed by LarGEST80, and by nine alternative methods are shown in Figure 2.2. In addition, we also show the accuracy of a predictor obtained on 805 examples without data imputation and we call this model Original. The results obtained by LarGEST80 were more accurate than alternatives and better than any previously reported accuracy of AOD retrieval.

CHAPTER 3

FEATURE SELECTION IN INCOMPLETE DATA

This chapter is about feature selection in incomplete data. The straightforward method is imputing the missing value first, then applying traditional feature selection method on the complete data. However, imputation methods work a lot better when data is missing completely at random, or when there is small fraction of data are missing, or when there is prior knowledge about the data distribution. My current work is aimed to feature selection in incomplete data without imputation. The method is based on only observed data and is not including the error/bias in imputation methods. The preliminary results show that our method can choose better features than the ones based on enlarged data by imputing missing values.

3.1. Introduction to Feature Selection in Incomplete Data

Feature selection methods can be broadly categorized into filtering models [46] and wrapper models [33]. 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 filtering methods. In [44] a margin-based method is proposed as a feature-weighting algorithm that is a new interpretation of a RELIEF-based method [28]. 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 [38] 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 [43]. 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 [36]. A method is presented to handle incomplete data where the missing features are structurally absent for some of the instances [26]. Instances are considered as sets of (feature value) pairs that naturally handle the missing value case [29]. 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.

3.2. Uncertainty Margin

Let $D = \{(x_n, y_n | n = 1, \dots, N) \subset \mathbb{R}^M \times \pm 1\}$ 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

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

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.

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 $\mathbf{w}$, 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 $\rho_D(\mathbf{x}_n|\mathbf{w}) = \mathbf{w}^T\beta_n$, where $\beta_n = |\mathbf{x}_n - nearmiss(\mathbf{x}_n)| - |\mathbf{x}_n - nearhit(\mathbf{x}_n)|$, and $|\cdot|$ is the element-wise absolute operator.

In an incomplete data set, we cannot apply a uniform weight $\mathbf{w}$ 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 $\boldsymbol{s}_n = ||\mathbf{w}_n||_1/||\mathbf{w}||_1$. Therefore, the instance-based margin can be written as:

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

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, 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:

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

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:

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

Please note that $d(\mathbf{x}_n, \mathbf{x}_i | \mathbf{w}_n, \mathbf{I}_n) = ||\mathbf{x}_n - \mathbf{x}_i||_{\mathbf{w}_n, \mathbf{I}_n}$ 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 $\mathbf{w}_n$ where missing values are indicated by $\mathbf{I}_n$. 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:

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

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.

3.3. 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:

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

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

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

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 $\boldsymbol{w}$. The optimization formulation (10) can also be written as:

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

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

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

For each solution to (12), there is a parameter θ, corresponding to the obtained λ in (12), which gives the same solution in (11). Formulation (12) is actually the optimization problem with ℓ_1 regularization. The benefits of adding the ℓ_1 penalty have been well studied and it showed that the ℓ_1 penalty can effectively handle sparse data and huge amounts of irrelevant features.

3.4. Learning Feature Weights Using Uncertainty Margin

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:

$$f : \mathbf{w} \rightarrow \mathbf{u}, \text{ where } \mathbf{w}(i) = \mathbf{u}(i)^2, \ \forall i = 1, 2, \ldots M \qquad (13)$$

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

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

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

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

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 3.1.

Table 3.1. 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	

3.5. Conclusion and Preliminary Results

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 [45] that we call LBFS, Simba [28] and Relief [31] based on applying the following three popular imputation methods [26] 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 [27].

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 an *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 [45]. 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. 3.1). 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. 3.1 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. 3.1, the SID method becomes unstable much later than the alternatives.

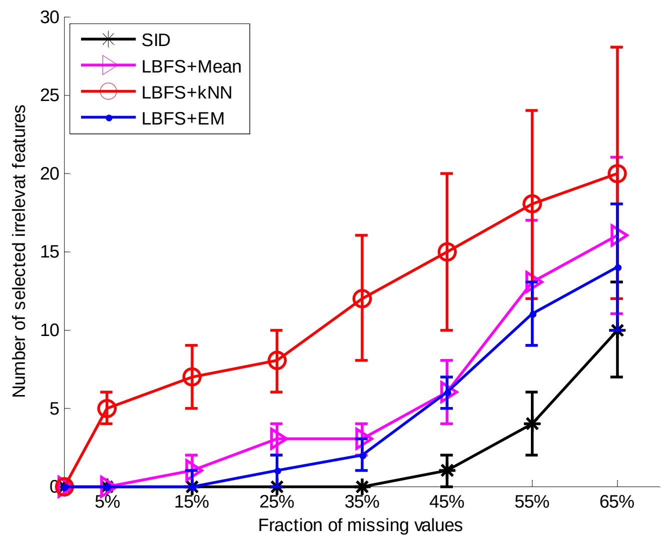

Figure 3.1: The Number of Irrelevant Features Selected Together With Two Relevant Features via. Fraction of Missing Values (Synthetic Data)

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 3.2. 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.

Table 3.2. 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

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.

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. 3.2 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. 3.3 and Fig. 4 the results of SID are compared to three alternatives based on kNN and EM imputation.

The results summarized at Fig. 3.2 and Fig. 3.3 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. 3.4 in the experiments where 2000 completely independent irrelevant features were present.

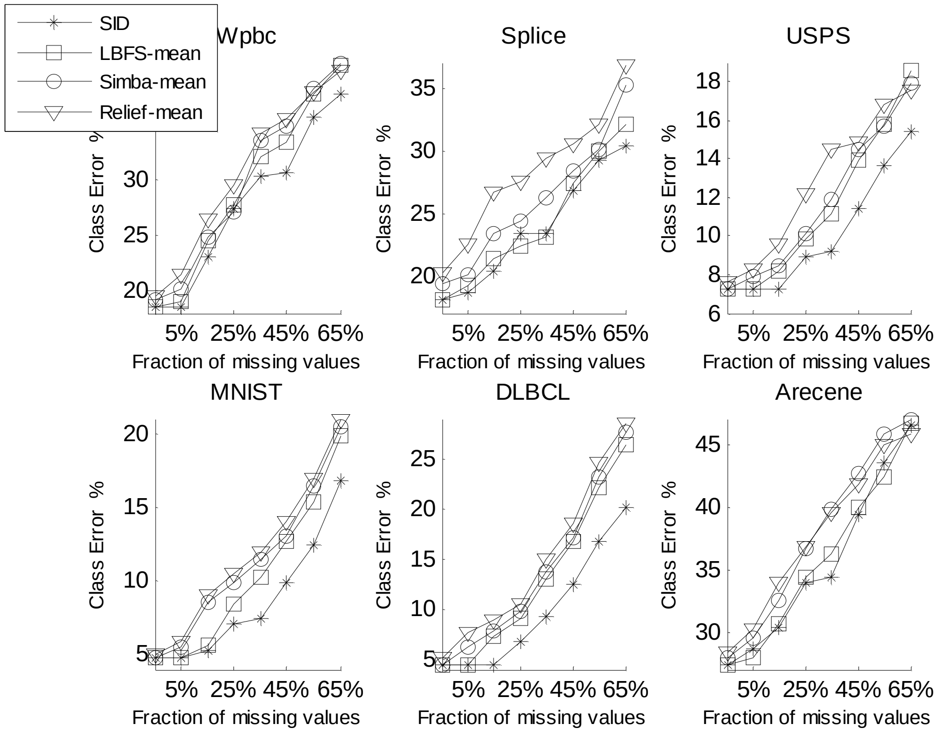

Figure 3.2. Classification error with respect to fraction of missing values by SID compared to three alternative feature selection methods that used Mean to perform data imputation

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. 3.2 and Fig. 3.3). However, these methods failed on *DLBCL* and *Arecene,* as most features in *these datasets* are irrelevant.

Fig. 3.2 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. 3.2, Fig. 3.3 and Fig. 3.4 also provide evidence that **SID** method outperformed alternatives in all data sets for different fractions of missing values.

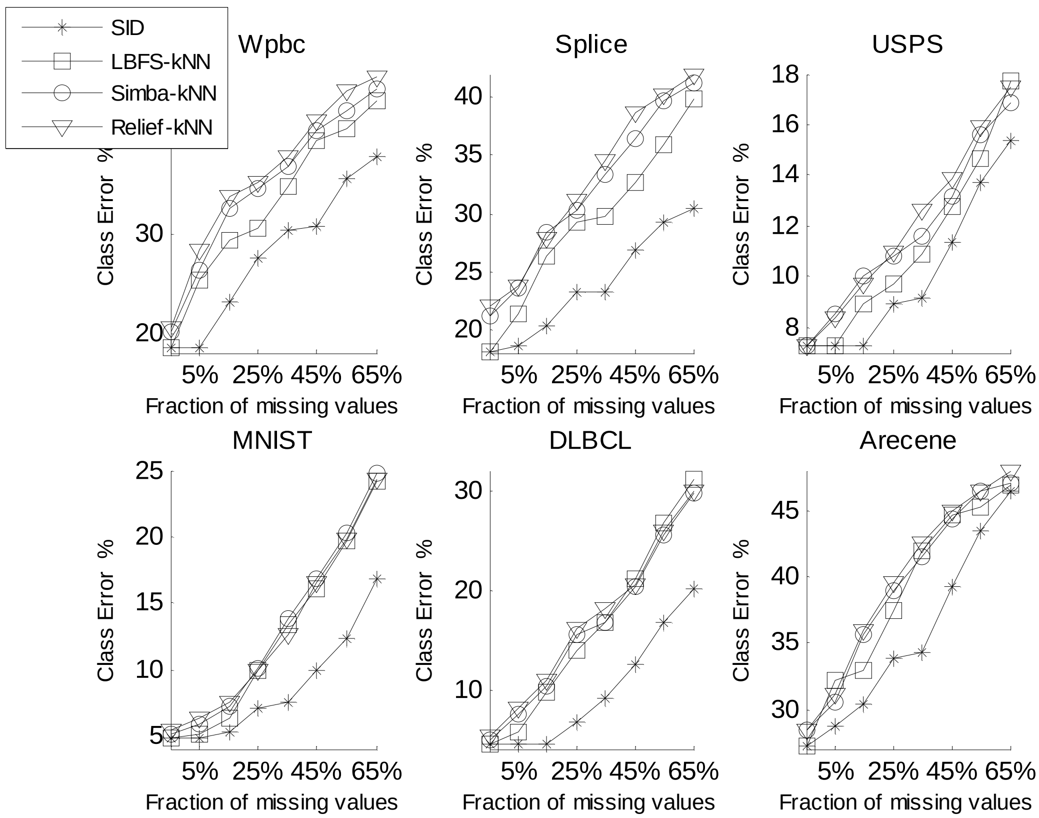

Figure 3.3 Classification error with respect to fraction of missing values by SID compared to three alternative feature selection methods that used KNN to perform data imputation

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.

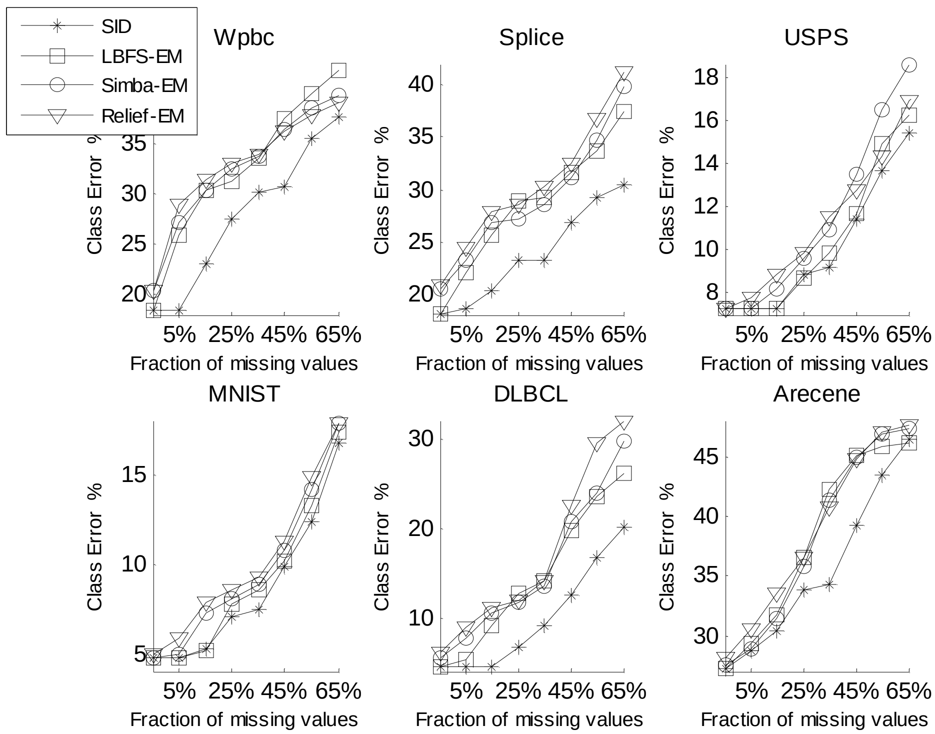

Figure 3.4. Classification error with respect to fraction of missing values by SID compared to three alternative feature selection methods that used EM to perform data imputation

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. 3.5. 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).

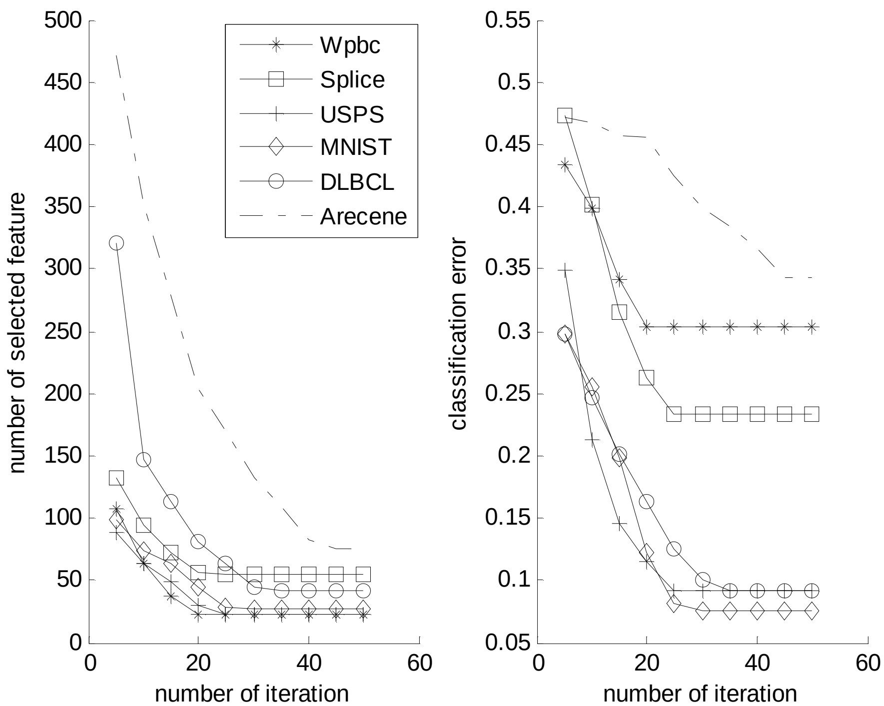

Figure 3.5. Convergence analysis. The number of selected features and classification error with respect to the number of iterations are shown at the left and the right panel, respectively.

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.

CHAPTER 4

LEARNING FROM TEMPORAL HIGH DIMENSIONAL DATA

In this part, I introduce my method handling more challenging situation where the high-dimensional data varies in time. Existing way to handle such data is to flatten temporal data into single static data matrix, and then applying traditional feature selection method. In order to keep the dynamics in the time series data, our method avoid flattening the data in advance. We propose a way to measure the distance between multivariate temporal data from two instances. Based on this distance, we define the new objective function based on the temporal margin of each data instance. A fixed-point gradient descent method is proposed to solve the formulated objective function to learn the optimal feature weights. The experimental results on real temporal microarray data provide evidence that the proposed method can identify more informative features than the alternatives that flatten the temporal data in advance.

4.1. Introduction

Microarray technology has the ability to simultaneously measure expression levels of thousands of genes for a given biological sample. There is often interest in the analysis of dynamic biological processes with data from DNA gene expression microarray chips. In order to predict an individual's health status, it is very helpful to analyze such high dimensional gene expression data that varies with time. There are two major challenges in prediction from such temporal microarray data. One is dealing with small-sample high-dimensional data where the number of biomarkers used as features is typically much larger than the number of labeled subjects. A common way to address this problem is to perform feature selection methods as a preprocessing step, followed by a classification method on selected features to predict the health status of an individual.

Another challenge of analyzing dynamic biological processes is that the data gathered is temporal. For example, in the two real flu data sets we used in experiments section, the data records for each individual are multivariate time series. The whole data set consists of many such multivariate time series from different individuals. However, traditional feature selection methods cannot handle such multivariate time series data. The most straightforward method of handling this is to apply some techniques to flatten the temporal data, and then perform traditional feature selection methods in the flattened data. Obviously, the flattening process may result in loss of some information among temporal data. Such straightforward methods tend to select features that are not informative enough.

In this study, we proposed a feature selection filter that can directly select informative features from temporal high-dimensional biomarkers. We defined a temporal margin for each subject based on a measure of distance between two multivariate time series data from two different subjects. The objective function of the proposed selection method is to maximize each subject's temporal margin in its own relevant subspace. We applied stochastic gradient ascent to solve the optimization problem and get the optimal weight for each feature. Features with large weights are selected to build the prediction model to predict the health status of each individual. The experimental results show that our method outperforms the alternatives, which apply traditional feature selection methods after flattening the temporal multivariate gene expression data.

There are some recent works about learning features from temporal gene expression data [55, 56]. However, they are different from this study. First, those methods treat the records for an individual at different time steps as different new subjects. Secondly, most of those works project the data to another space and learn features from the new space (factors or principal component). This method is difficult to use to help one understand the data based on selected factors since the factors are not in original space any more.

4.2. Related Work

Feature selection methods can be broadly categorized into filtering models [47] and wrapper models [48]. 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 widely used kinds of filtering methods. In [49, 51] a margin-based method is proposed as a feature-weighting algorithm that is a new interpretation of a RELIEF-based method [50]. The method in [51] is an online algorithm that solves a convex optimization problem with a margin-based objective function.

Markov Blanket-based methods [47, 52, 53] perform feature selection by searching an optimal set of features using Markov Blanket approximation. The method proposed in [47] used symmetrical uncertainty to measure the relationship between variables. For a pair of features, the method measured symmetrical uncertainty and also the symmetrical uncertainty between either of them and the target variable. If the measured value between these two variables is greater than the measured value between either one of them and the target variable, the variable with the larger symmetrical uncertainty to the target is regarded as the Markov Blanket of the other variable, which then is removed. The method proposed at [52] removed the feature whose Markov Blanket can be found in the rest of features. Method [53] applies Markov Blanket-based method in SNP data to select informative SNP to predict the efficacy of the drug.

Dependence estimation-based methods use the Hilbert-Schmide Independence Criterion as a measure of dependence between the features and the labels [54]. The key idea in this method is that good features should maximize such dependence. However, all these methods assume that the data is static without varying on time. They cannot be applied in temporal gene expression data that is the main problem of this study.

Several feature learning methods [55, 56] have recently been proposed to handle the temporal gene expression data, without imputing missing values in advance. Method in [55] applied the Beta Process to the factor scores, or to the singular values of a SVD construction to infer the number of factors in gene-expression data. The method was tested on several gene-expression data including a temporal one. Method in [56] developed a time-aligned Bayesian dynamic factor analysis methodology. By using a nonparametric cure rate model for the latent initiation times, the method allowed selection of the genes in the viral response pathway, variability among individuals in infection times, and s subset of individuals who are not infected. However, those two methods are different from the proposed method in this study. First, those methods treat the records for an individual at different time steps independently, which will result in loss of temporal information among the data. Secondly, all those works project the data to another space and learn features from the new space (factors or principal component). Those methods are actually methods for dimension reduction, rather than feature selection. Due to this, we will not compare our method with them in this study.

4.3. Proposed Method

Let $\mathbf{D} = \{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1,\ldots,I} \subset \Re^{n \times T_i} \times \pm 1$ be the data set with I individuals. $\mathbf{X}_i \in \Re^{n \times Ti}$ represents n observed biomarkers (e.g. gene expression data) for individual i measured at T_i time steps. $\mathbf{Y}_i \in \{1, -1\}$ represents the class label (e.g. health status) for individual i. Let $\mathbf{X}_i^{(r)}$ be the r^{th} column of $\mathbf{X}_\text{i}$ that corresponds n biomarkers measured at time t_r.

We will first define the measure of distance between multivariate time series data of two subjects, and then present the temporal margin based on the distance measure as well as the objective function of proposed feature selection method and algorithm for solving the corresponding optimization problem.

4.3.1. Measure Distance Among Multivariate Time Series

Given $\mathbf{X}_\text{i}$, $\mathbf{X}_\text{j}$ corresponding to the observed biomarkers measured at different time steps for individual i and individual j, respectively, the distance (we call Temporal distance, represented as *Tdist*) between two multivariate time series $\mathbf{X}_\text{i}$ and $\mathbf{X}_\text{j}$ is defined as:

$$Tdist(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{T_i \times T_j} \sum_{r=1}^{T_i} \sum_{s=1}^{T_j} d(\mathbf{X}_i^{(r)}, \mathbf{X}_j^{(s)})$$

where T_i and T_j are the number of time steps of individual *i* and individual *j*, respectively; $\mathbf{x}_i^{(r)}$ is the vector consists of biomarkers measured at time steps *r* for individual *i*; $\mathbf{x}_i^{(s)}$ is the vector of biomarkers measured at time steps *s* for individual *j*; for any two vectors **v** and **z**, function *d*(**v**, **z**) is defined as $d(\mathbf{v}, \mathbf{z}) = \sqrt{\sum_p (\mathbf{v}_p - \mathbf{z}_p)^2}$.

4.3.2. Maximize Temporal Margin

It has been shown [50] that margins play a crucial role in many machine learning methods. They measure the confidence of a classifier when making classification decision. Margins have been used both for theoretic generalization bounds and as guideline for algorithm design. Usually, the larger margin an instance has, the more easily it can be classified correctly. If many sample points have large margin, a good generalization can be guaranteed about the data, and the data will be easily classified.

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 $\mathbf{X}_i$, we find two nearest neighbors for $\mathbf{X}_i$, one with the same class label (called ***nearhit***), and the other with different class label (called ***nearmiss***). The hypothesis-margin of a given instance $\mathbf{X}_i$ in data set D is defined as:

$$L_D(\mathbf{X}_i) = \frac{1}{2}(Tdist(\mathbf{X}_i, nearmiss(\mathbf{X}_i)) \\ - Tdist(\mathbf{X}_i, nearhit(\mathbf{X}_i)))$$

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 $\mathbf{X}_i$ from D in a weighted feature space as:

$$\rho_D(\mathbf{X}_i \mid \mathbf{w}) = \frac{1}{2}(Tdist(\mathbf{X}_i, nearmiss(\mathbf{X}_i) \mid \mathbf{w}) \\ - Tdist(\mathbf{X}_i, nearhit(\mathbf{X}_i) \mid \mathbf{w}))$$

which is equivalent to:

$$\rho_D(\mathbf{X}_i \mid \mathbf{w}) = \frac{1}{2T_i \times T_j}\sum_{r=1}^{T_i}\sum_{s=1}^{T_M} d(\mathbf{X}_i^{(r)}, nearmiss(\mathbf{X}_i)^{(s)} \mid \mathbf{w}) \\ - \frac{1}{2T_i \times T_j}\sum_{r=1}^{T_i}\sum_{s=1}^{T_H} d(\mathbf{X}_i^{(r)}, nearhit(\mathbf{X}_i)^{(s)} \mid \mathbf{w})$$

where T_M and T_H are the number of time steps of nearmiss($\mathbf{X}_i$) and nearhit($\mathbf{X}_i$), respectively; for any two vectors **v** and **z**, function $d(\mathbf{v}, \mathbf{z}|\mathbf{w})$ is defined as:

$$d(\mathbf{v}, \mathbf{z} \mid w) = \sqrt{\sum_p (\mathbf{v}_p - \mathbf{z}_p)^2 w_p^2}\ .$$

We already define the instance margin for each subject $\mathbf{X}_i$. Therefore, we can define the temporal margin of the entire data D that has *I* subjects as the sum of all instance margins, which can be written as:

$$\rho_{D|\mathbf{w}} = \sum_{i=1}^{I} \rho_D(\mathbf{X}_i \mid \mathbf{w})$$

The feature weights can be learned by solving an optimization problem that maximizes the temporal margin of entire data D. Therefore the most informative features can be chosen based on the feature weights learned. The bigger weight a feature has, the more important the feature is. This optimization problem can be represented as:

$$\max_{\mathbf{w}} \sum_{i=1}^{I} \rho_D(\mathbf{X}_i \mid \mathbf{w})$$

which is equivalent to:

$$\begin{aligned}\max_{\mathbf{w}} \sum_{i=1}^{I} (&\frac{1}{2T_i \times T_j} \sum_{r=1}^{T_i} \sum_{s=1}^{T_M} d(\mathbf{X}_i^{(r)}, nearmiss(\mathbf{X}_i)^{(s)} \mid \mathbf{w}) \\ &- \frac{1}{2T_i \times T_j} \sum_{r=1}^{T_i} \sum_{s=1}^{T_H} d(\mathbf{X}_i^{(r)}, nearhit(\mathbf{X}_i)^{(s)} \mid \mathbf{w}))\end{aligned} \quad (1)$$

We solve this optimization problem to get the optimal weights **w** by applying stochastic gradient ascent. The gradient of $\rho_{D|\mathbf{w}}$ when evaluated on a data D is:

$$(\nabla\rho_{D|\mathbf{w}})_i = \frac{\partial\rho_{D|\mathbf{w}}}{\partial\mathbf{w}_i} = \sum_{j=1}^{I}\frac{\partial\rho_D(\mathbf{X}_j)}{\partial\mathbf{w}_i}$$

$$= \frac{1}{2}\sum_{j=1}^{I}\Bigg(\frac{\frac{1}{T_i \times T_M}\sum_{r=1}^{T_i}\sum_{s=1}^{T_M}| X_i^{(r)} - nearmiss(X_i)^{(r)} |_j^2}{Tdist(\mathbf{X}_i, nearmiss(\mathbf{X}_i) \mid \mathbf{w})} - \frac{\frac{1}{T_i \times T_H}\sum_{r=1}^{T_i}\sum_{s=1}^{T_H}| X_i^{(r)} - nearhit(X_i)^{(r)} |_j^2}{Tdist(\mathbf{X}_i, nearhit(\mathbf{X}_i) \mid \mathbf{w})}\Bigg) \tag{2}$$

where, for two vectors **v** and **z**, the function $|\,\mathbf{v}-\mathbf{z}\,|_j^2$ is defined as: $|\,\mathbf{v}-\mathbf{z}\,|_j^2 = (\mathbf{v}_j - \mathbf{z}_j)^2$.

4.3.3. Feature Selection Algorithm

In this section we will introduce our feature selection method, which solves the optimization problem introduced in previous section.

The proposed algorithm for **Feature Selection** in **Temporal** microarray data (we call it **FST**) is shown in Table 4.1. The **FST** algorithm starts with initializing the values of **w** to be 1. With such initialization we can estimate the instance margin for each instance $\mathbf{X}_i$. Then, in each iteration, the weights vector **w** is updated by solving the optimization problem introduced in previous section. We repeat the iteration until convergence or using all instances to update the weights.

The complexity of the **FST** algorithm is O(*TNM*) where T is the total number of iterations, *N* is the number of features, and *M* is the number of subjects. Usually, we let algorithm iterate on all training instances, therefore the complexity of FST algorithm is O(NM^2).

TABLE 4.1 FST FEATURE SELECTION METHOD

Input: data set $D = \{\mathbf{X}_i, y_i\}_{i=1,\ldots,I}$ **Output:** feature weights w

Initialization: set $w^{(0)}=1$, $t = 1$
 For each subject $\mathbf{X}_i$, Do
 Calculate Tdist($\mathbf{X}_{i,}\ \mathbf{X}_r|\mathbf{w}^{(t-1)}$) when $r \neq i$
 Calculate nearmiss($\mathbf{X}_i$) and nearhit($\mathbf{X}_i$)
 For each dimensionality j, Do
 Calculate ∇_j
 End For
 $t = t + 1$;
 $\mathbf{w}(t) = \mathbf{w}(t-1) + \nabla$
 End For
 $\mathbf{w} \leftarrow \mathbf{w}^2/\|\mathbf{w}^2\|$

4.4. Experiments and Results

To characterize the proposed algorithm, we conducted large-scale experiments on both synthetic and 2 real flu data sets [55, 56]. All experiments of this study were performed on a PC with 3 GB of memory. We compared our proposed **FST** algorithm in temporal gene expression data with three traditional feature selection methods (the method proposed in [47] that we call **FCBF**, **HSMB** [52] and **Relief** [57]) after flattening temporal multivariate data into one single matrix.

For the prediction method, we apply a Nearest Neighbor classifier on all features and select features by different feature selection methods. We compare results on both synthetic data and real data.

4.4.1. Results on Synthetic Data

We generate synthetic data simulating 20 subjects. Each subject has 50 dimensional records at 20 different time steps. Each subject i is generated according the following process. We first generate 50 dimensional random data Xi for subject i at time step 1. Label Yi is complete decided by the first four features following $Y_i = (\mathbf{X}_{i1} \vee \mathbf{X}_{i2}) \wedge (\mathbf{X}_{i3} \vee \mathbf{X}_{i4})$. We then generate records for subject i at other time steps using formula: $\mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t+1)} + \varepsilon$, where $\varepsilon \sim \mathrm{N}(0, \frac{t}{10})$ is the Gaussian noise that is also a function of time steps.

The results on synthetic data are shown in Figure 4.1 and Table 4.2. Figure 4.1 shows the feature weights for each feature learned by our FST algorithm. It clearly shows that our method assigns significantly larger weights to the first four features used to decide the Label than to most other features.

Table 4.2 shows the results comparing our method to three alternatives (three alternatives are applied after flattening the temporal data). We choose the top 10 features selected by each method, and compare the number of features correctly selected among these top 10 features. For FCBF and HSMB, top 10 features means 10 features first selected into the optimal set. For Relief and FST, top 10 features means 10 features with biggest feature weight. We can see from Table 4.2 that our method included all 4 informative features in the top 10 features, whereas FCBF hits none, HSMB hits only one and Relief hits 2. Out method outperforms alternatives on this synthetic data.

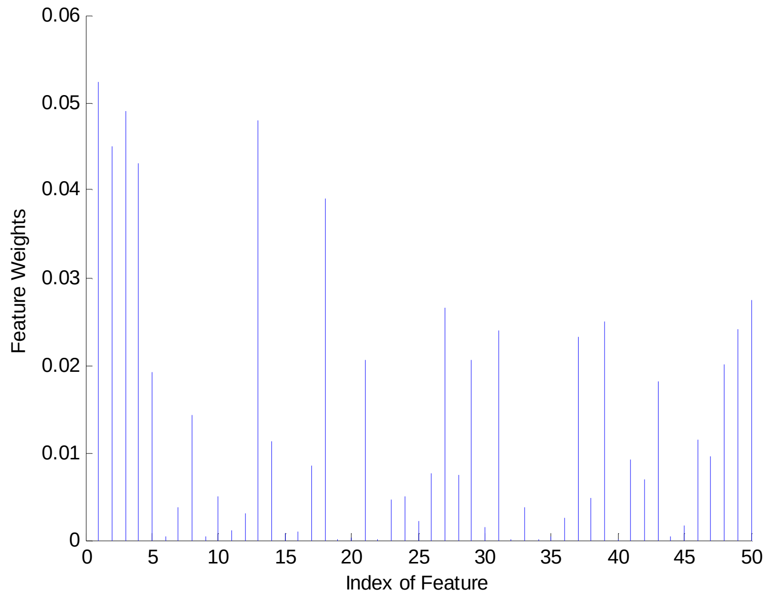

Figure 4.1. Feature weights learned on synthetic data by our **FST** method

TABLE 4.2 NUMBER OF CORRECTLY SELECTED FEATURES AMONG TOP 10 FEATURES

	FCBF	**HSMB**	**Relief**	**FST**
number correct features	0	1	2	4

4.4.2. Results on Two Real Flu Datasets

In this section we compare our method and other alternatives on two real flu datasets that have been used in some other studies [55, 56].

In summary, H3N2 data consists of records of 17 subjects collected at 16 different time steps. H1N1 data consists of records of 24 subjects collected at 16 different time steps. For H3N2 and H1N1 gene expression data, the same 12,023 genes are considered for analysis for each subject at each time step.

Since we don't know in advance which genes among these two datasets are deciding an individual's health status, we evaluate our method and three alternatives in a different way than that applied to Synthetic data. We apply all methods on both data sets, and build the prediction model on selected genes. We compare the accuracy of the prediction models built from different methods. We believe that the selected features tend to be more correct if the prediction model built on these features is more accurate.

For the feature selection and learning-prediction process, we apply leave-one-out schema because of the low number of subjects in both two data set. To avoid overfitting, in each iteration of leave-one-out schema, the training set is used to perform feature selection and learn the prediction model, and the one test subject is only touched in prediction process. We applied a Nearest Neighbor classifier to build the prediction model because it is easy to perform on multivariate temporal gene expression data sets.

The results on H3N3 and H1N1 data sets are listed in Table 4.3 and Table 4.4. Since H1N1 data set is imbalanced data (8 negative subjects and 16 positive subjects). We report sensitivity, specificity, and balanced accuracy to evaluate the results from all methods. Sensitivity measures the proportion of actual positives which are correctly identified as positive (e.g. the percentage of infected subjects who are correctly identified as having virus infected). Sensitivity represents the probability of a positive test given that the patient is ill. Specificity measures the proportion of negatives which are correctly identified as negatives (e.g. the percentage of non-infected subjects who are correctly identified as not having virus infected). Specificity represents the probability of a negative test given that the patient is well. The balanced accuracy is the average of sensitivity and specificity. The balanced accuracy tend to drop the chance that the classifier takes advantage of an imbalanced test set.

TABLE 4.3. RESULTS ON H3N2 DATA

(a) Classification Accuracy (mean ± std)

	All feature	**FCBF**	**HSMB**	**Relief**	**FST**
Sensitivity	0.667 ± 0	0.755 ± 0.242	0.750 ± 0.175	0.875 ± 0.063	**1.000** ± 0
Specificity	0.811 ± 0	0.556 ± 0.046	0.667 ± 0.135	0.778 ± 0.118	**0.889** ± 0.130
Balanced_Accuracy	0.771 ± 0	0.653 ± 0.149	0.708 ± 0.162	0.826 ± 0.065	**0.944** ± 0.064

(b) Number of Selected Features

FCBF	**HSMB**	**Relief**	**FST**
15	50	Top 50	Top 50

TABLE 4.4. RESULTS ON H1N1 DATA

(a) Classification Accuracy(mean ± std)

	All feature	**FCBF**	**HSMB**	**Relief**	**FST**
Sensitivity	0.938 ± 0	0.813 ± 0.068	0.813 ± 0.136	1.000 ± 0	**1.000** ± 0
Specificity	0.125 ± 0	0.375 ± 0.146	0.500 ± 0.128	0.500 ± 0.132	**0.750** ± 0.151
Balanced_Accuracy	0.531 ± 0	0.594 ± 0.102	0.656 ± 0.065	0.750 ± 0.074	**0.875** ± 0.101

(b) Number of Selected Features

FCBF	**HSMB**	**Relief**	**FST**
23	43	Top 43	Top 43

The classification results on H3N3 and H1N1 are shown at the top sub-table of Table 4.3 and Table 4.4. The number of selected features from different methods are shown at the bottom sub-table of Table 4.3 and Table 4.4. **FCBF** and **HSMB** can automatically select the optimal set of features, whereas **Relief** and **FST** are feature ranking features. For comparison, we let **Relief** and **FST** selects the same number of features as the bigger one among the number of features **FCBF** and **HSMB** returns automatically. We repeat experiments 20 times and report the mean ± std values for classification results (sensitivity, specificity, and balanced accuracy).

Table 4.3 shows the results on H3N3 data. We can see there that the accuracy of predictor built on the features selected by out proposed **FST** method outperforms all alternatives including the predictor built on all features. This proves that our **FST** method selects more accurate features. The bottom sub table of Table 4.3 shows that **FCBF** selects the smallest number of features among all methods, which is consistent to the one of widely know drawbacks of **FCBF**: **FCBF** tend to remove features too aggressively.

We got similar results, shown in Table 4.4, on H1N1 to the results on H3N2. Moreover, H1N1 is an unbalanced dataset (with large fraction of positive subjects). We can see from Table 4.4 that if we build a predictor on all features, we will tend to predict most negative subjects as positive subjects. The specificity results from **FCBF**, **HSMB** and **Relief** are also small, because they didn't select most informative features. The predictors built on these selected features suffered from imbalanced data, and treated most negative subjects as positive subjects.

4.4.3. Analysis of Our Results

In follow up experiments we built predictors from Top 1 to Top 1000 selected genes on both H3N3 and H1N1 data sets. The results on H3N2 and on H1N1 data are shown in Figure 4.2 and Figure 4.3, respectively. The results on both Figure 4.2 and

Figure 4.3 are the average results of 20 repeated experiments. To avoid messy, we didn't plot the error baron the figures. The stand derivation of accuracy of 20 experiments on H3N3 data is in the range of [0.045 0.026], whereas the range of [0.050 0.186]. This shows our method is quite standard on both data sets. Actually the results in Table 4.3 and Table 4.4 also show that our method can get similar results while running multiple times.

Figure 4.2 and Figure 4.3 show that, for H3N3 data, we got the highest accuracy by selecting about 25 genes, whereas we got the highest accuracy by selecting about 50 genes on H1N1 data. The results provide evidence that, for these two high dimensional gene expression data, selecting too few genes will result in losing information on the data, whereas selecting too many genes will bring in too much noisy information about the data.

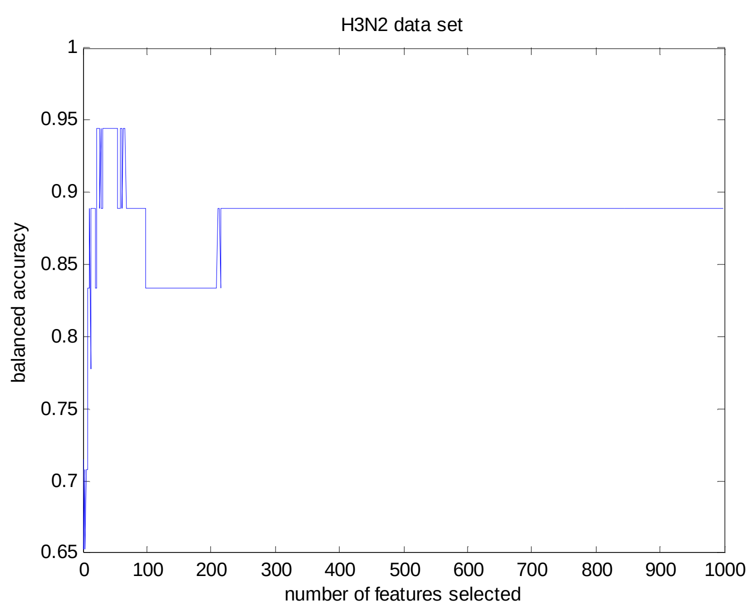

Figure 4.2. Accuracy vs. number of selected features by **FST** method (H3N2 data set).

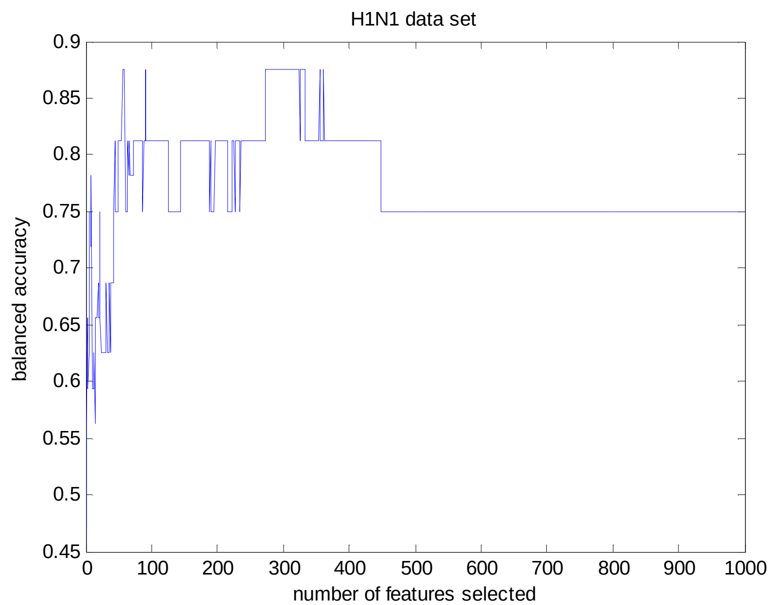

Figure 4.3. Accuracy vs. number of selected features by **FST** method (H1N1 data set).

4.5. Improving Optimization

In this section we further enhance proposed method **FST** in last subsection by improving optimization method. We called the improved method **MSTM**.

4.5.1. Improved Optimization

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 $\mathbf{X}_i$ from D in a weighted feature space as:

$$\rho_D(\mathbf{X}_i \mid \mathbf{w}) = \frac{1}{2}(Tdist(\mathbf{X}_i, nearmiss(\mathbf{X}_i) \mid \mathbf{w}) - Tdist(\mathbf{X}_i, nearhit(\mathbf{X}_i) \mid \mathbf{w})) \tag{3}$$

which is equivalent to:

$$\begin{aligned} \rho_D(\mathbf{X}_i \mid \mathbf{w}) &= \frac{1}{2T_i \times T_M} \sum_{r=1}^{T_i} \sum_{s=1}^{T_M} d(\mathbf{X}_i^{(r)}, nearmiss(\mathbf{X}_i)^{(s)} \mid \mathbf{w}) \\ &\quad - \frac{1}{2T_i \times T_H} \sum_{r=1}^{T_i} \sum_{s=1}^{T_H} d(\mathbf{X}_i^{(r)}, nearhit(\mathbf{X}_i)^{(s)} \mid \mathbf{w}) \\ &= \mathbf{w}^T \boldsymbol{\beta}_i \end{aligned} \tag{4}$$

where T_M and T_H are the number of time steps of nearmiss($\mathbf{X}_i$) and nearhit($\mathbf{X}_i$), respectively; for each instance Xi, the corresponding $\boldsymbol{\beta}_i$ is defined as:

$$\boldsymbol{\beta}_i = \frac{1}{2T_i \times T_M} \sum_{r=1}^{T_i} \sum_{s=1}^{T_M} | \mathbf{X}_i^{(r)} - nearmiss(\mathbf{X}_i)^{(s)} | - \frac{1}{2T_i \times T_H} \sum_{r=1}^{T_i} \sum_{s=1}^{T_H} | \mathbf{X}_i^{(r)} - nearhit(\mathbf{X}_i)^{(s)} | \tag{5}$$

where $|\cdot|$ is the element-wise absolute operator.

One possible problem may exist in the current definition of instance margin. The nearest neighbors we calculate for each instance might not be the real nearest neighbors, since we calculate the nearest neighbor for each instance in the weighted space which changes each time the weights got updated. To solve this problem, we take into account the uncertainty of calculating nearest neighbors when calculating instance margin. 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:

$$U_{\text{nearhit}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}) = \frac{\exp\left(\frac{1}{2T_i \times T_M} \sum_{r=1}^{T_i} \sum_{s=1}^{T_M} d(\mathbf{X}_i^{(r)}, (\mathbf{X}_n)^{(s)} \mid \mathbf{w}) / \sigma\right)}{\sum_j \exp\left(\frac{1}{2T_i \times T_M} \sum_{r=1}^{T_i} \sum_{s=1}^{T_M} d(\mathbf{X}_j^{(r)}, (\mathbf{X}_n)^{(s)} \mid \mathbf{w}) / \sigma\right)}$$
$$\text{where } 1 \le i \le N, i \ne n, y_i = y_n$$
$$\text{and } 1 \le j \le N, y_j = y_n \qquad (6)$$

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:

$$U_{\text{nearmiss}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}) = \frac{\exp\left(\frac{1}{2T_i \times T_H} \sum_{r=1}^{T_i} \sum_{s=1}^{T_H} d(\mathbf{X}_i^{(r)}, (\mathbf{X}_n)^{(s)} \mid \mathbf{w}) / \sigma\right)}{\sum_j \exp\left(\frac{1}{2T_i \times T_H} \sum_{r=1}^{T_i} \sum_{s=1}^{T_H} d(\mathbf{X}_j^{(r)}, (\mathbf{X}_n)^{(s)} \mid \mathbf{w}) / \sigma\right)}$$
$$\text{where } 1 \le i \le N, y_i \ne y_n$$
$$\text{and } 1 \le j \le N, y_j \ne y_n \qquad (7)$$

Please note that distance in equations (6) and (7) 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 **final Temporal margin with uncertainty** as the expectation of the instance margin of $\mathbf{x}_n$, which can be written as:

$$E_{\rho_D}(\mathbf{x}_n \mid \mathbf{w}) = \mathbf{w}^T \mathbf{E}_{\beta_n}$$
$$\text{where}$$
$$\mathbf{E}_{\beta_n} = \sum_{i, when\ y_i \ne y_n} U_{\text{nearmiss}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}) \cdot \frac{1}{2T_i \times T_M} \sum_{r=1}^{T_i} \sum_{s=1}^{T_M} | \mathbf{X}_i^{(r)} - nearmiss(\mathbf{X}_i)^{(s)} |$$
$$- \sum_{i, when\ y_i = y_n} U_{\text{nearhit}}(\mathbf{x}_i \mid \mathbf{x}_n, \mathbf{w}) \cdot \frac{1}{2T_i \times T_H} \sum_{r=1}^{T_i} \sum_{s=1}^{T_H} | \mathbf{X}_i^{(r)} - nearhit(\mathbf{X}_i)^{(s)} | \qquad (8)$$

As we mentioned before, our temporal margin incorporates the uncertainty in calculating two nearest neighbors ($E_{\beta n}$).

We already define the instance margin for each subject $\mathbf{X}_n$. Therefore, we can define the temporal margin of the entire data D that has *I* subjects as the sum of all instance margins, which can be written as:

$$\rho_{D|\mathbf{w}} = \sum_{n=1}^{I} \rho_D(\mathbf{X}_n \mid \mathbf{w}) \tag{9}$$

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:

$$\max_{\mathbf{w}} \sum_{n=1}^{N} E_{\rho_D}(\mathbf{x}_n \mid \mathbf{w}) \quad \text{subject to } \mathbf{w} \geq 0 \tag{10}$$

We followed logistic regression formulation framework. In order to avoid huge values in weight vector ***w***, we add a normalization condition $\|\mathbf{w}\|_1 \leq \theta$. Therefore, we can rewrite the optimization problem as:

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \log(1+\exp(-E_{\rho_D}(\mathbf{x}_n \mid \mathbf{w})) \quad \text{subject to } \mathbf{w} \geq 0, \|\mathbf{w}\|_1 \leq \theta \tag{11}$$

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

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

For each solution to (12), there is a parameter θ, corresponding to the obtained λ in (12), which gives the same solution in (11). Formulation (12) is actually the optimization problem with ℓ_1 regularization. The benefits of adding the ℓ_1 penalty have been well studied [58] and it is shown that the ℓ_1 penalty can effectively handle sparse data and huge amounts of irrelevant features.

4.5.2. Improved Results

To characterize the proposed algorithm, we conducted large-scale experiments on both synthetic and 2 real flu data sets [55, 56]. All experiments of this study were performed on a PC with 3 GB of memory. We compared our proposed **MSTM** algorithm in temporal gene expression data with four traditional feature selection methods (the method proposed in [60] that we call **BAHSIC**, **SIMBA** [50], **Relief** [57] and **FST** [61]) after flattening temporal multivariate data into one single matrix.

For the prediction method, we apply a Nearest Neighbor classifier on all features and select features by different feature selection methods. We compare results on both synthetic data and real data (same data sets as we used in Chapter 4.4).

The results on synthetic data are shown in Figure 1 and Table II. Figure 4.4 shows the feature weights for each feature learned by our proposed **MSTM** and three alternatives. It clearly shows that our method assigns significantly larger weights to the first four features used to decide the Label than to most other features. Moreover, our method applied L1 regularization so that the feature weights learned are sparse (most of feature weights are tend to zero).

Table 4.5 shows the results comparing our method to three alternatives (three alternatives are applied after flattening the temporal data). We choose the top 4 features selected by each method, and compare the number of features correctly selected among these top 4 features. Top 10 features means 10 features with biggest feature weight. We can see from Table II that our method included all 4 informative features in the top 4 features, whereas **SIMBA** hits 2, **BAHSIC** hits only one and **Relief** hits 2. Out method outperforms alternatives on this synthetic data.

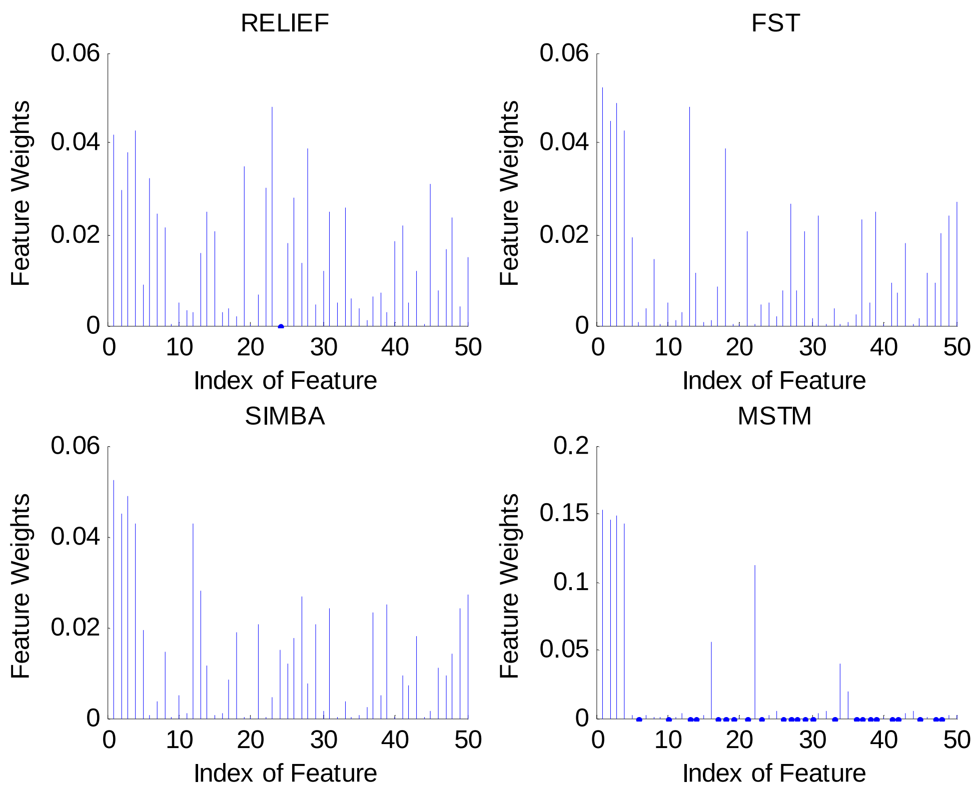

Figure 4.4. MSTM Results-Feature weights learned on synthetic data.

TABLE 4.5 NUMBER OF CORRECTLY SELECTED FEATURES AMONG TOP 4 FEATURES

	Relief	**FST**	**Simba**	**MSTM**
# correct features	2	2	3	4

The results on the same real flu data sets as we used in Chapter 4.4 are shown in Table 4.6 and Table 4.7. We repeat experiments 20 times and report the mean ± std values for classification results (sensitivity, specificity, and balanced accuracy). We can see there that the accuracy of the predictor built on the features selected by our proposed

MSTM method outperforms all alternatives including the predictor built on all features. This proves that our **MSTM** method selects more accurate features.

TABLE 4.6. MSTM RESULTS ON H3N2 DATA

(a) Classification Accuracy (mean ± std)

	All feature	**BAHSIC**	**Relief**	**Simba**	**FST**	**MSTM**
Sensitivity	0.667 ± 0	0.735 ± 0.202	0.875 ± 0.063	0.882 ± 0.073	1.000 ± 0	**1.000 ± 0**
Specificity	0.811 ± 0	0.582 ± 0.056	0.778 ± 0.118	0.763 ± 0.053	0.889 ± 0.130	**0.922 ± 0.150**
Balanced_Accuracy	0.771 ± 0	0.659 ± 0.129	0.826 ± 0.065	0.823 ± 0.063	0.944 ± 0.064	**0.961 ± 0.084**

(b) Number of Selected Features

BAHSIC	**Relief**	**Simba**	**FST**	**MSTM**
217	154	135	50	55

TABLE 4.7. MSTM RESULTS ON H1N1 DATA

(a) Classification Accuracy (mean ± std)

	All feature	**BAHSIC**	**Relief**	**Simba**	**FST**	**MSTM**
Sensitivity	0.938 ± 0	0.806 ± 0.052	1.000 ± 0	0.948 ± 0.003	1.000 ± 0	**1.000 ± 0**
Specificity	0.125 ± 0	0.405 ± 0.131	0.500 ± 0.132	0.605 ± 0.163	0.750 ± 0.151	**0.801 ± 0.131**
Balanced_Accuracy	0.531 ± 0	0.606 ± 0.092	0.750 ± 0.074	0.777 ± 0.085	0.875 ± 0.101	**0.901 ± 0.065**

(b) Number of Selected Features

BAHSIC	**Relief**	**Simba**	**FST**	**MSTM**
346	121	141	43	27

The number of selected features from different methods is shown at the bottom sub-table of Table 4.6 and Table 4.7. Our **MSTM** method can automatically select the optimal feature set by eliminating features with weight zero. **MSTM** selected 55 genes out of 12,023 features on *H3N3*, and 27 genes out of 12,023 features on *H1N1*. However, **FST**, **Simba, BAHSIC** and **Relief** cannot select the optimal feature set automatically, since they are all feature ranking methods. We report the number of top features where we get the highest accuracy for these three methods. The number of selected features is listed at the bottom of Table 4.6 and Table 4.7. Our method forces the weights of most irrelevant features to be zero, and it therefore selects much fewer features than the alternatives.

CONCLUSION

Data sets with irrelevant and redundant features and large fraction of missing values are common in the real life application. Learning such data usually requires some preprocess such as selecting informative features and imputing missing values based on observed data. These processes can provide more accurate and more efficient prediction as well as better understanding of the data distribution. In this presentation I will describe my previous work in both of these aspects and also my current work on feature selection in incomplete dataset without imputing missing values.

My previous work focus on handling such data in a straightforward way: imputing missing values first, and then applying traditional feature selection method to select informative features. We proposed two novel methods, one for imputing missing values and the other one for selecting informative features. We proposed a new method that imputes the missing attributes by exploiting temporal correlation of attributes, correlations among multiple attributes collected at the same time and space, and spatial correlations among attributes from multiple sources. The proposed feature selection method aims to find a minimum subset of the most informative variables for classification/regression by efficiently approximating the Markov Blanket which is a set of variables that can shield a certain variable from the target.

The straightforward method is imputing the missing value first, then applying traditional feature selection method on the enlarged data. However, imputation methods only work a lot better when data is missing completely at random, when fraction of missing values is small, or when there is prior knowledge about the data distribution. My recent work is aimed to feature selection in incomplete data without imputation. In the new method, we show how to perform feature selection directly, without imputing missing values. We define the objective function of the uncertainty margin-based feature selection method to maximize each instance's uncertainty margin in its own relevant subspace. In optimization, we take into account the uncertainty of each instance due to the missing values. The experimental 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 alternative 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 outperforms the alternatives, which impute missing values first.

In the last part of my dissertation, I introduce my method handling more challenging situation where the high-dimensional data varies in time. Traditional way to handle such data is to flatten temporal data into single static data matrix, and then applying traditional feature selection method. I proposed a method that avoids flattening the data in advance so that it can keep the dynamics in time. We propose a way to measure the distance between multivariate time series data from two instances. Therefore, we define the new objective function based on the temporal margin of each data instance. A fixed-point gradient descent method is proposed to solve the formulated objective function to learn the optimal feature weights.

BIBLIOGRAPHY

[1] Lozlov, A. V., Singh, and J. P. Sensitivities, ”An alternative to conditional probabilities for Bayesian Belief networks,” *In proceedings of the eleventh annual conference on Uncertainty in Artificial Intelligence.*, 1995, pp.376-385.

[2] Scholkopf, B. and Smola, A. “Learning with Kernels”, *Cambridge, MA, MIT Press*, 2002.

[3] Gretton, A., Bousquet, O., Smola, A., and Scholkopf, B., “Measuring statistical dependence with Hilbert-Schmidt Norms”, *Algorithmic Learning Theory*, 2005, pp. 63-78.

[4] Koller, D. and Sahami, M., “Toward optimal feature selection”, *International conference on Machine Learning*, 1996, pp. 284-292.

[5] Margaritis, D., and Thrun, S. “Bayesian Network Induction via Local Neighborhoods”, *Neural Information Processing Systems*, 1999, pp. 12: 505-511.

[6] Kohave, R. and John, G., “Wrappers for Feature Subset Selection”, *Artificial Intelligence,* 1997, 1-2: 273-324.

[7] Guyon, I. and Elisseeff A., “An Introduction to Variable and Feature Selection”, *Journal of Machine Learning Research*, 2000, 3:1157-1182.

[8] Tsamardinos, I. and Aliferis, C., “Toward Principled Feature Selection: Relevancy, Filters, and Wrappers”, *International Workshop on Artificial Intelligence and Statistics*, 2003.

[9] Fukumizu, K., Bach, F. R., and Jordan, M. I. “Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces” *Journal of Machine Learning Research,* 2004, 5: 73-99.

[10] Kim, Y., W. Nich Street, and Menczer, F., "Feature Selection for Unsupervised Learning via Evolutionary Search" *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 365-369.

[11] Shen J., Li, L. and Wong, W. "Markov Blanket Feature Selection for Support Vector Machines" *AAAI Conference on Artificial Intelligence*, 2008.

[12] Song, L., Smola, A., Gretton, A. and Borgwardt, K. M. "A Dependence Maximization View of Clustering" *International Conference on Machine Learning*, Corvallis, OR, 2007.

[13] Song, L., Smola, A., Gretton, A., Borgwardt, K. M., and Bego, J. "Supervised Feature Selection via Dependence Estimation" *International Conference on Machine Learning*, 2007.

[14] Yu, L., and Liu, H. "Feature Selection for High-dimensional Data: A Fast Correlation-based Filter Solution" *Twentieth International Conference on Machine Learning*, 2003, pp. 856-863.

[15] Yu, L., and Liu, H. "Efficient Feature Selection via Analysis of Relevance and Redundancy", *Journal of Machine Learning Research*, 2004, 5: 1205-1224.

[16] Yaramakala, S. and Margaritis, D. "Speculative Markov Blanket Discovery for Optimal Feature Selection" *International Conference on Machine Learning*, 2005.

[17] Ayuyev, V., Jupin, J., Harris, P., Obradovic, Z. "Dynamic clustering based estimation of missing values in mixed type data," *Proc. 11th Int'l Conf. Data Warehousing and Knowledge Discovery*, Linz, Austria, 2009, pp. 366-377.

[18] Beckers, J. and Rixen, M., "EOF calculations and data filling from incomplete oceanographic data sets," *J. Atmos. Ocean. Technol*., vol. 20, 2003, pp. 1839–1856,.

[19] Bishop, C. M., "Pattern Recognition and Machine Learning," *Springer*, Aug. 2006.

[20] http://aeronet.gsfc.nasa.gov/new_web

[21] http://modis.gsfc.nasa.gov

[22] Khan, R.A., Nelson, D.L., Garay, M.J., Levy, R.C., Bull, M.A., Diner, D.J., Martonchik, J.V., Paradise, S.R., Hansen, E.G., Remer, L.A.,"MISR aerosol

product attributes and statistical comparisons with MODIS," *IEEE Tran. Geoscience and Remote Sensing,* vol. 47(2), no, 12, 2009, pp. 4095-4114.,.

[23] Kondrashov, .D and Ghil, M, "Spatio-temporal filling of missing points in geophysical data sets," *Nonlinear Processes Geophys*, 13, 2006, pp. 151-159.

[24] Li, L., McCann, J., Pollard, N., Faloutsos,C. "DynaMMo: Mining and summarization of Coevolving Sequences with missing values," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, , Paris, France, 2009, pp. 507-516.

[25] Radosavljevic, V., Vucetic, S., Obradovic, Z. "A data mining technique for aerosol retrieval across multiple accuracy measures," *IEEE Geoscience and Remote Sensing Letters,* vol. 7, no. 2, 2010, pp. 411-415.

[26] 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.

[27] 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.

[28] 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.

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

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

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

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

[33] Kohave, R. and John, G. 1997. Wrappers for Feature Subset Selection. *Artificial Intelligence*, vol 1-2, pp. 273-324.

[34] Koller, D. and Sahami, M. 1996. Toward Optimal Feature Selection. In *International Conference on Machine Learning*, pp. 284-292 .

[35] Kress, R. 1998. Numerical Analysis. *New York:springger-Verlag*.

[36] 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.*

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

[38] Lou, Q. and Obradovic, Z. 2010. Feature Selection by Approximating the Markov Blanket in a Kernel-Induced Space. *Europe Conference on Artificial Intelligence*.

[39] Pannagadatta, S. K. Bhattacharyya, C. and Smola, A. J. 2006. Second Order Cone Programming Approaches for Handling Missing and Uncertain Data. *Journal of Machine Learning Research*, vol 7, pp. 1283-1314.

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

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

[42] 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*.

[43] 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.

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

[45] 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.*

[46] 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.

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

[48] Kohave, R. and John, G.. Wrappers for Feature Subset Selection. *Artificial Intelligence*, vol 1-2, 1997, pp. 273-324

[49] Lou, Q, and Obradovic, Z. (in press) "Margin-Based Feature Selection in Incomplete Data,". In Proc. Of 26th AAAI Conference on Artificial Intelligence (AAAI-12). July 2012, Toronto, Ontario, Canada

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

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

[52] Lou, Q, and Obradovic, Z, "Feature Selection by Approximating the Markov Blanekt in a Kernel-Induced Space", Proc. 19th European Conference on Artificial Intelligence, Lisbon, Portugal, 2010,

[53] Lou, Q, parkman, H.P., Jacobs, M.R, Krynetskiy, E. and Obradovic, Z. "Exploring Genetic Variability in Drug Therapy by Selecting a Minimum Subset of the Most Informative Single Nucleotide Polymorphisms through Approximation of a Markov Blanket in a Kernel-induced Space," Proc. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, San Deigo, CA, May 2012.

[54] Song, L, Smola, A, Gretton, A. and Borgwardt, K.L.. "A dependence maximization view of clustering", *In Proceedings of the 24th International Conference on Machine Learning,* Corvallis, OR, 2007.

[55] Chen, B., Chen, M., Paisley, J., Zass, A., Woods, C., Ginsburg, G.S.,Lucas, J., Dunson, D., and Carin, L. "Bayesian Inference of the Number of Factors in

Gene-expression Analysis: Application to Human Virus Challenge Studies,", BMC bioinformatics, 2010.

[56] Chen, M., Zaas, A., Woods, C., Ginsburg, G.S., Lucas, J., Dunson, D., and Carin, L. "Predicting Viral Infection From High-Dimensional Biomarkers Trajectories" Journal of the American Statistical Association, vol. 106, No. 496. December 2011.

[57] Sun, J. and Li, J. "Iterative RELIEF for Feature Weighting." In 23rd International Conference on Machine Learning, 2006, Pittsb.

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

[59] Kress, R. "Numerical Analysis." *New York: springger-Verlag*. 1998.

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

[61] Lou, Q and Obradovic, Z. "Prediciting Viral Infection by Selecting Informative Biomarkers From Temporal High-Dimensional Gene Expression Data" *IEEE International Confenece on Bioinformatics and Biomedicine,* October 2012, Philadelphia, USA