Feature Subset Selection on Multivariate Time Series with Extremely Large Spatial Features *

Hyunjin Yoon and Cyrus Shahabi
Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0781
{hjy,cshahabi}@usc.edu

Abstract

Several spatio-temporal data collected in many applications, such as fMRI data in medical applications, can be represented as a Multivariate Time Series (MTS) matrix with m rows (capturing the spatial features) and n columns (capturing the temporal observations). Any data mining task such as clustering or classification on MTS datasets are usually hindered by the large size (i.e., dimensions) of these MTS items. In order to reduce the dimensions without losing the useful discriminative features of the dataset, feature selection techniques are usually preferred by domain experts since the relation of the selected subset of features to the originally acquired features is maintained. In this paper, we propose a new feature selection technique for MTS datasets where their spatial features (i.e., number of rows) are much larger than their temporal observations (i.e., number of columns), or $m \gg n$. Our approach is based on Principal Component Analysis, Recursive Feature Elimination and Support Vector Machines. Our empirical results on real-world datasets show that our technique significantly outperforms the closest competitor technique.

1 Introduction

Feature subset selection (FSS) is a pre-processing technique to identify a subset of $original$ input features (or variables) from a given dataset by removing irrelevant and/or redundant ones [1]. Feature extraction (FE) is, however, to derive new features by linearly/non-linearly mapping the original input features into more effective ones. Both FSS and FE aim at providing better features of less numbers to improve the computational cost and the generalization performance of subsequent predictors (e.g., classifier). As contrary to FE that still requires all the original input features to be measured and stored for the mapping, FSS is more cost-effective in that only the selected features can be acquired after the identification, discarding all the other features for good [2].

Multivariate time series (MTS) is a series of observations, $\mathbf{x}_i(t); [i = 2, \cdots, m; t = 1, \cdots, n]$, made sequentially through time where i indexes the variables measured at each time point t. A natural representation of a single MTS is therefore an $m \times n$ matrix and a set of such data matrices with a fixed m but a variable n is the type of dataset in which we are interested. Note that MTS nicely represents *spatio-temporal* data since the observed variables (the m rows of the matrix) are in general acquired from sensors spread over a particular region and their values (the n columns of the matrix) are measured through definite time.

MTS is in general extremely high dimensional data. For example, in the EEG dataset [4] where 39 electrodes measured brain signals at 256Hz sampling rate during a 5-second imaginary task, each MTS becomes a matrix of 39 $\times$ 1280 dimensions, equivalently to a 49,920 dimensional vector. In this paper, we propose a feature subset selection method for MTS datasets to reduce their dimensions. In addition to the aforementioned advantage of FSS over FE, selecting relevant original variables helps to make insightful interpretation and easier verification in the context of the original application domain. For example, in the EEG data, the selected original features (i.e., electrodes or channels) can be exploited to localize the *neural correlates*, which are not known in such detail in the neuroscience literature [4]

Recursive feature elimination (RFE) embedding support vector machine (SVM) classifiers (SVM-RFE) [2] have become a popular feature subset selection technique for many datasets including MTS [4][2][6][11]. SVM-RFE starts

*This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC), IIS-0238560 (PECASE) and IIS-0307908, and unrestricted cash gifts from Microsoft and Google. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

with all the features and repeatedly removes a feature at a time based on a ranking criterion until the required number of features are left. In order to utilize SVM-RFE on MTS dataset, each MTS data matrix in the set must be first transformed into one row or column vector while retaining the correspondence to the original features - we called this process *vectorization* [11]. In [4], each row of 39 channel EEG data was encoded by an autoregressive (AR) model of order 3, resulting in a 117 dimensional vector. In our own previous work named *Corona* [11], we vectorized each MTS using its sample correlation matrix in order to explicitly incorporate the correlation information among the original features, which was presumably unconsidered in the former method. Empirical evidence shows that using the correlation matrix is a simple yet very effective vectorization method and the performance of subsequent classification is strongly affected by not only the selected features but also the vetorization itself, i.e., how each MTS data is encoded as an input to the SVM classifier. In practice, vectorization using the sample correlation matrix is however limited by the number of features m and unfeasible for the MTS dataset where m is over several hundreds and thousands due to its quadratic complexity to m. For example, the fMRI dataset [5] in which about 5000 voxels are measured every 0.5 second during an 8-second cognitive task requires approximately 200MB memory just to load a single 5000×5000 correlation matrix. Even in the case where the space complexity is not an issue, the resulting sample correlation matrix is an *unstable* estimate because the number of features m (≈ 5000) is much larger than the number of observations n (≈ 16), which is well known as the *undersampled* problem [1].

In this paper, we propose another extension of SVM-RFE for MTS, named RFE-Loft (**R**ecursive **F**eature **E**limination using Principal Component **Lo**adings as **F**ea**t**ures), which exploits an alternate vectorization method suited for the MTS data with extremely larger number of *spatial* features as compared to the number of *temporal* observations (i.e., $m \gg n$). Interestingly, the maximum benefit of feature subset selection can be gained from this type of MTS datasets (e.g., fMRI data) with an extremely high *spatial* resolution. In addition, we incorporate a new feature ranking criterion in the RFE procedure. While the ranking criterion used by SVM-RFE is determined by a single SVM classifier trained over all the training data samples, the proposed score criterion is based on multiple SVM classifiers obtained from a cross-validation procedure in order to make the feature ranking (and thus selection) *generalized* even for the unseen data samples.

The remainder of this paper is organized as follows. Section 2 discusses the background. Our proposed method is described in Section 3, followed by the experiments and results in Section 4. Conclusions are presented in Section 5.

2 Background

RFE-Loft utilizes the principal components for the vectorization and SVM-RFE with a new ranking criterion for the feature subset selection of MTS datasets, which are briefly described in this section.

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a process to identify the directions called principal components (PCs) subject to being uncorrelated with each other, which best accounts for the *variability* of the underlying data in non-increasing order. Geometrically, a principal component is a linear combination of original variables. Let the original variables be denoted by $x_1, x_2, \ldots, x_m$, then a principal component takes the form $u = \sum_i^m l_i x_i$, where $l_i (i = 1, 2, \ldots, m)$ are often referred to as the *principal component loadings* and can be interpreted as the contributions or weights of original features *loaded* on determining the principal directions.

In practice, PCA is performed by applying Singular Value Decomposition (SVD) to either a covariance matrix or a correlation matrix of an input data matrix. Let A be a mean-centered MTS data of $m \times n$ dimensions and AA^T be roughly its $m \times m$ sample covariance matrix. Then, SVD decomposes the real, symmetric matrix AA^T as follows:

$$\mathrm{AA}^T = U \Lambda U^T \tag{1}$$

where the columns of orthonormal matrix U are the m number of principal components and a diagonal matrix Λ has the corresponding variances along the diagonal. Equivalently, the principal components and the corresponding variances of A are the eigenvectors and the eigenvalues of AA^T, respectively, since $\mathrm{AA}^T U = U\Lambda$ is satisfied [1]. Computing the principal components by SVD on the sample covariance matrix will scale roughly as $O(nm^2 + m^3)$ [3]. Therefore, when the number of features m is large, the computation is not manageable. In the case where $m \gg n$, the principal components can be computed instead from the SVD on $\mathrm{A}^T\mathrm{A}$ of $n \times n$ dimensions as follows [7]:

$$\mathrm{A}^T\mathrm{A} = V S V^T \tag{2}$$

Post-multiply both sides by V, followed by pre-multiplying both sides by A,

$$\mathrm{A}^T\mathrm{A}V = VS \Rightarrow \mathrm{AA}^T\mathrm{A}V = \mathrm{A}VS \tag{3}$$

from which we can see that $\mathrm{A}V$ and S are the eigenvectors and the eigenvalues of AA^T, hence the principal components and the corresponding variances of A, respectively. Therefore, the principal components of A, i.e., U in Equation 1, can be finally obtained by $U = \mathrm{A}V$, where V are the

eigenvectors of $A^T A$ as in Equation 2. Note that the number of principal components solved by this computation is thus at most n ($n \ll m$) while it is m by the regular PCA computation in Equation 1. However, the rest $m-n$ principal components are not *informative* anyway even in the case where they can be calculated because their corresponding eigenvalues (variances) are all zero [7].

2.2 Support Vector Machine Classifier

Support Vector Machine (SVM) classifier[1] is a binary classification algorithm by Vapnik [8]. Geometrically, SVM classifier seeks for an *optimal hyperplane* that linearly separates two classes by maximizing the *margin*, i.e., the distance to the closest data point from both classes. This hyperplane can be described as a decision function as follows [8]: $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, where $\mathbf{w}$ is the norm or the weight vector of the hyperplane $f(\mathbf{x})$ and $b/\|\mathbf{w}\|$ is the distance from the origin to the hyperplane. Given a new data $\mathbf{x}_i$, the sign of $f(\mathbf{x}_i)$ determines the class of $\mathbf{x}_i$. Finding the maximum-margin hyperplane of a SVM classifier can be formulated into the following optimization problem [8]:

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{4}$$

under the constraints $y_i f(\mathbf{x}_i) \geq 1 - \xi_i (i = 1, \ldots, n)$, where $\mathbf{x}_i$ are all the training data and $y_i \in \{-1, 1\}$ are their corresponding class labels. Once this problem is solved using the Lagrangian theory, the optimal weight vector $\mathbf{w}$ is of the form:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{5}$$

where α_i are Lagrangian multipliers.

2.3 Recursive Feature Elimination

Based on SVM, Guyon *et al* [2] proposed a feature subset selection method called Recursive Feature Elimination (RFE). SVM-RFE is a *stepwise backward feature elimination* method [3]. The procedure of SVM-RFE can be briefly described as follows: 1) train a single SVM classifier with all the training data, 2) rank the features based on a ranking criterion, 3) eliminate the feature with the lowest ranking score, and 4) repeat until the required number of features are retained [2].

In order to rank features, SVM-RFE utilizes the *sensitivity analysis* based on the weight vector $\mathbf{w}$ of the trained SVM classifier. That is, at each iteration, SVM-RFE eliminates the feature whose removal minimizes the *change* of the following object function [2]: $J = (1/2)\|\mathbf{w}\|^2$, equivalently, $(1/2)\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$ by Equation 5. Hence, for a given feature k, its sensitivity or ranking score can be measured by the gradient of this object function with respect to k, which can be computed by introducing a virtual scaling factor v as follows [6]:

$$\frac{\partial J}{\partial v_k} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \frac{\partial(v_i\mathbf{x}_i \cdot v_j\mathbf{x}_j)}{\partial v_k} \tag{6}$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (2 v_k x_k^2) = w_k^2 \tag{7}$$

where the virtual scaling factors v_i and v_j are 1 if $i, j = k$, and 0 otherwise. Hence, the squared coefficients of the weight vector $\mathbf{w}$ of the trained SVM classifier are used as the feature ranking criterion of SVM-RFE. Intuitively, the feature with the minimum sensitivity would least influence the weight vector of the optimal hyperplane, and is therefore to be removed [11].

3 Proposed Method

In this section we describe RFE-Loft, which is an extension of SVM-RFE for MTS datasets. In particular, the target MTS dataset consists of N *labeled* MTS data items, each of which is an $m \times n$ dimensional matrix, where the number of original spatial features m are extremely larger than the number of temporal observations n (i.e., $m \gg n$) and m is fixed yet n is not necessarily the same across different MTS items. Given such an MTS dataset, RFE-Loft aims at selecting k number of original features out of m that collectively have the most discriminant power for the given classification problem. RFE-Loft first encodes each MTS data into a vector using principal components and then recursively eliminates one feature at a time until k features are left as in SVM-RFE. The details of the proposed method are presented in the following sections.

3.1 RFE-Loft: Vectorization

RFE-Loft utilizes the principal components to vectorize each MTS data matrix as an input to the RFE procedure. The intuition behind using the principal components for the vectorization comes from our previous work, which has shown that the similarity between two MTS data is effectively measured by comparing their principal components weighted by the corresponding variances [10]. The interpretation of principal component loadings as the weights of original features plays a significant role in the *unsupervised* feature subset selection [12].

The principal components of an $m \times n$ MTS data A, where m is much larger than n, can be obtained by the sin-

[1] In this paper, SVMs with linear kernel are considered. Therefore, the term SVM classifier is used to denote a linear SVM classifier except where specified otherwise.

gular value decomposition on its $\mathrm{A}^T\mathrm{A}$ matrix as in Equation 2∼3 rather than on its sample covariance matrix $\mathrm{A}\mathrm{A}^T$, which makes the PCA computation quite manageable by greatly reducing the required calculations from the order of features m to the order of observations n. For example, for the fMRI data of 5000 × 16 dimensions in [5], its PCA computation is performed on the 16 × 16 symmetric matrix as opposed to the 5000 × 5000 sample covariance matrix, which results in at most 16 principal components of 5000 dimensions. Note that as we discussed in Section 2.1 and Equation 2– 3, the obtained eigenvectors and eigenvalues are still for the original $\mathrm{A}\mathrm{A}^T$. We just used $\mathrm{A}^T\mathrm{A}$ as an intermediate step to reduce the size of data structures we are dealing with. Once the principal components of each MTS data are obtained, only the principal components with non-zero corresponding variance will be utilized in the subsequent process. Note that the number of obtained principal components with non-zero variance of an MTS data depends on its *rank* and it is thus not necessarily equivalent across all the MTS items in the set. RFE-Loft simply takes the minimum value among all the obtained numbers from the MTS dataset. Hence, the entire target MTS dataset with N MTS data items is finally encoded into an $N \times pm$ matrix, where p is the minimum number of principal components and m is the number of original features. We refer to this matrix as a *vectorized MTS matrix*, where several principal component loadings are associated with one original feature. This matrix will be utilized in the subsequent recursive feature elimination step.

3.2 RFE-Loft: Recursive Feature Elimination

Recall that at each iteration of SVM-RFE, a single SVM classifier is trained with all the data samples and the feature with the minimum squared coefficient of the weight vector $\mathbf{w}$ (i.e., w_i^2) of the obtained hyperplane is eliminated. The limitation of this approach is that the obtained weight vector $\mathbf{w}$ that SVM-RFE ranking criterion is based on might overfit to the training data and not be *generalized* to the unseen testing data samples. To overcome this, RFE-Loft utilizes multiple SVM classifiers obtained from a cross-validation procedure to approximate the weight vector $\mathbf{w}$ with high *generalization* ability. Intuitively, the ranking criterion based on multiple SVM classifiers *stabilizes* the feature ranking process, and hence the feature elimination.

Cross-validation (CV) is an effective practice to estimate the generalization performance of a classifier [1]. For example, n-fold CV randomly divides all the data samples into n folds of nearly equal size, from which one fold is taken out, all the left (n-1) folds are used to train a classifier, and the trained model is tested on the one being left out yielding a CV test error. This is repeated n times and the resulting n CV test errors are averaged to be reported as the estimated generalization performance. In our experiment, we utilized leave-one-out CV (LOO-CV), which is a special case of n-fold CV where n equals to the number of data samples.

Algorithm 1 RFE-Loft

Require: labeled MTS dataset,
N {number of MTS data in the set}, m {number of original features}, p {number of principal component loadings per feature}, k {required number of features}

$R \leftarrow []$, $S \leftarrow [1, \ldots, m]$, $cvErr \leftarrow []$;
$MTS_{vec} \leftarrow$ Vectorization of MTS dataset;
for i = 1 to N **do**
  $trMTS \leftarrow MTS_{vec}$ with ith row being left out;
  $tsMTS \leftarrow i$th row of MTS_{vec};
  $[model_i, cvW_i] \leftarrow$ Train a SVM classifier with $trMTS$;
  $tsErr(i,1) \leftarrow$ Test $model_i$ on $tsMTS$;
  $W \leftarrow [W;\ (1 - tsErr(i,1)) \times (cvW_i)^2]$;
end for
$cvErr \leftarrow [cvErr;\ \text{Mean}(tsErr)]$;
for i = 1 to $(p \times m)$ **do**
  $meanW_i \leftarrow \text{Mean}(W(:,i))$;
  $stdW_i \leftarrow \text{Std}(W(:,i))$;
  $RC(1,i) \leftarrow meanW_i / stdW_i$;
end for
for i = 1 to m **do**
  $RC_{agg}(1,i) \leftarrow max(RC(1, [i\ \ (1*m+i)\ \ (2*m+i) \ldots ((p-1)*m+i)]))$;
end for
$f \leftarrow$ feature with the lowest score in RC_{agg};
$R \leftarrow [f\ R]$;
$S \leftarrow S$ - $[f]$;
Repeat until k variables remain in S;

Let $\mathbf{w}_k$ be the weight vector of the SVM classifier obtained from the LOO-CV procedure with kth data sample being left out, ϵ_k be the corresponding CV test error, and w_{ki} be the corresponding weight coefficient value associated with the ith feature. Subsequently, the ranking score of ith feature is computed by aggregating the multiple weight vector coefficients w_{ki} weighed by the corresponding CV test accuracy $(1 - \epsilon_k)$ as follows:

$$RC_i = \frac{\overline{w_i^2}}{\sigma_{w_i^2}} = \frac{\frac{1}{n}\sum_{k=1}^{n}(1-\epsilon_k)w_{ki}^2}{\sqrt{\frac{\sum_{k=1}^{n}((1-\epsilon_k)w_{ki}^2 - \overline{w_i^2})^2}{n-1}}} \quad (8)$$

where $\overline{w_i^2}$ and $\sigma_{w_i^2}$ are mean and standard deviation of the squared weight vector coefficients weighted by the corresponding CV test accuracy, respectively. Note that in LOO-CV, the CV test error is either 0 or 1 since only one data sample taken out is tested with the trained model. Consequently, only the weight vectors that correctly predicts the test data sample are incorporated in the score computation. Intuitively, this new ranking score captures how consistent and how sensitive a feature's influence is on multiple dis-

criminant hyperplanes. Algorithm 1 describes how the features are scored based on the new ranking criterion in Lines 3–15. In general, the cross-validation is often performed to monitor the generalization performance of ranked features. As RFE-Loft already computes the cross-validation error during the feature ranking step as in Line 10 in Algorithm 1, the proposed ranking criterion based on multiple SVM classifiers from a cross-validation procedure does not impose any extra computation cost.

Note that the features that have been scored so far by the new ranking criterion are principal component loadings, not the original input features. In order to determine the ranks of the original input *variables*, all the weights of the *features*, i.e., the principal component loadings, with which the ith *variable* is associated, are aggregated and one score is obtained per variable. Finally, the variable to eliminate is decided based on this aggregated value. RFE-Loft takes the *greedy* approach as in Corona [11], and identifies a variable whose maximum ranking score of its corresponding encoded features (i.e., principal component loadings) is the minimum among the maximum ranking scores of all the variables (Lines 17, 19 in Algorithm 1). This variable whose maximum ranking score is the minimum is then to be removed. The entire feature elimination process is repeated until the required number of original features are left or all the variables have been ranked.

4 Performance Evaluation

In order to evaluate the effectiveness of RFE-Loft, we conducted experiments on the fMRI datasets[2] [5], which is a real-world dataset collected in Carnegie Mellon University's Center for Cognitive Brain Imaging (CCBI) to study the human brain activation in the cortex during a high-level cognitive task (i.e., sentence comprehension). In particular, the fMRI data is a series of brain images consisting of several thousand voxels ($\approx$ 5000) scanned at the rate of 2Hz for 8 seconds while a subject performs either a sentence or a picture comprehension task. Therefore, one training data sample is approximately a 5000×16 MTS data matrix with a label of either 'sentence' or 'picture'. Table 1 shows the details of fMRI datasets collected from 6 subjects, each of which was individually evaluated in our experiment due to the variability among subjects. The number of features (voxels) is different across the subjects yet the number of training data samples is the same as 80, i.e., 40 per class. In addition, specific regions within the brain of each subject, referred to as regions of interest (ROIs), have been anatomically defined in each of datasets [5].

First, RFE-Loft was applied to each of 6 fMRI datasets, where the resulting number of principal components with non-zero variance used for vectorization was 15 (out of 16) across all 6 fMRI datasets. Recall that the ranking criterion of RFE-Loft is based on the multiple SVM classifiers obtained from a leave-one-out cross-validation (LOO-CV). Therefore, RFE-Loft does not require any extra cross-validation procedure to report the CV classification error of each selected feature subset. However, in the competing technique (RFE-ARF), where the autoregressive (AR) fit coefficients of order 3 using the forward backward linear prediction [4] are used for the MTS vectorization and RFE with a single SVM classifier is used for the feature elimination [4], the LOO-CV must be performed to report the classification error after obtaining a subset of features. Subsequently, we compared the classification performance of RFE-Loft with those of RFE-ARF, using only the features in ROIs (ROI), as well as with all the available features (ALL). In addition, in order to explore whether the performance gain of RFE-Loft comes from the new vectorization using principal components (PCs) or the new criterion using multiple SVM classifiers, a combination of the new vectorization using PCs and the original ranking criterion using a single SVM classifier was also performed (RFE-noCV). Note that in order to speed up the feature subset selection procedure in our experiments, we eliminated the half of remaining features at each iteration at a possible performance degradation and obtained the corresponding cross-validation errors. The algorithms of RFE-Loft as well as other methods are implemented in $Matlab^{TM}$ using *The Spider* package [9]. For the classifier, SVM classifier with linear kernel is consistently used in every method, where the hyperparameter C has been set sufficiently high (C=100000) in order to keep training error low as in [6]. The entire experiments were performed on a machine with Pentium IV 3.2GHz CPU and 1 GB of RAM.

Subject ID	# of features	# of features in ROI
04799	4949	154
04820	5015	229
04847	4698	215
05675	5135	77
05680	5062	71
05710	4634	155

Table 1. fMRI datasets for experiments.

In Figure 1, the six plots, each corresponding to one subject, show the individual classification performance of the different numbers of features selected by RFE-Loft and the other comparative methods. The X axis is the number of selected features, i.e., the remaining features at each iterated elimination, and the Y axis is the classification error. As shown in Figure 1, the number of features can be significantly reduced from around 5000 down to 9 in subject 04799 and 8 in the other subjects to achieve the minimum

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www/

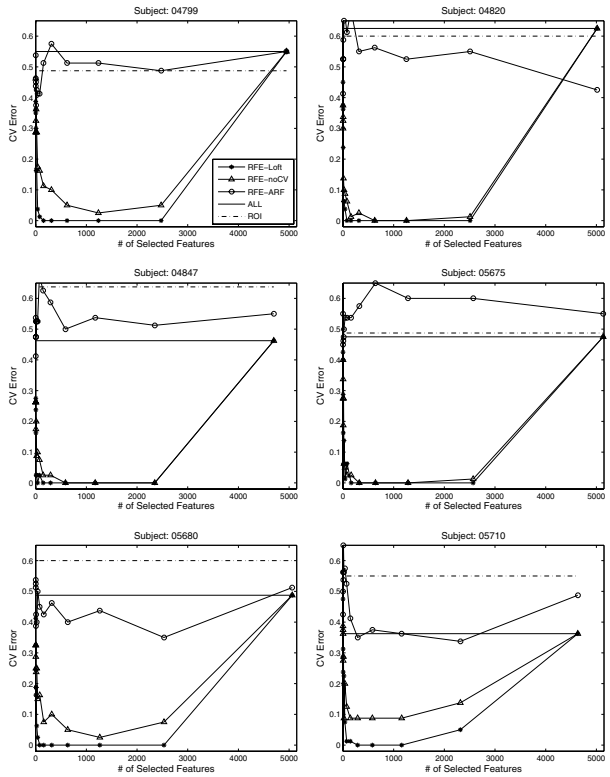

Figure 1. Classification results of RFE-Loft, RFE-noCV, RFE-ARF, ROI, and ALL for six subjects.

classification error 0. This is an improvement as compared to the results reported by the owner of this fMRI datasets, where the average error obtained for the most successful trained classifier, using the most successful feature selection strategy, was 0.11, averaged over 13 subjects[3], with the best subject reaching 0.04 [5]. The performances of using all the features as well as using the features in ROIs denoted by a solid and a dashed line, respectively, are even worse than the random classification, which is 0.5 in this binary classification problem. This result therefore strongly supports why feature subset selection, hence dimension reduction, is critical for the extremely high dimensional datasets. The performance of RFE-noCV using a single SVM classifier for the ranking criterion is comparable with the one of RFE-Loft in 3 subjects and is marginally worse in the other subject. This implies that the vectorization using the principal components plays a significant role in the performance gain of RFE-Loft. The performance by the closest competing method, RFE-ARF, is much worse than RFE-Loft. Even within RFE-ARF, the performance is not improved much as the number of features is reduced. This may indicate that the vectorization using the autoregressive coefficients does not maintain the discriminant and/or correlation information as much as RFE-Loft does, by considering each input feature separately.

[3]The fMRI datasets obtained only from 6 out of 13 subjects are open to public.

5 Conclusion

In this paper, we propose a new feature selection technique for MTS datasets where their spatial features are much larger than their temporal observations, termed RFE-Loft. RFE-Loft first vectorizes each MTS item using its principal components with non-zero corresponding variance, and then repeatedly eliminates one variable at a time based on the ranking criterion determined by multiple SVM classifiers until the required number of variables are retained. Our experiments on the fMRI datasets show that RFE-Loft consistently outperforms the closest competitor technique in terms of classification performance by up to a factor of 6.

References

[1] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., 1990.

[2] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[3] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*, chapter 3, page 121. Morgan Kaufmann, 2000.

[4] T. N. Lal, M. Schröder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schölkopf. Support vector channel selection in BCI. *IEEE Trans. Biomed. Eng.*, 51(6), June 2004.

[5] T. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman. Leaning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004.

[6] A. Rakotomamonjy. Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3(7):1357 – 1370, 2003.

[7] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[8] V. N. Vapnik. *Statistical Learning Theory*. Wiley, Sep. 1998.

[9] J. Weston, A. Elisseeff, G. BakIr, and F. Sinz. Spider: object-orientated machine learning library. http://www.kyb.tuebingen.mpg.de/bs/people/spider, 2004.

[10] K. Yang and C. Shahabi. A PCA-based similarity measure for multivariate time series. In *The 2nd ACM MMDB*, 2004.

[11] K. Yang, H. Yoon, and C. Shahabi. A supervised feature subset selection technique for multivariate time series. In *FSDM*, Newport Beach, CA, USA, April 2005.

[12] H. Yoon, K. Yang, and C. Shahabi. Feature subset selection and feature ranking for multivariate time series. *IEEE Trans. Knowl. Data Eng.*, 17(9):1186–1198, 2005.