A Supervised Feature Subset Selection Technique for Multivariate Time Series

Kiyoung Yang* Hyunjin Yoon† Cyrus Shahabi‡

Abstract

Feature subset selection (FSS) is a known technique to pre-process the data before performing any data mining tasks, e.g., classification and clustering. FSS provides both cost-effective predictors and a better understanding of the underlying process that generated data. We propose *Corona*, a simple yet effective supervised feature subset selection technique for Multivariate Time Series (MTS). Traditional FSS techniques, such as Recursive Feature Elimination (RFE) and Fisher Criterion (FC), have been applied to MTS datasets, e.g., Brain Computer Interface (BCI) datasets. However, these techniques may lose the correlation information among MTS variables, since each variable is considered separately when an MTS item is *vectorized* before applying RFE and FC. *Corona* maintains the correlation information by utilizing the correlation coefficient matrix of each MTS item as features to be employed for SVM. Our exhaustive sets of experiments show that *Corona* consistently outperforms RFE and FC by up to 100% in terms of classification accuracy, and takes more than one order of magnitude less time than RFE and FC in terms of the overall processing time.

Keywords

multivariate time series, feature subset selection, support vector machine, recursive feature elimination, correlation coefficient matrix

1 Introduction

Feature subset selection (FSS) is one of the techniques to pre-precess the data before we perform any data mining tasks, e.g., classification and clustering. FSS is to identify a subset of original features from a given dataset while removing irrelevant and/or redundant features [1]. The objectives of FSS are [2]:

- to improve the prediction performance of the predictors
- to provide faster and more cost-effective predictors
- to provide a better understanding of the underlying process that generated the data

The FSS methods choose a subset of the original features to be used for the subsequent processes. Hence, only the data generated from those features need to be collected. The differences between feature *extraction* and FSS are:

- Feature subset selection maintains information on the original features while this information is usually lost when feature extraction is used.
- After identifying the subset of original features, only those features can be measured and collected ignoring all the other features. However, feature extraction in general requires measuring all the original features.

A time series is a series of observations, $x_i(t); [i = 1, \cdots, n; t = 1, \cdots, m]$, made sequentially through time where i indexes the measurements made at each time point t [3]. It is called a univariate time series when n is equal to 1, and a multivariate time series (MTS) when n is equal to, or greater than 2.

MTS datasets are common in various fields, such as in multimedia and medicine. For example, in multimedia, Cybergloves used in the Human and Computer Interface applications have around 20 sensors, each of which generates 50∼100 values in a second [4, 5]. In [6], 22 markers are spread over the human body to measure the movements of human parts while walking. The dataset collected is then used to recognize and identify the person at a distance by how he or she walks. In the Neuro-rehabilitation domain, kinematics datasets generated from sensors are collected and analyzed to evaluate the functional behavior (i.e., the movement of upper extremity) of post-stroke patients [7]. In medicine, Electro Encephalogram (EEG) from 64 electrodes placed on

[*] Computer Science Department, University of Southern California, Los Angeles, CA 90089, U.S.A., kiyoungy@usc.edu

[†] Computer Science Department, University of Southern California, Los Angeles, CA 90089, U.S.A., hjy@usc.edu

[‡] Computer Science Department, University of Southern California, Los Angeles, CA 90089, U.S.A., shahabi@usc.edu

the scalp are measured to examine the correlation of genetic predisposition to alcoholism [8]. Functional Magnetic Resonance Imaging (fMRI) from 696 voxels out of 4391 has been used to detect similarities in activation between voxels in [9].

The size of an MTS dataset can become very large quickly. For example, the EEG dataset in [10] utilizes tens of electrodes and the sampling rate is 256Hz. In order to process MTS datasets efficiently, it is therefore inevitable to preprocess the datasets to obtain the relevant subset of features which will be subsequently employed for further processing. In the field of Brain Computer Interfaces (BCIs), the selection of relevant features is considered absolutely necessary for the EEG dataset, since the *neural correlates* are not known in such detail [10]. Identifying optimal and valid features that differentiate the post-stroke patients from the healthy subjects is also challenging in the Neurorehabilitation applications.

An MTS item is naturally represented as an $m \times n$ matrix, where m is the number of observations and n is the number of *variables*, e.g., sensors. However, the state of the art feature subset selection techniques, such as Recursive Feature elimination (RFE) [2], require each item to be represented in one row. Consequently, to utilize these techniques on MTS datasets, each MTS item needs to be first transformed into one row or column vector, which we call *vectorization*. For example, in [10] where an EEG dataset with 39 channels is used, an autoregressive (AR) model of order 3 is utilized to represent each channel. Hence, each 39 channel EEG time series is transformed into a 117 dimensional vector. However, if each channel of EEG is considered separately, we will lose the correlation information among the variables.

Information theory (IT) based feature subset selection methods, such as information gain and information gain ratio, have been extensively studied and employed in the data mining and machine learning community [11, 12]. However, IT based feature subset selection methods are also not directly applicable to MTS items, because, again, an MTS item is not a vector, and also each value of an MTS item is continuous, not discrete. Hence, each MTS item should first be transformed into a vector and also be discretized, which usually results in loss of important information.

In this paper, we propose a simple yet quite effective subset selection method for multivariate time series (MTS)[1], termed *Corona* (*Cor*relati*on* as Fe*a*tures). *Corona* is based on RFE. Recall that RFE, which utilizes SVM, requires each item to be represented as a vector. The performance of RFE will therefore heavily rely on how the MTS dataset is fed into SVM, i.e., how each MTS item is transformed to be utilized by SVM. *Corona* employs the correlation coefficients of an MTS item as features for SVM and hence for RFE. The intuition is based on our previous work [13] which has shown that the correlation information among the variables plays an important role in obtaining the similarity between two MTS items. Hence, *Corona* first computes the pairwise correlation coefficients of all the variables, i.e., the correlation coefficient matrix, of each MTS item. Since the correlation coefficient matrix is symmetric and its diagonal values are all 1s, only the upper triangle of the correlation coefficient matrix except the diagonal values is utilized to *vectorize* an MTS item. Consequently, an MTS dataset is transformed into a matrix, which we call a *feature matrix*, where each row represents an MTS item. *Corona* subsequently trains SVM on the feature matrix, which will produce the weights of each feature. Note that each feature in the feature matrix is the correlation coefficient of two variables. *Corona* then aggregates the weights for each variable and ranks the variables based on the aggregated weights. Subsequently, *Corona* eliminates the variable with the lowest rank. This process is repeated until the required number of variables is obtained. Our experiments show that the classification performance of the variable subsets selected by *Corona* is up to about 100% better than those selected by other feature subset selection methods, such as Recursive Feature Elimination (RFE) and Fisher Criterion (FC). Moreover, *Corona* takes more than one order of magnitude less time than RFE and FC in terms of the overall processing time which includes the time to *vectorize* an MTS dataset.

The remainder of this paper is organized as follows. Section 2 discusses the background. Our proposed method is described in Section 3, which is followed by the experiments and results in Section 4. Related work is presented in Section 5 followed by conclusions and future work in Section 6.

2 Background

Corona utilizes the correlation coefficient matrix and RFE for feature subset selection of MTS datasets. In this section, we briefly describe the correlation coefficient matrix, Support Vector Machine and Recursive Feature Elimination.

2.1 Correlation Coefficient Matrix The correlation represents how strongly one variable implies the other, based on the available data [14]. Assume that

[1] For multivariate time series, each *variable* is regarded as a feature [10]. Hence, the terms *feature* and *variable* are interchangeably used throughout this paper, when there is no ambiguity.

a and **b** are two vectors of length n. The correlation between **a** and **b** is then defined as follows [14]:

$$Corr(\mathbf{a},\mathbf{b}) = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{(n-1)\sigma_a\sigma_b} \tag{2.1}$$

where $\bar{a}$ and $\bar{b}$ are the averages of vector **a** and **b**, respectively; σ_a and σ_b are the standard deviations of **a** and **b**, respectively. The correlation value ranges from -1 to 1. A value greater than 0 means that there is a positive correlation. That is, if the values of **a** increase, then the values of **b** would also increase. If the correlation is 0, then there is no correlation between **a** and **b** meaning that they are independent. The negative correlation value means that there is a negative correlation between **a** and **b**. That is, if the values of **a** increase, then the values of **b** would decrease, or vice versa.

A correlation coefficient matrix is a symmetric matrix, where the $(i,j)th$ entry in the matrix represents the correlation between the ith and jth variables. Our proposed supervised feature subset selection technique, *Corona*, utilizes the correlation coefficient matrix of each MTS item as features for SVM to obtain the *weights* of each variable, which is described in Section 3.

2.2 Support Vector Machine Support Vector Machine (SVM) is a classification technique by Vapnik [15]. SVM performs classification by obtaining and utilizing the *optimal separating hyperplane* that separates two classes and maximizes the distance to the closest point from either class, called *margin* [15, 16]. Figure 1 represents the training result of an SVM model for a simple two class dataset[2].

The hyperplane that separates the two classes shown in Figure 1 can be described as follows [18]:

$$g(\mathbf{x}) = \mathbf{w}'\mathbf{x} + w_0 \tag{2.2}$$

where **w** is the norm vector of the hyperplane $g(\mathbf{x})$ and $w_0/||\mathbf{w}||$ is the distance from the origin to the hyperplane. Given new data $\mathbf{x}_i$, the sign of $g(\mathbf{x}_i)$ determines the class of $\mathbf{x}_i$. For simplicity, we described only the case where the classes are *linearly separable*. For more details, please refer to [18, 16].

2.3 Recursive Feature Elimination Based on SVM, Guyon *et al* [19] proposed a feature subset selection method called Recursive Feature Elimination (RFE). RFE is a *stepwise backward feature elimination* method [14]. That is, RFE starts with all the features and removes features based on a ranking criterion until the required number of features are obtained. The procedure can be briefly described as in Algorithm 1 [19]:

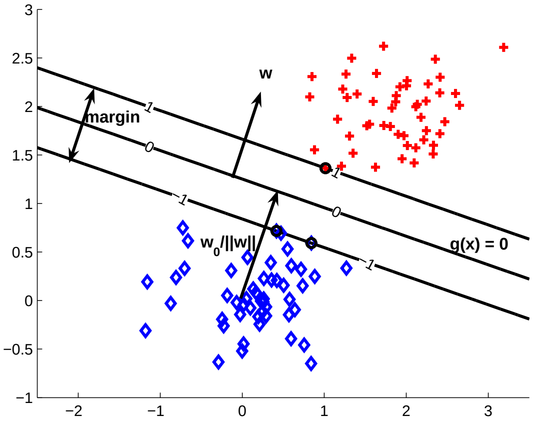


Figure 1: Two classes are linearly separable.

Algorithm 1 Recursive Feature Elimination

```
1: Train SVM;
2: Rank the features;
3: Eliminate the feature with the lowest rank;
4: Repeat until the required number of features are retained;
```

In order to rank the features, RFE utilizes the *sensitivity analysis* based on the weight vector **w** in Equation 2.2. That is, at each iteration, RFE eliminates one feature with the minimum weight. The intuition is that the feature with the minimum weight would least influence the weight vector norm [20], and is therefore to be removed.

RFE, however, cannot be used with MTS datasets *as is*, since an MTS item is represented as a matrix, while RFE requires each item to be represented as a vector. In [10], for example, each variable, i.e., channel, is transformed separately using the autoregressive fit coefficients of order 3. By doing so, however, the correlation information among the variables would be lost. In the following section, we propose an extension of RFE to MTS datasets, called *Corona*.

3 Proposed Method

In this section, we describe *Corona*, which is a simple yet effective feature subset selection technique for MTS datasets based on RFE. Recall that SVM, hence RFE, requires each MTS item to be represented as a vector.

[2] SVM and Kernel Methods Matlab Toolbox [17] is utilized to generate the figure.

Corona utilizes the correlation coefficients as features for an MTS item to be used for SVM. The intuition using the correlation coefficients as features for MTS items to be used for SVM comes from our previous work [13] which has shown that the correlation information of an MTS item plays a significant role in computing the similarity between two MTS items.

Hence, *Corona* first computes the correlation coefficient matrix for each MTS item. A correlation coefficient matrix is symmetric and its diagonal values, which represent the *autocorrelations* of all the variables, are all 1s. Hence, as features for an MTS item, the correlation coefficients in the upper triangle of the correlation coefficient matrix except the diagonal values are utilized, which are then vectorized as in Algorithm 2. For an n-variate MTS item, the number of features to be used for SVM is $\sum_{i=1}^{n-1} i = n \times (n-1)/2$. For example, for the HumanGait dataset where n is 66, the number of features is $66 \times 65/2 = 2145$. For an MTS dataset which has N items, this transformation results in an $N \times p$ matrix, where $p = n \times (n-1)/2$. We denote this matrix a *feature matrix*.

Corona subsequently trains SVM using the feature matrix. Utilizing the model resulted from the SVM training, we obtain the weight vector $\mathbf{w}$ for the features that have been employed in the SVM training. Note that each feature utilized for SVM training is a correlation of two variables. In order to determine the ranks of the *variables*, we construct a symmetric matrix using the weights obtained by SVM, which we call a *weight matrix* (Lines 1–10 in Algorithm 4). This is similar to *un-vectorizing* the vectorized correlation coefficient matrix except that the *weights* obtained from SVM are used, not the correlation coefficients. Hence, the ith column in the weight matrix represents all the weights of the *features*, i.e., the correlation coefficients, with which the ith *variable* is associated. After obtaining the weight matrix, *Corona* aggregates all the weights of each variable and obtains one value per variable. Finally, based on the aggregated values, *Corona* decides which variable to eliminate. In our experiments, we took the *greedy* approach, and identified a variable whose maximum weight is the minimum among the maximum weights of all the variables (Lines 11–12 in Algorithm 4). The variable whose maximum weight is the minimum is then to be removed. The intuition behind using the *max* aggregate function is to retain the variables that are associated with the correlation coefficients which contribute most to the SVM training result.

Algorithm 3 describes the overall process of *Corona*. Given an MTS dataset, *Corona* first computes the feature matrix T by vectorizing the upper triangle of the correlation coefficient matrix of each MTS item (Lines 1–4 of Algorithm 3, and Algorithm 2). Subsequently, it performs SVM on the feature matrix. Using the feature weights obtained from SVM, *Corona* ranks the variables as in Algorithm 4. The entire process is repeated until the required number of variables are identified.

Algorithm 2 Vectorize a correlation coefficient matrix using the upper triangle

```
Require: $C$ {a correlation coefficient matrix of an $n$-variate MTS item};
$C_{vectorized} \leftarrow []$;
for $i = 1$ to $n$ do
    $C_{vectorized} \leftarrow [C_{vectorized} \quad C[i, (i+1) : n]]$;
end for
```

Algorithm 3 Corona

```
Require: MTS dataset, $N$ {the number of items in the dataset}, $k$ {the required number of variables};
1: for $i = 1$ to $N$ do
2:     $C$ ← correlation coefficient matrix of the $i$TtH MTS item;
3:     $T[i, :]$ ← vectorize $C$ using the upper triangle of $C$;
4: end for
5: $[rank_{SVM}, weights_{SVM}]$ ← Train SVM on $T$;
6: Rank variables using $weights_{SVM}$;
7: Remove one variable with the lowest rank;
8: Repeat until $k$ variables remain;
```

4 Performance Evaluation

In order to evaluate the effectiveness of Corona in terms of classification performance and overall processing time, we conducted several experiments on three real-world datasets. After obtaining a subset of variables using Corona, we performed classification using SVM with linear kernel as in [10]. Subsequently, we compared the performance of Corona with those of RFE [2, 10], Fisher Criterion (FC), Exhaustive Search Selection (ESS) when possible, and using all the available variables (ALL). The algorithm of Corona for the experiments is implemented in *Matlab* and in[3] *R* using[4] *e*1071 and[5] *RFE* packages.

4.1 Datasets

The **HumanGait dataset** [6] has been used for identifying a person by recognizing his/her gait at a distance. In order to capture the gait data, a

[3] http://www.r-project.org/

[4] http://cran.r-project.org/src/contrib/Descriptions/e1071.html

[5] http://www.hds.utc.fr/~ambroise/softwares/RFE/

Algorithm 4 Rank variables using $weights_{SVM}$

```
Require: $weights_{SVM}$ {weights obtained by SVM}, $n$ {the number of variables for an MTS item};
$W \leftarrow []$;
$count \leftarrow 1$;
for $i = 1$ to $n$ do
    $W[i, (i+1) : n] \leftarrow weights_{SVM}[count : (count + n - i - 1)]$;
    $count \leftarrow count + n - i$;
end for
$W \leftarrow W + transpose(W)$;
for $i = 1$ to $n$ do
    $W(i, i) \leftarrow 1$;
end for
$weights_{Corona} \leftarrow$ Aggregate $W$ in column-wise;
$rank_{Corona} \leftarrow sort(weights_{Corona})$;
```

	HumanGait	BCAR	BCI MPI
# of variables	66	11	39
average length	133	454	1280
# of labels	15	2	2
# of items per label	36	22/17	1000
total # of items	540	39	2000

Table 1: Summary of datasets used in the experiments

twelve-camera VICON system was utilized with 22 reflective markers attached to each subject. For each reflective marker, 3D position, i.e., x, y and z, are acquired at 120Hz, generating 66 values at each timestamp. 15 subjects, which are the labels assigned to the dataset, participated in the experiments and were required to walk at four different speeds, nine times for each speed. The total number of data items is 540 (15 × 4 × 9) and the average length is 133.

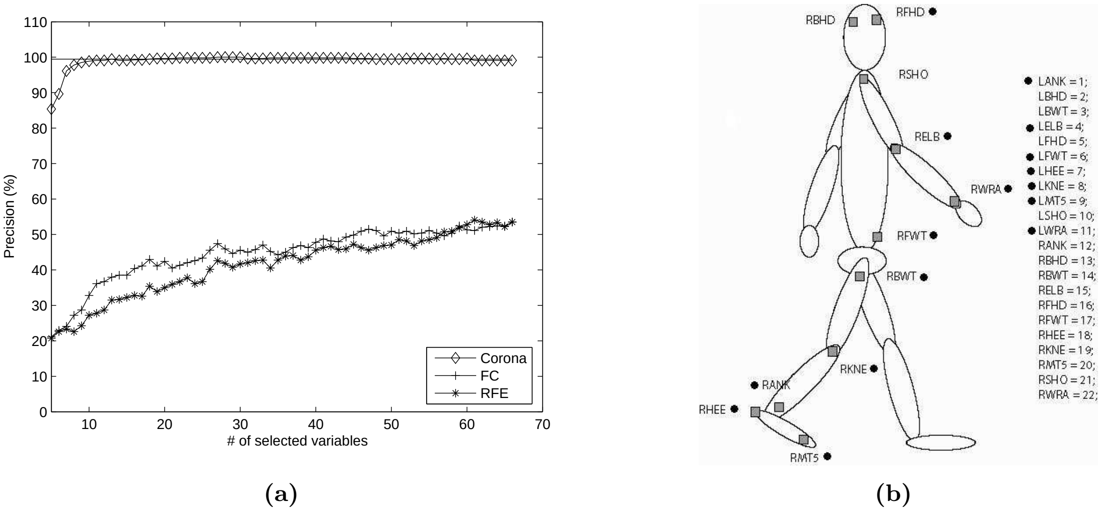
Motor Behavior and Rehabilitation Laboratory, University of Southern California collected **Brain and Behavior Correlates of Arm Rehabilitation (BCAR) kinematics dataset** to study the effect of Constraint-Induced (CI) physical therapy on the post-stroke patients' control of upper extremity [7]. The functional specific task performed by subjects was a continuous 3 phase reach-grasp-place action; a subject sits on a chair pressing down the starting switch with his or her impaired forearm. She or he is then supposed to reach for a target object, either a cylinder or a card, grasp it, place it into a designated hole, release it, and finally bring her or his hand back to the starting switch. This specific task is repeated five times per subject under four different conditions, i.e., for 2 different objects (Cylinder/Card) by posing 2 different forearm postures (pronation/supination). The performance is traced by six *miniBIRD* trackers attached on the index nail, thumb nail, dorsal hand, distal dorsal forearm, lateral mid upper arm and shoulder, respectively. Then, 11 dependent variables are measured from the raw data, sampled at 120Hz and filtered using a 0-lag Butterworth low-pass filter with a 20Hz cut-off frequency. Unlike other datasets, BCAR dataset kept the record of 11 dependent features rather than 36 *raw* variables at each timestamp. They were defined by experts in advance and calculated from the raw variables by the device software provided with the trackers; some of them were just raw variables (e.g., wrist tracker X, Y, and Z coordinates) while others were synthesized from raw variables (e.g., aperture was computed as tangential displacement of two trackers on thumb and index nail). Note that these 11 variables were considered as original variables throughout the experiments. Four control (i.e., healthy) subjects and three post-stroke subjects experiencing a different level of impairment participated in the experiments. For each of the 4 conditions, the total number of data items is 39, and their average length is about 454 (i.e., about 3.78 seconds).

The **Brain Computer interface (BCI) dataset** at the **Max Planck Institute (MPI)** [10] was collected to examine the relationship between the brain activity and the motor imagery, i.e., the imagination of limb movements. Eight right handed male subjects participated in the experiments, out of which three subjects were filtered out after pre-analysis [10]. 39 electrodes were placed on the scalp to record the EEG signals at the rate of 256Hz. The total number of items is 2000, i.e., 400 items per subject.

Table 1 summarizes the datasets used in the experiments.

4.2 Classification Performance We evaluated the effectiveness of Corona in terms of classification accuracy. Support Vector Machine (SVM) with linear kernel was adopted as the classifier. Using SVM, we performed leave-one-out cross validation for the BCAR dataset and 10 fold cross validation [14] for the rest since they have too large number of items to conduct leave-one-out cross validation.

For RFE and FC, we vectorized each MTS item as in [10]. That is, each variable is represented as the autoregressive (AR) fit coefficients of order 3 using the forward backward linear prediction [21]. Therefore, each MTS item with n variables is represented in a vector of size $n \times 3$. *The Spider* [22] implementation of FC is subsequently employed. For small datasets, i.e., BCAR and HumanGait, RFE in *The Spider* [22] was employed, while for large dataset, i.e., BCI MPI, RFE package for R is utilized. Note that Exhaustive Search Selection



(a) **(b)**

Figure 2: (a) HumanGait dataset, Classification Evaluation (b) 22 markers for the HumanGait dataset. The markers with a filled circle represent 16 markers from which the 27 variables are selected by *Corona*, which yields better performance accuracy than using all the 66 variables.

(ESS) method was performed only on BCAR dataset due to the intractability of ESS for the large datasets. The ESS methods simply searches exhaustively among all possible combinations of variables and selects the best combination. Obviously, this is an impractical approach due to its high complexity and we only used it here (when possible) to generate the ground truth.

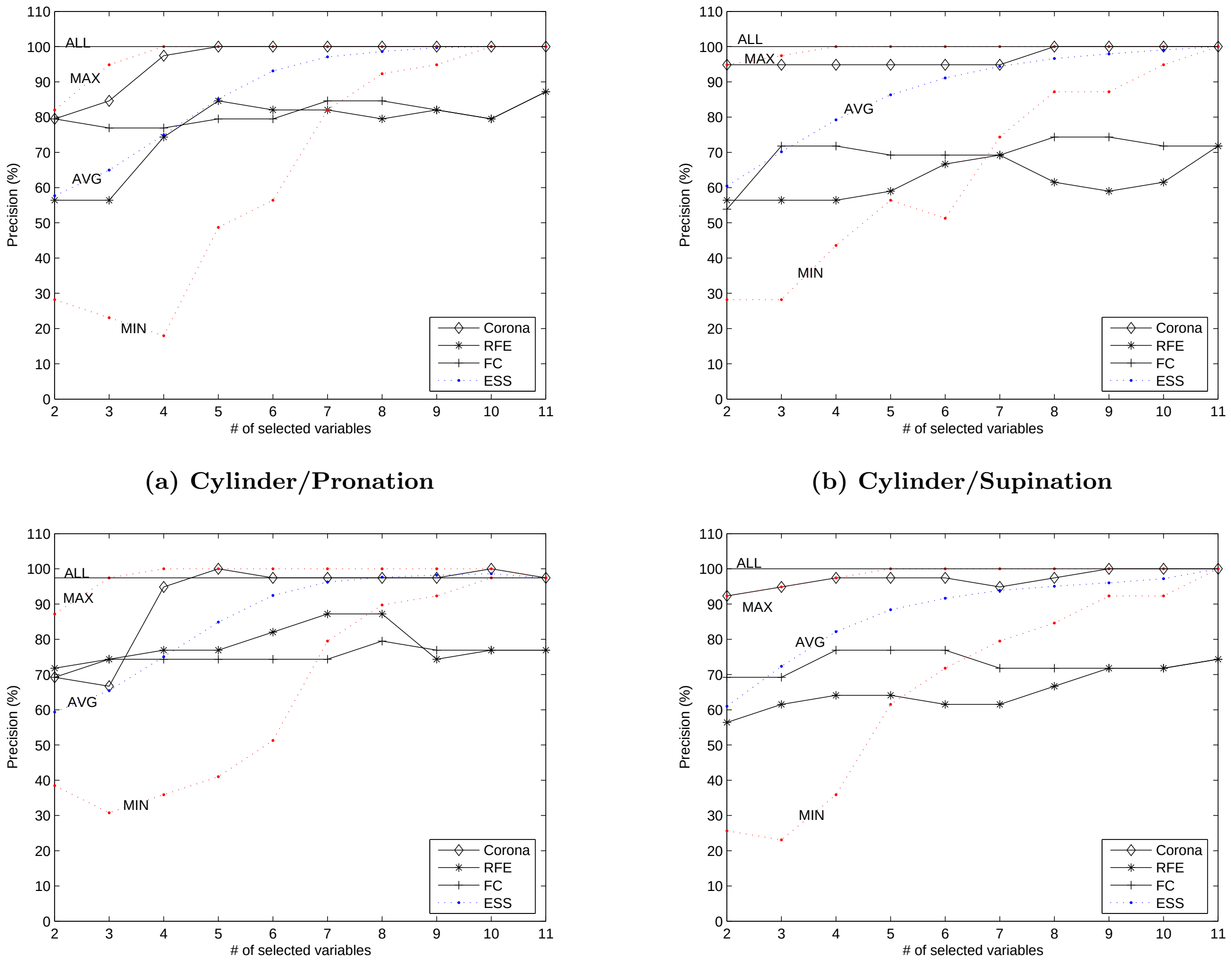
Figure 2(a) presents the generalization performances on the HumanGait dataset. It shows that a subset of 11 variables selected by *Corona* out of 66 performs the same as the one using all the variables (99.0741% accuracy), which is represented as a solid horizontal line. Moreover, a subset of 27 variables yields 100% accuracy. The 27 variables selected by Corona are from only 16 markers (marked with a filled circle in Figure 2(b)) out of 22, which would mean that the values generated by the remaining 6 markers does not contribute much to the identification of the person. From this information we may be able to better understand the characteristics of the human walking.

The performances by RFE and FC for the HumanGait dataset is much worse than Corona. Even when using all the variables, the classification accuracy is around 55%. Considering the fact that RFE on 3 AR coefficients performed well in [10], this may indicate that for the HumanGait dataset the correlation information among variables is more important than for the BCI MPI dataset. Hence, each variable should not be taken out separately to compute the autoregressive coefficients, by which the correlation information would be lost. Note that in [10], the order 3 for the autoregressive fit is identified after proper model selection experiments, which would mean that for the HumanGait dataset, the order of the autoregressive fit should be determined, again, after comparing different order models. Hence, it is not a trivial task to transform an MTS item into a vector, after which the traditional machine learning techniques, such as Support Vector Machine (SVM), can be applied.

Figure 3 shows the classification performance of the selected variables on the BCAR dataset for 4 different conditions. For example, Figure 3(c) represents that a card was used as a target object and the pronated forearm posture was taken by a subject to perform the continuous reach-grasp-place task in [7].

The BCAR is the simplest dataset with 11 original variables and the number of MTS items for each condition is just 39. Hence, we applied the Exhaustive Search Selection (ESS) method to find all the possible variable subset combinations, for each of which we performed leave-one-out cross validation. It took about 87 minutes to complete the whole ESS experiments. The result of ESS shows that 100% classification accuracy can be achieved by either 4 or 5 variables out of 11. The dotted lines represent the best, the average, and the worst performance obtained by ESS, respectively, given the number of selected variables.

Figure 3 again shows that Corona consistently outperforms RFE and FC methods. The figure also depicts that the 5 variables selected by *Corona* produce



(a) Cylinder/Pronation

(b) Cylinder/Supination

(c) Card/Pronation

(d) Card/Supination

Figure 3: BCAR dataset, Classification Evaluation

100% classification accuracy for Cylinder/Pronation and Card/Pronation conditions. Besides, *Corona* outperforms or performs the same as the one using all the variables, which is represented as a horizontal solid line. This implies that *Corona* never eliminates useful information in its variable selection. For the Cylinder/Pronation condition, for example, Figure 3(a) shows that only the 4 variables selected by *Corona* produce about 98% classification accuracy, which is the same as using all the 11 variables. Moreover, the overall performance of *Corona* is close to the best performance of ESS, which is far from the average performance.

As illustrated in the figure, FC method never beats the *Corona* for 3 conditions, and for the Card/Pronation condition, *Corona* by far outperforms FC when more than 3 variables are selected. As compared to RFE, *Corona* again shows consistently better classification performance almost always.

Figure 4 represents the performance comparison using the BCI MPI dataset. Note that unlike in [10] where they applied the feature subset selection per subject, the whole items from the 5 subjects were utilized in our experiments, which would make the subset of variables selected by *Corona* more applicable for subsequent data mining tasks. Moreover, the regularization parameter $\mathcal{C}_s$ for SVM was estimated via 10 fold cross validation from the training datasets in [10], while we used the default value, which is 1. The figure again depicts that Corona performs far better than RFE and FC.

For the BCI MPI dataset, it is intractable to try all the combinations of the 39 channels to identify the best combination. Therefore, to find the ground-truth,

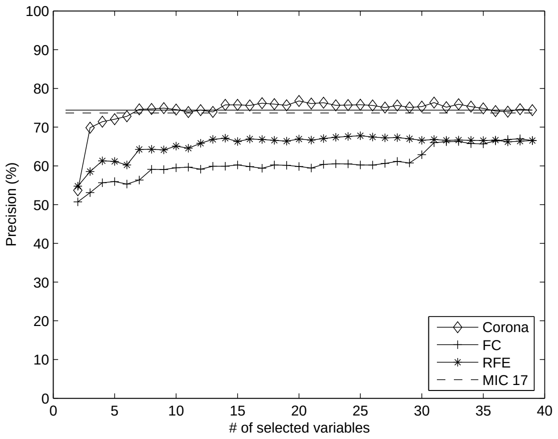


Figure 4: BCI MPI dataset, Classification Evaluation

in [10], the 17 channels located over or close to the motor cortex were manually identified as the best combination of channels using the domain knowledge. In Figure 4, the classification performance using those 17 motor imagery channels (termed MIC 17) is presented in dashed lines, while the performance using all the variables is shown in a solid horizontal line. Using the 17 variables selected by Corona, the classification accuracy is 75.45%, which is even better than the expert-selected channels of MIC 17 whose accuracy is 73.65%.

4.3 Processing Time *Corona* in fact utilizes a lot more number of features than RFE to *vectorize* an MTS item. For example, for the HumanGait dataset where there are 66 variables, each MTS item is represented with $66 \times 65/2 = 2145$ features by *Corona*, while RFE represents each MTS item with $66 \times 3 = 198$ features. Obviously, this would result in more training time for SVM on which both *Corona* and RFE are based. However, RFE takes a considerable amount of time to compute and obtain the AR coefficients of order 3. Hence, the overall processing time of *Corona*, including the time to transform the MTS dataset, is one order of magnitude less than that of RFE.

For the BCI MPI dataset, for example, it takes only 4.562 seconds to compute all the 2000 correlation coefficient matrices for *Corona*, while it takes about 7600 seconds to compute the AR coefficients of order 3 for RFE, both using Matlab. The total processing time including the transformation for *Corona* of the BCI MPI dataset is less than 480 seconds, while that of RFE is more than 7800 seconds. Table 2 summarizes the processing time of the 3 feature selection methods employed for the experiments.

Table 2: Comparison of processing time in seconds for different feature selection methods on 3 different datasets

	HumanGait	BCAR	BCI MPI
Corona	422.688	0.191	472.953
RFE	962.063	9.039	7886.844
FC	113.907	6.469	7594.941

5 Related Work

In the field of Brain Computer Interfaces (BCIs), extensive research has been conducted on Electroencephalogram (EEG) datasets. The EEG dataset is collected using multiple electrodes placed on the scalp. The sampling rate is hundreds of Hertz. The selection of relevant features is considered absolutely necessary for the EEG dataset, since the neural correlates are not known in such detail [10].

In [10], feature selection is performed on the 39 channel EEG dataset. Each EEG item is broken into 39 separate channels, and for each channel, autoregressive (AR) fit of order 3 is computed. Subsequently, each channel is represented by 3 autoregressive coefficients. Feature selection using Recursive Feature Elimination (RFE) is then performed on these transformed dataset. As shown in Section 4.2, by considering the channels separately, they lose the correlation information among channels.

In [23], EEG dataset from UCI KDD Archive [24] has been used for the experiments. EEG-1 dataset contains only 20 measurements for two subjects from two arbitrary electrodes (F4 and P8). EEG-2 dataset contains 20 measurements from the same 2 electrodes for each subject. It is not clear how the two subjects out of 122 subjects and the two electrodes out of 64 are chosen. The best accuracies obtained are $90.0 \pm 0.0\%$ using DCHMM-exact, $90.5 \pm 5.6\%$ using Multivariate HMM for the EEG-1 dataset. $78.5 \pm 8.0\%$ using Multivariate HMM.

In [25], a subset of the HumanGait dataset, a total of 45 items of 15 subjects, was used for an HMM-based clustering. They, however, achieved only 75% classification accuracy, which could have been achieved by *Corona* using only 9 variables out of 66 as shown in Figure 2(a).

In [26], Genetic Algorithm (GA) and Support Vector Machine (SVM) are used for feature subset selection. Two EEG datasets are used, TTD and NIPS 2001. The TTD (Thought Translation Device) EEG dataset were generated with 6 channels, and the other EEG dataset which was submitted to Neural Information Processing Systems (NIPS) Conference in 2001, were collected with

27 channels. For the EEG dataset with 6 channels, they also performed the exhaustive search to find out the best channels. The advantage of GA is that the optimal subset of variables is produced as output, and hence, one does not have to specify how many variables she would like to select. However, GA is known to be very time consuming.

In [27], features are firstly extracted from the original dataset, and then feature subset selection are performed using mutual information. The accuracy from training set is less than 70% and from test set is less than 85%. The EEG data used was obtained from Graz University of Technology, Austria, and Artificial Neural Network (ANN) is used for classification. Note that this approach, i.e., performing feature extraction and then feature selection, may work well in terms of classification accuracy. However, we cannot reduce the amount of data to be collected, if the features are global features for which all the *raw* data would be required.

6 Conclusion and Future Work

In this paper, we proposed a simple yet quite effective feature subset selection method for multivariate time series (MTS), termed *Corona*. *Corona* first vectorizes the correlation coefficient matrix of each MTS item to be used as features for SVM, and yields a *feature matrix*. After training SVM on the feature matrix, *Corona* computes the *weight matrix*, from which the ranks for the variables are identified. Based on the ranks, *Corona* eliminates one variable with the lowest rank, and repeats itself until the required number of variables are retained. Our experiments on the three real-world datasets show that *Corona* consistently outperforms other feature selection methods, such as Recursive Feature Elimination (RFE) and Fisher Criterion (FC) in terms of classification performance by up to 100%. Moreover, *Corona* takes more than one order of magnitude less time than RFE in terms of the overall processing time.

We intend to extend this technique to the stream of data where the feature subset selection can be performed incrementally adjusting itself based on the observations collected thus far.

Acknowledgements

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. 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. The authors would like to thank Dr. Carolee Winstein and Jarugool Tretiluxana for providing us the BCAR dataset and valuable feedbacks, and Thomas Navin Lal for providing us the BCI MPI dataset. The authors would also like to thank the anonymous reviewers for their valuable comments.

References

[1] Liu, H., Yu, L., Dash, M., Motoda, H.: Active feature selection using classes. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. (2003)

[2] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182

[3] Tucker, A., Swift, S., Liu, X.: Variable grouping in multivariate time series via correlation. IEEE Trans. on Systems, Man, and Cybernetics, Part B **31** (2001)

[4] Kadous, M.W.: Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series. PhD thesis, University of New South Wales (2002)

[5] Shahabi, C.: AIMS: An immersidata management system. In: VLDB Biennial Conference on Innovative Data Systems Research. (2003)

[6] Tanawongsuwan, Bobick: Performance analysis of time-distance gait parameters under different speeds. In: 4th International Conference on Audio- and Video Based Biometric Person Authentication, Guildford, UK (2003)

[7] Winstein, C., Tretriluxana, J.: Motor skill learning after rehabilitative therapy: Kinematics of a reach-grasp task. In: the Society For Neuroscience, San Diego, USA (2004)

[8] Zhang, X.L., Begleiter, H., Porjesz, B., Wang, W., Litke, A.: Event related potentials during object recognition tasks. Brain Research Bulletin **38** (1995)

[9] Goutte, C., Toft, P., Rostrup, E., Nielsen, F.A., Hansen, L.K.: On clustering fmri time series. NeuroImage **9** (1999)

[10] Lal, T.N., Schröder, M., Hinterberger, T., Weston, J., Bogdan, M., Birbaumer, N., Schölkopf, B.: Support vector channel selection in BCI. IEEE Trans. on Biomedical Engineering **51** (2004)

[11] Mitchel, T.M.: Machine Learning. McGraw Hill (1997)

[12] Witten, I.H., Frank, E.: 7. In: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999)

[13] Yang, K., Shahabi, C.: A PCA-based similarity measure for multivariate time series. In: The Second ACM MMDB. (2004)

[14] Han, J., Kamber, M.: 3. In: Data Mining: Concepts and Techniques. Morgan Kaufmann (2000) 121

[15] Vapnik, V.N.: Statistical Learning Theory. Wiley (1998)

[16] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)

[17] Canu, S., Grandvalet, Y., Rakotomamonjy, A.: Svm and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France (2003)

[18] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Second edn. Wiley Interscience (2001)

[19] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46** (2002) 389–422

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

[21] Moon, T.K., Stirling, W.C.: Mathematical Methods and Algorithms for Signal Processing. Prentice Hall (2000)

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

[23] Zhong, S., Ghosh, J.: HMMs and coupled HMMs for multi-channel EEG classification. In: International Joint Conference on Neural Networks. (2002)

[24] Hettich, S., Bay, S.D.: The UCI KDD Archive. http://kdd.ics.uci.edu (1999)

[25] Alon, J., Sclaroff, S., Kollios, G., Pavlovic, V.: Discovering clusters in motion time-series data. In: IEEE CVPR. (2003)

[26] Schröder, M., Bogdan, M., Hinterberger, T., Birbaumer, N.: Automated EEG feature selection for brain computer interfaces. In: IEEE EMBS International Conference on Neural Engineering. (2003)

[27] Deriche, M., Al-Ani, A.: A new algorithm for EEG feature selection using mutual information. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. (2001)