AN EFFICIENT FEATURE SELECTION ALGORITHM FOR
COMPUTER-AIDED POLYP DETECTION

JIANG LI, Ph.D, JIANHUA YAO, Ph.D, RONALD M. SUMMERS, M.D, Ph.D*
Clinical Center, National Institutes of Health
Bethesda, MD 20892-1182, USA
{lj3, jyao, rms}@cc.nih.gov

NICHOLAS PETRICK, Ph.D
NIBIB/CDRH, Joint Laboratory for the Assessment of
Medical Imaging System, FDA
HFZ-140, 12720 Twinbrook Parkway
Rockville, MD 20852, USA
nicholas.petrick@fda.hhs.gov

MICHAEL T. MANRY, Ph.D
Electrical Engineering Department
University of Texas at Arlington
Arlington, Texas, 76019 USA
manry@uta.edu

AMY K. HARA, M.D
Mayo Clinic Scottsdale
Scottsdale, AZ, USA
hara.amy@mayo.edu

We present an efficient feature selection algorithm for computer aided detection (CAD) computed tomographic (CT) colonography. The algorithm (1) determines an appropriate piecewise linear network (PLN) model by cross validation, (2) applies the orthonormal least square (OLS) procedure to the PLN model utilizing a Modified Schmidt procedure, and (3) uses a floating search algorithm to select features that minimize the output variance. The undesirable “nesting effect” is prevented by the floating search approach, and the piecewise linear OLS procedure makes this algorithm very computationally efficient because the Modified Schmidt procedure only requires one data pass during the whole searching process. The selected features are compared to those obtained by other methods, through cross validation with support vector machines (SVMs).

Keywords: Feature selection; CAD; piecewise linear network; orthonormal least squares; branch and bound; floating search.

*This research was supported by the Intramural Research Program of the NIH Clinical Center.
1. Introduction

Colon cancer is the second leading cause of cancer death in the United States. Research for the development of computer aided procedures for screening patients for colonic carcinoma has grown as a result of recognized disadvantages that accompany the current standard procedure, colonoscopy. CAD combined with CT colonography is an alternative. In order for an alternative screening procedure to be successful, it must be both sensitive and specific. There is an ongoing effort by several institutions to develop classification schemes that optimize the performance of CAD methods for colon polyp detection. Summers et al.\textsuperscript{1} describes recent work on a version of computer aided polyp detection that uses geometric and volumetric features, acquired from the CT data, as the basis for polyp identification. Our software first segments the colon using a region growing algorithm, after which regions of interest along the colon wall are identified. A total of more than 100 different quantitative features are currently calculated for each polyp candidate. However, many of these features are based on heuristic techniques and are not eventually useful. Irrelevant or redundant features can lead to several problems: (1) training an unnecessarily large classifier requires more computational resources and memory, (2) high dimensional data may have the \textit{curse of dimensionality} problem if the available data is limited, and (3) training algorithms for large networks can also have convergence difficulties and poor generalization. The work presented in this paper focuses on feature selection for computer-aided polyp detection with the goal of selecting a compact subset of features that leads to a good prediction of classification accuracy based on available data.

Feature selection algorithms usually evaluate fitness of the features first, then search different combinations of features in the whole feature space with the goal of obtaining maximizing a fitness value. Feature selection algorithms may be divided into \textit{filter}, \textit{wrapper} and \textit{embedded} categories. A filter approach is unsupervised. It performs preprocessing of the data without any actual classifier involved; examples include the FOCUS,\textsuperscript{2} Fisher criterion\textsuperscript{3} and RELIEF\textsuperscript{4} algorithms. A filter approach has the advantage of computational efficiency but the selected feature subset may not perform well. A wrapper approach is supervised in that it determines the fitness of a feature subset by actually training a classifier.\textsuperscript{5} Usually wrapper approaches give better results than filter approaches. However, they have a higher computational complexity. Finally, in the embedded approach, the feature selection process is done inside the induction algorithm itself. Such examples are ID3,\textsuperscript{6} C4.5\textsuperscript{7} and CART.\textsuperscript{8}

All of the above mentioned methods are deterministic approaches. In order to overcome problems associated with local maxima, randomness has been introduced into feature selection algorithms. Examples of non-deterministic approaches include genetic algorithms,\textsuperscript{9} evolutionary computation,\textsuperscript{10} and simulated annealing.\textsuperscript{11}

Search engines for feature selection often use \textit{forward} or \textit{backward} methods. Both approaches suffer from the so-called \textit{“nesting effect”}, i.e., in the forward search the
discarded feature cannot be re-selected, whereas in the case of the backward search a feature cannot be discarded once selected. The fitness of a set of features depends on the interaction among the features; the single best feature does not necessarily assure its membership in an optimal feature subset. We know that the optimal search algorithm is the branch and bound algorithm, when the criterion function satisfies monotonicity. Monotonicity implies that a subset of features should be not better than a larger set containing the subset. The branch and bound approach is very efficient compared to the exhaustive search. However, it still becomes impractical for data sets with large numbers of features. Attempts to prevent the “nesting effect” and to attain algorithm efficiency include the plus-l-minus-r (l-r) search method and the floating search algorithm. The drawback of the l-r algorithm is that there is no theoretical way of predicting the values of l and r to achieve the best feature set. The floating search algorithm is an excellent tradeoff between the “nesting effect” and computational efficiency, and there are no parameters to be determined.

We present an efficient wrapper type feature selection algorithm, which acts as a filter approach. The floating search method is used to prevent the “nesting effect”, and it evaluates features based upon a piecewise linear orthonormal (PLO) model. We have applied this idea for regression problems. In the following sections, we first review the OLS procedure for forward selection. We then describe our proposed piecewise linear orthonormal floating search (PLOFS) algorithm. Finally, results for colonic polyps detection data are presented and conclusions are given.

2. OLS Procedure for Forward Selection

In this section, we give the concept of classifier design through regression, the problem formulation of orthonormal linear system and a brief review of the forward OLS procedure.

2.1. Classifier design through regression

Neural network classifiers trained using the mean squared error (MSE) objective function have been shown to approximate the optimal Bayesian classifier. Although the expectation value of the classification error rate is considered to be the ideal training criteria, training algorithms that are based on the minimization of the expected squared error criteria are often easier to mechanize and better understood.

Given a set of data pairs \( \{x_p, i_p\}_{p=1}^{N} \), where the feature vector \( x_p \in \mathbb{R}^N \), and \( i_p \) is an integer class number associated with \( x_p \). We convert \( i_p \) to a real vector \( t_p \in \mathbb{R}^M \) as \( t_p(i_c) = b \) and \( t_p(i_d) = -b \), where \( b \) is a positive constant, \( i_c \) denotes the correct class number for the current training sample, \( i_d \) denotes any incorrect class number for that sample, and \( M \) is the number of classes. Now, the classifier is designed by mapping from \( x \) to \( t \). For example, we designed a mapping, \( f \),

\[
f(x_p) \rightarrow [b, -b]^T, \text{ if } x_p \in \text{ class 1}
\]
for dealing with our two class problem in our feature selection procedure.

2.2. Orthonormal linear system

For the set of data pairs \( \{ \mathbf{x}_p, \mathbf{t}_p \}_{p=1}^{N_v} \), where \( \mathbf{x}_p \in \mathbb{R}^N \) and \( \mathbf{t}_p \in \mathbb{R}^M \). Consider the multiple input multiple output (MIMO) regression model of the form,

\[
y_p(k) = \sum_{i=1}^{N+1} w(k,i)x_p(i)
\]

(3)

\[
t_p(k) - y_p(k) = \xi_p(k),
\]

(4)

where \( 1 \leq k \leq M, \ 1 \leq p \leq N_v \), \( t_p(k) \) is the desired output of the \( k \)-th output for \( p \)-th pattern, \( \xi_p(k) \) is the error between \( t_p(k) \) and the model output \( y_p(k) \). Here \( x_p(N+1) = 1 \) provides a bias term for the output. \( w(k,i) \) is the model weight from the \( i \)-th feature to the \( k \)-th output, \( x_p(i) \) is the \( i \)-th feature or regressor, \( N \) is the total number of candidate features, and \( M \) the number of outputs. Substituting (3) into (4) yields,

\[
t_p(k) = \sum_{i=1}^{N+1} w(k,i)x_p(i) + \xi_p(k), \ 1 \leq k \leq M .
\]

(5)

By defining

\[
\mathbf{t} = \begin{bmatrix}
t_1(1) & t_1(2) & \cdots & t_1(M) \\
t_2(1) & t_2(2) & \cdots & t_2(M) \\
\vdots & \vdots & \ddots & \vdots \\
t_{N_v}(1) & t_{N_v}(2) & \cdots & t_{N_v}(M)
\end{bmatrix}
\]

(6)

\[
\mathbf{x} = \begin{bmatrix}
x_1(1) & x_1(2) & \cdots & x_1(N) & 1 \\
x_2(1) & x_2(2) & \cdots & x_2(N) & 1 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
x_{N_v}(1) & x_{N_v}(2) & \cdots & x_{N_v}(N) & 1
\end{bmatrix}
\]

(7)

\[
\mathbf{w} = \begin{bmatrix}
w(1,1) & w(2,1) & \cdots & w(M,1) \\
w(1,2) & w(2,2) & \cdots & w(M,2) \\
\vdots & \vdots & \ddots & \vdots \\
w(1,N+1) & w(2,N+1) & \cdots & w(M,N+1)
\end{bmatrix}
\]

(8)

\[
\mathbf{\Xi} = \begin{bmatrix}
\xi_1(1) & \xi_1(2) & \cdots & \xi_1(M) \\
\xi_2(1) & \xi_2(2) & \cdots & \xi_2(M) \\
\vdots & \vdots & \ddots & \vdots \\
\xi_{N_v}(1) & \xi_{N_v}(2) & \cdots & \xi_{N_v}(M)
\end{bmatrix}
\]

(9)
the model (5) now can be written in a matrix form,

\[ \mathbf{t} = \mathbf{x}\mathbf{w} + \Xi. \]  

(10)

Suppose \( \mathbf{x} \) can be decomposed as

\[ \mathbf{x} = \Theta \mathbf{A}, \]  

(11)

where

\[
\mathbf{A} = \begin{bmatrix}
    a_{11} & a_{12} & \cdots & \cdots & a_{1,N+1} \\
    0 & a_{22} & \cdots & \cdots & a_{2,N+1} \\
    \vdots & \vdots & \ddots & \ddots & \vdots \\
    0 & \cdots & 0 & a_{N,N} & a_{N,N+1} \\
    0 & \cdots & 0 & 0 & a_{N+1,N+1}
\end{bmatrix} \]  

(12)

\[
\Theta = [\Theta_1, \Theta_2, \cdots, \Theta_{N+1}]. \]  

(13)

Here orthonormal columns satisfy \( \Theta_i^T \Theta_j = I \), \( I \) denotes the identity matrix. The regression model of (10) now becomes

\[ \mathbf{t} = \Theta \mathbf{A}\mathbf{w} + \Xi = \Theta\mathbf{w}_0 + \Xi, \]  

(14)

where \( \mathbf{w}_0 \) denotes the weights for the orthonormal system. The least square solution for (14) is

\[ \mathbf{w}_0 = \Theta^T \mathbf{t}, \]  

(15)

which is projection of the desired output onto the orthonormal bases. The weights \( \mathbf{w} \) for the original system can be easily obtained as

\[ \mathbf{w} = \mathbf{A}^{-1} \mathbf{w}_0. \]  

(16)

Note that in Ref. 16, \( \mathbf{x} \) is decomposed to an orthogonal basis as

\[ \mathbf{x} = \Theta^g \mathbf{A}^g, \]  

(17)

where

\[ \Theta^g = [\Theta_1^g, \Theta_2^g, \cdots, \Theta_{N+1}^g] \]  

(18)

with

\[ \Theta_i^g \Theta_j^g = \begin{cases} 
    0, & i \neq j \\
    \lambda_i, & i = j
\end{cases} \]  

(19)

where \( \lambda_i \) is the square of the length of \( \Theta_i^g \). Define

\[ \Lambda = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_{N+1}\}, \]  

(20)

we get

\[ \Theta = \Theta^g \Lambda^{-1} \]  

(21)
and

$$A = \Lambda A^g.$$  \hspace{1cm} (22)

In other words, in order to get the orthonormal basis $\Theta$, the $i$th row in the transformation matrix $A^g$ is normalized by the length of $\Theta^g$.

Based on a Modified Schmidt procedure (Appendix A), all of the orthonormal process can be fulfilled in terms of the autocorrelation and cross-correlation matrices, which can be obtained by passing through the data only once. The orthonormal procedure utilizes only these matrices, whose sizes are usually much smaller than those of the original data. Therefore, a very efficient algorithm can be implemented.

This orthonormal linear system can be used for feature selection for regression. The following subsection gives a brief review.

2.3. Forward OLS procedure

The task of feature selection for regression is to select the most significant features from the set of $N$ available features. The OLS procedure selects a set of $N_s$ features to form a set of orthonormal bases $\Theta_i$, $0 \leq i \leq N_s$, in a forward manner.

System (14) consists of $M$ subsystems and can be denoted as follows,

$$t(k) = \Theta w_0(k) + \Xi(k), \hspace{1cm} 1 \leq k \leq M,$$ \hspace{1cm} (23)

where $t(k) = \{t_1(k), t_2(k), \ldots, t_{N_v}(k)\}^T$, $w_0(k)$ is the weight matrix connecting to the $k$ outputs, and $\Xi(k) = \{\xi_1(k), \xi_2(k), \ldots, \xi_{N_v}(k)\}^T$. Multiplying equation (23) by its transpose and time averaging, the following equation is easily derived,

$$\frac{1}{N_v} t(k)^T t(k) = \frac{1}{N_v} w_0^T(k) w_0(k) + \frac{1}{N_v} \Xi^T(k) \Xi(k).$$ \hspace{1cm} (24)

The second moment energy for the $k$th output $E[t^2(k)] = \frac{1}{N_v} t(k)^T t(k)$ contains two parts, $\frac{1}{N_v} w_0^T(k) w_0(k)$, explained by the features and $\frac{1}{N_v} \Xi^T(k) \Xi(k)$, the unexplained variance for the $k$th output. The error reduction ratio for the outputs due to the $i$th feature is defined as

$$Err(i) = \frac{\sum_{k=1}^{M} w_0^2(k, i)}{\sum_{k=1}^{M} t(k)^T t(k)}, \hspace{1cm} 1 \leq i \leq N.$$ \hspace{1cm} (25)

The most significant features are forwardly selected according to the value of $Err(i)$. At the first step, $w_0(k, i), 1 \leq k \leq M$, is calculated for the $i$th feature treating it as the first feature to be orthonormal, then $Err(i)$ is obtained using equation (25). For example, we calculate the residual error if only the $k$th feature is used in the regression model. This process is repeated for $N$ times, the feature is selected if it produces the largest value of $Err(i)$. At the second step, the above steps are repeated for the remaining features. For multiple output systems, one could apply
the above procedure for each output separately and obtain different feature subsets for each output system. This is necessary if each subsystem has different dynamic. However, if the multiple output system has similar dynamics, the selected feature subset for each output may not differ much from one to another. We can then make a trade-off between the complexity and the accuracy using equation (25).

We have reviewed the OLS procedure for feature selection that is based on a linear model. However, a linear model cannot adequately describe nonlinear systems which constitute the majority of classification problems. It has been shown that PLN models can approximate nonlinear systems adequately. Thus, we propose a feature selection algorithm based on a PLN model, and the floating search engine is used to avoid the “nesting effect”.

3. The Proposed PLOFS Algorithm

In this section we integrate a PLO model into the forward floating search algorithm. Some important issues about this algorithm are also addressed.

3.1. Piecewise Linear Orthonormal (PLO) system

PLN often employs a clustering method to partition the feature space into a hierarchy of regions (or clusters), where simple hyperplanes are fit to the local data. Thus local linear models construct local approximations to nonlinear mappings.

The regression model (10) for a PLN can be written as

$$ t = x^{(q)} w^{(q)} + \Xi^{(q)} \tag{26} $$

where the superscript denotes that when the feature data belongs to the $q$th cluster, the weight $w^{(q)}$ and error $\Xi^{(q)}$ become valid. We apply the Modified Schmidt procedure to each cluster, yielding the PLO system

$$ t = \Theta^{(q)} A^{(q)} w^{(q)} + \Xi^{(q)} = \Theta^{(q)} w^{(q)} + \Xi^{(q)}. \tag{27} $$

Again, if output systems have similar dynamics, we could use the same partitions for all the output systems.

There are two important issues regarding the PLN model (26): (1) how many clusters in the model are adequate and (2) the way to partition the feature space.

3.2. Number of clusters and partition of feature space

Determining the number of clusters in a PLN for a given set of data is a model validation problem, which is a difficult task because it is not possible simply to choose the model that fits the data best: more complex models always fit data better, but poor generalization often results. In this paper we initially partitioned feature space into a large number of clusters using a Self-Organizing-Map (SOM) (For details see Appendix B). For each cluster a linear regression model is designed, and the classification error for the training data is calculated. The trained PLN is
then applied to the test data to get a testing error. Then a cluster is pruned if its elimination leads the smallest increase in training error. The pruning procedure continues until only one cluster remains. Finally, we produce curves of training and testing errors versus the number of clusters, and we find the minimum value on the testing error curve. The number of clusters corresponding to the minimum of the testing error is chosen for the PLN model thereby completing one implementation of the structural risk minimization principle.\(^\text{21}\)

### 3.3. Floating search through PLO system

To describe the search algorithm we first introduce the following definitions.

Let \( X(d) = \{ x(i) : 1 \leq i \leq d, x(i) \in Z \} \) be a set of \( d \) features from the set \( Z = \{ z(i) : 1 \leq i \leq N \} \) of \( N \) available features. Suppose we partitioned the feature space into \( N_c \) clusters and obtained its PLO system as (27).

**Definition 1:** The individual fitness of one feature, \( x(i) \), is

\[
S_0(x(i)) = \sum_{k=1}^{M} \sum_{q=1}^{N_c} (w_q^{(q)}(k,i))^2 ,
\]

which is the total variance explained for all outputs due to the \( i \)th feature. This is a general measure of the fitness value of one feature regardless of the order of the \( i \)th feature in the orthonormal procedure.

**Definition 2:** The fitness of a set of \( X(d) \) is measured as,

\[
J(X(d)) = \sum_{i=1}^{d} \sum_{k=1}^{M} \sum_{q=1}^{N_c} (w_q^{(q)}(k,i))^2 ,
\]

which is the total variance explained for all outputs due to all features in the set \( X(d) \). Here features in \( X(d) \) are made orthonormal to each other according to the feature search order when \( d \) features have been selected.

**Definition 3:** The fitness \( S_{d-1}(x(i)) \) of the feature \( x(i) \), \( 1 \leq i \leq d \), in the set \( X(d) \) is defined by

\[
S_{d-1}(x(i)) = \sum_{k=1}^{M} \sum_{q=1}^{N_c} (w_q^{(q)}(k,i))^2 ,
\]

where \( x(i) \) is the last feature in the set \( X(d) \) that is made orthonormal to the other bases in the modified Schmidt procedure. This measure is used to identify which feature in the selected feature pool is the least significant one, where the orthonormal weight \( w_q^{(q)}(k,i) \) is the weight from the \( i \)th feature to the \( k \)th output.

The least significant feature in the set \( X(d) \) can be identified in the following procedure: for each \( x(i) \in X(d) \), where \( 1 \leq i \leq d \), let \( x(i) \) be the last feature to be made orthonormal to other features in the feature subset \( X(d) \). Now calculate the
fitness of $x(i)$ as in (30). This procedure is repeated $d$ times, $x(j)$ is identified as the least significant feature in the set $X(d)$ if

$$S_{d-1}(x(j)) = \min_{1 \leq i \leq d} S_{d-1}(x(i)).$$

(31)

Definition 4: The fitness $S_{d+1}(x(i))$ of the feature $x(i)$ with respect of $X(d)$, where $x(i) \in Z - X(d)$, is

$$S_{d+1}(x(i)) = \sum_{k=1}^{M} \sum_{q=1}^{N_c} (w_{o}^{(q)}(k, i))^2,$$

(32)

where $x(i)$ is made orthonormal to $X(d)$, to get $w_{o}(k, i), k = 1, 2, \ldots, M$. The goal of this measure is to identify which feature in the remaining feature pool is the most significant one, so that it can be selected in the current step. The orthonormal weight $w_{o}(k, i)$ is the weight from the $i$th feature to the $k$th output.

In contrast, the most significant feature in the set $Z - X(d)$ is identified as follows: For each feature $x(i)$ from the set $Z - X(d)$, let it be the first feature in the set $Z - X(d)$ to be made orthonormal to $X(d)$ and calculate the fitness of $x(i)$ as in (32). This process is repeated $N - d$ times and the most significant feature with respect to the set $X(d)$ is identified as $x(j)$ if

$$S_{d+1}(x(j)) = \max_{1 \leq i \leq N-d} S_{d+1}(x(i)).$$

(33)

Note that in these definitions, the weights $w_{o}(k, i)$s have different meanings though they have the same notation: they have been obtained in a orthonormal procedure but the features are made to be orthornomal in a different order, and these measures are defined for different purposes. Definition 1 is a general measure that constructs basis for the other three definitions. Definition 2 is a measure of the stop criterion for the Continuation of the conditional deletion, definition 3 is used in the conditional deletion, and definition 4 is used in the Adding one feature in the forward floating search algorithm described in the following subsection.

3.4. Algorithm description

We are now ready to describe the proposed PLOFS algorithm for selecting $N_s$ features from $N$ available features.

(1) Determine the number of clusters, $N_c$, for the PLN model.
(2) Design an $N_c$ cluster PLN model for the training data, accumulate autocorrelation and cross-correlation matrices for each of the clusters.
(3) Initialize $d=0$, and use the forward least square method to form $X(1)$ and $X(2)$.

The fitness value $J(X(d))$ and corresponding members for each subset feature are stored.
(4) Adding one feature. Find the most significant feature, say $x(d+1)$, in the set of $Z - x(d)$ with respect to $X(d)$ using (33), and update

$$X(d+1) = X(d) + x(d+1)$$

(34)
(5) **Conditional deletion.** Using (31) to find the least significant feature, say $x_{(d+1)}$, in the set of $X_{(d+1)}$. Update

$$J(X_{(d+1)}) = J(X_{(d)}) + S_{d+1}(x_{(d+1)})$$

(35)

and set $d = d + 1$. If $d = N_s$, Stop. Otherwise return to step 4. However, if $x_{(m)}$, $m \neq d + 1$, is the least significant feature in the set of $X_{(d+1)}$, delete $x_{(m)}$ from $X_{(d+1)}$ and form $X'_{(d)}$ as

$$X'_{(d)} = X_{(d+1)} - x_{(m)}.$$  

(36)

Update $J(X_{(d)})$ as

$$J(X_{(d)}) = J(X_{(d+1)}) + S_{d+1}(x_{(d+1)}) - S_{d+1}(x_{(m)}).$$

(37)

(6) **Continuation of the conditional deletion.** Find the least significant feature, say $x_{(n)}$, in the set of $X'_{(d)}$. If $J(X'_{(d)} - x_{(n)}) \leq J(X_{(d-1)})$, then set $X_{(d)} = X'_{(d)}$ and return to step 4. Otherwise delete $x_{(n)}$ from $X'_{(d)}$ to form a new set $X'_{(d-1)}$, update $J(X_{(d-1)}) = J((X'_{(d)} - x_{(n)})$ and set $X_{(d-1)} = X'_{(d-1)}$. Set $d = d - 1$, if $d = 2$, return to step 4. Otherwise repeat step 6.

### 4. Related Work

In this section, we briefly review six feature selection algorithms which were used for comparison in this paper. Two algorithms are filter based and four are wrapper based in which an SVM was used as a wrapper for the evaluation of the fitness of the features.

#### 4.1. Filter algorithms for feature selection

Two heuristic algorithms were used to evaluate the proposed algorithm. One algorithm is based on the PLN model we described in the previous section, and the other is based on the Fisher criterion. Both algorithms belong to the filter based feature selection category.

##### 4.1.1. Importance

Many feature selection algorithms are based on the magnitude of the weights in the trained network. We implement a similar method (called “Importance”) based on a PLN model. We first design a PLN model for the given data, where the number of cluster is determined by cross-validation as described in section 3.2. The importance of one feature is calculated as the summation of the magnitudes of the weights to all outputs among all clusters in the PLN. The selection order of the features is based on their importance, and the most important feature is selected first. We use a conjugate gradient method to obtain the weights by solving a set of linear equations for each cluster in the PLN.
4.1.2. Fisher criterion

The Fisher criterion is a classical measure to assess the degree of separation between classes. For our two-class case, it can be stated as,

\[ J(x(i)) = \frac{(m_1(i) - m_2(i))^2}{\sigma_1(i)^2 + \sigma_2(i)^2} \]  

(38)

where \( m_1(i) \) and \( m_2(i) \) are means of the \( i \)th feature for the two classes, and \( \sigma_1(i) \) and \( \sigma_2(i) \) are their corresponding standard deviations. The Fisher criterion gives a high score for a feature if the two classes are far apart compared to the within class variance. The feature with the highest score is selected first.

4.2. Wrapper algorithms for feature selection

Four wrapper based algorithms were used for comparison in this paper, where each algorithm used a SVM classifier to evaluate features. The fitness criterion for these algorithms was defined as the average of sensitivity and specificity of the involved SVM. Sensitivity denotes classification accuracy for positive cases whereas specificity represents classification accuracy for negative cases in the data.

4.2.1. Genetic algorithm

(GA): The basic idea is derived from the Darwinian theory of survival of the fittest, and three fundamental mechanisms drive the evolutionary process: selection, crossover and mutation within chromosomes. Each mechanism occurs with a certain probability that allows for some randomness. For each generation of the GA, individual solutions (feature vectors) are evaluated using a fitness function that was the average of sensitivity and specificity given by an SVM trained on the feature vector. The crossover and mutation within chromosomes then produce the next generation. In our experiments, we set the necessary parameter of the GA algorithm as in Table 1. For details about this algorithm please see Refs. 9 and 23.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>100</td>
</tr>
<tr>
<td>Tournament Selection Size</td>
<td>10</td>
</tr>
<tr>
<td>Probability of Crossover</td>
<td>0.9</td>
</tr>
<tr>
<td>Probability of Mutation</td>
<td>0.1</td>
</tr>
<tr>
<td>Number of Generation</td>
<td>20</td>
</tr>
</tbody>
</table>
4.2.2. Exhaustive search

(ES): This algorithm searches every possible combination of features among all the features. It is optimal in the sense of the chosen fitness value but longer search time is required.

4.2.3. Forward stepwise search

(FSS): Suppose we want to select \( N_s \) features from \( N \) available features, first, a group of \( N_s \) feature vectors are randomly generated. Then one feature in the vector is substituted by a new feature. If the substitution improves the performance, the new vector replaces the old one. The process is iterated on each feature in the vector over and over until no further improvement can be made.

4.2.4. Progressive search

(PS): \( N \)-feature vectors are formed progressively in \( N \) stages. In the first stage, 1-feature vectors are ranked based on their performance, and the maximum top 1000 vectors are passed to the next stage. In the \( N \)th stage, \( N \)-feature vectors are formed by adding one feature to the \((N-1)\)-feature vectors selected in the \((N-1)\)th stage.

5. Results

Different algorithms usually select different feature subsets, so we evaluated these feature subsets using a 10-fold cross validation method followed by a \( t \)-test. The time efficiency of each algorithm was also compared. In ten-fold cross validation, we first randomly divided the available data into 10 equal-sized parts. Each of the ten parts was held out as a test set, and the remaining nine-tenths were used to train the SVM classifiers. The training and testing procedures were repeated 10 times, and we obtained 10 testing results for each feature subset. The \( t \)-test was used to verify if the testing results of SVMs trained by different feature subset were significantly different at the 95% confidence level.

5.1. Data acquisition

CTC procedure was performed on 29 patients with a high suspicion of colonic polyps or masses. There were 19 males and 10 females. The mean age was 69 years (std. dev. 11 years; range 41 to 86 years). All patients had at least one polyp and 27 of them had at least one polyp 1 cm or larger. These patients were chosen from a larger cohort who underwent contrast-enhanced CTC. Selection criteria included that patients had at least 1 polyp \( > 5 \) mm, a majority of which were identified on both the prone and supine views. Optical colonoscopy was performed just after CTC for each patient on the same day. The polyps identified by a radiologist and confirmed by optical colonoscopy were considered as ground truth.
The software first segmented the colon using a region growing algorithm. Next, regions of interest along the colon wall were identified. Polyp segmentation was performed on the regions of interest and a total of 102 different quantitative features were calculated for each polyp candidate based on 2-D or 3-D segmentation algorithms.\textsuperscript{24,25} We therefore had in total four data sets: Supine\textsubscript{2D}, Supine\textsubscript{3D}, Prone\textsubscript{2D} and Prone\textsubscript{3D}. Table 2 shows basic information for these data sets.

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Segmentation</th>
<th>True Positives</th>
<th>False Positives</th>
<th>Feature Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prone\textsubscript{2D}</td>
<td>2D</td>
<td>144</td>
<td>1013</td>
<td>102</td>
</tr>
<tr>
<td>Supine\textsubscript{2D}</td>
<td>2D</td>
<td>214</td>
<td>1021</td>
<td>102</td>
</tr>
<tr>
<td>Prone\textsubscript{3D}</td>
<td>3D</td>
<td>148</td>
<td>1022</td>
<td>102</td>
</tr>
<tr>
<td>Supine\textsubscript{3D}</td>
<td>3D</td>
<td>221</td>
<td>1034</td>
<td>102</td>
</tr>
</tbody>
</table>

5.2. Experiment 1

In Ref. 26, it had been studied that a committee of SVMs with 7 SVM and each SVM using 3 features is statistically adequate for our data sets. For an SVM using 4 features did not provide statistically significant improvements for the classification accuracy over 3 features, where several feature selection algorithms, including GA, were tested. The purpose of this first experiment was to test if the previously tested algorithm, GA, the proposed algorithm, PLOFS, along with the two heuristic algorithms can identify three relevant features. In this experiment, each of the four algorithms selected three features, and those three features were evaluated using 10-fold cross validation followed by a t-test. In the PLOFS algorithm, the number of clusters was determined to be 4 for all the data sets by the cross validation method described in section 3.2.

Tables 3 and 4 show the results of 10-fold cross validation. The fitness value was defined as the average of sensitivity and specificity of SVM. The accuracy in Table 4 and thereafter is referred to as the overall classification accuracy for both the positive and the negative cases in the test data. Table 3 shows fitness comparison of the four algorithms. It is clear that the Importance algorithm was the worst and the PLOFS algorithm was the best. Importance is significantly worse than the best for three data sets. PLOFS won twice, and it is not significantly worse than the best in the cases that other algorithms won. Fisher criterion method won once and it is significantly worse than the best for the Supine\textsubscript{3D} data set. GA was the second best, it won once and is never significantly worse than the best for any of the data sets.

The superiority of PLOFS is more apparent when the overall accuracy was considered. As shown in Table 4, PLOFS won three times, and it was not significantly worse than GA for Supine\textsubscript{3D} data set which was the only data set that GA won. All algorithms except PLOFS were significantly worse than the best for at least two data sets.
Table 3. Fitness Comparison of PLOFS with Two Heuristic Algorithms and GA by a Single SVM Utilizing Three Features.

<table>
<thead>
<tr>
<th>Data</th>
<th>Importance Fitness (%)</th>
<th>Fisher Criterion Fitness (%)</th>
<th>PLOFS Fitness (%)</th>
<th>GA Fitness (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prone_2D</td>
<td>66.32−</td>
<td>81.08</td>
<td>82.03</td>
<td>82.98</td>
</tr>
<tr>
<td>Prone_3D</td>
<td>67.78−</td>
<td>83.18</td>
<td>82.92</td>
<td>80.72</td>
</tr>
<tr>
<td>Supine_2D</td>
<td>59.41−</td>
<td>84.16</td>
<td>87.52</td>
<td>85.89</td>
</tr>
<tr>
<td>Supine_3D</td>
<td>83.59</td>
<td>82.71−</td>
<td>88.22</td>
<td>86.89</td>
</tr>
</tbody>
</table>

*Note: All fitness values are the mean of the ten runs, and each fitness value is the average of sensitivity and specificity of SVMs. The maximum mean fitness value for each data set is shown in bold, and a ‘−’ sign means that the mean value is significantly worse than the corresponding maximum mean at the 95% confidence level.*

Table 4. Accuracy Comparison of PLOFS with Two Heuristic Algorithms and GA by a Single SVM Utilizing Three Features.

<table>
<thead>
<tr>
<th>Temp Data</th>
<th>Importance Accuracy (%)</th>
<th>Fisher Criterion Accuracy (%)</th>
<th>PLOFS Accuracy (%)</th>
<th>GA Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prone_2D</td>
<td>69.51−</td>
<td>81.50−</td>
<td>85.33</td>
<td>81.58−</td>
</tr>
<tr>
<td>Prone_3D</td>
<td>68.05−</td>
<td>81.30−</td>
<td>86.34</td>
<td>80.16−</td>
</tr>
<tr>
<td>Supine_2D</td>
<td>58.45−</td>
<td>87.62</td>
<td>88.95</td>
<td>88.74</td>
</tr>
<tr>
<td>Supine_3D</td>
<td>87.43−</td>
<td>84.84−</td>
<td>90.64</td>
<td>90.72</td>
</tr>
</tbody>
</table>

*Note: All accuracy values are the average of the ten runs, and each accuracy value is the classification accuracy of SVMs. The maximum mean accuracy for each data set is shown in bold, and a ‘−’ sign means that the mean value is significantly worse than the corresponding maximum mean at the 95% confidence level.*

5.3. *Experiments 2*

In this experiment, we compared the wrapper feature selection algorithms (ES, FSS, GA and PS) reviewed in section 4. We used a committee of 7 SVMs as the classifier, and each classifier used 3 features. This configuration is considered as statistically optimal for our data sets. Each of the four algorithms generated many 3-feature combinations, the algorithms kept the first 1000 such combinations according to their fitness values. The best SVM committee with 7 members was then searched among those 1000 combinations. We again used 10-fold cross validation to evaluate the selected feature set using the committee of SVMs.

Table 5 shows the results for this experiment. It is observed that all fitness and overall accuracy values were very similar. The *t*-tests show that there are no significant differences between the best result and any other result for each data set. These four feature selection algorithms are then considered as a set of statistically optimal feature selection algorithm for our data sets.

Although all four algorithms can provide good solutions for our data sets, some of them are very time consuming. For example, searching a set of 3 features among
Table 5. Ten Fold Cross Validation Results of Four Feature Selection Algorithms.

<table>
<thead>
<tr>
<th>Data</th>
<th>ES Fitness (%)</th>
<th>ES Acc. (%)</th>
<th>FSS Fitness (%)</th>
<th>FSS Acc. (%)</th>
<th>GA Fitness (%)</th>
<th>GA Acc. (%)</th>
<th>PS Fitness (%)</th>
<th>PS Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prone_2D</td>
<td>84.41</td>
<td>86.65</td>
<td>84.71</td>
<td>84.37</td>
<td>84.96</td>
<td>83.78</td>
<td>84.96</td>
<td>83.94</td>
</tr>
<tr>
<td>Prone_3D</td>
<td><strong>85.81</strong></td>
<td>85.35</td>
<td>85.01</td>
<td>85.01</td>
<td>83.83</td>
<td>84.02</td>
<td>85.23</td>
<td><strong>85.40</strong></td>
</tr>
<tr>
<td>Supine_2D</td>
<td>86.69</td>
<td>87.78</td>
<td>87.01</td>
<td>87.69</td>
<td>85.83</td>
<td>87.00</td>
<td><strong>87.68</strong></td>
<td><strong>89.08</strong></td>
</tr>
<tr>
<td>Supine_3D</td>
<td>88.31</td>
<td><strong>93.99</strong></td>
<td>88.31</td>
<td><strong>93.99</strong></td>
<td>87.50</td>
<td>90.01</td>
<td><strong>88.41</strong></td>
<td>93.86</td>
</tr>
</tbody>
</table>

Note: Fitness represents all numbers are the average value of the ten runs. Acc. denotes the overall accuracy, and fitness is the average of sensitivity and specificity of SVMs. The maximum mean for a data set is shown in bold, and there is no significant difference between the best result and any other result for each data set at the 95% confidence level.

5.4. Experiment 3

In order to improve the efficiency of the feature selection algorithm, we first used filter based algorithms to reduce the 102 features to 25 as a preprocessing step. ES then searched all possible combinations of three features among the remaining 25 features using SVMs for evaluating features. The final SVM committee was generated the same way as before. The algorithms used for feature reduction included the Importance, Fisher criterion, PLOFS, and PLO-Forward algorithms. The PLO-Forward algorithm is the same as the PLOFS algorithm except that the former used stepwise forward method to search useful features and the latter used the floating method for the same purpose. Again, we used 10-fold cross validation followed by a t-test to evaluate the feature subset selected by each algorithm.

Tables 6 and 7 show the fitness and overall accuracy comparisons for the four hybrid methods and the ES algorithm. Where the last column, the ES algorithm, means that we used the ES algorithm to search for useful feature set among all the available features. For the fitness comparison, only the importance+ES algorithm can not provide a comparable solution for the problem. The other three hybrid algorithms were comparable to the optimal algorithm, the ES algorithm. If overall accuracy was considered, however, only the PLOFS+ES algorithm was comparable to the ES algorithm (See Table 7).

Table 8 lists some feature subsets selected by the PLOFS algorithm for Prone_3D data set during the search procedure. The best 5-feature subset does not contain all members of the best 4-feature subset, and the best 9-feature subset does not contain all members of the best 8-feature subset either, both are clearly signs that
Table 6. Fitness Comparison between the ES Algorithm and Four Hybrid Algorithms.

<table>
<thead>
<tr>
<th>Data</th>
<th>Importance +ES Fitness (%)</th>
<th>Fisher Criterion +ES Fitness (%)</th>
<th>PLO-Foward +ES Fitness (%)</th>
<th>FLOFS +ES Fitness (%)</th>
<th>ES Fitness (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prone_2D</td>
<td>76.75^−</td>
<td>83.27</td>
<td>84.75</td>
<td>84.49</td>
<td>84.41</td>
</tr>
<tr>
<td>Prone_3D</td>
<td>77.47</td>
<td>82.39</td>
<td>84.60</td>
<td>85.15</td>
<td>85.81</td>
</tr>
<tr>
<td>Supine_2D</td>
<td>79.36^−</td>
<td>87.01</td>
<td>86.10</td>
<td>87.58</td>
<td>86.69</td>
</tr>
<tr>
<td>Supine_3D</td>
<td>83.46</td>
<td>88.96</td>
<td>87.74</td>
<td>88.43</td>
<td>88.31</td>
</tr>
</tbody>
</table>

Note: ‘Importance+ES’ denotes that the Importance algorithm first reduced the 102 features down to 25 and the ES algorithm searched all combinations among the 25 features. All Fitness values are the average of the ten runs. The maximum mean value for a data set is shown in bold, and a ‘^−’ sign means that the mean value is significantly worse than the corresponding maximum mean at the 95% confidence level.

Table 7. Accuracy Comparison between the ES Algorithm and Four Hybrid Algorithms.

<table>
<thead>
<tr>
<th>Data</th>
<th>Importance +ES Acc. (%)</th>
<th>Fisher Criterion +ES Acc. (%)</th>
<th>PLO-Foward +ES Acc. (%)</th>
<th>FLOFS +ES Acc. (%)</th>
<th>ES Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prone_2D</td>
<td>82.89^−</td>
<td>83.05^−</td>
<td>81.89^−</td>
<td>86.78</td>
<td>86.65</td>
</tr>
<tr>
<td>Prone_3D</td>
<td>84.65</td>
<td>81.50^−</td>
<td>83.24^−</td>
<td>84.20</td>
<td>85.35</td>
</tr>
<tr>
<td>Supine_2D</td>
<td>86.41^−</td>
<td>90.04</td>
<td>89.77</td>
<td>90.05</td>
<td>87.78</td>
</tr>
<tr>
<td>Supine_3D</td>
<td>87.47^−</td>
<td>93.58</td>
<td>91.87</td>
<td>92.41</td>
<td>93.99</td>
</tr>
</tbody>
</table>

Note: ‘Importance+ES’ denotes that the Importance algorithm first reduced the 102 features down to 25 and the ES algorithm searched all combinations among the 25 features. All accuracy values are the average of the ten runs. The maximum mean accuracy for a data set is shown in bold, and a ‘^−’ sign means that the mean value is significantly worse than the corresponding maximum mean at the 95% confidence level.

Table 8. Sample Feature Subsets Selected by PLOFS for The Prone_3D Data Set.

<table>
<thead>
<tr>
<th>Feature Subset Size</th>
<th>Indexes of Feature Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{100}</td>
</tr>
<tr>
<td>2</td>
<td>{100, 80}</td>
</tr>
<tr>
<td>3</td>
<td>{100, 80, 101}</td>
</tr>
<tr>
<td>4</td>
<td>{100, 80, 101, 69}</td>
</tr>
<tr>
<td>5</td>
<td>{100, 80, 101, 71, 2}</td>
</tr>
<tr>
<td>6</td>
<td>{100, 80, 101, 71, 2, 69}</td>
</tr>
<tr>
<td>7</td>
<td>{100, 80, 101, 71, 2, 69, 84}</td>
</tr>
<tr>
<td>8</td>
<td>{100, 80, 101, 71, 2, 69, 84, 102}</td>
</tr>
<tr>
<td>9</td>
<td>{100, 80, 101, 71, 96, 69, 84, 98, 43}</td>
</tr>
</tbody>
</table>

Note: Each feature subset was considered as optimal by PLOFS algorithm during the search procedure.
the nesting effect was successfully avoided. The 25 features selected by the PLOFS algorithm contains all necessary information to compose the optimal SVM committee, which was searched by the ES algorithm, for our data sets.

Table 9 shows time efficiencies of the algorithms, where significant differences are clearly demonstrated. The most efficient algorithm, PLOFS+ES, only required about 1 or 2 minutes. The second efficient one, GA, needed 6 or 7 minutes. The other three algorithms took much longer time to complete. The committee selection took the same amount of time (around 20 mins) for each algorithm, and was excluded from time-efficiency comparisons.

<table>
<thead>
<tr>
<th>Data</th>
<th>ES</th>
<th>FSS</th>
<th>GA</th>
<th>PS</th>
<th>PLOFS+ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prone 2D</td>
<td>90</td>
<td>23</td>
<td>6</td>
<td>35</td>
<td>1 (+5 sec)</td>
</tr>
<tr>
<td>Supine 2D</td>
<td>134</td>
<td>54</td>
<td>7</td>
<td>70</td>
<td>2 (+6 sec)</td>
</tr>
<tr>
<td>Prone 3D</td>
<td>92</td>
<td>28</td>
<td>6</td>
<td>33</td>
<td>1 (+5 sec)</td>
</tr>
<tr>
<td>Supine 3D</td>
<td>144</td>
<td>47</td>
<td>7</td>
<td>75</td>
<td>2 (+6 sec)</td>
</tr>
</tbody>
</table>

Note: All values are in minutes, except times in parentheses which are in seconds, e.g., ‘1(+5sec)’ denotes that ES requires 1 minute to search 1000 combinations among 25 features which are selected by PLOFS within 5 seconds.

5.5. Discussion

Decreasing the dimensionality of the feature vector before using a wrapper algorithm for exhaustive searching is very important in practical applications. CTC data used in this paper have 102 features which is quite large for many feature selection algorithms. It is very time consuming if we try to search useful features among all the available features using an exhaustive search method. By reducing the dimensionality of the data to 25 using the PLOFS algorithm, we significantly increased the time efficiency because search time increases exponentially with dimensionality. The FSS and PS algorithms improved time efficiency compared to that of ES, however, for a large size data they still may not be feasible. Since all the algorithms give statistically similar performances, time efficiency becomes the critical factor.

The Importance and Fisher criterion feature selection algorithms are essentially filter algorithms, where each feature obtained a rank. The selection order is then based on the rank of each feature. Those algorithms are very time efficient which cost only about 1 second for our data sets. The rank of features may represent discriminative power of each feature when they are examined individually. However, rank of features does not take into account the interaction among features, and therefore suffers from the nesting effect. We clearly showed that if we reduced the 102 features to 25 by either the Importance or Fisher criterion algorithm followed
by the ES algorithm, both hybrid algorithms could not provide solution comparable with the ES algorithm alone. By incorporating the floating search algorithm, the PLOFS algorithm successfully avoided the nesting effect, and it provided solutions comparable with the ES algorithm in a time-efficient fashion.

Feature selection is a very difficult task, where many factors may affect performance of combinations of features. Guyon and Elisseeff\textsuperscript{27} showed that in some cases the combination of some random features may significantly improve the classification accuracy though each individual feature has no discriminative power at all. Stepwise forward or rank based feature selection algorithms will have less chance to pick up an individually useless feature even it could be useful when combined with other features. The proposed PLOFS algorithm had a high possibility to avoid this effect by using the floating search. It also improved time efficiency upon the ES algorithm or GA algorithm alone. It is worthy to note that PLOFS is not optimal because it did not try all possible feature combinations. It only tried the combinations with potential high discriminative power, and provided a good trade-off between performance and complexity.

The PLOFS algorithm is essentially a wrapper-type feature selection algorithm, where a piecewise linear classifier (PLC) was involved for evaluating each feature combination. The procedure of solving weights for the PLC was bypassed by the modified Schmidt procedure, which made PLOFS act as a filter feature selection algorithm because it did not need to pass through the whole data during the search process. Note that if the number of instances is less than the number of features in one data set, this PLOFS algorithm is not appropriate. For this case, we need to pass through its auto-correlation and cross-correlation matrices which are bigger than the original data!

6. Conclusions

We proposed a novel approach for feature selection in colonic polyp detection. First, we determined an appropriate PLN model for the given data. Then, we applied the floating search algorithm on the PLN model to select a feature subset, through an OLS procedure.

We showed that the proposed algorithm was better than the other two heuristic algorithms, the Importance and Fisher criterion algorithm, when selecting 3 features for classification. The PLOFS algorithm was comparable to the GA algorithm which uses an SVM classifier as a wrapper for selecting features.

We compared the proposed PLOFS+ES algorithm with 4 other wrapper-type algorithms on the CTC data sets. Our results showed that all 5 algorithms gave statistically similar fitness values, but the proposed algorithm was the most efficient one, increasing time efficiency by a factor of three. We also showed that the two heuristic algorithms combined with ES algorithm did not give comparable results with that of the optimal methods for our data sets, though the two heuristic algorithms were the most time efficient ones.
Appendix A. The Modified Gram-Schmidt Procedure

The normal or modified Gram-Schmidt procedure\(^{28}\) is a recursive process that requires scalar products between raw basis functions and orthonormal basis functions. The disadvantage of this is that one pass through the training data is required to obtain each new basis function. In the following, a more useful form of the Schmidt process is reviewed, which enables us to express the orthonormal system in terms of autocorrelation elements.

Rewrite (11) as

\[ \Theta = xA^{-1}, \]  
(A.1)

since \( A \) is an upper triangular matrix so that \( A^{-1} \) is also upper triangular. Define

\[ A^{-1} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \cdots & \alpha_{1,N+1} \\ 0 & \alpha_{22} & \cdots & \cdots & \alpha_{2,N+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \alpha_{N,N} & \alpha_{N,N+1} \\ 0 & \cdots & 0 & \alpha_{N+1,N+1} \end{bmatrix} \]  
(A.2)

From (A.1) and (A.2), the \( m \)th orthonormal basis function \( \Theta_m \) can be expressed as

\[ \Theta_m = \sum_{k=1}^{m} x(k)a_{mk}, \]  
(A.3)

for \( m = 1 \), the first basis function is obtained as

\[ \Theta_1 = \alpha_{11}x(1) \]  
(A.4)

where

\[ \alpha_{11} = \frac{1}{\|x(1)\|} = \frac{1}{r(1,1)^{1/2}} \]  
(A.5)

and

\[ r(i,j) = \langle x(i), x(j) \rangle = \frac{1}{N_v} \sum_{j=1}^{N_v} x_p(i)x_p(j) \]  
(A.6)

is the autocorrelation of \( x(i) \) and \( x(j) \). For values of \( m \) between 2 and \( N + 1 \), \( c_i \) is first found for \( 1 \leq i \leq m - 1 \) as

\[ c_i = \sum_{q=1}^{i} \alpha_{iq}r(q,m), \]  
(A.7)

then obtain \( m \) coefficients \( b_k \) as

\[ b_k = \begin{cases} -\sum_{i=k}^{m-1} c_i\alpha_{ik}, & 1 \leq k \leq m - 1 \\ b_m = 1 \end{cases} \]  
(A.8)
Finally for the \( m \)th basis function the new \( \alpha_{mk} \) coefficients are found as

\[
\alpha_{mk} = \frac{b_k}{|r(m, m) - \sum_{i=1}^{m-1} C_i|^2/2}, \quad 1 \leq k \leq m. \tag{A.9}
\]

Using (15) the weights for the orthonormal system can be obtained as

\[
w_0 = \Theta^T t = (A^{-1})^T X^T t = (A^{-1})^T C \tag{A.10}
\]

where \( C \) is the cross-correlation matrix defined as

\[
C(i, k) = \frac{1}{N_v} \sum_{p=1}^{N_v} x_p(i)t_p(k) \tag{A.11}
\]

specifically, the weights from the \( m \)th basis to the \( k \)th output can be expressed as

\[
w_0(k, m) = \sum_{q=1}^{m} \alpha_{mq} c(k, q). \tag{A.12}
\]

If the \( m \)th feature is linearly dependent with the previous \( m - 1 \) basis functions, we just obtain all zero-valued weights for the \( m \)th feature. This will eliminate the numerical problem and the linearly dependent feature will not contribute to the explanation of the output variance. Using (16) the weights for the original system can be readily found as

\[
w(k, m) = \sum_{q=k}^{N+1} w_0(k, q) \alpha_{qm}. \tag{A.13}
\]

The above process enables us to calculate all of orthonormal bases in terms of the autocorrelation matrix \( R \) and cross-correlation matrix \( C \), which can be obtained by passing through the data only once before the orthonormal process.

**Appendix B. Self-organizing Map**

SOM is an unsupervised learning (clustering) algorithm and has many successful applications.\(^{29,30,31}\) In a SOM model, \( x \) denotes the \( m \)-dimensional input vector, \( w_j \) denotes the \( j \)th cluster center vector, where \( x, w_j \in \mathbb{R}^m \) and \( 1 \leq j \leq l \). Here \( m \) is the number of dimensionality of \( x \) and \( l \) is the total number of clusters.

SOM uses a competitive, cooperative and adaptive unsupervised learning procedure to adjust these cluster centers and the final spatial coordinates of the cluster centers can provide an interpretation of the input data information. For each input vector \( x \), the distance between \( x \) and \( w_j \) is computed as \( \| w_j - x \| \), where any norm may be used. The \( i \)th center is declared to be the winning center if \( w_i \) has the smallest distance to \( x \). In addition, there are some neighbors of the winning center declared to be excited centers under the control of a topological neighborhood function \( h_{ji} \). Let \( d_{ji} = |j - i| \) denote the lateral distance between the winning center
i and the candidate center $j$ to be excited, a typical choice of $h_{ji}$ is
\[ h_{ji} = \exp^{-\frac{d^2_{ji}}{2\sigma^2(n)}} \]  
(B.1)
where $\sigma(n)$ is the effective width of the topological neighborhood which shrinks with time $n$ as
\[ \sigma(n) = \sigma_0 \exp^{-\frac{n}{\tau_1}} \]  
(B.2)
here $\tau_1$ is a time constant with an initial value of $\tau_0$. After the above competitive and cooperative process, the winning center and its excited neighbor centers are moved towards $x$ by performing
\[ w_j(n+1) = w_j(n) + \eta(n)h_{ji}(x)(n)(x - w_j(n)) \]  
(B.3)
where $w_j$ represents the winning or excited center, $\eta(n)$ is the learning rate, which starts at a small initial value $\eta_0$ and decreases with time $n$ as
\[ \eta = \eta_0 \cdot \exp^{-\frac{n}{\tau_2}} \]  
(B.4)
where $\tau_2$ is a time constant.

Once the SOM algorithm converged, the resulted feature map (cluster centers) display important statistical characteristics of the data: the cluster centers converged to the locations that represent the probability distribution of the data, and the feature map is topologically ordered. For example, the cluster centers with adjacent indexes are grouped adjacent.

References
15. Jiang Li, Michael T. Manry, and Chuanghua Yu, “Feature selection using a piecewise linear network,” *Accepted for publication, IEEE Transactions on Neural Networks*.