

An Efficient Feature Selection Algorithm for Computer-Aided Polyp Detection

Jiang Li, Jianhua Yao, Ronald M. Summers
Clinical Center, National Institutes of Health, MD

Amy Hara
Mayo Clinic, Scottsdale, AZ

Abstract

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 based on a learning theorem for the given data set, 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 selected by other methods, through cross-validation with a committee of support vector machines (SVMs).

Introduction

Colon cancer is the third leading cause of cancer deaths in the US. 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 prevail, 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. (Summers *et al.* 2002) describes recent work on a version of computer automated 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 and 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 *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 is centered in feature selection for computer-aided polyp detection with the goal of selecting a compact subset of features that leads to the best prediction of classification accuracy based on available data at hand.

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 maximum fitness value. Feature selection algorithms may be divided into *filter*, *wrapper* and *embedded* categories. A filter approach performs some preprocessing of the data without any actual classifier involved; examples include FOCUS (Almuallin & Dietterich 1991), RELIEF (Kira & Rendell 1992). A filter approach has the advantage of computational efficiency but the selected feature subset may not have a good performance. A wrapper approach determines the fitness of a feature subset by actually training a classifier (Kohavi & John 1997). Usually wrapper approaches give better results than that of filter approaches. However, they have a higher computational complexity. Finally, in the case of embedded approach, the feature selection process is done inside the induction algorithm itself. Such examples are ID3 (Quinlan 1986), C4.5 (Quinlan 1993) and CART (Breiman *et al.* 1983).

All of the above mentioned methods are deterministic approaches. On the other side, randomness has been introduced into feature selection algorithms to escape from local maximum. Examples of non-deterministic approaches include genetic algorithms (Raymer *et al.* 2000), evolutionary computation (Etxeberria *et al.* 2000), and simulated annealing (Siedlecki & Sklansky 1988).

Search engines used for feature selection are often *forward* or *backward* method. However, both approaches suffer from the so-called “*nesting effect*”, i.e., in the forward search the discarded feature cannot be re-selected while in the case of the backward search the 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* (Narendra & Fukunaga 1977), which requires a monotonic criterion function. 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 (Stearns 1976) and the *floating* search algorithm (Pudil, Novovičová, & Kittler 1994). 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 without passing through data. We have applied this idea for regression problems (Li, Manry, & Yu). 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.

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 (Chen, Billings, & Luo 1989).

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 (Ruck *et al.* 1990). 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 $\{\mathbf{x}_p, i_p\}_{p=1}^{N_v}$, where the feature vector $\mathbf{x}_p \in \mathbf{R}^N$, and i_p is an integer class number associated with \mathbf{x}_p . We convert i_p to a real vector $\mathbf{t}_p \in \mathbf{R}^M$ 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 fitting a mapping from \mathbf{x} to \mathbf{t} .

Orthonormal Linear System

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

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

$$t_p(k) - y_p(k) = \xi_p(k) \quad (2)$$

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 $x_p(N+1) = 1$ handles output threshold. $w(k, i)$ 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 (1) into (2) yields,

$$t_p(k) = \sum_{i=1}^{N+1} w(k, i)x_p(i) + \xi_p(k), 1 \leq k \leq M \quad (3)$$

By defining

$$\mathbf{t} = \begin{bmatrix} t_1(1) & t_1(2) & \cdots & t_1(M) \\ t_2(1) & t_2(2) & \cdots & t_2(M) \\ \cdots & \cdots & \cdots & \cdots \\ t_{N_v}(1) & t_{N_v}(2) & \cdots & t_{N_v}(M) \end{bmatrix} \quad (4)$$

$$\mathbf{x} = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(N) & 1 \\ x_2(1) & x_2(2) & \cdots & x_2(N) & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{N_v}(1) & x_{N_v}(2) & \cdots & x_{N_v}(N) & 1 \end{bmatrix} \quad (5)$$

$$\mathbf{w} = \begin{bmatrix} w(1,1) & w(2,1) & \cdots & w(M,1) \\ w(1,2) & w(2,2) & \cdots & w(M,2) \\ \cdots & \cdots & \cdots & \cdots \\ w(1,N+1) & w(2,N+1) & \cdots & w(M,N+1) \end{bmatrix} \quad (6)$$

$$\Xi = \begin{bmatrix} \xi_1(1) & \xi_1(2) & \cdots & \xi_1(M) \\ \xi_2(1) & \xi_2(2) & \cdots & \xi_2(M) \\ \cdots & \cdots & \cdots & \cdots \\ \xi_{N_v}(1) & \xi_{N_v}(2) & \cdots & \xi_{N_v}(M) \end{bmatrix} \quad (7)$$

the model (3) now can be written in a matrix form,

$$\mathbf{t} = \mathbf{x}\mathbf{w} + \Xi \quad (8)$$

Suppose \mathbf{x} can be decomposed as

$$\mathbf{x} = \Theta\mathbf{A} \quad (9)$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1,N+1} \\ 0 & a_{22} & \cdots & \cdots & a_{2,N+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & a_{N,N} & a_{N,N+1} \\ 0 & 0 & \cdots & 0 & a_{N+1,N+1} \end{bmatrix} \quad (10)$$

$$\Theta = [\theta_1, \theta_2, \cdots, \theta_{N+1}] \quad (11)$$

Here orthonormal columns satisfy $\theta_i^T \theta_j = I$, I denotes the identity matrix. The regression model of (8) now becomes

$$\mathbf{t} = \Theta\mathbf{A}\mathbf{w} + \Xi = \Theta\mathbf{w}_0 + \Xi \quad (12)$$

where \mathbf{w}_0 denotes the weights for the orthonormal system. The least square solution for (12) is

$$\mathbf{w}_0 = \Theta^T \mathbf{t} \quad (13)$$

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

$$\mathbf{w} = \mathbf{A}^{-1} \mathbf{w}_0 \quad (14)$$

Based on a Modified Schmidt procedure (Maldonado & Manry 2002), all of orthonormal process can be fulfilled in terms of autocorrelation matrix \mathbf{r} and cross-correlation matrix \mathbf{C} , which can be obtained by pass through data only once, where \mathbf{r} and \mathbf{C} are defined as,

$$r(i, j) = \langle \mathbf{x}(i), \mathbf{x}(j) \rangle = \frac{1}{N_v} \sum_{p=1}^{N_v} x_p(i)x_p(j) \quad (15)$$

and

$$c(i, k) = \frac{1}{N_v} \sum_{p=1}^{N_v} x_p(i)t_p(k) \quad (16)$$

Once these matrices are calculated through one data pass, the orthonormal procedure only utilizes 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 (Chen *et al.* 2004). The following subsection gives a brief review.

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 θ_i , $0 \leq i \leq N_s$, in a forward manner.

System (12) consists of M subsystems and can be denoted as follows,

$$\mathbf{t}(k) = \Theta \mathbf{w}_0(k) + \Xi(k), 1 \leq k \leq M. \quad (17)$$

where $\mathbf{t}(k) = \{t_1(k), t_2(k), \dots, t_{N_v}(k)\}^T$, $\mathbf{w}_0(k)$ is the weight matrix connecting to the k outputs, and $\Xi(k) = \{\xi_1(k), \xi_2(k), \dots, \xi_{N_v}(k)\}^T$. Multiplying (17) by its transpose and time averaging, the following equation is easily derived,

$$\frac{1}{N_v} \mathbf{t}(k)^T \mathbf{t}(k) = \frac{1}{N_v} \mathbf{w}_0^T(k) \mathbf{w}_0(k) + \frac{1}{N_v} \Xi^T(k) \Xi(k). \quad (18)$$

The variance or energy for the k th output $E[\mathbf{t}^2(k)] = \frac{1}{N_v} \mathbf{t}^T(k) \mathbf{t}(k)$ contains two parts, $\frac{1}{N_v} \mathbf{w}_0^T(k) \mathbf{w}_0(k)$, the variance 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

$$Err(i) = \frac{\sum_{k=1}^M w_0^2(k, i)}{\sum_{k=1}^M \mathbf{t}^T(k) \mathbf{t}(k)}, 1 \leq i \leq N. \quad (19)$$

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 (19). 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 subset for each output

system. This is necessary if each subsystem has a different dynamics (see (Hong & Harris 2001)). However, if the multiple output system has similar dynamics, the selected subset features for each output may not differ much from one to another. We can then make a trade-off between the complexity and the accuracy as of (19), which is the summation of error reduction ration for all the outputs due to the i th feature.

We have reviewed the OLS procedure for feature selection which is based on a linear model. However, linear model cannot adequately describe nonlinear system which are usually the case for classification problems. It has been shown that PLN models can approximate nonlinear systems adequately (Billings & Voon 1987). Thus, we propose a feature selection algorithm based on a PLN model, and the floating search engine is used to avoid the ‘‘nesting effect’’.

The Proposed PLOFS Algorithm

In this section we integrate a PLO model into the forward floating search algorithm (Pudil, Novovičová, & Kittler 1994). Some important issues about this algorithm are also addressed.

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 (8) for a PLN can be written as

$$\mathbf{t} = \mathbf{x}^{(q)} \mathbf{w}^{(q)} + \Xi^{(q)} \quad (20)$$

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

$$\mathbf{t} = \Theta^{(q)} \mathbf{A}^{(q)} \mathbf{w}^{(q)} + \Xi^{(q)} = \Theta^{(q)} \mathbf{w}_o^{(q)} + \Xi^{(q)}. \quad (21)$$

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 (20): how many clusters in the model are adequate and the way to partition the feature space.

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 bad generalization often results. In this paper we initially partitioned space partitioned into a large number of clusters using a Self-Organizing-Map (SOM) (Kohonen 1989). For each cluster a linear regression model is designed, and the classification error for the training data is calculated. The trained PLN is applied to the test data to get testing error. Then a cluster is pruned if its elimination leads the smallest increases of training error. The pruning procedure continues till there is only one cluster remains. Finally we produce curves of training error and testing error 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* (Vapnik 1998).

Floating Search Through PLO System

In order to describe the search algorithm we first introduce the following definitions.

Let $\mathbf{X}(d) = \{x(i) : 1 \leq i \leq d, x(i) \in \mathbf{Z}\}$ be a set of d features from the set $\mathbf{Z} = \{z(i) : 1 \leq i \leq N\}$ of N available features. Suppose we partitioned the feature space into N_c clusters and obtained its PLO system as (21),

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

$$\mathbf{S}_0(\mathbf{x}(i)) = \sum_{k=1}^M \sum_{q=1}^{N_c} (w_o^{(q)}(k, i))^2 \quad (22)$$

which is the total variance explained for all outputs due to the i th feature.

Definition 2: The fitness of a set of $\mathbf{X}(d)$ is

$$J(\mathbf{X}(d)) = \sum_{i=1}^d \sum_{k=1}^M \sum_{q=1}^{N_c} (w_o^{(q)}(k, i))^2 \quad (23)$$

which is the total variance explained for all outputs due to all features in the set $\mathbf{X}(d)$.

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

$$\mathbf{S}_{d-1}(\mathbf{x}(i)) = \sum_{k=1}^M \sum_{q=1}^{N_c} (w_o^{(q)}(k, i))^2 \quad (24)$$

where $x(i)$ is the last feature in the set $\mathbf{X}(d)$ that is made orthonormal to the other bases in the Schmidt procedure.

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

$$\mathbf{S}_{d+1}(\mathbf{x}(i)) = \sum_{k=1}^M \sum_{q=1}^{N_c} (w_o^{(q)}(k, i))^2 \quad (25)$$

here $x(i)$ is made orthonormal to $\mathbf{X}(d)$, to get $w_o(k, i)$, $k = 1, 2, \dots, M$.

Definition 1 is used to evaluate the general fitness of one feature in the set $\mathbf{X}(d)$. *Definition 2* is used to calculate the total fitness of one set of features. *Definition 3* is used to identify which feature in the set $\mathbf{X}(d)$ is the least significant feature. *Definition 4* is used to determine which feature in the $\mathbf{Z} - \mathbf{X}(d)$ is the most significant feature.

The least significant feature in the set $\mathbf{X}(d)$ can be identified in the following procedure: for each $x(i) \in \mathbf{X}(d)$, where $1 \leq i \leq d$, let $x(i)$ be the last feature to be made orthonormal to other features in the feature subset $\mathbf{X}(d)$. Now calculate the fitness of $x(i)$ as in (24). This procedure is repeated d times, $x(j)$ is identified as the least significant feature in the set $\mathbf{X}(d)$ if

$$S_{d-1}(x(j)) = \min_{1 \leq i \leq d} S_{d-1}(x(i)). \quad (26)$$

In contrast, the most significant feature in the set $\mathbf{Z} - \mathbf{X}(d)$ is identified as follows: For each feature $x(i)$ from the set $\mathbf{Z} - \mathbf{X}(d)$, let it be the first feature in the set $\mathbf{Z} - \mathbf{X}(d)$ to be made orthonormal to $\mathbf{X}(d)$ and calculate the fitness of $x(i)$ as (25). This process is repeated $N - d$ times and the most significant feature with respect to the set $\mathbf{X}(d)$ is identified as $x(j)$ if

$$\mathbf{S}_{d+1}(\mathbf{x}(j)) = \max_{1 \leq i \leq N-d} \mathbf{S}_{d+1}(\mathbf{x}(i)). \quad (27)$$

Algorithm Description

Now we are 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 $\mathbf{X}(1)$ and $\mathbf{X}(2)$. The fitness value $J(\mathbf{X}(d))$ and corresponding members for each subset feature are stored.
4. *Adding one feature.* Find the most significant feature, say $\mathbf{x}(d+1)$, in the set of $\mathbf{Z} - \mathbf{x}(d)$ with respect to $\mathbf{X}(d)$ using (27), and update

$$\mathbf{X}(d+1) = \mathbf{X}(d) + \mathbf{x}(d+1) \quad (28)$$

5. *Conditional deletion.* Using (26) to find the least significant feature, say $\mathbf{x}(d+1)$, in the set of $\mathbf{X}(d+1)$. Update

$$J(\mathbf{X}(d+1)) = J(\mathbf{X}(d)) + \mathbf{S}_{d+1}(\mathbf{x}(d+1)) \quad (29)$$

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

$$\mathbf{X}'(d) = \mathbf{X}(d+1) - \mathbf{x}(m). \quad (30)$$

Update $J(\mathbf{X}(d))$ as

$$J(\mathbf{X}(d)) = J(\mathbf{X}(d+1)) + \mathbf{S}_{d+1}(\mathbf{x}(d+1)) - \mathbf{S}_{d+1}(\mathbf{x}(m)). \quad (31)$$

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

Results

We test our proposed algorithm with four other algorithms on CT colonography (CTC) data sets. Ten-fold crossvalidation was performed using a committee of 7 support vector machines classifier, where each SVM takes 3 features as inputs. This configuration is chosen based on statistical analysis to achieve the best classification performance with the least complexity.

Table 1: Basic Information for CTC Data

Data Name	SM	NTP	NFP	Feature Size
Prone_2D	2D	144	1013	102
Supine_2D	2D	214	1021	102
Prone_3D	3D	148	1022	102
Supine_3D	3D	221	1034	102

SM stands for “segmentation method”. NTP represents “number of truth positive” and NFP denotes “number of false positive” in the data set.

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 (st. dev. 11 years; range 41 to 86 years). All patients had at least one polyp and 27 of them had at least one polyp of mass 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.

The software first segments the colon using a region growing algorithm, after which, regions of interest along the colon wall are identified. A total of 102 different quantitative features are calculated for each polyp candidate based on a 2-D or 3-D segmentation algorithms (Yao & Summers 2004). We have in total four data sets: Supine_2D, Supine_3D, Prone_2D and Prone_3D. Table 1 shows basic information for these data sets.

Feature Selection Algorithms

The algorithms we compared include four wrapper type feature selection methods, using a SVM as classifier. The fitness criterion for these algorithms is defined as the average of sensitivity and specificity of the involved SVM. Sensitivity denote classification accuracy for truth positive instances whereas specificity represents classification accuracy for false positive instances in the data.

Exhaustive Search (ES): This algorithm searches every possible combination of 3 features among all the features. It is optimal in the sense of chosen fitness value but longer search time is required.

Forward Stepwise Search (FSS): First, a group of 3 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.

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. For details about this algorithm please see (Raymer *et al.* 2000).

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

Table 2: Ten Fold Cross Validation Results of 5 Feature Selection Algorithms

Data	ES	FSS	GA	PS	PLOFS+ES
Prone_2D	84.4	84.7	84.4	83.8	85.1
Supine_2D	86.7	87.0	85.8	87.7	86.3
Prone_3D	85.8	85.9	85.5	85.2	85.6
Supine_3D	88.3	88.3	87.5	88.4	87.5

All numbers are the average fitness value (in %) of the ten runs, and each fitness value is the average of sensitivity and specificity of SVMs.

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. N is 3 in the experiment.

The proposed algorithm, PLOFS, is combined with ES to compose the PLOFS-ES algorithm. A total of 25 features are first selected by PLOFS, ES then search all possible combinations of three features among the remaining 25 features using SVMs for evaluating features.

Each of the five algorithms generate a group of combinations of 3 features, we keep the first 1000 such combinations according to its fitness values. The best SVM committee with 7 members is then searched among those 1000 combinations. The committee selection takes the same amount of time (around 20 mins) for each algorithm, and it is excluded from algorithms time-efficiency comparisons.

Evaluation and Discussion

Different algorithm usually selects different feature subset, we evaluate these feature subsets using a ten-fold cross validation method followed by a t test. Time efficiency of each algorithm is also compared. In ten-fold cross validation, we first randomly divide the available data into 10 equal-sized parts. Each of the ten parts is held out as a test set, and the remaining nine tenths are used to train the committee of 7 SVMs. The training and testing procedures are repeated 10 times, and we get 10 testing results for each feature subset. Each testing result is the average of sensitivity and specificity of the trained SVMs. The t test is used to verify if the testing results of SVMs trained by different feature subset are significantly different.

Table 2 shows cross-validation results for all 5 algorithms on the 4 CT data sets. The number of clusters for the PLN model in PLOFS is 4 for all the data sets. It is observed that all fitness values are very similar. A t test showed there was no significant difference between any paired values at a 95%

Table 3: Time Efficiency of the 5 Algorithms

Data	ES	FSS	GA	PS	PLOFS+ES
Prone_2D	90	23	6	35	1(+5sec)
Supine_2D	134	54	7	70	2(+6sec)
Prone_3D	92	28	6	33	1(+5sec)
Supine_3D	144	47	7	75	2(+6sec)

The unit for the numbers is minute. ‘1(+5sec)’ denotes that ES requires 1 minute to search 1000 combinations among 25 features which are selected by PLOFS within 5 seconds.

confidence level. However, Table 3 shows time efficiencies of the algorithms, where significant differences are clearly demonstrated. The most efficient algorithm, PLOFS+ES, only requires about 1 or 2 minutes. The second efficient one, GA, needs 6 or 7 minutes. The other three algorithms take much longer time to complete.

CTC data used in this paper has 102 features which is quit large for many feature selection algorithms. It is very time consuming if we try to search useful features using an exhaustive search. By reducing the dimensionality of the data using the PLOFS algorithm down to 25, we have significantly increased the time efficiency because the search time increases exponentially with the dimensionality. The FSS and PS algorithms improve time efficiency compared to that of ES, however, for a large size data they still may not be feasible. The GA algorithm is quite efficient, and it maybe worthwhile to combine GA with PLOFS to see if they could give an even more efficient hybrid algorithm. Since all the algorithms give statistically similar performances, time efficiency becomes the critical factor.

Decreasing the dimensionality of the feature vector before using a wrapper algorithm for exhaustive searching is very important in practical applications. One of the most popular scheme for dimensionality deduction is principle component analysis. It is very interesting to compare PLOFS to such filter type feature selection algorithms. We are currently doing such comparisons and a journal paper is under preparation.

Conclusions

We proposed a novel approach for feature selection in colonic polyp detection. First, we determined an appropriate PLN model for the given data based on a learning theorem. Then, we applied the floating search algorithm on the PLN model to select a feature subset, through an OLS procedure. We compared the proposed algorithm with 4 other wrapper type algorithms on the same CTC data sets. Our results showed that all 5 algorithms gave statistically similar fitness value, but the proposed algorithm was the most efficient one, which increased time efficiency by three.

References

- Almuallin, H., and Dietterich, T. G. 1991. Learning with many irrelevant features. In *Proc. AAAI-91*, 547–552.
- Billings, S. A., and Voon, W. S. F. 1987. Piecewise linear identification of nonlinear systems. *Int. J. Contr.* 46:215–235.
- Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1983. *CART: Classification and Regression Trees*. CBelmont, California: Wadsworth.
- Chen, S.; Hong, X.; Harris, C. J.; and Sharkey, P. M. 2004. Sparse modelling using orthogonal forward regression with press statistic and regularization. *IEEE Trans. Systems, Man and Cybernetics-Part B: Cybernetics* 34(2):898–911.
- Chen, S.; Billings, S. A.; and Luo, W. 1989. Orthogonal least squares methods and their application to non-linear system identification. *Int. J. Control* 50(5):1873–1896.
- Etcheberria, R.; Inza, I.; Larranaga, P.; and Sierra, B. 2000. Feature subset selection by bayesian network-based optimization. *Artificial Intelligence* 123:157–184.
- Hong, X., and Harris, C. J. 2001. Variable selection algorithm for the construction of mimo operating point dependent neurofuzzy networks. *IEEE Trans. Fuzzy Systems* 9(1):88–101.
- Kira, K., and Rendell, L. A. 1992. The feature selection problem: traditional methods and a new algorithm. In *Proc. AAAI-92*, 122–126.
- Kohavi, R., and John, G. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97(1-2):273–324.
- Kohonen, T. 1989. *Self-Organization and Associative Memory*. Heidelberg: Springer, 3rd edition.
- Li, J.; Manry, M. T.; and Yu, C. Feature selection using a piecewise linear network. *Revised to IEEE Transactions on Neural Networks*.
- Maldonado, F. J., and Manry, M. T. 2002. Optimal pruning of feed-forward neural networks based upon the schmidt procedure. In *the 36th Asilomar Conference on Signals, Systems, & Computers*, 1024–1028.
- Narendra, P. M., and Fukunaga, K. 1977. A branch and bound algorithm for feature subset selection. *IEEE Trans. Comput.* 26:917–922.
- Pudil, P.; Novovičová, J.; and Kittler, J. 1994. Floating search methods in feature selection. *Pattern Recognition Letters* 15:1119–1125.
- Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1:81–106.
- Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann.
- Raymer, M.; Punch, W.; Goodman, E.; Kuhn, L.; and Jain, A. 2000. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation* 4:164–171.
- Ruck, D. W.; Rogers, S. K.; Kabrisky, M.; Oxley, M. E.; and Suter, B. W. 1990. The multilayer-perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks* 1(4):296–298.
- Siedlecki, W., and Sklansky, J. 1988. On automatic feature selection. *Int. J. of Pattern Recognition and Artif. Intell.* 2(2):197–220.
- Stearns, S. D. 1976. On selecting features for pattern classifiers. *Third Internat. Conf. on Pattern Recognition* 71–75.
- Summers, R. M.; Jerebko, A.; Franaszek, M.; Malley, J.; and Johnson, C. 2002. Colonic polyps: Complementary role of computer-aided detection in ct colonography. *Radiology* 225:391–399.
- Vapnik, V. N. 1998. *Statistical Learning Theory*. New York: Wiley.
- Yao, J., and Summers, R. 2004. 3d colonic polyp segmentation using dynamic deformable surfaces. In *SPIE Medical Imaging*.