IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
*IEEJ Trans* 2011; **6**: 173–179
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/tee.20641

**Paper**

# Feature Subset Selection: A Correlation-Based SVM Filter Approach

Boyang Li, Non-member
Qiangwei Wang, Non-member
Jinglu Hu[a], Member

The central criterion of feature selection is that good feature sets contain features that are highly correlated with the output, yet uncorrelated with each other. Based on this criterion, we address the problem of feature selection through correlation-based feature clustering and support vector machine (SVM) based feature ranking. Correlation-based clustering is proposed to group features into some clusters based on the correlation between two features. As a result, a feature is highly correlated to any other feature in the same cluster but uncorrelated to the features in other clusters. From each cluster, we select a feature as the delegate based on its influence quantities on the output. The influence quantities are measured by the feature sensitivity in the SVM. The proposed approach can identify relevant features and eliminate redundancy among them effectively. The effectiveness of the proposed approach is demonstrated through comparisons with other methods using real-world data with different dimensions. 

**Keywords:** feature selection, correlation-based clustering, support vector machine, feature ranking

*Received 16 June 2009; Revised 27 October 2009*

## 1. Introduction

In machine learning, it is important to build a robust learning model for high-dimensional data. One of the main tasks is dimension reduction, which can be divided into feature selection and feature extraction [1]. Feature selection tries to find a subset of the original variables. Feature extraction tries to map the multidimensional space into a space of fewer dimensions. Feature selection has been proven to be faster and more suitable for data with some redundant features [2]. Especially in some real-world applications, feature selection is effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. Feature selection algorithms typically fall into two categories: feature ranking and subset selection. Feature ranking ranks the features by a metric and eliminates all features that do not achieve an adequate threshold [3]. Subset selection searches the set of possible features for the optimal subset [4].

Subset selection algorithms can be mainly divided into wrappers and filters [5]. Wrappers apply an unsupervised learning algorithm to each subset of features and then evaluate the subset of features by criterion functions, such as evolutionary algorithms [6]. The disadvantages of wrappers are the high computational cost and the risk of overfitting, since they must repeatedly call the induction algorithm along with a statistical resampling technique and must be rerun when a different induction algorithm is used [7].

Filters are similar to wrappers in the search approach, but instead of evaluating against a model, a simpler filter is evaluated. In recent years, data have become increasingly larger in number of features in many applications such as genome projects, text categorization, image retrieval, and customer relationship management [8]. As discussed frequently in the literature, filters are ultimately more feasible than wrappers in these cases [9]. Therefore, in this paper, we adopt a filter model to work on the feature selection problems.

[a] Correspondence to: Jinglu Hu. E-mail: jinglu@waseda.jp

Graduate School of Information, Production and Systems, Waseda University, Hibikino 2-7, Wakamatsu-ku, Kitakyushu-shi, Fukuoka-ken, Japan

Some existing feature-ranking algorithms, evolutionary algorithms, and some other evaluation measures have been shown to be effective in feature selection on training and testing data, for example, sequential forward search (SFS), plus and take away (PTA), gradient-based feature selection (GFS), genetic algorithms (GAs), Chaotic BPSO with K-NN (CBPSO-KNN), and PSO-SVM [7, 10–15]. However, most of them only score subsets of features according to their predictive power but ignore the correlation analysis of features. As a result of ignoring the correlation analysis, these methods cannot preserve the diversity of features, and thus they cannot be generalized well for actual problems.

In Hall *et al*. and Yu *et al*.'s papers, correlation measures are applied to evaluate the goodness of feature subsets [5,9,16]. A hypothesis is also given as a rule to select good features: a good feature subset is one that contains features highly correlated to the class, yet uncorrelated to each other [5,9]. Although the correlation analysis is applied in their approaches, the classification performances are not good enough. That is because their approach does not adapt any classifier, and their predictive power cannot be measured.

To overcome the problems of these algorithms and meet the demand for feature selection for high-dimensional data, we develop a hybrid approach which contains a supervised learning process and is designed on the basis of the correlation analysis. The supervised learning is implemented as a ranking feature and eliminates features with low influence quantities on the output. The correlation analysis is implemented as a correlation-based clustering to divide the features into some clusters, and selects the feature that is ranked first in its cluster.

In some existing feature selection methods, neural network (NN) based feature ranking has been used [17,18] and proven effective. However, it does not stand mass noise and easily falls into a local optimum. In order to obtain better performances, we choose support vector machines (SVMs) to develop an improved feature-ranking approach. As an algorithm with outstanding performance, SVMs are not negatively affected by high dimensionality and can overcome overfitting. In the proposed SVM feature ranking, a feature sensitivity is defined and calculated on the basis of the

Lagrange multipliers, and SVs are calculated from the training process. Features are ranked by their sensitivities. A threshold is set to eliminate features with very low sensitivities. In this approach, the rank of the features reflects the influence quantities of the features of the output.

After feature ranking, we use a clustering method to divide and select the features. Many existing methods can be used to do feature clustering. Here, we select a new method called affinity propagation [19], which takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a high-quality set of exemplars and the corresponding clusters gradually emerge. Affinity propagation can find clusters with much lower error than other methods, and it is also faster. In this clustering method, we define two kinds of correlation measures: *linear correlation coefficient* and *information gain*. Linear correlation coefficient is suitable for most problems, but it is not able to capture correlations that are not linear in nature. Therefore we introduced information gain as another correlation measure for some real-world data. In each cluster, we select the feature that is ranked first.

As a combination of SVM-based feature ranking and correlation-based clustering, the proposed approach can preserve the diversity of selected features and improve the adaptability of the classifiers. In comprehensive experiments, we obtained better performances by the proposed approach on several real-world problems.

The remainder of this paper is organized as follows: the next section provides the idea and the introduction of the proposed approach. SVM-based feature ranking and correlation-based clustering are also described in detail. In Section 3, we evaluate the effectiveness of the proposed approach via experiments on various real-world datasets, and discuss the implications of the findings. In Section 4, we conclude our work with some possible extensions.

## 2. Correlation-Based Feature Selection

**2.1. Structure of the proposed approach** As mentioned in the previous section, correlation analysis is important in feature selection. If we adopt the correlation between two features as a criterion, the purpose of feature selection becomes selecting a feature subset in which each feature is highly correlated to the output but uncorrelated to any other features. In other words, if a feature has a high enough influence quantity on the output to make it predictive and is uncorrelated to any other relevant selected features, it will be regarded as a suitable feature to be selected for the classification task. In this sense, the problem of feature selection boils down to finding a suitable measure of correlations between features and a selection procedure to select features from the relevant features.

In order to solve this problem, we proposed a two-stage modular model. The proposed model consists of two main modules: the SVM-based feature-ranking module and the correlation-based clustering module.

SVM-based feature ranking is proposed to rank the features based on their sensitivities on the output. The features with very low sensitivities are eliminated. The rank of features is also used after the correlation-based clustering to select features from clusters.

Correlation-based clustering is used to split the feature space into some fragments. In other words, it is used to divide the features of the data into some clusters based on the correlation of features. In each cluster, a feature has high correlation with the other features, but low correlation with the features in the other clusters. So the features from the same cluster have similar characteristics. Therefore, we select a feature to delegate its cluster based on the rank of features. The flowchart of the proposed model is shown in Fig. 1.

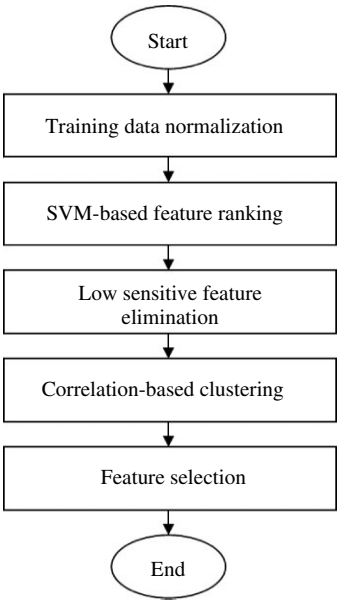


Fig. 1. Flowchart of the proposed approach

**2.2. SVM-based feature ranking** In order to eliminate the features with low influence quantities on the output and give a criterion of feature selection after the clustering, we proposed a feature-ranking approach based on SVM. That is because SVM performs well in most real-world problems.

The basic idea of SVM is to find the maximum margin, which is defined as the distance between bounding planes of different classes. The margin maximization process is equivalent to finding the optimal separation boundary [20]. In SVM, the input vectors are firstly mapped into a high-dimensional feature space, and then the separation boundary is calculated by solving a quadratic programming (QP) problem in its dual form. Because the SVM implements the structural risk minimization principle, it can overcome overfitting and obtain a global optimal solution.

*2.2.1. Sensitivity in SVM* Assume that we have a binary input dataset denoted as $\{X_i, y_i\}$, where $X_i \in R^n$, $i = 1, 2, \ldots, N$. $X_i$ is the $i$th input vector and $y_i$ is its class label ($+1$ or $-1$). The input dataset is divided into two different classes $A$ and $B$ which have labels $+1$ and $-1$, respectively. There are two bounding planes for these two classes. The distance between them is the margin. Maximizing this margin could improve the capability of the classifier model generally [20]. If the dataset is nonlinear and nonseparable, we should attempt to simultaneously minimize separation error and maximize the margin [21,22].

In SVM, SVs from $A$ are defined as those $A_i$ in the halfspace $\{X \in R^n | w^{\mathrm{T}} X \leq b + 1\}$ and SVs from $B$ are the points $B_i$ in the halfspace $\{X \in R^n | w^{\mathrm{T}} X \geq b - 1\}$, where $w$ and $b$ are weights and bias. These points are the only solutions that are relevant to the calculation of the optimal separation boundary.

To find the separation boundary in a nonlinear dataset, we transform input data from a low-dimensional space to a high-dimensional feature space using nonlinear mapping function $\varphi(x)$, and then construct the bounding plane function (1):

$$y_i[w^{\mathrm{T}}\varphi(X_i) + b] \geq 1 - \xi_i, \ \forall i \tag{1}$$

The optimum separation boundary problem becomes an optimization problem (2):

$$\min_{w,b,\xi} J(w, \xi) = \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i=1}^{N} \xi_i \tag{2}$$

$$s.t.\begin{cases} y_i[w^{\mathrm{T}}\varphi(X_i)+b] \geq 1-\xi_i \\ \xi_i \geq 0, i=1,\ldots,N \end{cases}$$

where parameter $C$ is used to control the degree of tolerance, which is the only changeable parameter in the SVM.

For nonlinear problems, by introducing a vector of Lagrange multipliers $\alpha = (\alpha_1,\ldots,\alpha_N)$, the problem (2) is rebuilt as a QP problem in dual space [23]:

$$\max_{\alpha} Q(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{N} y_i y_j K(X_i,X_j)\alpha_i\alpha_j + \sum_{j=1}^{N}\alpha_j \tag{3}$$

$$s.t.\begin{cases} \sum_{i=1}^{N}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C,\ \forall i \end{cases}$$

where $K(X_i,X_j) = \varphi(X_i)^{\mathrm{T}}\varphi(X_j)$ is the kernel function [24]. After training, we obtain the vector of Lagrange multipliers $\alpha$. If $\alpha_i \neq 0$, then sample-$i$ is an SV.

The decision value is defined as follows:

$$f(X) = \sum_{i=1}^{N}\alpha_i y_i K(X,X_i) + b \tag{4}$$

where $X_i$ are samples from training data and $X$ is the input vector. Only SVs have nonzero Lagrange multipliers, so using them to determine the separation boundary is sufficient. The structure of an SVM can be drawn as a network shown in Fig. 2.

In order to rank features, we introduce the definition of sensitivity into SVM. In general, the sensitivity of function $y$ to variable $x$ is defined as follows [25]:

$$\eta(x|y) = \left|\frac{\partial y}{\partial x}\right| \tag{5}$$

In SVM, we can consider that $f(X)$ is a mapping function of $X$. Then the sensitivity can be used as a measure of the feature's ability to influence the decision value $f(X)$. $X$ is the set of features, so the sensitivity of SVM to $X$ is defined by the formula

$$\eta(X|f(X)) = \left|\frac{\partial f(X)}{\partial X}\right| = |\nabla_X f(X)| \tag{6}$$

According to the definition of decision value (4), the sensitivity is rewritten as follows:

$$\eta(X|f(X)) = \left|\sum_{i=1}^{N}\alpha_i y_i \nabla_X K(X,X_i)\right| \tag{7}$$

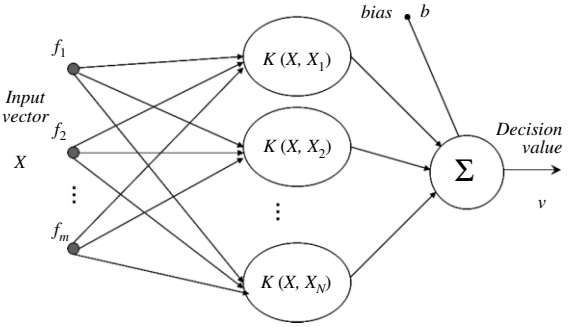


Fig. 2. Structure of SVM

Table I. Kernels and their gradients

| Kernel | $K(X,X_i)$ | $\nabla_X K(X,X_i)$ |
|---|---|---|
| Linear | $X\cdot X_i$ | $X_i$ |
| Poly | $(1+X\cdot X_i)^c$ | $c(1+X\cdot X_i)^{c-1}X_i$ |
| RBF | $\exp\left(\frac{-\|X_i-X\|^2}{2\sigma^2}\right)$ | $\frac{(X_i-X)}{\sigma^2}\exp\left(\frac{-\|X_i-X\|^2}{2\sigma^2}\right)$ |
| Sigmoid | $\tanh(b(X\cdot X_i)-c)$ | $\frac{b}{\cosh^2(X\cdot X_i-c)}X_i$ |

*2.2.2. Sensitivity-based feature ranking* In the feature-ranking process, the sensitivity of each feature is used as a criterion to rank and select features.

In SVMs, many kernels can be selected to calculate the feature sensitivity. As shown in Table I, we can select a linear kernel, ploy kernel, radial basis function (RBF) kernel, or sigmoid kernel to calculate the feature sensitivity. Their gradients are also shown in Table I. Therefore, we can easily calculate the feature sensitivity of each feature with these kernels.

As a practical example, here we select a linear function as the kernel of the SVM to explain the feature-ranking process. The linear kernel is defined as

$$K(X,X_i) = X\cdot X_i \tag{8}$$

Assume that $X = [f_1,f_2,\ldots,f_m]$ and $X_i = [d_{i1},d_{i2},\ldots,d_{im}]$, and the kernel function is rewritten as follows:

$$K(X,X_i) = X\cdot X_i = \sum_{j=1}^{m} f_j d_{ij} \tag{9}$$

Using the linear kernel, the formula of the decision value is rewritten as

$$\begin{aligned} f(X) &= \sum_{i=1}^{N}\sum_{j=1}^{m}\alpha_i y_i f_j d_{ij} + b \\ &= \sum_{j=1}^{m}\left(\sum_{i=1}^{N}\alpha_i y_i d_{ij}\right)f_j + b \end{aligned} \tag{10}$$

Based on (7) and (10), we can calculate the feature sensitivity of each feature:

$$\eta_j = \left|\sum_{i=1}^{N}\alpha_i y_i d_{ij}\right|, \qquad j = 1,2,\ldots,m \tag{11}$$

Using a linear kernel SVM, the feature sensitivity of each feature is calculated by the Lagrange multipliers $\alpha_i$, the instances $X_i$ from the training data, and their corresponding labels $y_i$, where $i = 1,\ldots,N$. The feature sensitivities are used to rank features. The feature that has a higher feature sensitivity value is arranged at a higher position in the sequence.

On the other hand, we can redraw the structure of SVM as Fig. 3 based on (10). The figure shows that the decision value can be written as a linear combination of the features. Each feature has a weight. The weight of a feature is just its feature sensitivity. So we can also consider that the criterion in linear kernel SVM is the absolute value of each feature's weight.

As in the linear kernel case, we can also calculate feature sensitivities based on other kernels. Because it is very hard to select appropriate types of kernel functions for a given problem in SVMs [26], we use both linear and RBF kernels to calculate feature sensitivities in the experiments.

In order to make feature ranking feasible, we normalize the features before SVM training. Using the proposed approach, we

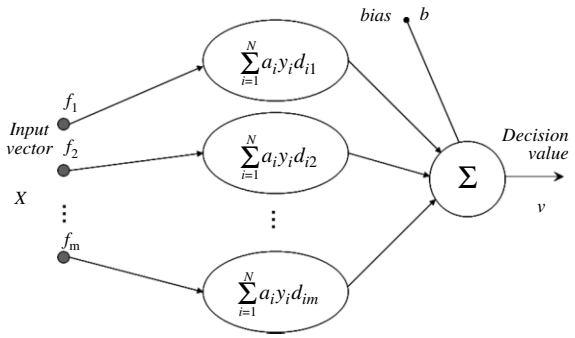


Fig. 3. Structure of an SVM with linear kernel

rank the features based on their sensitivities. As in some other feature ranking methods, we also define a threshold $\varepsilon$ to eliminate the features with low sensitivities. In the simulations, we normalize the sensitivity values to the range $[0, 1]$. Because we only want to eliminate the features that have very low predictive power, $\varepsilon$ could be given any very small positive value. In the proposed approach, the classification results are not highly sensitive to the threshold $\varepsilon$. As an example, we set $\varepsilon = 0.03$ in our simulations.

**2.3. Correlation-based clustering** In order to measure the correlations between reserved features after feature ranking, we select a new method called affinity propagation, which was proposed by Frey and Dueck [19]. Here, all features are simultaneously considered as potential exemplars. By viewing each feature as a node in a network, this method recursively transmits real-valued messages along the edges of the network until a good set of exemplars and corresponding clusters emerge. Messages are updated on the basis of simple formulas that search for minima of an appropriately chosen energy function. In this method, we should define a function as the criterion to measure the correlation between each two features.

There exist broadly two approaches to measure the correlation between two features. One is based on classical linear correlation and the other is based on information theory. Under the first approach, the most well-known measure is the linear correlation coefficient [27]. For a pair of variables $(X, Y)$, the correlation coefficient $r$ is given by the formula

$$r = \mathcal{C}(X, Y) = \frac{\sum_i (x_i - \overline{x}_i)(y_i - \overline{y}_i)}{\sqrt{\sum_i (x_i - \overline{x}_i)^2}\sqrt{\sum_i (y_i - \overline{y}_i)^2}} \tag{12}$$

where $\overline{x_i}$ is the mean of $x_i$, and $\overline{y_i}$ is the mean of $y_i$. The correlation coefficient is a symmetrical measure for two variables. The value of $r$ lies between $-1$ and 1, inclusive. If $X$ and $Y$ are completely correlated, $r$ takes the value of 1 or $-1$; if $X$ and $Y$ are totally independent, $r$ is zero.

Although linear correlation coefficient is applicable in most cases, it is not suitable for some real-world data. That is because it is unable to capture correlations that are not linear in nature. In order to overcome this problem, in our approach we also adopt a criterion called information gain. In information theory, the expected value of the information gain is the mutual information for a pair of variables $(X, Y)$. It denotes the reduction in the entropy of $X$ achieved by learning the state of the random variable $Y$. In other words, information gain reflects the correlation between $X$ and $Y$. The calculation of information gain is based on the information-theoretical concept of entropy. The entropy of a variable $X$ is defined as

$$H(X) = -\sum_i P(x_i) \log_2(P(x_i)) \tag{13}$$

and the entropy of X after observing values of $Y$ is defined as

$$H(X|Y) = -\sum_i \sum_j P(y_j) P(x_i|y_j) \log_2(P(x_i|y_j)) \tag{14}$$

where $P(y_j)$ is the prior probabilities of $Y$, and $P(x_i|y_j)$ is the posterior probabilities of $X$, given the values of $Y$. Information gain is given as

$$IG(X|Y) = H(X) - H(X|Y) \tag{15}$$

From this equation, it is clear that information gain is the amount by which the entropy of $X$ decreases, which reflects additional information about $X$ provided by $Y$. Thus, a feature $Y$ is regarded more correlated to feature $X$ than to feature $Z$ if $IG(X|Y) > IG(Z|Y)$.

By using the linear correlation coefficient and information gain as the correlation measurement function, affinity propagation is implemented to find feature clusters in the training data. In affinity propagation, a preference parameter $p$ can be changed to increase or decrease the number of identified clusters. As in other feature selection approaches, the identification of the number of clusters is still a problem.

A common choice parameter $p$ is the median of similarity values $p' = \text{median}(s)$, where $s$ is the vector of similarity values. In our simulations, we tuned the preference parameter $p$ by a parameter selection methodology. The training data is fist divided into five partitions. Then the procedure of parameter $p$ optimization follows Rätsch's methodology [28], which trains the algorithm with each candidate parameter by a fivefold cross-validation procedure on the five training partitions of the training data and selects the model parameters to be the median over those five estimates. In our simulations, the candidate parameters are set to $\{\frac{p'}{3}, \frac{p'}{2}, p', 2p',$ and $3p'\}$. The computational cost of this way of estimating the parameter is a little high, so we will commit ourselves to solving this problem in our future work. However, this method will make our comparison more robust and the results more reliable.

In each cluster produced by the correlation-based clustering, a feature will be selected if it is ranked first. As a result of the proposed approach, the selected features are not highly correlated with each other, but have the highest influence quantity on the output in their clusters.

## 3. Simulations and Results

In our simulations, we evaluate the proposed approach in terms of the number of selected features, classification accuracy, and the most important feature subset on several real-world data.

In this section, the proposed correlation-based SVM filter that uses the linear correlation coefficient as the correlation measure is abbreviated to *CSF-r*. And the proposed approach that uses the information gain as the correlation measure is abbreviated to *CSF-IG*. By using linear kernel and RBF kernel in SVM-based feature ranking, we construct four feature selection models: CSF-r (Lin), CSF-IG (Lin), CSF-r (RBF), and CSF-IG (RBF).

In order to demonstrate the effectiveness of the proposed approach, we employ some commonly used approaches (SFS, PTA), GAs (sequential genetic algorithm (SGA), hybrid genetic algorithm (HGA)), some novel evolutionary algorithms (CBPSO-KNN, PSO-SVM), and a feature-ranking method (gradient-based feature selection (GFS)) from the literature in the comparison experiments. [7,10–15].

**3.1. Ionosphere dataset** We first test our proposed approach on the Ionosphere dataset from the UCI database [29]. This radar data was collected by a system in Goose Bay, Labrador.

Table II. Format of ionosphere dataset

| Dataset | Instances | Classes | Features |
|---|---|---|---|
| Ionosphere | 351 (201/150) | 2 | 34 |

*Note: x/y*: Number of test samples/Number of train samples.

Table III. Clustering results (CSF-r)

| Clusters | Features |
|---|---|
| I | 1 |
| II | 3, 5 |
| III | 4 |
| IV | 8, 10, 12, 14, 16 |
| V | 6, 7, 9, 11, 13, 15, 17, 19, 21, 23 |
| VI | 18, 20, 22, 26, 30 |
| VII | 24 |
| VIII | 25, 27, 29, 31, 33 |
| IX | 28, 32, 34 |

Table IV. Clustering results (CSF-IG)

| Clusters | Features |
|---|---|
| I | 1, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 29, 31, 33, 34 |
| II | 11, 17 |
| III | 13, 15, 21 |
| IV | 16 |
| V | 18, 25, 27, 30 |
| VI | 20 |
| VII | 19, 22, 23 |
| VIII | 24 |
| IX | 26 |
| X | 28 |
| XI | 32 |

The data format was arranged as shown in Table II. The size of training data is 150 and the size of testing data is 201.

This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kW. The targets were free electrons in the ionosphere. 'Good' radar returns are those showing evidence of some type of structure in the ionosphere. 'Bad' returns are those that do not; their signals pass through the ionosphere. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by two attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.

Using the proposed approach, 34 features are first ranked by the SVM-based feature ranking (in simulations, we use two common kernels: linear kernel and RBF kernel). Some features with low sensitivities are eliminated. The reserved features are divided into some clusters by correlation-based clustering (in our simulations, we use two kinds of correlation measures: linear correlation coefficient and information gain). The number of clusters is identified by a preference parameter $p$ of affinity propagation. The value $p$ is determined by Rätsch's parameter selection methodology. And then, based on the rank, we select the features from the clusters.

The clustering results of CSF-r and CSF-IG are shown in Tables III and IV, respectively. Features selected by the proposed four models (CSF-r (Lin), CSF-IG (Lin), CSF-r (RBF), and CSF-IG (RBF)) are shown in Table V. The number of selected features of CSF-r and CSF-IG are 9 and 11. In some papers, the second feature is commonly omitted in the experiments, because it is zero for all observations and thus does not contribute to the variance. In our simulations, the sensitivity of the second feature is the lowest. In feature ranking, because its sensitivity is smaller than the threshold, it is eliminated.

We can find that the feature subset $\{3, 22, 24\}$ is selected by all four feature selection models. So in our simulations, we consider this subset as the most important feature subset. That means the features in this subset contains the most important information of the dataset. Of course, we can not only use this subset to solve classification problems, but some other selected significant features will also be used. Using the selected feature subset, an SVM classifier is trained to classify the test data. Table VI compares the best experimental results obtained by other approaches with the proposed models. It is clear that the proposed approach can obtain a higher accuracy than the other approaches on the ionosphere dataset.

**3.2. Other real-world datasets** We also used some other datasets from the UCI Repository in our simulations [29]. These datasets are all collected from the real world. A summary of the datasets is presented in Table VII. We use tenfold cross validation to evaluate the proposed approach. Table VIII compares the best experimental results obtained by other approaches with the proposed approach. The features selected by our proposed approach for each dataset are shown in Table IX.

**3.3. Results and discussion** From Tables VI and VIII, we can find that the proposed approach obtained the highest

Table V. Selected features

| Approaches | $d^*$ | Selected features |
|---|---|---|
| CSF-r (Lin) | 9 | 1, **3**, 4, 7, 8, **22**, **24**, 27, 34 |
| CSF-IG (Lin) | 11 | **3**, 11, 16, 20, 21, **22**, **24**, 26, 27, 28, 32 |
| CSF-r (RBF) | 9 | 1, **3**, 4, 7, 8, **22**, **24**, 29, 34 |
| CSF-IG (RBF) | 11 | **3**, 11, 13, 16, 20, **22**, **24**, 26, 27, 28, 32 |

*Note:* Features selected by four models (CSF-r (Lin), CSF-IG (Lin), CSF-r (RBF), CSF-IG (RBF)) are in bold type.

Table VI. Accuracy of classification for ionosphere dataset

| Approaches | $d^*$ | accuracy(%) |
|---|---|---|
| SFS | 7 | 93.45 |
| PTA | 7 | 93.45 |
| SGA | 7 | 95.44 |
| HGA | 7 | 95.73 |
| CBPSO-KNN | 15 | 97.33 |
| PSO-SVM | 15 | 97.33 |
| GFS | 12 | 97.33 |
| CSF-r (Lin) | 9 | **98.67** |
| CSF-IG (Lin) | 11 | 98.00 |
| CSF-r (RBF) | 9 | 97.33 |
| CSF-IG (RBF) | 11 | 97.33 |

*Note:* The best result is in Bold type.

Table VII. Summary of classification problems

| Datasets | Instances | Classes | Features |
|---|---|---|---|
| Vowel | 990 | 11 | 10 |
| Wine | 178 | 3 | 13 |
| WDBC | 569 | 2 | 30 |
| Sonar | 208 | 2 | 60 |

Table VIII. Accuracy of classification for test datasets

| | Vowel | | Wine | | WDBC | | Sonar | |
|---|---|---|---|---|---|---|---|---|
| | $d^*$ | Accuracy (%) | $d^*$ | Accuracy (%) | $d^*$ | Accuracy (%) | $d^*$ | Accuracy (%) |
| SFS | 8 | **99.70** | 8 | 95.51 | 18 | 94.02 | 48 | 91.82 |
| PTA | 8 | **99.70** | 8 | 95.51 | 18 | 94.20 | 48 | 92.31 |
| SGA | 8 | **99.70** | 5 | 95.51 | 12 | 94.38 | 24 | 95.67 |
| HGA | 8 | 99.70 | 5 | 95.51 | 6 | 94.90 | 24 | **97.12** |
| CBPSO-KNN | 6 | 97.88 | 8 | 99.44 | 8 | 97.54 | 27 | 93.27 |
| PSO-SVM | 7 | 99.49 | 8 | **100** | 13 | 95.61 | 34 | 96.15 |
| GFS | 5 | 99.49 | 3 | **100** | 7 | **98.24** | 39 | 93.27 |
| CSF-r (Lin) | 5 | 99.49 | **2** | **100** | 4 | **98.24** | **9** | 96.15 |
| CSF-IG (Lin) | **4** | 99.49 | **2** | **100** | **2** | **98.24** | 28 | 95.67 |
| CSF-r (RBF) | 6 | 99.49 | **2** | 95.51 | 4 | **98.24** | 11 | 96.15 |
| CSF-IG (RBF) | **4** | **99.70** | **2** | **100** | **2** | **98.24** | 43 | 92.31 |

*Note:* Better results are in Bold type.

Table IX. Selected features for test datasets

| Datasets | Approaches | $d^*$ | Selected features |
|---|---|---|---|
| Vowel | CSF-r (Lin) | 5 | **1**, **2**, 3, 6, **9** |
| | CSF-IG (Lin) | 4 | **1**, **2**, 7, **9** |
| | CSF-r (RBF) | 6 | **1**, **2**, 5, 6, **9**, 10 |
| | CSF-IG (RBF) | 4 | **1**, **2**, 5, **9** |
| Wine | CSF-r (Lin) | 2 | 7, **13** |
| | CSF-IG (Lin) | 2 | 4, **13** |
| | CSF-r (RBF) | 2 | 7, **13** |
| | CSF-IG (RBF) | 2 | 5, **13** |
| WDBC | CSF-r (Lin) | 4 | 2, 17, **24**, 27 |
| | CSF-IG (Lin) | 2 | 4, **24** |
| | CSF-r (RBF) | 4 | 17, 22, **24**, 27 |
| | CSF-IG (RBF) | 2 | 4, **24** |
| Sonar | CSF-r (Lin) | 9 | 12, 18, 22, 27, 30, 31, 35, 44, 47 |
| | CSF-IG (Lin) | 28 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 28, 40, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51 |
| | CSF-r (RBF) | 11 | 4, 8, 11, 14, 21, 23, 28, 34, 36, 43, 45 |
| | CSF-IG (RBF) | 43 | 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 |

*Note:* Features selected by four models (CSF-r (Lin), CSF-IG (Lin), CSF-r (RBF), CSF-IG (RBF)) are in Bold type.

classification accuracy for the Ionosphere, Vowel, Wine, and WDBC classification problems.

The classification accuracies of the Ionosphere dataset obtained by CSF-r (Lin) and CSF-IG (Lin) are 98.67 and 98.00%, respectively, an increase of 1.3 and 0.7% compared to the other existing approaches. The results of CSF-r (RBF) and CSF-IG (RBF) are 97.33%, which equals the accuracy obtained by CBPSO-KNN, PSO-SVM, and GFS but is better than other approaches. In the Vowel classification problem, only CSF-IG (RBF) gives the best result. Although the classification accuracies obtained by CSF-r (Lin), CSF-IG (Lin), and CSF-r (RBF) are worse than those of some other approaches, they are still comparable. For the Wine and WDBC classification problems, the proposed models all obtained good performances. In addition, the number of selected features by using the proposed approach is much smaller than in the other approaches. This means that only a few features are needed in these classification problems. For the Sonar classification problem, though the classification accuracy obtained by the proposed approach is worse than that of some other approaches, it is also still comparable.

From Table IX, we can also select the most important feature subsets of these problems. The selection of the most important feature subset may be useful for some other data analysis problems. The most important feature subsets of Vowel, Wine, and WDBC are $\{1, 2, 9\}$, $\{13\}$, and $\{24\}$.

These results indicate that, for different classification problems, the proposed approach can serve as a preprocessing step and optimize the feature selection process. The features selected by the proposed approach can lead to an increase in classification accuracy effectively.

## 4. Conclusions and Future Works

In this paper, we proposed a feature selection approach based on correlation analysis and SVM. The affinity propagation clustering approach is used to measure the correlation between features and group the features with high correlations into clusters. In the clustering process, two correlation measures (linear correlation coefficient and information gain) are defined as the criteria for correlation. For different datasets, the two correlation measures can give different performances. Based on the structure of SVM and the definition of sensitivity, we proposed a feature-ranking approach to rank features and select the significant features from the clusters built by correlation-based clustering.

The proposed approach is a general model for many problems, because it considers both correlation of features and their influence quantities on the output. The proposed approach was tested on some real-world datasets. Simulation results show that our method simplifies features more effectively and obtains a higher classification accuracy compared to the other feature selection methods.

In our future work, we will improve the identification of the number of clusters. And some more effective correlation analysis algorithms will be introduced to our proposed approach.

Developing some more suitable sensitivity algorithms for different kernels is also the focus in our future research.

## References


(1) Gorban A, Kegl B, Wunsch D, Zinovyev A. *Principal Manifolds for Data Visualisation and Dimension Reduction*. Lecture Notes in Computational Science and Engineering, Vol. 58, Springer: Berlin; 2007.

(2) Huber R, Dutra LV. Feature selection for ERS-1/2 InSAR classification: high dimensionality case. Proceedings of the International Geoscience and Remote Sensing Symposium, 1998; 1605–1607.

(3) Jong K, Mary J, Cornujeols A, Marchiori E, Sebag M. Ensemble feature ranking. *Lecture Notes in Computer Science* 2004; **3202**:267–278.

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

(5) Yu L, Liu H. Feature selection for high-dimensional data: a fast correlation-based filter solution. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003; 856–863.

(6) Liu H, Dougherty ER, Dy JG, Torkkola K, Tuv E, Peng H, Ding C, Long F, Berens M, Parsons L. Evolving feature selection. *IEEE Intelligent Systems* 2005; **20**(**6**):64–76.

(7) Chuang LY, Li JC, Yang CH. Chaotic binary particle swarm optimization for feature selection using logistic map. Proceedings of the International MultiConference of Engineers and Computer Scientists 2008, Volume I, IMECS, 2008.

(8) Blum A, Langley P. Selection of relevant features and examples in machine learning. *Artificial Intelligence* 1997; **97**:245–271.

(9) Hall MA, Smith LA. Feature subset selection: a correlation based filter approach. International Conference on Neural Information Processing and Intelligent Information Systems 1997, Berlin, 1997; 855858.

(10) Raymer ML, Punch WF, Goodman ED, Kuhn LA, Jain AK. Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation* 2000; **4**(**2**):164–171.

(11) Narendra PM, Fukunage K. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers* Volume 1977; **6**(**9**):917–922.

(12) Pudil P, Novovicova J, Kittler J. Floating search methods in feature selection. *Pattern Recognition Letters* 1994; **15**:1119–1125.

(13) Roberto B. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks* 1994; **5**(**4**):537–550.

(14) Tu CJ, Chuang LY, Chang JY, Yang CH. Feature Selection using PSO-SVM. *IAENG International Journal of Computer Science* 2007; **33**(**1**):IJCS–33118.

(15) Rakotomamonjy A. Variable selection using SVM-based criteria. *Journal of Machine Learning Research* 2003; **3**:1357–1370.

(16) Hall MA. Correlation-based feature selection for discrete and numeric class machine learning. Proceedings of the Seventeenth International Conference on Machine Learning, 2000; 359–366.

(17) Gallinari P. Gallinari P, Feature selection with neural networks. *Behaviormetrika* 1999; **26**:6–16.

(18) Ruck DW, Rogers SK, Kabrisky M. Feature selection using a multilayer perceptron. *Neural Network Computing* 1990; **2**:40–48.

(19) Frey BJ, Dueck D. Clustering by passing messages between data points. *Science* 2007; **315**:972–976.

(20) Chapelle O, Vapnik V. In *Model selection for Support Vector Machines*. Adv. Neural Inf. Proc. Syst., vol. 12, Solla S, Leen T, Muler K-R (eds). MIT Press: Cambridge, MA; 2000.

(21) Gunn SR. Support vector machines for classification and regression. Technical Report, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, 10 May, 1998.

(22) Osuna E, Freund R, Girosi F. Support vector machines: training and applications. A.I. Memo 1602, MIT A.I.Lab., 1997.

(23) Suykens J. Least squares support vector machines, Tutorial IJCNN, 2003.

(24) Smola AJ, Schökopf B. On a kernelbased method for pattern recognition, regression, approximation and operator inversion. Algorithmica, **22**:211–231, 1998. Technical Report 1064, GMD FIRST, April 1997.

(25) Shen KQ, Ong CJ, Li XP, Smith EW. Feature selection via sensitivity analysis of SVM probabilistic outputs. *Machine Learning* 2008; **70**(**1**):1–20.

(26) Li Huaqing, Wang Shaoyu, Qi Feihu. SVM model selection with the VC bound. *Lecture Notes in Computer Science* 2005; **3314/2005**:1067–1071.

(27) Cohen J, Cohen P, West SG, Aiken LS. *Applied Multiple Regression/Correlation Analysis For The Behavioral Sciences*. 3rd ed. Lawrence Erlbaum Associates: Hillsdale, NJ; 2003.

(28) Rätsch G, Onoda T, Müller KR. Soft margins for adaboost. *Machine Learning* 2001; **42**(**3**):287–319.

(29) Murphy PM, Aha DW. UCI repository of machine learning databases. Technical report, Department of Information and Computer Science, University of California, Irvine, California. http://archive.ics.uci.edu/ml/ [Last accessed June 2009].



**Boyang Li** (Non-member) received the B.S. degree in electrical engineering from Dalian University of Technology, China, in 2004. From 2004 to 2005, he studied in Dalian University of Technology as a mater's student. From 2005 to 2006, he studied in the Graduate School of Information, Production and Systems, Waseda University, Japan. He received the M.S. degree from Waseda University in 2006 and another M.S. degree from Dalian University of Technology in 2007. Since September 2006, he has been a doctoral student in the Graduate School of Information, Production and Systems, Waseda University. His current research interests include neural networks, machine learning, and kernel algorithm and their applications.

**Qiangwei Wang** (Non-member) received the B.B.A. degree in human resources management from Renmin University, China, in 2005. From 2005 to 2007, she worked in China International Telecommunication Construction Corporation. In April 2007, she joined the Graduate School of Information, Production and Systems, Waseda University. She received the M.S. degree from Waseda University in 2009. Since 2009, she has been a doctoral student in the Graduate School of Information, Production and Systems, Waseda University. Her current research interests include neural networks, machine learning, and intelligence algorithm and their applications.

**Jinglu Hu** (Member) received the M.S. degree in electronic engineering from Zhongshan University, China, in 1986, and the Ph.D. degree in computer science and engineering from Kyushu Institute of Technology, Japan, in 1997. From 1986 to 1993, he worked in Zhongshan University, where he was a Research Associate and then a Lecturer. From 1997 to 2003, he worked as a Research Associate at Kyushu University. From 2003 to 2008, he was an Associate Professor and, since April 2008, he has been a Professor at the Graduate School of Information, Production and Systems, Waseda University. His research interests include computational intelligence, neural networks and genetic algorithms and their applications to system modeling and identification, bioinformatics, and time series prediction. He is a member of the IEEE, IEEJ, SICE, and IEICE.