Expert Systems with Applications 31 (2006) 231–240

Expert Systems with Applications

www.elsevier.com/locate/eswa

A GA-based feature selection and parameters optimization for support vector machines

Cheng-Lung Huang [a,*], Chieh-Jen Wang [b]

[a] *Department of Information Management, National Kaohsiung First University of Science and Technology, 2, Juoyue Rd., Nantz District, Kaohsiung 811, Taiwan, ROC*

[b] *Department of Information Management, Huafan University, 1, Huafan Rd., Shihtin Hsiang, Taipei Hsien 223, Taiwan, ROC*

Abstract

Support Vector Machines, one of the new techniques for pattern classification, have been widely used in many application areas. The kernel parameters setting for SVM in a training process impacts on the classification accuracy. Feature selection is another factor that impacts classification accuracy. The objective of this research is to simultaneously optimize the parameters and feature subset without degrading the SVM classification accuracy. We present a genetic algorithm approach for feature selection and parameters optimization to solve this kind of problem.

We tried several real-world datasets using the proposed GA-based approach and the Grid algorithm, a traditional method of performing parameters searching. Compared with the Grid algorithm, our proposed GA-based approach significantly improves the classification accuracy and has fewer input features for support vector machines.

Keywords: Support vector machines; Classification; Feature selection; Genetic algorithm; Data mining

1. Introduction

Support vector machines (SVM) were first suggested by Vapnik (1995) and have recently been used in a range of problems including pattern recognition (Pontil and Verri, 1998), bioinformatics (Yu, Ostrouchov, Geist, & Samatova, 1999), and text categorization (Joachims, 1998). SVM classifies data with different class labels by determining a set of support vectors that are members of the set of training inputs that outline a hyperplane in the feature space. SVM provides a generic mechanism that fits the hyperplane surface to the training data using a kernel function. The user may select a kernel function (e.g. linear, polynomial, or sigmoid) for the SVM during the training process that selects support vectors along the surface of this function.

When using SVM, two problems are confronted: how to choose the optimal input feature subset for SVM, and how to set the best kernel parameters. These two problems are crucial, because the feature subset choice influences the appropriate kernel parameters and vice versa (Fröhlich and Chapelle, 2003). Therefore, obtaining the optimal feature subset and SVM parameters must occur simultaneously.

Many practical pattern classification tasks require learning an appropriate classification function that assigns a given input pattern, typically represented by a vector of attribute values to a finite set of classes. Feature selection is used to identify a powerfully predictive subset of fields within a database and reduce the number of fields presented to the mining process. By extracting as much information as possible from a given data set while using the smallest number of features, we can save significant computation time and build models that generalize better for unseen data points. According to Yang and Honavar (1998), the choice of features used to represent patterns that are presented to a classifier affects several pattern classification aspects, including the accuracy of the learned classification algorithm, the time needed for learning a classification function, the number of examples needed for learning, and the cost associated with the features.

In addition to the feature selection, proper parameters setting can improve the SVM classification accuracy. The parameters that should be optimized include penalty parameter C and the kernel function parameters such as the gamma (γ) for the radial basis function (RBF) kernel. To design a SVM, one must choose a kernel function, set the kernel parameters and determine a soft margin constant

* Corresponding author. Tel.: +1 2483 702 831; fax: +1 2483 704 275. *E-mail address:* clhuang@ccms.nkfust.edu.tw (C.-L. Huang).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.09.024

C (penalty parameter). The Grid algorithm is an alternative to finding the best C and gamma when using the RBF kernel function. However, this method is time consuming and does not perform well (Hsu and Lin, 2002; LaValle and Branicky, 2002). Moreover, the Grid algorithm cannot perform the feature selection task.

Genetic algorithms have the potential to generate both the optimal feature subset and SVM parameters at the same time. Our research objective is to optimize the parameters and feature subset simultaneously, without degrading the SVM classification accuracy. The proposed method performs feature selection and parameters setting in an evolutionary way. Based on whether or not feature selection is performed independently of the learning algorithm that constructs the classifier, feature subset selection algorithms can be classified into two categories: the filter approach and the wrapper approach (John, Kohavi, & Peger, 1994; Kohavi and John, 1997). The wrapper approach to feature subset selection is used in this paper because of the accuracy.

In the literature, only a few algorithms have been proposed for SVM feature selection (Bradley, Mangasarian, & Street, 1998; Bradley and Mangasarian, 1998; Weston et al., 2001; Guyon, Weston, Barnhill, & Bapnik, 2002; Mao, 2004). Some other GA-based feature selection methods were proposed (Raymer, Punch, Goodman, Kuhn, & Jain, 2000; Yang and Honavar, 1998; Salcedo-Sanz, Prado-Cumplido, Pérez-Cruz, & Bousoño-Calzón, 2002). However, these papers focused on feature selection and did not deal with parameters optimization for the SVM classifier. Fröhlich and Chapelle (2003) proposed a GA-based feature selection approach that used the theoretical bounds on the generalization error for SVMs. The SVM regularization parameter can also be optimized using GAs in Fröhlich's paper.

This paper is organized as follows: a brief introduction to the SVM is given in Section 2. Section 3 describes basic GA concepts. Section 4 describes the GA-based feature selection and parameter optimization. Section 5 presents the experimental results from using the proposed method to classify several real world datasets. Section 6 summarizes the results and draws a general conclusion.

2. Brief introduction of support vector machines

2.1. The optimal hyperplane (linear SVM)

In this section we will briefly describe the basic SVM concepts for typical two-class classification problems. These concepts can also be found in (Kecman, 2001; Schőlkopf and Smola, 2000; Cristianini and Shawe-Taylor, 2000). Given a training set of instance-label pairs (x_i, y_i), $i=1, 2,\ldots, m$ where $x_i \in R^n$ and $y_i \in \{+1, -1\}$, for the linearly separable case, the data points will be correctly classified by

$$\langle w \cdot x_i \rangle + b \geq +1 \quad \text{for } y_i = +1 \tag{1}$$

$$\langle w \cdot x_i \rangle + b \leq -1 \quad \text{for } y_i = -1 \tag{2}$$

Eqs. (1) and (2) can be combined into one set of inequalities.

$$y_i(\langle w \cdot x_i \rangle + b) - 1 \geq 0 \quad \forall\, i = 1, \ldots, m \tag{3}$$

The SVM finds an optimal separating hyperplane with the maximum margin by solving the following optimization problem:

$$\min_{w,b} \tfrac{1}{2} w^{\mathrm{T}} w \quad \text{subject to}: \ y_i(\langle w \cdot x_i \rangle + b) - 1 \geq 0 \tag{4}$$

It is known that to solve this quadratic optimization problem one must find the saddle point of the Lagrange function:

$$L_p(w, b, \alpha) = \frac{1}{2} w^{\mathrm{T}} \cdot w - \sum_{i=1}^{m} (\alpha_i y_i (\langle w \cdot x_i \rangle + b) - 1) \tag{5}$$

where the α_i denotes Lagrange multipliers, hence $\alpha_i \geq 0$. The search for an optimal saddle point is necessary because the L_p must be minimized with respect to the primal variables w and b and maximized with respect to the non-negative dual variable α_i. By differentiating with respect to w and b, the following equations are obtained:

$$\frac{\partial}{\partial w} L_p = 0, \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{6}$$

$$\frac{\partial}{\partial b} L_p = 0, \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{7}$$

The Karush Kuhn–Tucker (KTT) conditions for the optimum constrained function are necessary and sufficient for a maximum of Eq. (5). The corresponding KKT complementarity conditions are

$$\alpha_i [y_i (\langle w \cdot x_i \rangle + b) - 1] = 0 \quad \forall\, i \tag{8}$$

Substitute Eqs. (6) and (7) into Eq. (5), then L_P is transformed to the dual Lagrangian $L_D(\alpha)$:

$$\max_{\alpha} L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle$$
$$\text{subject to}: \ \alpha_i \geq 0 \ \ i = 1, \ldots, m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{9}$$

To find the optimal hyperplane, a dual Lagrangian $L_D(\alpha)$ must be maximized with respect to non-negative α_i. This is a standard quadratic optimization problem that can be solved by using some standard optimization programs. The solution α_i for the dual optimization problem determines the parameters w^* and b^* of the optimal hyperplane. Thus, we obtain an optimal decision hyperplane $f(x,\alpha^*,b^*)$ (Eq. (10)) and an indicator decision function sign $[f(x,\alpha^*,b^*)]$.

$$f(x, \alpha^*, b^*) = \sum_{i=1}^{m} y_i \alpha_i^* \langle x_i, x \rangle + b^* = \sum_{i \in sv} y_i \alpha_i^* \langle x_i, x \rangle + b^* \tag{10}$$

In a typical classification task, only a small subset of the Lagrange multipliers α_i usually tend to be greater than zero. Geometrically, these vectors are the closest to the optimal hyperplane. The respective training vectors having nonzero α_i are called support vectors, as the optimal decision hyperplane $f(x,\alpha^*,b^*)$ depends on them exclusively.

2.2. The optimal hyperplane for nonseparable data (linear generalized SVM)

The above concepts can also be extended to the non-separable case, i.e. when Eq. (3) there is no solution. The goal is to construct a hyperplane that makes the smallest number of errors. To get a formal setting of this problem we introduce the non-negative slack variables $\xi_i \geq 0$, $i=1,\ldots, m$. Such that

$$\langle w\cdot x_i\rangle + b \geq +1-\xi_i \quad \text{for } y_i = +1 \tag{11}$$

$$\langle w\cdot x_i\rangle + b \leq -1+\xi_i \quad \text{for } y_i = -1 \tag{12}$$

In terms of these slack variables, the problem of finding the hyperplane that provides the minimum number of training errors, i.e. to keep the constraint violation as small as possible, has the formal expression:

$$\underset{w,b,\xi}{\text{Min}} \frac{1}{2} w^{\mathrm{T}} w + C\sum_{i=1}^{m} \xi_i$$
$$\text{subject to}: \; y_i(\langle w\cdot x_i\rangle + b) + \xi_i - 1 \geq 0, \quad \xi_i \geq 0 \tag{13}$$

This optimization model can be solved using the Lagrangian method, which is almost equivalent to the method for solving the optimization problem in the separable case. One must maximize the same dual variables Lagrangian $L_D(\alpha)$ (Eq. (14)) as in the separable case.

$$\underset{\alpha}{\text{Max}}\, L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i\alpha_j y_i y_j \langle x_i\cdot x_j\rangle$$
$$\text{subject to}: \; 0\leq \alpha_i \leq C, \;\; i=1,\ldots,m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{14}$$

To find the optimal hyperplane, a dual Lagrangian $L_D(\alpha)$ must be maximized with respect to non-negative α_i under the constrains $\sum \alpha_i y_i = 0$ and $0\leq\alpha_i\leq C$, $i=1,\ldots,m$. The penalty parameter C, which is now the upper bound on α_i, is determined by the user. Finally, the optimal decision hyperplane is the same as Eq. (10).

2.3. Non-linear SVM

The nonlinear SVM maps the training samples from the input space into a higher-dimensional feature space via a mapping function Φ, which are also called kernel function. In the dual Lagrange (9), the inner products are replaced by the kernel function (15), and the non-linear SVM dual Lagrangian $L_D(\alpha)$ (Eq. (16)) is similar with that in the linear generalized case.

$$(\Phi(x_i)\cdot\Phi(x_j)) := k(x_i, x_j) \tag{15}$$

$$L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i\alpha_j y_i y_j k(x_i\cdot x_j)$$
$$\text{subject to}: \; 0\leq \alpha_i \leq C, \;\; i=1,\ldots,m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{16}$$

This optimization model can be solved using the method for solving the optimization in the separable case. Therefore, the optimal hyperplane has the form Eq. (17). Depending upon the applied kernel, the bias b can be implicitly part of the kernel function. Therefore, if a bias term can be accommodated within the kernel function, the nonlinear SV classifier can be shown as Eq. (18).

$$f(x,\alpha^*,b^*) = \sum_{i=1}^{m} y_i\alpha_i^*\langle\Phi(x_i),\Phi(x)\rangle + b^* = \sum_{i=1}^{m} y_i\alpha_i^* k(x_i,x) + b^* \tag{17}$$

$$f(x,\alpha^*,b^*) = \sum_{i\in sv} y_i\alpha_i^*\langle\Phi(x_i),\Phi(x)\rangle = \sum_{i\in sv} y_i\alpha_i^* k(x_i,x) \tag{18}$$

Some kernel functions include polynomial, radial basis function (RBF) and sigmoid kernel (Burges, 1998), which are shown as functions (19), (20), and (21). In order to improve classification accuracy, these kernel parameters in the kernel functions should be properly set.

Polynomial kernel:

$$k(x_i,x_j) = (1 + x_i\cdot x_j)^d \tag{19}$$

Radial basis function kernel:

$$k(x_i,x_j) = \exp(-\gamma\|x_i - x_j\|^2) \tag{20}$$

Sigmoid kernel:

$$k(x_i,x_j) = \tanh(kx_i\cdot x_j - \delta) \tag{21}$$

3. Genetic algorithm

Genetic algorithms (GA), a general adaptive optimization search methodology based on a direct analogy to Darwinian natural selection and genetics in biological systems, is a promising alternative to conventional heuristic methods. GA work with a set of candidate solutions called a population. Based on the Darwinian principle of 'survival of the fittest', the GA obtains the optimal solution after a series of iterative computations. GA generates successive populations of alternate solutions that are represented by a chromosome, i.e. a solution to the problem, until acceptable results are obtained. Associated with the characteristics of exploitation and exploration search, GA can deal with large search spaces efficiently, and hence has less chance to get local optimal solution than other algorithms.

A fitness function assesses the quality of a solution in the evaluation step. The crossover and mutation functions are

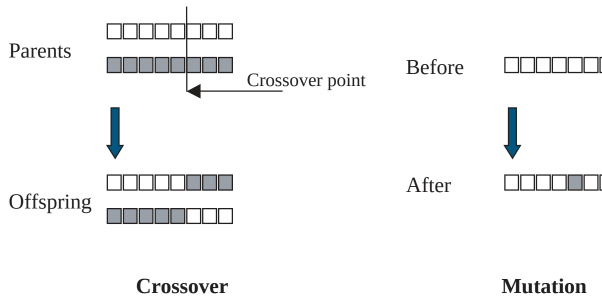

Fig. 1. Genetic crossover and mutation operation.

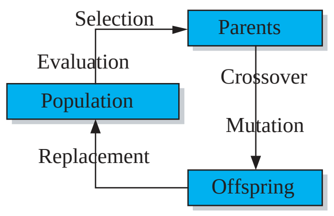

Fig. 2. Evolutionary cycle.

the main operators that randomly impact the fitness value. Chromosomes are selected for reproduction by evaluating the fitness value. The fitter chromosomes have higher probability to be selected into the recombination pool using the roulette wheel or the tournament selection methods.

Fig. 1 illustrates the genetic operators of crossover and mutation. Crossover, the critical genetic operator that allows new solution regions in the search space to be explored, is a random mechanism for exchanging genes between two chromosomes using the one point crossover, two point crossover, or homologue crossover. In mutation the genes may occasionally be altered, i.e. in binary code genes changing genes code from 0 to 1 or vice versa.

Offspring replaces the old population using the elitism or diversity replacement strategy and forms a new population in the next generation.

The evolutionary process operates many generations until termination condition satisfy. Fig. 2 depicts the GA evolutionary process mentioned above (Goldberg, 1989; Davis, 1991).

4. GA-based feature selection and parameters optimization

The chromosome design, fitness function, and system architecture for the proposed GA-based feature selection and parameter optimization are described as follows.

4.1. Chromosome design

To implement our proposed approach, this research used the RBF kernel function for the SVM classifier because the RBF kernel function can analysis higher-dimensional data and requires that only two parameters, C and γ be defined (Hsu, Chang, & Lin, 2003; Lin and Lin, 2003). When the RBF kernel is selected, the parameters (C and γ) and features used as input attributes must be optimized using our proposed GA-based

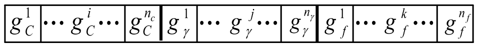
$g_C^1 \cdots g_C^i \cdots g_C^{n_c}$
$g_\gamma^1 \cdots g_\gamma^j \cdots g_\gamma^{n_\gamma}$
$g_f^1 \cdots g_f^k \cdots g_f^{n_f}$

Fig. 3. The chromosome comprises three parts, C, γ, and the features mask.

system. Therefore, the chromosome comprises three parts, C, γ, and the features mask. However, these chromosomes have different parameters when other types of kernel functions are selected. The binary coding system was used to represent the chromosome. Fig. 3 shows the binary chromosome representation of our design. In Fig. 3, $g_C^1 \sim g_C^{n_c}$ represents the value of parameter C, $g_\gamma^1 \sim g_\gamma^{n_\gamma}$ represents the parameter value γ, and $g_f^1 \sim g_f^{n_f}$ represents the feature mask. n_c is the number of bits representing parameter C, n_r is the number of bits representing parameter γ, and n_f is the number of bits representing the features. Note that we can choose n_c and n_γ according to the calculation precision required, and that n_γ equals the number of features varying from the different datasets.

In Fig. 3, the bit strings representing the genotype of parameter C and γ should be transformed into phenotype by Eq. (22). Note that the precision of representing parameter depends on the length of the bit string (n_c and n_r); and the minimum and maximum value of the parameter is determined by the user. For chromosome representing the feature mask, the bit with value '1' represents the feature is selected, and '0' indicates feature is not selected.

$$p = \min_p + \frac{\max_p - \min_p}{2^l - 1} \times d \tag{22}$$

P	phenotype of bit string
$\min_p$	minimum value of the parameter
$\max_p$	maximum value of the parameter
d	decimal value of bit string
l	length of bit string

4.2. Fitness function

Classification accuracy, the number of selected features, and the feature cost are the three criteria used to design a fitness function. Thus, for the individual (chromosome) with high classification accuracy, a small number of features, and low total feature cost produce a high fitness value. We solve the multiple criteria problem by creating a single objective fitness function that combines the three goals into one. As defined by formula (23), the fitness has two predefined weights: (i) W_A for the classification accuracy; (ii) W_F for the summation of the selected feature (with nonzero F_i) multiplying its cost. The weight accuracy can be adjusted to 100% if accuracy is the most important. Generally, W_A can be set from 75 to 100% according to user's requirements. Each feature has different feature cost in the dataset from the UCI. If we do not have the feature cost information, the cost C_i can be set to the same value, e.g. '1' or another number. The chromosome with high fitness value has high probability to be preserved to the next generation, so user should appropriately define these settings according to his requirements.

$$\text{fitness} = W_A \times \text{SVM\_accuracy} + W_F \times \left(\sum_{i=1}^{n_f} C_i \times F_i \right)^{-1} \quad (23)$$

W_A	SVM classification accuracy weight
SVM_accuracy	SVM classification accuracy
W_F	weight for the number of features
C_i	cost of feature *i*
F_i	'1' represents that feature *i* is selected; '0' represents that feature *i* is not selected

4.3. System architectures for the proposed GA-based approach

To precisely establish a GA-based feature selection and parameter optimization system, the following main steps (as shown in Fig. 4) must be proceeded. The detailed explanation is as follows:

(1) *Data preprocess: scaling*. The main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation (Hsu et al., 2003). Feature value scaling can help to increase SVM accuracy according to our experimental results. Generally, each feature can be linearly scaled to the range [−1, +1] or [0, 1] by formula (24), where v is original value, v' is scaled value, $\max_a$ is upper bound of the feature value, and $\min_a$ is low bound of the feature value.

$$v' = \frac{v - \min_a}{\max_a - \min_a} \quad (24)$$

(2) *Converting genotype to phenotype*. This step will convert each parameter and feature chromosome from its genotype into a phenotype.

(3) *Feature subset*. After the genetic operation and converting each feature subset chromosome from the genotype into the phenotype, a feature subset can be determined.

(4) *Fitness evaluation*. For each chromosome representing C, γ and selected features, training dataset is used to train the SVM classifier, while the testing dataset is used to calculate classification accuracy. When the classification accuracy is obtained, each chromosome is evaluated by fitness function— formula (23).

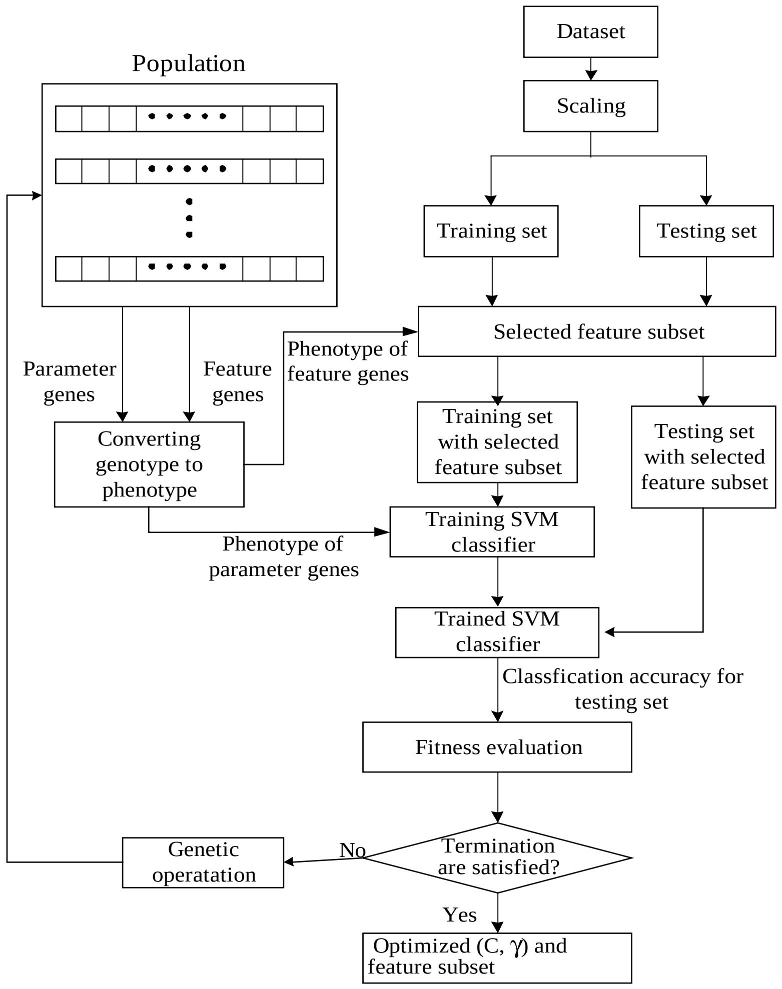
Population
Dataset
Scaling
Training set
Testing set
Selected feature subset
Parameter genes
Feature genes
Phenotype of feature genes
Converting genotype to phenotype
Training set with selected feature subset
Testing set with selected feature subset
Training SVM classifier
Phenotype of parameter genes
Trained SVM classifier
Classfication accuracy for testing set
Fitness evaluation
Genetic operatation
No
Termination are satisfied?
Yes
Optimized (C, γ) and feature subset

Fig. 4. System architectures of the proposed GA-based feature selection and parameters optimization for support vector machines.

Table 1
Datasets from the UCI repository

No.	Names	No. of classes	No. of instances	Nominal features	Numeric features	Total features
1	German (credit card)	2	1000	0	24	24
2	Australian (credit card)	2	690	6	8	14
3	Pima-Indian diabetes	2	760	0	8	8
4	Heart disease (Statlog Project)	2	270	7	6	13
5	Breast cancer(Wisconsin)	2	699	0	10	10
6	Contraceptive Method Choice	3	1473	7	2	9
7	Ionosphere	2	351	0	34	34
8	Iris	3	150	0	4	4
9	Sonar	2	208	0	60	60
10	Statlog project: vehicle	4	940	0	18	18
11	Vowel	11	990	3	10	13

(5) *Termination criteria*. When the termination criteria are satisfied, the process ends; otherwise, we proceed with the next generation.
(6) *Genetic operation*. In this step, the system searches for better solutions by genetic operations, including selection, crossover, mutation, and replacement.

5. Numerical illustrations

5.1. Experiment descriptions

To evaluate the classification accuracy of the proposed system in different classification tasks, we tried several real-world datasets from the UCI database (Hettich, Blake, & Merz, 1998). These data sets have been frequently used as benchmarks to compare the performance of different classification methods in the literature. These datasets consist of numeric and nominal attributes. Table 1 summarizes the number of numeric attributes, number of nominal attributes, number of classes, and number of instances for these datasets.

To guarantee valid results for making predictions regarding new data, the data set was further randomly partitioned into training sets and independent test sets via a k-fold cross validation. Each of the k subsets acted as an independent holdout test set for the model trained with the remaining $k-1$ subsets. The advantages of cross validation are that all of the test sets were independent and the reliability of the results could be improved. The data set is divided into k subsets for cross validation. A typical experiment uses $k=10$. Other values may be used according to the data set size. For a small data set, it may be better to set larger k, because this leaves more examples in the training set (Salzberg, 1997). This study used $k=10$, meaning that all of the data will be divided into 10 parts, each of which will take turns at being the testing data set. The other nine data parts serve as the training data set for adjusting the model prediction parameters.

Our implementation was carried out on the Matlab 6.5 development environment by extending the Libsvm which is originally designed by Chang and Lin (2001). The empirical evaluation was performed on Intel Pentium IV CPU running at 1.6 GHz and 256 MB RAM.

The Grid search algorithm is a common method for searching for the best C and γ. Fig. 5 shows the process of Grid algorithm combined with SVM classifier. In the Grid algorithm, pairs of (C, γ) are tried and the one with the best cross-validation accuracy is chosen. After identifying a 'better' region on the grid, a finer grid search on that region can be conducted (Hsu et al., 2003).

This research conducted the experiments using the proposed GA-based approach and the Grid algorithm. The results from

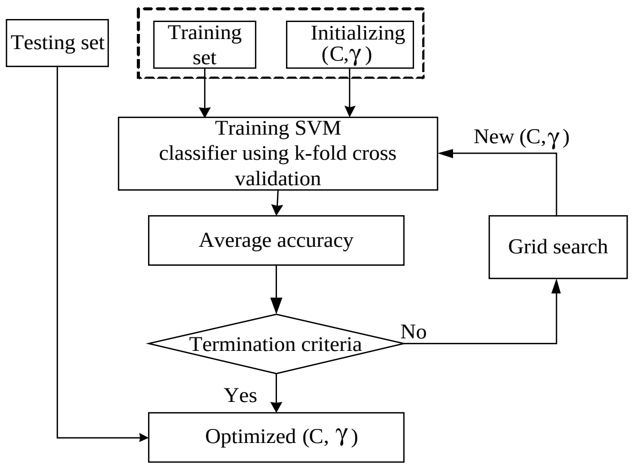

Fig. 5. Parameters setting using Grid algorithm.

Table 2
A 2×2 contingency table

		Target (or disease)	
		+	−
Predicted (or Test)	+	True Positive (TP)	False Positive (FP)
	−	False Negative (FN)	True Negative (TN)

the proposed method were compared with that from the Grid algorithm. In all of the experiments 10-fold cross validation was used to estimate the accuracy of each learned classifier. Some empirical results are reported in the following sections.

5.2. Accuracy calculation

Accuracy using the binary target datasets can be demonstrated by the positive hit rate (sensitivity), the negative hit rate (specificity), and the overall hit rate. For the multiple class datasets, the accuracy is demonstrated only by the average hit rate. A two by two table with the classification results on the left side and the target status on top is as shown in Table 2. Some cases with the 'positive' class (with disease) correctly classified as positive (TP = True Positive fraction), however, some cases with the 'positive' class will be classified negative (FN = False Negative fraction). Conversely, some cases with the 'negative' class (without the disease) will be correctly classified as negative (TN = True Negative fraction), while some cases with the 'negative' class will be classified as positive (FP = False Positive fraction). TP and FP are the two important evaluation performances for classifiers (Woods and Bowyer, 1997). Sensitivity and specificity describe how well the classifier discriminates between case with positive and with negative class (with and without disease).

Sensitivity is the proportion of cases with positive class that are classified as positive (true positive rate, expressed as a percentage). In probability notation for sensitivity: $P(T^+|D^+) = TP/(TP + FN)$. *Specificity* is the proportion of cases with the negative class, classified as negative (true negative rate, expressed as a percentage). In probability notation: $P(T^-|D^-) = TN/(TN + FP)$. The overall hit rate is the overall accuracy which is calculated by $(TP + TN)/(TN + FP + FN + FP)$.

The *SVM_accuracy* of the fitness in function (23) is measured by sensitivity × specificity for the datasets with two classes (positive or negative), and by the overall hit rate for the datasets with multiple classes.

5.3. Experimental results and comparison

The detail parameter setting for genetic algorithm is as the following: population size 500, crossover rate 0.7, mutation rate 0.02, two-point crossover, roulette wheel selection, and elitism replacement. We set $n_c = 20$ and $n_r = 20$; the value of n_f depends on the experimental datasets stated in Section 5.1. According to the fitness function of Eq. (23), W_A and W_F can influence the experiment result. The higher W_A is; the higher classification accuracy is. The higher W_F is; the smaller the number of features is. We can compromise between weight W_A and W_F. Taking the German and Australia datasets, for example, as shown in Figs. 6 and 7, the accuracy (measured by overall hit rate) is high with large numbers of features when high W_A and low W_F are defined. We defined $W_A = 0.8$ and $W_F = 0.2$ for all experiments. The user can choose different weight values; however, the results could be different.

The termination criteria are that the generation number reaches generation 600 or that the fitness value does not improve during the last 100 generations. The best chromosome is obtained when the termination criteria satisfy. Taking the German dataset, for example, the positive hit rate, negative hit rate, overall hit rate, number of selected features, and the best pairs of (C, γ) for each fold using GA-based approach and Grid algorithm are shown in Table 3. For GA-based approach, its average positive hit rate is 89.6%, average negative hit rate is 76.6%, average overall hit rate is 85.6%, and average number of features is 13. For Grid algorithm, its average positive hit rate is 88.8%, average negative hit rate is 46.2%, average overall hit rate is 76.0%, and all 24 features are used. Note that the weight W_A and W_F are 0.8 and 0.2 in all of our experiments. Table 4 shows the summary results for the positive hit rate, negative hit rate, overall hit rate, number of selected features, and running time for the 11 UCI datasets using the two approaches. In Table 4, the accuracy and average number of features are illustrated with the form of 'average ± standard deviation.' The GA-based approach generated small feature subsets while Grid algorithm uses all of the features.

To compare the overall hit rate of the proposed GA-based approach with the Grid algorithm, we used the nonparametric Wilconxon signed rank test for all of the datasets. As shown in Table 4, the p-values for diabetes, breast cancer, and vehicle are larger than the prescribed statistical significance level of 0.05, but other p-values are smaller than the significance level of 0.05. Generally, compared with the Grid algorithm, the

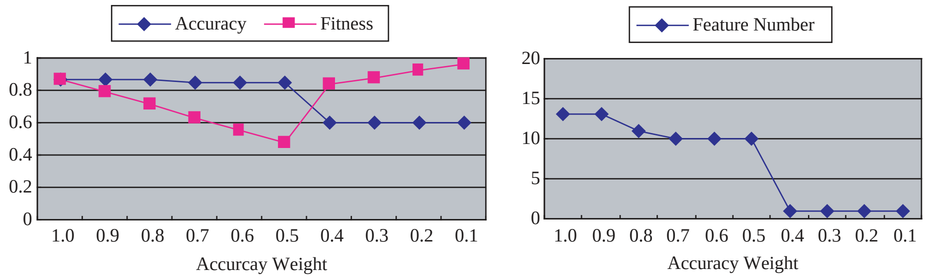

Fig. 6. Illustration of the classification accuracy versus the accuracy weight, W_A, for fold #4 of German dataset.

Table 3
Experimental results for German dataset using GA-based approach and Grid algorithm

Fold #	GA-based approach						Grid algorithm				
	Positive hit rate	Negative hit rate	Overall hit rate%	Optimized C	Optimized γ	Selected features	Positive hit rate	Negative hit rate	Overall hit rate%	Optimized C	Optimized γ
1	0.863	0.7407	83	170.9992278	0.00491136	16	0.8493151	0.55555556	77	2048	0.000488
2	0.8592	0.7931	84	9.29520704	0.5189297	13	0.8591549	0.44827586	74	8	0.007813
3	0.9028	0.75	86	171.9769019	0.05343727	11	0.875	0.5	77	512	0.001953
4	0.8919	0.8462	88	60.24731324	0.05299713	11	0.9324324	0.46153846	81	8192	0.000122
5	0.9091	0.7059	84	174.4961022	0.02207236	13	0.9545455	0.32352941	74	512	0.000122
6	0.8904	0.7778	86	20.99907683	0.04938617	15	0.890411	0.48148148	78	128	0.000488
7	0.9041	0.7778	87	219.8157576	0.03912631	12	0.9452055	0.44444444	81	32768	0.000035
8	0.9155	0.7931	88	95.16782536	0.12330259	11	0.8309859	0.44827586	72	8	0.007813
9	0.871	0.7632	83	255.9134212	0.27580662	15	0.8225806	0.44736842	68	2048	0.000122
10	0.9538	0.7143	87	174.313561	0.01230963	13	0.9230769	0.51428571	78	512	0.000122
Average	0.89608	0.76621	85.6			13	0.8882708	0.46247552	76		

Table 4
Experimental results summary of GA-based approach and Grid algorithm on the test sets

Names	GA-based approach					Grid algorithm			p-value for Wilcoxon testing
	Number of original features	Number of selected features	Average positive hit rate	Average negative hit rate	Average overall hit rate%	Average positive hit rate	Average negative hit rate	Average overall hit rate%	
German	24	13±1.83	0.89608	0.76621	85.6±1.96	0.888271	0.462476	76±4.06	0.005*
Australian	14	3±2.45	0.8472	0.92182	88.1±2.25	0.885714	0.823529	84.7±4.74	0.028*
Diabetes	8	3.7±0.95	0.78346	0.87035	81.5±7.13	0.592593	0.88	77.3±3.03	0.139
Heart disease	13	5.4±1.85	0.94467	0.95108	94.8±3.32	0.75	0.909091	83.7±6.34	0.005*
breast cancer	10	1±0	0.9878	0.8996	96.19±1.24	0.98	0.944444	95.3±2.28	0.435
Contraceptive	9	5.4±0.53	N/A	N/A	71.22±4.15	N/A	N/A	53.53±2.43	0.005*
Ionosphere	34	6±0	0.9963	0.9876	98.56±2.03	0.94	0.9	89.44±3.58	0.005*
Iris	4	1±0	N/A	N/A	100±0	N/A	N/A	97.37±3.46	0.046*
Sonar	60	15±1.1	0.9863	0.9842	98±3.5	0.65555	0.9	87±4.22	0.004*
Vehicle	18	9.2±1.4	N/A	N/A	84.06±3.54	N/A	N/A	83.33±2.74	0.944
Vowel	13	7.8±1	N/A	N/A	99.3±0.82	N/A	N/A	95.95±2.91	0.02*

proposed GA-based approach has good accuracy performance with fewer features.

The ability of a classifier to discriminate between ‘positive’ cases (+) and ‘negative’ cases (−) is evaluated using Receiver Operating Characteristic (ROC) curve analysis. ROC curves can also be used to compare the diagnostic performance of two or more diagnostic classifiers (DeLeo and Rosenfeld, 2001). For every possible cut-off point or criterion value we select to discriminate between the two populations (with positive or negative class value) there will generate a pair of sensitivity and specificity. An ROC curve shows the trade-off between sensitivity and specificity, and demonstrates that the closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the classifier. The area under the curve (AUC) is the evaluation criteria for the classifier.

Taking fold #4 of German dataset, for example, ROC curve of GA-based approach and Grid algorithm are shown in Fig. 8, where AUC is 0.85, 0.82, respectively. For the ROC curve for the other nine folds can also be plotted in the same manner. In short, the average AUC for the 10 folds of testing dataset of German dataset are 0.8424 for

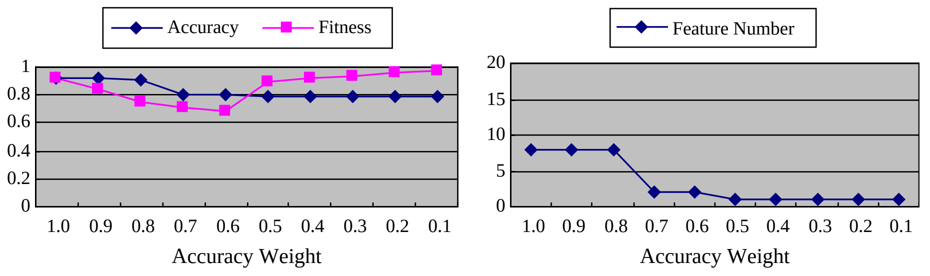

Fig. 7. Illustration of the classification accuracy versus the accuracy weight, W_A, for fold #3 of Australia dataset.

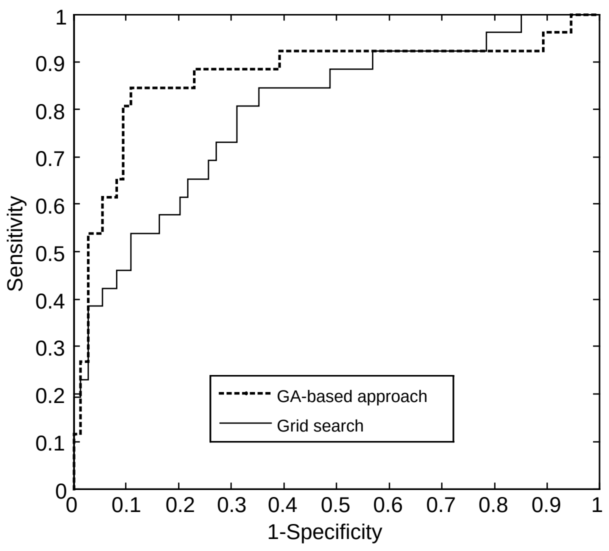

Fig. 8. ROC curve for fold #4 of German Credit Dataset.

Table 5
Average AUC for datasets with two classes

	GA-based approach	Grid algorithm
German	0.8424	0.7886
Australian	0.9019	0.8729
Diabetes	0.8298	0.7647
Heart disease	0.9458	0.8331
breast cancer	0.9423	0.9078
Contraceptive	0.7701	0.6078
Ionosphere	0.9661	0.8709
Iris	0.9756	0.9572
Sonar	0.9522	0.8898
Vehicle	0.8587	0.8311
Vowel	0.9513	0.9205

GA-based approach and 0.7886 for Grid algorithm. We summarize the average AUC for other datasets with two classes in Table 5. The average AUC shows that GA-based approach outperforms the Grid algorithm.

The average running time for GA-based approach is slightly inferior to that of the Grid algorithm; however, the software environment for the two approaches and the predefined searching precision of the Grid algorithm affect the running time. The Grid algorithm is performed under the Python, while the proposed GA-based approach is implemented by using the Matlab in our research. Generally, compared with other systems, the running time is much longer when using the Matlab. Although the proposed approach performed under the Matlab did not outperform the Grid algorithm, it significantly improves the classification accuracy and has fewer input features for support vector machines.

6. Conclusion

SVM parameters and feature subsets were optimized simultaneously in this work because the selected feature subset has an influence on the appropriate kernel parameters and vice versa. We proposed a GA-based strategy to select the feature subset and set the parameters for SVM classification. As far as we know, previous researches did not perform simultaneous feature selection and parameters optimization for support vector machines.

We conducted experiments to evaluate the classification accuracy of the proposed GA-based approach with RBF kernel and the Grid algorithm on 11 real-world datasets from UCI database. Generally, compared with the Grid algorithm, the proposed GA-based approach has good accuracy performance with fewer features.

This study showed experimental results with the RBF kernel. However, other kernel parameters can also be optimized using the same approach. The proposed approach can also be applied to support vector regression (SVR). Because the kernel parameters and input features heavily influence the predictive accuracy of the SVR with different kernel functions; we can use the same GA-based feature selection and parameters optimization procedures to improve the SVR accuracy.

Acknowledgements

The author would like to thank Dr C.-J. Lin and C.-W. Hsu (National Taiwan University, Taipei, ROC) that the support vector machines were constructed with LIBSVM (Version 2.8) by them.

References

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *Proceedings of the 13th international conference on machine learning* (pp. 82–90). San Francisco, CA.

Bradley, P. S., Mangasarian, O. L., & Street, W. N. (1998). Feature selection via mathematical programming. *INFORMS Journal on Computing*, *10*, 209–217.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Chang, C. C., & Lin, C.J. (2001). *LIBSVM: A library for support vector machines*. Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge: Cambridge University Press.

Davis, L. (1991). *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold.

DeLeo, J.M., & Rosenfeld, S.J. (2001). Essential roles for receiver operating characteristic (ROC) methodology in classifier neural network applications. In *Proceedings of the international joint conference on neural networks (IJCNN'01)* (pp. 2730–2731, Vol. 4). Washington, DC, USA.

Fröhlich, H., & Chapelle, O. (2003). Feature selection for support vector machines by means of genetic algorithms. *Proceedings of the 15th IEEE international conference on tools with artificial intelligence, Sacramento, CA, USA* pp. 142–148.

Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*. Reading, MA: Addison-Wesley.

Guyon, I., Weston, J., Barnhill, S., & Bapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422.

Hettich, S., Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases* , Department of Information and Computer Science, University of California, Irvine, CA. http//www.ics.uci.edu/~mlearn/MLRepository.html.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Hsu, C. W., & Lin, C. J. (2002). A simple decomposition method for support vector machine. *Machine Learning*, *46*(1–3), 219–314.

Joachims, T. (1998). Text categorization with support vector machines. In *Proceedings of European conference on machine learning (ECML)* (pp. 137–142). Chemintz, DE.

John, G., Kohavi, R., & Peger, K. (1994). Irrelevant features and the subset selection problem. *Proceedings of the 11th international conference on machine learning, San Mateo, CA* pp. 121–129.

Kecman, V. (2001). *Learning and soft computing*. Cambridge, MA: The MIT Press.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324.

LaValle, S. M., & Branicky, M. S. (2002). On the relationship between classical grid search and probabilistic roadmaps. *International Journal of Robotics Research*, *23*(7–8), 673–692.

Lin, H. T., & Lin, C. J. (2003). *A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods*. Technical report, Department of Computer Science and Information Engineering, National Taiwan University. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf.

Mao, K. Z. (2004). Feature subset selection for support vector machines through discriminative function pruning analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, *34*(1), 60–67.

Pontil, M., & Verri, A. (1998). Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(6), 637–646.

Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *4*(2), 164–171.

Salcedo-Sanz, S., Prado-Cumplido, M., Pérez-Cruz, F., & Bousoño-Calzón, C. (2002). *Feature selection via genetic optimization Proceedings of the ICANN international conference on artificial neural networks, Madrid, Span* pp. 547–552.

Salzberg, S. L. (1997). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, *1*, 317–327.

Schőlkopf, B., & Smola, A. J. (2000). *Statistical learning and kernel methods*. Cambridge, MA: MIT Press.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVM. In S. A. Solla, T. K. Leen, & K.-R. Muller, *Advances in neural information processing systems* (Vol. 13) (pp. 668–674). Cambridge, MA: MIT Press.

Woods, K., & Bowyer, K. W. (1997). Generating ROC curves for artificial neural networks. *IEEE Transactions on Medical Imaging*, *16*(3), 329–337.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, *13*(2), 44–49.

Yu, G. X., Ostrouchov, G., Geist, A., & Samatova, N. F. (2003). An SVM-based algorithm for identification of photosynthesis-specific genome features. *Second IEEE computer society bioinformatics conference, CA, USA* pp. 235–243.