An Embedded Method for Feature Selection Using Kernel Parameter Descent Support Vector Machine

Haiqing Zhu[1], Ning Bi[1], Jun Tan[1(✉)], and Dongjie Fan[2]

[1] School of Mathematics, Sun Yat-sen University, Guangzhou 510275, China
mcstj@mail.sysu.edu.cn

[2] Center for Urban Science and Progress, New York University, New York 10012, USA

Abstract. We introduce a novel embedded algorithm for feature selection, using Support Vector Machine (SVM) with kernel functions. Our method, called Kernel Parameter Descent SVM (KPD-SVM), is taking parameters of kernel functions as variables to optimize the target functions in SVM model training. KPD-SVM use sequential minimal optimization, which breaks the large quadratic optimization problem into some smaller possible optimization problem, avoids inner loop on time-consuming numerical computation. Additionally, KPD-SVM optimize the shape of RBF kernel to eliminate features which have low relevance for the class label. Through kernel selection and execution of improved algorithm in each case, we simultaneously find the optimal solution of selected features in the modeling process. We compare our method with algorithms like filter method (Fisher Criterion Score) or wrapper method (Recursive Feature Elimination SVM) to demonstrate its effectiveness and efficiency.

Keywords: Feature selection · Support vector machine
Kernel function

1 Introduction

Feature Selection is a vital issue in machine learning. It is common to apply feature selection methods to classification problems, especially when those original data sets have redundant features [1].

According to [2], there are three main directions for feature selection: filter, wrapper, and embedded methods.

Filter takes statistical analysis to filter out poorly informative features, it is usually done before the samples taken into a classifier. Relief [3] is a typical filter method which is statistically relevant to the target concept and feeds features into the classifier.

Wrapper approach searches the whole set of samples to score feature subset, therefore it naturally entails training and implementation of learning algorithms

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11258, pp. 351–362, 2018.
https://doi.org/10.1007/978-3-030-03338-5_30

during the procedure of feature selection, wrappers use different classifier such as naive Bayes [4], neural networks [5] and nearest neighbor [6]. The random forests based wrapper approaches [7,8] are widely used to identify important features from feature subset.

In embedded method, feature selection is embedded into the classifier [9], feature is selected by the internal function of an algorithm such as least absolute shrinkage and selection operator (LASSO) [10] and decision tree [11].

Above methods have their limitation, wrapper algorithms are complex in computation, but usually obtain more accurate results than filter methods [12], the problem of a wrapper is high computational cost because it involves repeated training. The robustness of above methods in high dimension data set is a crucial problem. Therefore some features select approaches constructed by combining multiple classifiers, their robust more than the approaches with a single classifier [13]. In addition, support vector machines (SVM) have been proposed as a wrapper classifier for feature selection [14].

Although standard implementation of SVM shows good performance in classification prediction, it cannot rank each features' importance for feature elimination. Thus we introduce a novel approach which selects features according to the descent path of kernel parameters, indirectly figuring out the importance of each features as well as optimizing the model predicting ability. The method we called Kernel Parameter Descent Support Vector Machine (KPD-SVM), the approach not only optimizes the parameter of SVM, but also obtains a subset of features for specific objective. KPD-SVM will be talked in detail and be compared with other characteristic approaches of feature selection in SVM.

2 Related Works

2.1 Support Vector Machine

In this section, we will simply review the development of SVM method.

Support Vector Machine (SVM) is a strictly math-based machine learning model, raised by Vapnik [15]. The principle of SVM classifier is obvious. It tries to find out the optimal hyperplane for the optimization problem with "soft margin" as follows:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

$$s.t. \quad y_i \cdot (\mathbf{w}^T \cdot \phi(\mathbf{x}_i) + b) \geq (1 - \xi_i) \qquad i = 1, \ldots, n,$$

$$\xi_i \geq 0, \qquad i = 1, \ldots, n$$

Here we denote ξ_i as slack variable. The training data can be transformed into higher dimensional space through kernel function $x \rightarrow \phi(x)$. So the decision function can be rewritten as:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b \tag{2}$$

Since the scalar products $\langle \phi(x), \phi(y) \rangle$ are the only value to be calculated, kernel function

$$K(x, y) = \langle \phi(x), \phi(y) \rangle \tag{3}$$

is used to solve them. As result the optimization problem can be rewritten as:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{4}$$

$$s.t. \quad 0 \leq \alpha_i \leq C \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

2.2 Feature Selection in SVMs

Typically, there are three methods in SVM based feature selection process, Filter, Wrapper and Embedded [1]. Here we review each of them briefly and stress one representative algorithm of each method, for experimental comparison in next section.

- **Filter Method:** Among all the measurement in Filter method, Fisher Criterion Score (F-Score) is one of the most common indicator to use. It computes the significance of each feature independently of the other feature by comparing that feature's correlation to the output labels. The respective score $F(j)$ of feature j is given by:

$$F(j) = \left| \frac{\mu_j^+ + \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2} \right| \tag{5}$$

  Where μ_j^+ $(\mu_j^-)$is the mean value for the jth feature in positive(negative) class. And $\sigma_j^+(\sigma_j^-)$ is the standard deviation. When the $F(j)$ is large, it means jth feature has much more information to discriminate itself from other features, which suggests it ranks top of the feature list and would be more likely not to be eliminate and vice versa. The disadvantage of filter method is time consuming and skillful because you need to choose a suitable measurement method.
- **Wrapper Method:** One representative wrapper method is Recursive Feature Elimination SVM (RFE-SVM), which is raised by Guyon [16]. RFE-SVM aims to find out the r-feature subset among the original n-feature set through backward greedy algorithm, which build model by the whole feature at the beginning then cut off one feature according the ranking order. The disadvantage of Wrapper method is that it is more time consuming than filter method because it need to train models on different feature subsets.

– **Embedded Method:** The last method for feature selection is embedded method. The most different novelty between embedded and others is that it conducts the selection in the process of model training. One common embedded method is to add a penalty item to the target function which limits the model complexity [17]. Compared with filter method and wrapper method, we choose embedded method in our model because it is less time consuming.

3 The Proposed Method: KPD-SVM

The principle of proposed method aims to improve the classification performance as well as to eliminate less important features by optimizing parameter/parameters in kernel function. This method use penalty item like $L0-norm$ or $L1-norm$ of the parameter to punish the large number of feature we consider in modeling which is more likely to cause over-fitting problems. Through gradient descent algorithm, we can find out the best solution (which means the best classification performance) of the vector of kernel parameters. During this iteration process, we set the parameters whose values are lower than a small criterion as 0. Thus we can deal with the feature selection task.

3.1 Kernel Function

Among the kernel function SVM commonly uses, we pay attention to the following mostly-used kernels:

Gaussian Kernel function we write the kernel function in the form of the summation in each feature:

$$K(x,y) = exp\Big(-\sum_{j=1}^{d}\frac{(x_j-y_j)^2}{2\sigma_j^2}\Big) \tag{6}$$

where $\sigma = [\sigma_1, \sigma_2, \sigma_3 \ldots, \sigma_n]$ indicates the width of the kernel and determines the kernel shape. d is the number of features. For better demonstration, we denote:

$$\gamma = [\frac{1}{2\sigma_1^2}, \frac{1}{2\sigma_2^2}, \frac{1}{2\sigma_3^2}, \ldots, \frac{1}{2\sigma_d^2}] \tag{7}$$

which leads to

$$K(x,y) = exp\Big(-\sum_{j=1}^{d}\gamma_j(x_j-y_j)^2\Big) \tag{8}$$

Exponential kernel (Laplace) Similar with Gaussian kernel, it is shown as:

$$K(x,y) = exp\Big(-\sum_{j=1}^{d}\gamma_j(x_j-y_j)\Big) \tag{9}$$

Polynomial kernel its function as:

$$K(x,y) = (\alpha x^T y + c)^D \tag{10}$$

Here we fix D and let $c = 1$ in our proposed method, hence we only need to consider the vector of α:

$$K(x,y) = ((\sum_{j=1}^{d} \alpha_j x_j y_j) + 1)^D \tag{11}$$

3.2 Target Function in KPD-SVM

According the previous definition, the set of Lagrange multipliers α is considered, and adding the new parameter γ in kernel function and penalty item of model complexity, therefore the optimization problem $\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$ is minimized with a penalty function and some constrains.Our target function G is as follows:

$$\min_{\alpha,\gamma} G(\alpha,\gamma) = \min_{\alpha} \Psi(\alpha) + \min_{\alpha,\gamma} \Phi(\alpha,\gamma) \tag{12}$$

where the $\Psi(\alpha)$ are transformed from the target optimization function (4) of the standard SVM:

$$\min_{\alpha} \Psi(\alpha) = \min_{\alpha} -\sum_{i=1}^{n} \alpha_i + \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{13}$$

$$s.t. \quad 0 \le \alpha_i \le C \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

and $\Phi(\alpha,\gamma)$ is penalized function, the first item of Eq. (14) is transformed from the second item of Eq. (13), the second item of Eq. (14) is penalized item:

$$\min_{\alpha,\gamma} \Phi(\alpha,\gamma) = \min_{\alpha,\gamma} \frac{1}{2} \sum_{i,s=1}^{n} \alpha_i \alpha_s y_i y_s K(x_i, x_s, \gamma) + C_2 f(\gamma) \tag{14}$$

$$s.t. \quad 0 \le \alpha_i \le C \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\gamma_j \ge 0 \qquad i = 1, \ldots, d$$

where γ_j need to be non-negative and we use $L0 - norm$ as $f(\gamma)$, which is approximately equal to [9]:

$$f(\gamma) = \mathbf{e}^T(\mathbf{e} - exp(-\beta\gamma)) = \sum_{j=1}^{d} [1 - exp(-\beta\gamma_j)] \tag{15}$$

C_2 is the strength of the penalty of the complexity of our model which is different from C for penalty of training error(slack variable ξ). Also $L0-norm$ can be replaced with $L1-norm$ or $L2-norm$ in our target function.

Because this optimization problem is not convex [17], it may be hard to search the globally optimal solution. So that we propose an algorithm to search a locally optimal solution. Then we use a method to solve this optimization problem in two step [17]:

[Step 1] Given a set of fixed kernel parameter γ, calculate the value of α in optimal function $\min_\alpha \Psi(\alpha)$, here sequential minimal optimization(SMO) [18] is a method to solve the SVM QP problem.

For convenience, all quantities that refer to the first multiplier will have a subscript 1, while the other refers to the second multiplier α_2. Without loss of generality, the second multiplier α_2 will be computed firstly. The following bounds W, H apply to α_2 while the target y_1 does not equal the target y_2:

$$W = max(0, \alpha_2 - \alpha_1), H = min(C, C + \alpha_2 - \alpha_1). \tag{16}$$

If the target $y_1 = y_2$, the bounds apply to α_2 is shown as:

$$W = max(0, \alpha_2 + \alpha_1 - C), H = min(C, \alpha_2 + \alpha_1). \tag{17}$$

The second derivative of the objective function $\min_\alpha \Psi(\alpha)$ along the diagonal line can be conducted as:

$$\eta = K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2). \tag{18}$$

Under the normal condition, the objective function is positive definite, there will be a minimum along the direction of the linear constraint, and η is greater than 0. The new minimum is computed along the direction of the constraint as follow:

$$\alpha_2^{opt} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta} \tag{19}$$

where $E_i = u_i - y_i, i = 1, 2$ is the error on the i-th training example, as a next step, the constrained minimum is clipped by the bound W, H. Let $s = y_1 y_2$. The optimal α_1 is computed by the optimized and clipped α_2:

$$\alpha_1^{opt} = \alpha_1 + s(\alpha_2 - \alpha_2^{opt}) \tag{20}$$

Under unusual condition, η will not be positive, which can cause the objective function to become indefinite.

[Step 2] Find out the best γ for given fixed α in step 1, solve the objective function $\min_{\alpha,\gamma} \Phi(\alpha, \gamma)$ using gradient descent algorithm. And if the renewed γ_j is below the criterion we set, eliminate the feature j and loop for next iteration until reaching the stop criterion. For given j the gradient of $F(\gamma_j^*)$ is:

Gaussian

$$\begin{aligned} \Delta_j \Phi(\gamma^*) = \frac{1}{2} \sum_{i,s=1}^{n} \gamma_j^* (x_{i,j} - x_{s,j})^2 \alpha_i \alpha_s y_i y_s K(x_i, x_s, \gamma^*) \\ + C_2 \beta exp(-\beta \gamma_j^*) \end{aligned} \tag{21}$$

Polynomial

$$\begin{aligned}\Delta_j \Phi(\gamma^{poly}) = \frac{1}{2}\sum_{i,s=1}^{n} D x_{i,j} x_{s,j} \alpha_i \alpha_s y_i y_s K(x_i, x_s, \gamma^{poly}, D-1) \\ + C_2 \beta exp(-\beta \gamma_j^{poly})\end{aligned} \tag{22}$$

To avoid misunderstandings of γ in polynomial kernel and target function, we set γ^{poly} in polynomial kernel. **Exponential Kernel (Laplace)**

$$\begin{aligned}\Delta_j \Phi(\gamma^*) = \frac{1}{2}\sum_{i,s=1}^{n} (x_{i,j} - x_{s,j}) \alpha_i \alpha_s y_i y_s K(x_i, x_s, \gamma^*) \\ + C_2 \beta exp(-\beta \gamma_j^*)\end{aligned} \tag{23}$$

The algorithm adjust the kernel components using gradient descent procedure, specially to parameter γ, which is set to be small to avoid negative at the first iterations.

3.3 Detailed Process of Proposed Algorithm

The pseudo code is shown as below:

Algorithm 1. KPD-SVM

kernel selection: we take **Gaussian kernel** as an example.
input:
parameter of gentle update strategy:d_1, d_2, θ;
parameter of update:$\varepsilon_1, \varepsilon_2$

```
start: stop = False,  t = 0,
γ* = (γ*)^[0],  α_1^[0], α_2^[0]
WHILE  stop ≠ True
    train SVM for a given γ* using SMO
    FOR i = 1, ..., d_1
        compute E_1, E_2, η, s
        α_2^[i+1] = α_2^[i] + (y_2(E_1 − E_2))/η,
α_1^[i+1] = α_1^[i] + s * (α_2^[i] − α_2^[i+1])
        IF‖(α_1)^[t+1] − (α_1)^[t]‖ < ε_1
THEN α* = (α_1)^[t+1] Break  ENDIF
    ENDFOR
    train SVM for a given α*
    FOR j = 1, ..., d_2
        (γ_j*)^[t+1] = (γ_j*)^[t] − θΔ_jΦ((γ*)^[t])
        IF(γ_j*)^[t+1] < ε_2
THEN (γ_j*)^[t+1] = 0 Break ENDIF
    ENDFOR
    IF(γ*)^[t], (γ*)^[t+1]  meet the requirements of  ζ_absolute, ζ_relative
        stop = True
    ENDIF
    ENDWHILE
```

where

$$\gamma^* = \sqrt{(}2\gamma) = [\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_d}] \tag{24}$$

and

$$F(\gamma^*) = \sum_{i,s=1}^{n} \alpha_i \alpha_s y_i y_s K(x_i, x_j, \gamma^*) + C_2 f(\gamma^*) \tag{25}$$

In the algorithm, we may consider the following vital step, some details are given as follows:

Kernel Selection, Use the whole features to train model with different kernels (eg. Gaussian, Polynomial) and different parameter (γ, D, c). Calculate the average accuracy of each model with different kernels by cross validations. Then select the kernel with the best performance which is the most appropriate kernel of this data set.

Set Original Value, At the start of algorithm, we give the initial value of α, γ, and some parameter for update.

Calculate α, Based on standard SVM training process and may take SMO algorithm [18] to quickly and efficiently find out the answer α^*.

Update σ and γ, Apply gradient descent algorithm to renew σ_i or γ_i^*, the lines 10–13 of the algorithm shows the iteration process, one by one for fixed the optimal α.

Step size of gradient descent, We set θ as the step size of gradient descend in each iteration.

Elimination criterion, ε is the eliminate threshold which means we eliminate the feature j by setting $\gamma_j^* = 0$ if value γ_j^* is below ε.

Stop criterion, For the stop criterion, we set a relative stop criterion $\zeta_{relative}$ and an absolute stop criterion $\zeta_{absolute}$ in order to balance the time of iterations and the performance of the model. $\zeta_{relative}$ is defined as the ratio $\frac{\|(\gamma^*)^{[t+1]} - (\gamma^*)^{[t]}\|_1}{\|(\gamma^*)^{[t]}\|_1}$ and $\zeta_{absolute}$ is set as $\| (\gamma^*)^{[t]} \|_1$.

3.4 Discussion of Parameter

Our discussion mainly concentrates on one issue: Selection of parameter values in proposed method. Basically, the proposed method outperforms in its process of feature selection and modeling. However, there are some parameters we need to tune for the optimal solution of classification. In [17], it has already concluded that β, ε and $\gamma^{[0]}$ have less influence in the final solution. In terms of the penalty for slack variables, C, we use Leave-One-Out Cross-Validation to find the best value of C in each case.

Complexity Penalty C_2: C_2 is the coefficient of penalty item on the number of feature or model complexity. A large C_2 means a strict limitation to build greatly complicated model. We choose C_2 according to the balance of prediction performances and model complexity.

Step Size θ: θ represents the step size of gradient descend in each iteration.

We want to use an automatically adjusted step size in some cases. Hence,we denote θ_{auto} as $\frac{\varepsilon}{median\{\Delta_j F(\gamma^*)\}}$, $j = 1, \ldots, d$. And we may take $\theta = \min\{\theta_{original}, \theta_{auto}\}$ as step size in each iteration.

Stop Criterion $\zeta_{absolute}, \zeta_{relative}$**:** With the increasing number of iterations, the $1 - norm$ difference of kernel parameter in t and $t + 1$ iteration goes to convergence, which shows the algorithm can find out the best kernel parameter in certain countable iterations.

4 Experiments

In this section, we apply the proposed method to do experiments in some real-world dataset. Also we will compare our method with F-score and RFE-SVM, which represents the filter and wrapper algorithm in feature selection. The measurements we make comparison are as follows: First, model prediction performance. Second, the number of features in the optimal solution.

4.1 Data Set

The data sets we selected are from UCI Machine Learning Database. Detailed information of each data set is shown as follows:

- **Sonar**: This is the data set used by Gorman and Sejnowski in their study of the classification of sonar signals.
  The data set contains 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions. And it contains 97 patterns obtained from rocks under similar conditions. The label associated with each record contains the letter "R" if the object is a rock and "M" if it is a mine (metal cylinder).
- **WBC**: The Wisconsin Breast Cancer data set has 569 observations and 30 features. All feature values are recoded with four significant digits. In addition, people who are diagnosed are labeled as $M \quad (malignant \quad tumor)$ and the other are marked as $B \quad (benign \quad tumor)$.

We basically consider the following three kernel functions: Gaussian, Polynomial, Laplace (Exponential). Then the values of parameters in each kernel function we used are as follows:

- $\sigma_{Gaussian} = (0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 10, 50, 100, 500, 1000)$
- $D = (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$
- $\sigma_{Laplace} = (0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 10, 50, 100, 500, 1000)$

4.2 Case: Sonar

Basic information of this data set is shown in Table 1.

Table 1. Basic information of Sonar (mines vs. rocks) data set

	Features	Observations	Proportion	Predominant class prop.
Total	60	208	100%	53.4%
Train	60	145	70%	54.5%
Test	60	63	30%	50.8%

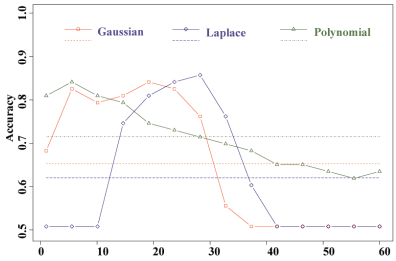

Fig. 1. The accuracy of Gaussian, Laplace and Polynomial in Sonar (horizontal axis represents feature numbers)

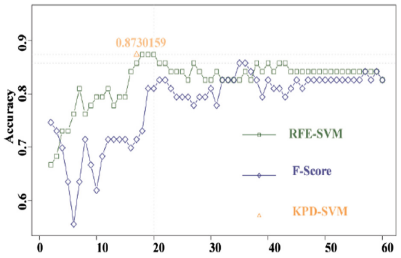

Fig. 2. The accuracy of KPD-SVM, F-Scores and RFE-SVM in Sonar (horizontal axis represents feature numbers)

First we carry out kernel selection. Fig. 1 shows the average performances of each kernel function applied in Sonar. Thus we choose Polynomial Kernel in this case.

Figure 2 shows the performance of proposed method KPD-SVM compared with F-Score and RFE-SVM. The optimal feature subset are selected by each method, and the number of these subsets are shown below: Filter(F1-Scores):24, Wrapper(RFE-SVM):18-20, Embedded(KPD-SVM):20.

In conclusion, KPD-SVM outperforms F-Score and RFE-SVM in this Sonar case.

4.3 Case: WBC

Basic information of this data set is shown in Table 2.

First we carry out kernel selection. In WBC we choose Polynomial Kernel in this case. Figure 3 shows the average performances of each kernel function applied in WBC.

The performance of proposed method KPD-SVM compared with F-Score and RFE-SVM shown in Fig. 4. The optimal feature subset are selected by each

Table 2. Basic information of WBC data set

	Features	Observations	Proportion	Predominant class prop.
Total	30	569	100%	62.7%
Train	30	512	90%	63.4%
Test	30	57	10%	52.6%

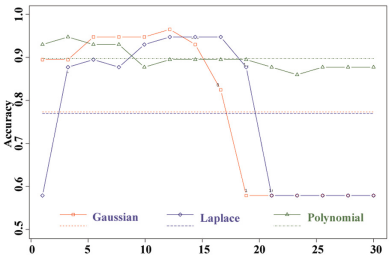

Fig. 3. The accuracy of Gaussian, Laplace and Polynomial in WBC (horizontal axis represents feature numbers)

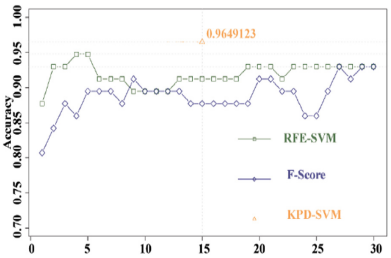

Fig. 4. The accuracy of KPD-SVM, F-Scores and RFE-SVM in WBC (horizontal axis represents feature numbers)

method, and the number of these subsets are shown below: Filter(F1-Scores):26, Wrapper(RFE-SVM):19, Embedded(KPD-SVM):15.

In conclusion, considering the model prediction accuracy and the model complexity (the number of features), we can say KPD-SVM outperforms in this WBC case.

5 Conclusion

In this paper, we have presented a novel method called Kernel Parameter Descent Support Vector Machine (KPD-SVM) for feature selection using kernel functions. Our embedded method can generalize a well-trained SVM classifier as well as a good solution for feature selecting. In addition, our KPD-SVM method outperforms other methods, like filter method (F-Score) and wrapper method (RFE-SVM). Besides, compared with former embedded algorithm by optimizing kernel parameters [1–4], our method has novelties in stop criterion and step size settings in executions, which performs better in time consuming.

Acknowledgements. We would like to acknowledge Professor Chih-Jen Lin from National Taiwan University for his research on Support Vector Machine and his work on software LIBSVM.

This work was supported by the Guangdong Provincial Government of China through the "Computational Science Innovative Research Team" program and Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University, and the National Science Foundation of China (11471012).

References

1. Chandrashekar, G., Sahin, F.: A Survey on Feature Selection Methods. Pergamon Press, Inc., Oxford (2014)
2. Cheriet, M., Kharma, N., Liu, C.L., et al.: Character Recognition Systems: A Guide for Students and Practitioners. Scitech Book News (2007)
3. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: International Workshop on Machine Learning, pp. 249–256. Morgan Kaufmann Publishers Inc., Burlington (1992)
4. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1–2), 273–324 (1997)
5. Sesmero, M.P., Alonso-Weber, J.M., Ledezma, A., et al.: A new artificial neural network ensemble based on feature selection and class recoding. Neural Comput. Appl. **21**(4), 771–783 (2012)
6. Yang, J., Yao, D., Zhan, X., Zhan, X.: Predicting disease risks using feature selection based on random forest and support vector machine. In: Basu, M., Pan, Y., Wang, J. (eds.) ISBRA 2014. LNCS, vol. 8492, pp. 1–11. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08171-7_1
7. Anaissi, A., Kennedy, P.J., Goyal, M., et al.: A balanced iterative random forest for gene selection from microarray data. Bmc Bioinform. **14**(1), 1–10 (2013)
8. Swan, A.L., Mobasheri, A., Allaway, D., et al.: Application of machine learning to proteomics data: classification and biomarker identification in postgenomics biology. Omics J. Integr. Biol. **17**(12), 595–610 (2013)
9. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artif. Intell. **97**(1–2), 245–271 (1997)
10. Tibshirani, R.: Regression shrinkage and selection via the lasso: a retrospective. J. R. Stat. Soc. **73**, 273–282 (2011)
11. Chan, H.P., Kim, S.B.: Sequential random k-nearest neighbor feature selection for high-dimensional data. Expert. Syst. Appl. **42**(5), 2336–2342 (2015)
12. Maldonado, S., Weber, R.: A wrapper method for feature selection using support vector machines. Inf. Sci. **179**(13), 2208–2217 (2009)
13. Tuv, E., Borisov, A., Runger, G., et al.: Feature selection with ensembles, artificial variables, and redundancy elimination. J. Mach. Learn. Res. **10**(3), 1341–1366 (2009)
14. Chen, P., Zhang, D.: Constructing support vector machines ensemble classification method for imbalanced datasets based on fuzzy integral. In: Ali, M., Pan, J.-S., Chen, S.-M., Horng, M.-F. (eds.) IEA/AIE 2014. LNCS (LNAI), vol. 8481, pp. 70–76. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07455-9_8
15. Vapnik, V.N., Vapnik, V.: Statistical Learning Theory, vol. 1. Wiley, New York (1998)
16. Guyon, I., Gunn, S., Nikravesh, M., et al.: Feature Extraction: Foundations and Applications. Studies in Fuzziness and Soft Computing. Springer, New York (2006). https://doi.org/10.1007/978-3-540-35488-8
17. Maldonado, S., Weber, R., Basak, J.: Simultaneous feature selection and classification using kernel-penalized support vector machines. Elsevier Science Inc. (2011)
18. Platt, J.C.: Sequential minimal optimization: a fast algorithm for training support vector machines. In: Advances in Kernel Methods-Support Vector Learning, pp. 212–223 (1998)