PSO-Based Hyper-Parameters Selection for LS-SVM Classifiers

X.C. Guo[1,2], Y.C. Liang[1,*], C.G. Wu[1,3], and C.Y. Wang[1]

[1] College of Computer Science and Technology, Jilin University, Key Laboratory of Symbol
Computation and Knowledge Engineering of the Ministry of Education,
Changchun 130012, P.R. China
ycliang@jlu.edu.cn
[2] College of Science, Northeast Dianli University, Jilin 132012, China
[3] The Key Laboratory of Information Science & Engineering of Railway ministry/The Key
Laboratory of Advanced information science and network technology of Beijing,
Beijing Jiaotong University, Beijing 100044, China

Abstract. The determination for hyper-parameters including kernel parameters and the regularization is important to the performance of least squares support vector machines (LS-SVMs). In this paper, the problem of model selection for LS-SVMs is discussed. The particle swarm optimization (PSO) is introduced to select the LS-SVMs hyper-parameters. In the proposed method we do not need to consider the analytic property of the generalization performance measure and the number of hyper-parameters. The feasibility of this method is evaluated on benchmark data sets. Experimental results show that better performance can be obtained. Moreover, different kinds of kernel families are investigated by using the proposed method. Experimental results also show that the best and good test performance could be obtained by using the SRBF and RBF kernel functions, respectively.

Keywords: least squares support vector machines; particle swarm optimization; fitness function; parameter selection; classification.

1 Introduction

Support vector machines (SVMs) were developed by Vapnik and his colleagues [1]. SVMs are based on the structural risk minimization principle (SRM), which has been shown to be superior to the traditional empirical risk minimization principle (ERM) employed by conventional neural networks. SRM minimizes an upper bound of generalization error as opposed to ERM that minimizes the error on training data. Therefore, the solution of SVM may be global optimum while other neural network models tend to fall into a local optimal solution, and overfitting is unlikely to occur with SVM [2, 3, 4]. The classical training algorithm of SVMs is equivalent to solving a quadratic programming with linearly constraints. During the last decade, many pattern recognitions have been tackled using SVMs. Least Squares Support Vector

[*] Corresponding author.

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 1138 – 1147, 2006.

Machines (LS-SVMs) are introduced by Suykens et. al as reformulations to standard SVMs [5] which lead to solving linear Karush-Kuhn-Tucker (KKT) systems for classification problems as well as regression. LS-SVM simplifies the solution process of standard SVM in a great extent by substituting the inequality constraints by equality counterparts. Consequently, the decision function can be obtained by solving a group of linear equalities rather than quadratic programming.

For the standard SVMs and its reformulations, LS-SVM, the regularization parameter and kernel parameter(s) are called hyper-parameters, which play a crucial role to the performance of the SVMs. There exist different techniques for tuning the hyper-parameters related to the regularization constant and the parameter of kernel function. These methods can be divided into two classes: one is the analytical and algebraic techniques, another is heuristic search algorithm (including grid search). The analytical and algebraic techniques are almost based on the gradient of some generalized error measure [6-13]. Recently, genetic algorithm, simulated annealing algorithm and other evolutionary strategy [14-19] are employed for the hyper-parameters of SVMs. Iterative gradient-based algorithms, which usually rely on smoothed approximations of a function, do not ensure that the search direction points exactly to an optimum of the generalization performance measure which is often discontinuous. Grid search which needs an exhaustive search over the space of hyper-parameters is often used to select parameters [20]. This procedure requires a grid search over the space of parameter values and needs to locate the interval of feasible solution and a suitable sampling step. This is a tricky task since a suitable sampling step varies from kernel to kernel and the grid interval may not be easy to locate without prior knowledge of the problem. Moreover, when there are more than two hyper-parameters, the manual model selection may become intractable.

In this paper, a new parameters selection algorithm is proposed based on the principles of the particle swarm optimization (PSO). The PSO is an evolutionary computation technique based on swarm intelligence. It follows a collaborative population-based search, which models over the social behavior of bird flocking. The PSO system combines experiences form both self and neighboring and attempts to balance exploration and exploitation. The PSO has many advantages over other heuristic techniques, e.g., it can be used effectively to exploit the distributed and parallel computing capabilities, to escape local optima, and to implement in a few lines of computer codes. The proposed method is applied to tuning kernels and regularization parameters of LS-SVMs.

2 LS-SVM Classifiers

Consider a given training set $\{(x_i, y_i) \mid x_i \in R^n, y_i \in \{-1,+1\}\}_{i=1}^{N}$, where x_i is input and y_i is the binary class label. The discriminant function takes the following form:

$$y = \text{sign}[w^T \phi(x) + b] \tag{1}$$

where the nonlinear function $\phi(\cdot)$, which is not explicitly constructed, maps the input into a higher dimensional feature space (can be infinite dimension). The coefficient vector w and bias term b need to be determined. In order to obtain the coefficient vector w and bias term b, the following optimization problem to be solved is as follows [5, 20]

$$\min_{w,e_i} J(w,e) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \tag{2}$$

subject to the equality constraints

$$y_i[w^T \phi(x_i) + b] = 1 - e_i,\ i = 1,2,\cdots,N \tag{3}$$

The Lagrangian corresponding to Eq. (2) can be defined as:

$$L(w,b,e_i,\alpha) = J(w,b,e) - \sum_{i=1}^{N} \alpha_i \{ y_i[w^T \phi(x_i) + b] - 1 + e_i \} \tag{4}$$

where α_i $(i = 1,2,\cdots,N)$ are Lagrange multipliers. The KK-T conditions can be expressed by

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 & \Rightarrow \ w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \\ \dfrac{\partial L}{\partial b} = 0 & \Rightarrow \ \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \dfrac{\partial L}{\partial e_i} = 0 & \Rightarrow \ \alpha_i = \lambda e_i \\ \dfrac{\partial L}{\partial \alpha_i} = 0 & \Rightarrow \ y_i[w^T \phi(x_i) + b] - 1 + e_i = 0 \end{cases} \quad i = 1,2,\cdots,N \tag{5}$$

Referring to Suykens and Gestel's work [5, 20], the solution of the optimization problem (2) can be obtained by solving the following linear equations:

$$\begin{bmatrix} 0 & y^{\mathrm{T}} \\ y & ZZ^T + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} \mathrm{b} \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{1} \end{bmatrix} \tag{6}$$

where $Z = [\phi(x_1) y_1, \cdots, \phi(x_N) y_N]^T$, $y = [y_1, y_2, \cdots, y_N]^T$, $\vec{1} = [1,1,\cdots,1]^T$, $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_N]^T$ and $\Omega = ZZ^T$ takes the form as $\Omega_{kl} = y_k y_l \phi(x_k)^T \phi(x_l) = \psi(x_k, x_l)$ $(k,l = 1,2,\cdots,N)$ according to mercer's condition.

Table 1. Classical common kernel functions

Name	Function Expression
Linear Kernel	$\psi(x, y) = x^T y$
Polynomial Kernel	$\psi(x, y) = (1 + x^T y / \sigma^2)^d$
RBF Kernel	$\psi(x, y) = \exp\{-\|x - y\|_2^2 / \sigma^2\}$
SRBF Kernel	$\psi(x, y) = \exp\{-\sum_{i=1}^{n} (x_i - y_i)^2 / \sigma_i^2\}$
MLP* Kernel	$\psi(x, y) = \tanh(kx^T y + \theta)$

For the choice of the kernel function $\psi(\cdot,\cdot)$ one has several alternatives. Some of common kernel functions are listed in **Table 1.**, where c , d , σ , k and θ are constants, and for the function with "*" symbol. A suitable choice for k and θ is needed to enable the kernel function to satisfy Mercer condition.

After solving the Eq. (6), the LS-SVM model for classification can be obtained as:

$$y = \mathrm{sign}\left[\sum_{i=1}^{N}\alpha_i y_i \psi(x, x_i) + b\right] \tag{7}$$

3 PSO-Based Hyper-Parameters Selection for LS-SVM

3.1 Brief Introduction to PSO

The particle swarm optimization (PSO), originally developed by Kennedy and Elberhart [21], is a method for optimizing hard numerical functions on metaphor of social behaviors of flocks of birds and schools of fish. It is an evolutionary computation technique based on swarm intelligence. A swarm consists of individuals, called particles, which change their positions over time. Each particle represents a potential solution to the problem. In a PSO system, particles fly around in a multi-dimensional searching space. During its flight each particle adjusts its position according to its own experience and the experience of its neighboring particles, making use of the best position encountered by itself and its neighbors. The effect is that particles move towards the better solution areas, while still having the ability to search a wide area around the better solution areas. The performance of each particle is measured according to a pre-defined fitness function, which is related to the problem being solved. The PSO has been found to be robust and fast in solving non-linear, non-differentiable and multi-modal problems [22]. The mathematical description and executive steps of the PSO are as follows.

Let the i th particle in a D-dimensional space be represented as $\vec{x}_i = (x_{i1},\ldots,x_{id},\ldots,x_{iD})$. The best previous position of the i th particle is recorded and represented as $\vec{p}_i = (p_{i1},\ldots,p_{id},\ldots,p_{iD})$, which gives the best fitness value and is also called $pbest$. The index of the best $pbest$ among all the particles is represented by the symbol g . The location P_g is also called $gbest$. The velocity for the i th particle is represented as $\vec{v}_i = (v_{i1},\ldots,v_{id},\ldots,v_{iD})$. The concept of the particle swarm optimization consists of changing the velocity and location of each particle towards its $pbest$ and $gbest$ locations according to Eqs. (1) and (2) at each time step:

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) , \tag{8}$$

$$x_{id} = x_{id} + v_{id} , \tag{9}$$

where w is the inertia coefficient which is a constant in the interval [0, 1] and can be adjusted in the direction of linear decrease [23]; c1 and c_2 are learning rates which are nonnegative constants; r_1 and r_2 are generated randomly in the interval [0, 1]; $v_{id} \in [-v_{\max}, v_{\max}]$, and $v_{\max}$ is a designated maximum velocity. The termination criterion for iterations is determined according to whether the maximum generation or a designated value of the fitness is reached.

3.2 PSO-Based Hyper-Parameters Selection

There are two key factors to determine the optimized hyper-parameters using particle swarm optimization (PSO): one is how to represent the hyper-parameters as the particle's position, namely how to encode. Another is how to define the fitness function which evaluates the goodness of a particle. The following will give the two key factors.

Encoding Hype-parameters: The optimized hyper-parameters for LS-SVMs include kernel parameter(s) (except for linear kernel) and regularization parameter. In solving hyper-parameters selection by the PSO, each particle is requested to represent a potential solution, namely hyper-parameters combination. So let us denote an m hyper-parameters combination as a vector of dimension m. For example, SRBF: v=(γ, σ_1, σ_2, ..., σ_{ninput}), Pol: v=(γ, σ, *d*). The method of encoding is very intuitionistic. In this study, v=($\log\gamma$, $\log\sigma_1$, $\log\sigma_2$, ..., $\log\sigma_{ninput}$) and v=($\log\gamma, \log\sigma$, log*d*) is used because this gives a more stable optimization. For different kernels the length of the parameters vector is different.

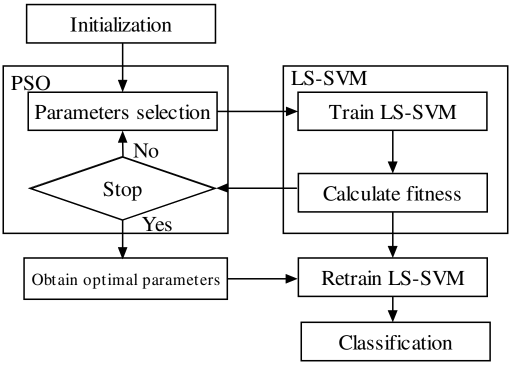

Fig. 1. Flow chart of PSO-based hyper-parameters algorithm

Fitness function: The fitness function is the generalization performance measure. For the generation performance measure, there are some different descriptions. Therefore the corresponding fitness can be determined. In this paper, the employed fitness function will be defined in Section 4.2.

The flow chart of the PSO-based hyper-parameters selection algorithm for the LS-SVM is shown in **Fig. 1**.

4 Numerical Experiments

4.1 Data sets and Its Preprocessing

Experiments are performed to evaluate the performance of PSOLS-SVM for binary classification. We selected the Diabetes (DB), Breast-Cancer (BC), Heart (HT), Thyroid (TD) and Titanic (TC) data sets from the UCI Machine Learning repository. The data sets used in this study are provided by G. Ratsch at http://ida.first.gmd.de/aetsch/data/benchmarks.htm. The detailed description of these data sets is reported in **Table 2**. These data sets have been referred to numerous times in the literature, which makes them very suitable for benchmarking purposes. The data are preprocessed and partitioned as in [24]: Each component of the input data is normalized to zero mean and unit standard deviation. It ensures the larger value input attributes do not overwhelm smaller value inputs; hence helps to reduce errors. After normalized to zero mean and unit standard deviation, each data set is divided randomly 100 times into different pairs of disjoint train and test sets.

Table 2. Description of the data sets

	DB	BC	HT	TD	TC
N_{train}	468	200	170	140	150
N_{test}	300	77	100	75	2051
N	768	277	270	215	2201
n_{input}	8	9	13	5	3

N_{train} and N_{test} denotes the number of train and test patterns, respectively. N stands for the total number of the patterns. n_{input} is the number of the input.

4.2 Determination of the Fitness Function

In PSO, the fitness value is used to evaluate goodness of the particles, namely hyper-parameter combination. So the determination of fitness function is important to the parameters of LS-SVM. The fitness should reflect the generalization performance of LS-SVM for different hyper-parameter combination. The fitness function is defined as follows: For each particle, five LS-SVMs are built using the training sets of the first five data partitions and the average of the classification correct rates on the corresponding five test sets determines the fitness value (training performance of LS-SVM). The particle with the largest fitness is chosen as the optimal parameters combination [25]. The test performance of LS-SVM with optimal parameters is measured as follows: 100 LS-SVMs are built using the optimal parameters using all the training sets and the average of the classification correct rates on the corresponding 100 test sets is define as the test performance of LS-SVM.

4.3 Experiment Results

All experiments are performed on a PC with Pentium IV 2.6GHz processor and 512MB memory. The optimal values for the regularization parameter and the kernel parameters with linear, polynomial, RBF and SRBF kernel are shown in **Table 3**. The first column is the parameters used with different kernels. The rest column is the optimal parameters values for different data sets. For polynomial kernel, the degree is denoted in bracket. The corresponding optimal values are not given because of the large number of parameters. In **Table 4** and **Table 5**, the first column lists the different kernels and the first row shows the benchmark data sets in our study, respectively. **Table 4** and **Table 5** show the performance of training and test of LS-SVM on different data sets, respectively. Experimental results in **Table 5** show that the SRBF kernel yields the best test performance and the polynomial and RBF kernel give good test performance. **Table 6**, in which test error found by PSO-based hyper-parameters selection of LS-SVM and other methods for different data sets is listed, shows that the results obtained from the proposed method for LS-SVM with SRBF kernel are better than those in literature [24].

Table 3. Optimized hyper-parameter values of the LS-SVM with linear, RBF and polynomial kernels for different data sets

	BC	HT	TiD	DS	TC
Lin: $\log_{10}(\gamma)$	-0.23	-2.26	0.68	-1.06	-1.67
Pol: $\log_{10}(\gamma)$	1.35	1.36	1.30	1.08	1.69
Pol: $\log_{10}(\sigma)$	1.39	2.16	0.62	1.34	-0.23
Pol: $\log_{10}(d)$	0.76 (5)	0.71 (5)	0.83 (6)	0.75 (5)	0.68 (4)
RBF: $\log_{10}(\gamma)$	0.33	1.41	1.27	1.92	4.00
RBF: $\log_{10}(\sigma)$	0.76	1.83	0.26	1.20	0.58

Table 4. LS-SVM training performance with different kernel functions by using the optimized parameters for different data sets

	BC	HT	TD	DS	TC
Lin	72.99	83.00	84.53	76.73	77.59
Pol	74.81	83.00	93.33	77.20	78.43
RBF	74.81	83.00	97.07	77.27	77.60
SRBF	79.74	86.20	98.40	78.00	77.55

Table 5. LS-SVM test performance with different kernel functions by using the optimized parameters for different data sets

	BC	HT	TD	DS	TC
Lin	72.98	84.41	85.21	76.63	77.33
Pol	73.66	84.41	92.01	77.01	77.02
RBF	73.82	84.42	95.99	77.05	78.10
SRBF	76.09	84.49	96.59	77.46	78.41

Table 6. Test performance found by PSO-based hyper-parameters selection of LS-SVM and other methods for different data sets

	BC	HT	TD	DS	TC
RBF-Network	72.36	82.45	95.48	76.71	76.74
AdaBoost with RBF-Network	69.64	79.71	95.60	73.53	77.42
LP_Reg-AdaBoost	73.21	82.51	95.41	75.89	76.02
QP_Reg-AdaBoost	74.09	82.83	95.65	74.61	77.29
AdaBoost_Reg	73.49	83.53	95.45	76.21	77.36
SVM with RBF-Kernel	73.96	84.05	95.20	76.47	77.58
KFD with RBF-Kernel	75.23	83.86	95.80	76.79	76.75
LS-SVM with SRBF-Kernel	**76.09**	**84.49**	**96.59**	**77.46**	**78.41**

5 Conclusions

A promising novel particle swarm optimization-based hyper-parameters selection for LS-SVM classifier is proposed. The presented method does not consider the analytic property of the generalization performance measure and the number of hyper-parameters. The feasibility of our presented method is evaluated on benchmark data sets. Experimental results show that better performance can be obtained. Experiments on SRBF kernel show that the proposed method can tune much more hyper-parameters. Experimental results also show that the SRBF kernel yields the best test performance and the polynomial and RBF kernel gives better test performance. Compared with the results of other methods, the proposed PSO-based hyper-parameters selection for LS-SVM yields higher accurate rate for all data sets tested in this paper.

Acknowledgment

The authors are grateful to the support of the National Natural Science Foundation of China (60433020), the science-technology development project of Jilin Province of China (20050705-2), the doctoral funds of the National Education Ministry of China (20030183060), the graduate innovation lab of Jilin University (503043), and “985”project of Jilin University.

References

1. Vapnik V. N.: Statistical Learning Theory. John Wiley, New York, USA (1998)
2. Christianimi N., Shawe-Taylor J.: An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
3. Gunn S. R.: Support Vector Machines for Classification and Regression. Technical Report. University of Southampton (1998)
4. Kim K. J.: Financial Time Series Forecasting Using Support Vector Machines. Neurocomputing. 1-2 (2003) 307-319

5. Suykens J. A. K, Vandewalle J.: Least Squares Support Vector Machine Classifiers. Neural Processing Letters. 9 (1999) 293-300
6. Chapelle O., Vapnik V. N., Bousquet O., Mukherjee S.: Choosing Multiple Parameters for Support Vector Machines. Machine Learning. 1-3 (2002) 131-159
7. Vapnik V. N., Chapelle O.: Bounds on Error Expectation for Support Vector Machines. Neural computation. 9 (2000) 2013-2036
8. Chapelle O., Vapnik V. N.: Model Selection for Support Vector Machines. In: Solla S., Leen T., Müller K.-R. (Eds.): Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference, Vol. 12. MIT Press, Cambridge, MA (2000) 230-236
9. Chung K. -M., Kao W. -C., Sun C. -L., Lin C.-J.: Radius Margin Bound for Support Vector Machines with RBF Kernel. Neural Computation. 11 (2003) 2643-2681
10. Wahba G., Lin X., Gao F., Xiang D., Klein R., Klein B.: The Bias-variance Trade-off and the Randomized gacv, In: Kearns M., Solla S., Cohn D. (Eds.): Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference. Vol. 11, MIT Press, Cambridge, MA (1999) 620–626
11. Sathiya Keerthi S.: Efficient Tuning of SVM Hyperparameters Using Radius/Margin Bound and Iterative Algorithms. IEEE Transactions on Neural Network. 5 (2002) 1225-1229
12. Ayat N. E., Cheriet M., Suen C. Y.: Automatic Model Selection for the Optimization of SVM Kernels. Pattern Recognition. 10 (2005) 1733-1745
13. Gold C., Sollich P.: Model Selection for Support Vector Classification. Neurocomputing. 1-2 (2003) 221-249
14. Eads D. R., Hill D., Davis S., Perkins S. J., Ma J., Porter R. B., Theiler J. P.. Genetic Algorithms and Support Vector Machines for Time Series Classification. In: Bosacchi B., Fogel D. B., Bezdek J. C. (Eds.): Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V. Proceedingsof the SPIE, Vol. 4787. (2002) 74–85
15. Frohlich H., Chapelle O., Scholkopf B.: Feature Selection for Support Vector Machines by Means of Genetic Algorithms. In: Proceedings of the 15th IEEE International Conference on Tools with AI (ICTAI 2003). IEEE Computer Society. Institute of Electrical and Electronics Engineers Inc., USA (2003) 142–148
16. Jong K., Marchiori E., r Vaart A. van de: Analysis of Proteomic Pattern Data for Cancer Detection. In: Raidl G. R., Cagnoni S., Branke J., Corne D. W., Drechsler R., Jin Y., Johnson C. G., Machado P., Marchiori E., Rothlauf F., Smith, G. D. Squillero G. (Eds.): Applications of Evolutionary Computing. Lecture Notes in Computer Science, Vol. 3005, Springer Berlin, Heidelberg (2004) 41–51
17. Miller M. T., Jerebko A. K., Malley J. D., Summers R. M.: In: Clough A. V., Amini A. A. (Eds.): Feature Selection for Computer-aided Polyp Detection Using Genetic Algorithms, Medical Imaging 2003: Physiology and Function: Methods, Systems, and Applications, Proceedings of the SPIE, Vol. 5031, (2003) 102–110.
18. Ping-Feng Pai, Wei-Chiang Hong: Support Vector Machines with Simulated Annealing Algorithms in Electricity Load Forecasting. Energy Conversion & Management. 17 (2005) 2669-2688
19. Frauke Friedrichs, Christian Igel: Evolutionary Tuning of Multiple SVM Parameters. Neurocomputing. 64 (2005) 107-117
20. Gestel T. V., Suykens J. A. K., Baesens B., Viaene S., Vanthienen J., Dedene G., Moor B. D., Vandewalle J.: Benchmarking Least Squares Support Vector Machine classifiers. Machine Learning. 1 (2004) 5-32

21. Kennedy J., Eberhart R.: Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia. IEEE Service Center, Vol. 4. Piscataway, NJ (1995) 1942–1948
22. Ge H. W., Liang Y. C., Zhou Y., Guo X. C.: A Particle Swarm Optimization-based Algorithm for Job-shop Scheduling Problem. International Journal of Computational Methods. 3 (2005) 419-430
23. Shi Y., Eberhart R.: A Modified Particle Swarm Optimizer. IEEE World Congress on Computational Intelligence, Alaska, ALTEC, Vol. 1 (1998) 69-73
24. Ratsch G., Onoda T., Muller K.-R.: Soft Margins for Adaboost. Machine Learning. 3 (2001) 287–330
25. Meinicke P., Twellmann T., Ritter H.: Discriminative Densities from Maximum Contrast Estimation. In: Becker S., Thrun S., Obermayer K. (Eds.): Advances in Neural Information Processing Systems, Vol. 15. MIT Press, Cambridge (2002) 985–992