An Improved Grid Search Algorithm of SVR Parameters Optimization

Qiujun Huang
Beijing University of Posts and Telecommunications
Beijing, 100876, China
huangqiujunbupt@gmail.com

Jingli Mao
Beijing University of Posts and Telecommunications
Beijing, 100876, China
mjl018@126.com

Yong Liu
Beijing University of Posts and Telecommunications
Beijing, 100876, China
liuyongamtf@yahoo.com.cn

Abstract—**The proper selection of parameters, kernel parameter g, penalty factor c, non-sensitive coefficient p of Support Vector Regression (SVR) model can optimize SVR's performance. The most commonly used approach is grid search. However, when the data set is large, a terribly long time will be introduced. Thus, we propose an improved grid algorithm to reduce searching time by reduce the number of doing cross-validation test. Firstly, the penalty factor c could be calculated by an empirical formula. Then the best kernel parameter g could be found by general grid search algorithm with the achieved c and a p-value selected randomly within a range. According to the achieved c and p, the grid search algorithm is used again to search the best non-sensitive coefficient p. Experiments on 5 benchmark datasets illustrate that the improved algorithm can reduce training time markedly in a good prediction accuracy.**

Keywords- SVR; kernel parameter g; penalty factor c; non-sensitive coefficient p; grid search.

I. INTRODUCTION

SVR (Support Vector Regression) [1] is a new type of machine learning algorithm based on statistical learning theory and structural risk minimization principle, which has good generalization ability. The parameters which contribute most to the performance of SVR include penalty coefficient c, kernel parameter g and non-sensitive coefficient p. Optimization of these parameters is the key point to improve the training ability of SVR model.

A general search algorithm of SVR parameters optimization is known as grid search algorithm, by which, global optimum parameters could be achieved. However it takes too much time. Now many scholars have approached a large number of exploration for SVR parameter optimization [2~5]. Most of these methods are based on some intelligent algorithms for heuristic searching such as genetic algorithms, ant colony algorithm, etc. But these algorithms are easy to go into the shortcomings of local optimum limit, and can't meet the general requirements. Jing-xu Liu et al [6] take heuristic searching algorithm, Li-ping Ni et al [7] use ant colony parameters optimization algorithm, the time of optimization were all reduced, but still too long, easily falling into local optimum and other shortcomings. Chien-ming Huang [8] et al use of nested uniform design optimization, efficient but risk of falling into local minimum. Xiao-fang Yuan [9] uses chaos optimization algorithm, Chang-ping Liu [10] take gradient algorithm (local area still using the chaos algorithm) for parameter optimization, improving efficiency; But because of the randomness of chaotic sequence, there is still risk of falling into local minimum. Zhi-ming Wang [11] makes a uniform-design table to calculate the index value, and then uses self-calling to construct the regression model between the evaluation index value and the factor, and obtains better results. However, the choice of the small sample parameter is still inflexible and experienced to select.

In this paper, we propose an improved grid algorithm not only getting the approximately global optimum parameters but also reducing training time markedly.

This paper is organized as follows. Section II gives a brief introduction to the theory of SVR. Section III describes the improved grid search algorithm of SVR parameters optimization. Section IV gives detailed simulation results and analysis. Finally, conclusion is given in Section V.

II. SUPPORT VECTOR REGRESSION [12, 13, 14]

Given a training set of instance-label pairs (x_i , y_i); i = 1L where $x_i \in R^n$, and y_i is any real number. Construct regression function:

$$f(x) = w^T \phi(x) + b \tag{1}$$

Nonlinear mapping $\phi(x)$, x is mapped to the data of high-dimensional feature space, w and b are the weight vector and bias in the single-output regression algorithm, ε-insensitive loss function, defined as :

$$c(x, y, g(x)) = \begin{cases} 0, \mid y - g(x) \mid < \varepsilon \\ \mid y - g(x) \mid - \varepsilon, others \end{cases} \tag{2}$$

This work is supported by Chinese University Scientific Fund Supplement, "Traffic analysis and modeling in the complex heterogeneous network environment".

978-1-4673-2101-3/12/$31.00 ©2012 IEEE

If the estimate is defined as the ε-insensitive loss function ($\varepsilon \geq 0$) for risk minimization problem, and the introduction of slack variables ξ_i and ξ_i^* , then the problem is reduced to solving the following quadratic programming problem:

$$R(w,\xi^*,\xi) = \frac{1}{2} w^T w + c\sum_{i=1}^{l}(\xi + \xi^*) \tag{3}$$

Subject to

$$\begin{cases} (w^T \Phi(x_i) + b) - y_i \leq \varepsilon + \xi_i \\ y_i - (w^T \Phi(x_i) + b) \leq \varepsilon + \xi^*{}_i \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{4}$$

Introduce Lagrange function:

$$\begin{aligned} L = {} & \frac{1}{2} w^T w + c\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ & -\sum_{i=1}^{l}\alpha_i (y_i - w^T \phi(x_i) - b + \varepsilon + \xi_i) \\ & -\sum_{i=1}^{l}\alpha_i^* (w^T \phi(x_i) + b - y_i + \varepsilon + \xi_i^*) \\ & -\sum_{i=1}^{l}(\gamma_i \xi + \gamma_i^* \xi^*{}_i) \end{aligned} \tag{5}$$

According to KKT conditions, the following equation holds:

$$w = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)\phi(x_i) \qquad \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{6}$$

$$\alpha_i, \alpha_i^* \in [0, c]; i = 1, 2, ..., l \tag{7}$$

Which α_i, α_i^* are Lagrangian multipliers, put expression (6), (7) into expression (5), and then the above quadratic programming problem can be turn into solving the following maximum value functional:

$$\begin{aligned} W(\alpha, \alpha^*) = {} & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) \\ & +\varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) - \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) \end{aligned} \tag{8}$$

Subject to

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{9}$$

Put α_i, α_i^* obtained from (16) into (14), we can gain the regression function:

$$f(x) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)K(x_i, x) \tag{10}$$

Where $\alpha_i^* - \alpha_i$ is not equal to zero corresponds to the sample support vector.

$K(x_i, x)$ is the kernel function.

One of the commonly used kernel functions is radial basis function (RBF).Radial basis functions have received significant attention, most commonly with a Gaussian of the form,

$$K(x_i, x_j) = \exp(-g \| x_i - x_j \|^2), g > 0 \tag{11}$$

Classical techniques utilising radial basis functions employ some method of determining a subset of centres. Typically a method of clustering is first employed to select a subset of centres. An attractive feature of the SVR is that this selection is implicit, with each support vectors contributing one local Gaussian function, centred at that data point. By further considerations it is possible to select the global basis function width, s, using the SRM [15] principle.

III. IMPROVED GRID SEARCH ALGORITHM

A. Grid Search Algorithm

The general grid search algorithm is trying all possible values of the(*c*, *g*, *p*) pair. To each (*c*, *g*, *p*) pair, cross-validation is used to get its MSE. In cross-validation [16], firstly, randomly divide the training set into k disjoint subsets of roughly equal size for each fold. Then use K-1 subsets as training subset, the remaining as testing subset of the regression model. A MSE is calculated for each testing subset. The average MSE of k testing subsets is as performance evaluation parameter.

After all possible values of (*c*, *g*, *p*) pair are tested, (*c*, *g*, *p*) pair with the minimum MSE is the best pair. However the exhaustive process needs a lot of time, which is not proper for many applications.

B. The proposed algorithm

Optimal choice of regularization parameter c can be derived from standard parameterization of SVR solution given by expression (10) [17]:

$$\begin{aligned} |f(x)| &\leq \sum_{i=1}^{l} |(a_i - a_i^*)| \cdot |K(x_i, x)| \\ &\leq \sum_{i=1}^{l} C \cdot |K(x_i, x)| \end{aligned} \tag{12}$$

Further we use kernel functions (upper) bounded in the input domain. To simplify the presentation, assume RBF kernel function (11), so that $K(x_i, x) \leq 1$. Hence we obtain the following upper bound on SVR function:

$$|f(x)| \leq c \cdot l \tag{13}$$

Expression (5) relates regularization parameter *c* and the number of support vectors (*l*), for a given value of ε .However, note that the relative number of support vectors depends on the ε -value. In order to estimate the value of *c* independently of (unknown) *l*, one can set its value so that $c \geq |f(x)|$ for all training samples, which leads to setting c equal to the range of response values of training data. However, such a setting is quite sensitive to outliers, so the following prescription is used instead:

$$c = \max(\left|\bar{y} + 3\sigma_y\right|, \left|\bar{y} - 3\sigma_y\right|) \quad (14)$$

Where $\bar{y}$ is the mean of training output and σ_y is the standard deviation of training outputs.

For different values of p, the best (c, g, p) pair with the minimum MSE by cross-validation is observed to have almost the fixed value of c and g [18]. Since c can be calculated by formula (14), we can firstly select a p-value randomly within a range, then get the best value of g.

The proposed improved grid search algorithm is described as follows.

Define the SVR default range of searching parameters: $\log_2 c$ = -1 : step : 6 ,$\log_2 g$ = -8 : step : 0, $\log_2 p$ = -8 : step : -1. And general step = 1. Fig.1 shows the flow of the algorithm.

Step1: Calculate the best c_f value according to formula (14).

Step2: Randomly select a p-value p_1. Within the default range, all possible values of g are selected. Then we can get (c_f, g, p_1) combinations with the fixed c_f, p_1.

Step3: These combinations (c_f, g, p_1) with the fixed c_f, p_1 are used to do the cross-validation with training set T to get the corresponding MSEs.

Step4: The g in the (c_f, g, p_1) combination with the minimum MSE is selected to be the best g_f.

Step5: Now we have c_f and g_f. Within the default range, all possible values of p are selected. Then we can get (c_f, g_f, p) combinations with the fixed c_f, g_f.

Step6: These combinations (c_f, g_f, p) with the fixed c_f and g_f are used to do the cross-validation with training set T to get the corresponding MSEs.

Step7: The p in the (c_f, g_f, p) combination with the minimum MSE is selected to be the best p_f.

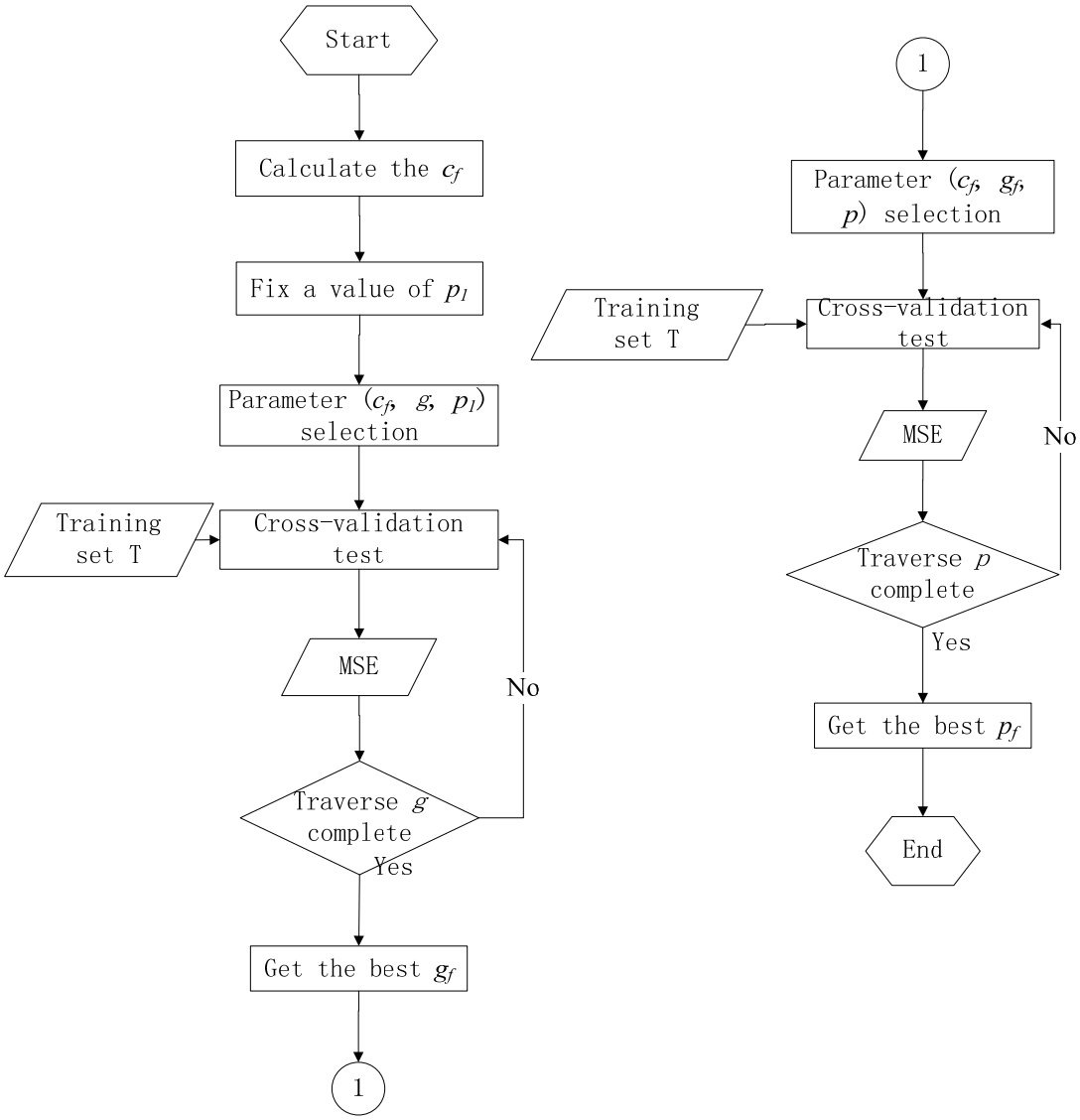

Figure 1. The flow chart of the best parameters (c, g, p) selection

IV. SIMULATION RESULTS AND ANALYSIS

We selected five data sets. They are mg, cpusmall, abalone, space_ga, triazines that are released on http://www.csie.ntu.edu.tw. For each set, we randomly select two-thirds data as the training data, the others as the prediction data. Firstly we separately use general grid search algorithm and the proposed algorithm to select the best parameters (*c*, *g*, *p*) of each set. Then we build a SVR model for each set using its (*c*, *g*, *p*) so as to predict the result of the input of prediction data of the set. Finally, we get the MSE of each set. Here are their spending times, MSEs and (*c*, *g*, *p*) by general grid search algorithm and the proposed algorithm in tables below.

TABLE I. Spending time of the two ALGORITHMS .unit(s)

Data Name	Total number of samples	Number of training data	Number of Prediction data	Grid Search	Improved Grid Search
triazines	186	120	66	53	2
mg	1385	1000	385	666	23
abalone	4177	3133	1044	3645	122
cpusmall	8192	7500	692	14219	473
space	3107	2000	1107	639	22

TABLE II. MSEs of THE two algorithms

Data Name	Total number of samples	Number of training data	Number of Prediction data	Grid Search	Improved Grid Search
triazines	186	120	66	0.019886	0.0201518
mg	1385	1000	385	0.014535	0.0145843
abalone	4177	3133	1044	4.75677	4.75677
cpusmall	8192	7500	692	8.6326	8.63263
space	3107	2000	1107	0.016978	0.0173914

TABLE III. (c, g, p) of the two algorithms

Data Name	Total number of samples	Number of training data	Number of Prediction data	Grid Search	Improved Grid Search
triazines	186	120	66	(-1 ,-5 ,-5)	(0,-6,-5)
mg	1385	1000	385	(3,-1,-3)	(1,0,-3)
abalone	4177	3133	1044	(5,0 ,-1)	(5,0,-1)
cpusmall	8192	7500	692	(6, 0, -1)	(6, 0, -1)
space	3107	2000	1107	(2, -1, -7)	(1,0,-6)

Table1 shows that the improved grid search can significantly decrease the parameter optimization time. From Table1 and Table2, we can see that improved grid search can get almost the same (*c*, *g*, *p*) and MSE as grid search algorithm.

V. CONCLUSION

The optimization of parameters (c, g, p) of SVR model is essential. This paper proposes an improved grid search algorithm, which not only can assure the prediction accuracy but also can reduce training time markedly.

The grid search algorithm mainly spends time on doing cross-validation test. The proposed algorithm only does 17 times cross-validation test while the general grid search has to do 576 times. Experiments show that the proposed algorithm is more efficient than the gird search. From the results shown in the tables above, we can see that parameter optimization time is reduced to almost one-thirtieth of the time spent by grid search. Experiments have demonstrated its effectiveness.

REFERENCES

[1] Vapnik V N, "Statistical Learning Theory[M]." New York, USA: Wiely, 1998.

[2] Tian peng, "Improved particle swarm optimazation application in support vector machine training [J]." Automation Technology and Application. 2009, vol. 28(3): pp. 5-8.

[3] Chapelle O, and Vapnik V. "Choosing multiple parameters for support vector machines [J]." Machine Learning, 2002, vol. 46(1): pp.131-160.

[4] Hong-gang Zhou, and chun-de Yang, "Based on Immune Algorithm and SVM anomaly detection method [J]." Computer application, 2006, vol. 26 (9): pp.2145-2147.

[5] Liang-min Li,Guang-rui Wen, and Sheng-chang Wang, "Based on genetic algorithm support vector regression parameters selection method [J]." Computer Engineering and Applications, 2008, vol. 44(7): pp.23-26.

[6] Jing-xu Liu, Huai-ping Cai, and Yue-jin Tan, “Heuristic Algorithm for Tuning Hyper parameters in Support Vector Regression.” Journal of System Simulation, 2007, vol. 17 No. 19.

[7] Ni Li-ping, Ni Zhi-wei, and Li Feng-gang, “SVM model selection based on ant colony algorithm[J].” Computer Technology and Develo pment,2007, vol.17(9): pp.95-98.

[8] Chien-Ming Huang, and Yuh-JyeLee, “Model selection for support vector machines via uniform design[J].” Computation al Statistics & Data Analysis (S0167-9473), 2007, vol. 52 (1): pp. 335 -346.

[9] Yuan Xiao-fang, and Wang Yao-nan. “Selection of SVM parameters using chaos optimization algorithms[J].” Control and Decision,2006, vol. 21(1): pp.111- 113,117.

[10] Liu Chang-ping, Fan Ming-yu, and Guang-wei Wang. “ Optimizing parameters of support vector machine based on gradient algorithm [J] . ” Control and Decision,2008, vol. 23(11): pp.1291-1295.

[11] Zhi-ming Wang, Xian-sheng Tan, Zhe-ming Yuan, and Chao-hua Wu, “Parameters Optimization of SVM Based on Self-calling SVR,” Journal of System Simulation,2010.2, vol. 22 No. 2.

[12] C. Cortes and V. Vapnik. Support-vector network. Machine Learning, 20:273-297, 1995.

[13] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin A Practical Guide to Support Vector Classification. Department of Computer Science National Taiwan University, Taipei 106, Taiwan.

[14] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation, 15(7):1667-1689, 2003.

[15] V. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, New York, NY, 1995.

[16] Guo-he Feng, “Support Vector Machine parameter selection method,” Computer Enginee- ring and Applications, 2011, vol. 47(3): pp. 123.

[17] Cherkassky V, Ma Y. Selection of Meta-parameters for Support Vector Regression [C]// Proceedings of the International Conference on Artificial Neural Networks. Madrid Spain: Springer. 2002.

[18] LIU Jingxu, CAI Huaiping, TAN Yuejin. Heuristic Algorithm for Tuning Hyperparameters in Support Vector Regression [J]. Journal of System Simulation (S1004-731X), 2007, 19(7): 1540-1543