CHEMICAL ENGINEERING TRANSACTIONS
VOL. 33, 2013
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at: www.aidic.it/cet
Guest Editors: Enrico Zio, Piero Baraldi

ISBN 978-88-95608-24-2; **ISSN** 1974-9791
DOI: 10.3303/CET1333088

Parameters Tuning in Support Vector Regression for Reliability Forecasting

Wei Zhao[a*], Tao Tao[a], Enrico Zio[b, c]

[a]Group 203, School of Electronic and Information Engineering, Beihang University, Beijing, 100191, China
[b]Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France, Ecole Centrale Paris and Supelec, Paris, France
[c]Dipartimento di Energia, Politecnico di Milano, Milano, Italy
[*]zhaowei203@buaa.edu.cn

The recent and promising machine learning technique called support vector machine (SVM) has become a hot research subject in time series forecasting, since proposed from Statistic Learning Theory by Vapnik. As an important application of time series forecasting, reliability prediction by analyzing the historical time series data of system condition to predict the future system behaviour and/or diagnose the possible system fault, has been solved successfully by SVM with high forecasting accuracy. For this, the critical problem is the selection of SVM parameters. Many methods have been proposed, such as genetic algorithm, particle swarm optimization and analytic selection; but there is no generally structured way, yet. In this paper, the capability of SVM to perform function fitting and reliability forecasting based on different methods is investigated by experimenting on both artificial and real-world data. A comparison of the methods is offered on criteria of prediction accuracy and robustness. Finally, an attempt is made to obtain a comparative optimal parameter selection method.

1. Introduction

Safe and reliable operation of engineering systems is very important. To guarantee this, reliability analysis and risk assessment offer sound technical frameworks for the study of component and system failures, with quantification of their probabilities and consequences (Zio, 2009). In this frameworks, one important goal is reliability prediction. Under certain conditions, reliability prediction can be seen as a time series prediction problem whose solution entails predicting the future values of reliability based on past data observations. A widely used prediction approach is the ARIMA model, with solid foundations in classical probability theory. However, the time-consuming off-line modelling efforts required for model identification and building limits its usefulness in practical applications (Lu et al., 2001). In recent years, neural network has emerged as a universal approximator for any nonlinear continuous function varying over a time or space domain, and has been applied successfully to various reliability problems such as software reliability prediction (Adnan and Yaacob, 1994) and complex system maintenance (Amjady and Ehsan, 1999). However, practical difficulties are encountered due to the need of large datasets for training, no guarantee of convergence to optimality and the danger of over-fitting (Chen, 2007, Sapankevych and Sankar, 2009).
Another powerful machine learning paradigm is the Support Vector Machine (SVM) developed by Vapnik and others in 1995 (Vapnik, 1995), based on statistics learning theory and VC theory. SVM embodies the idea of minimizing the Structure Risk Minimization (SRM) rather than the Empirical Risk Minimization (ERM) adopted in neural network training. Since the ERM principle is most suited for large training datasets, SVM has been proven to provide superior performances than neural networks on small datasets. For this reason, SVM has been applied to many machine learning tasks including time series prediction and reliability forecasting. For example, Hong applied the SVM method to predict engine reliability and compared the predicting performance with the Duane model, ARIMA model and general regression neural networks (Hong and Pai, 2006). Experiment results show that the SVM model has better performance over the other models.

Please cite this article as: Zhao W., Tao T., Zio E., 2013, Parameters tuning in support vector regression for reliability forecasting, Chemical Engineering Transactions, 33, 523-528 DOI: 10.3303/CET1333088

When applying SVM to regression and prediction problems, the performance depends heavily on the setting of the free meta-parameters of SVM. Then, how to select the parameters is a main issue for practitioners trying to apply SVM. The grid searching algorithms combined with k-fold cross validation are often used to find the best value set of the parameters. But the computational burden can be heavy, which renders this exhaustive method little practical. A simple but practical analytical selection approach (AS) can provide the basic form of the parameters (Cherkassky and Ma, 2004); advanced optimization algorithms such as simulated annealing (SA) (Pai and Hong, 2006), genetic algorithm (GA) (Chen, 2007) and particle swarm optimization (PSO) (Lins et al., 2011) have also been used for SVM parameters tuning. In this paper, we investigate the capability of SVM parameters tuning by AS, GA and PSO for function regression and reliability prediction. The investigation is carried out by way of some experiments on both artificial and real world data.

The remainder of the paper is organized as follows. Section 2 introduces background knowledge about SVR and the basic theory of AS, GA and PSO is presented in Section 3. Section 4 presents the experiments on artificial and real-world datasets through which the regression performances of the three methods are compared. Section 5 provides some discussions and conclusions on the experiment results.

2. Support vector machines for regression

Given a dataset $D = \{(\boldsymbol{s}_i, y_i\}_i^n$, where $\boldsymbol{s}_i \in R^l$ denotes the *l*-dimension input vector, y_i denotes the real-valued output and n is the number of data patterns,, we consider, first, an SVM to estimate the linear regression function:

$$f(\boldsymbol{s}_i) = \boldsymbol{w}^T \boldsymbol{s}_i + b \tag{1}$$

where $\boldsymbol{w}$ and b are respectively the weight vector and intercept of the model that one needs to find for optimal fitting of the data in D .

In the nonlinear case, by a nonlinear mapping $\Phi : R^l \to F$, where F is the feature space of Φ , the SVM transforms the complex nonlinear regression problem into the comparatively simple problem of finding the flattest function in the feature space F (Chen, 2007). Then, the regression function takes a general form suitable for both linear and nonlinear cases:

$$f(\boldsymbol{s}_i) = \boldsymbol{w}^T \Phi(\boldsymbol{s}_i) + b \tag{2}$$

Then, we introduce the ε -insensitive loss function (Vapnik, 1995):

$$l = |\, y_i - f(\boldsymbol{s}_i)\,|_{\varepsilon} = \begin{cases} 0, & |\, y_i - f(\boldsymbol{s}_i)\,| \le \varepsilon \\ |\, y_i - f(\boldsymbol{s}_i)\,| - \varepsilon, & otherwise \end{cases} \tag{3}$$

which ignores the error if the difference between the prediction value obtained by Eq.(2) and the real value is smaller than ε, which is a parameter to be tuned. For the error larger than ε , slack variables ξ, ξ^* are introduced to respectively represent the functional distance of two possible but mutually exclusive samples. By introducing the ε -insensitive loss function, we can measure the empirical error and set up a procedure for minimizing it. Besides, in SVM we must also minimize the Euclidean norm of the linear weight $\boldsymbol{w}$, $\|\boldsymbol{w}\|$, which is related to the generalisation ability of the SVM model trained. Then, a compromised optimal quadratic optimization problem to identify the regression model arises as follows:

$$\begin{aligned} &\min_{\boldsymbol{w},\xi,\xi^*} J(\boldsymbol{w},\xi,\xi^*) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\xi + \xi^*) \\ &s.t. \quad \begin{cases} y_i - \boldsymbol{w}^T\Phi(\boldsymbol{s}) - b \le \varepsilon + \xi_i \\ \boldsymbol{w}^T\Phi(\boldsymbol{s}) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \qquad i = 1,...,n \end{aligned} \tag{4}$$

where C denotes the penalty coefficient that modulates the trade-off between empirical and generalization errors, and must be also tuned by the analyst. The solution of this quadratic optimization problem obtained by the Lagrangian dual method gives the optimal $\boldsymbol{w}$ and b through which we can estimate the prediction value numerically:

$$f(\boldsymbol{s}) = \langle \boldsymbol{w} \bullet \Phi(\boldsymbol{s}) \rangle + b = \sum_{i=1}^{n} \alpha_i K(s, s_i) + b$$
$$K(\boldsymbol{s}_i, \boldsymbol{s}_j) = \Phi(\boldsymbol{s}_i)^T \Phi(\boldsymbol{s}_j) \quad (5)$$

where $K(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is the kernel function satisfying the Mercer condition (Boser et al., 1992). If not mentioned specifically, the kernel function used in this paper is the radial basis function with width γ also to be tuned by the analyst.

3. Parameter selection methods

3.1 AS method

The analytic selection (AS) method chooses the parameter triplet, $\boldsymbol{X} = [C, \varepsilon, \gamma]$, directly from the training data and (estimated) noise level analytically as follows (Cherkassky and Ma, 2004):

$$C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|), \quad \varepsilon = 3\sigma\sqrt{\frac{\ln n}{n}}, \quad \gamma \sim (0.1 - 0.5) \times range(\boldsymbol{s}) \quad (6)$$

where $\bar{y}$ and σ_y are the mean and the standard deviation of the y values, $range(\boldsymbol{s}) = |\max(\boldsymbol{s}) - \min(\boldsymbol{s})|$, σ is the estimated noise level of the training data obtained by the following prescription via the *k*-nearest-neighbour's method:

$$\sigma = \frac{n^{1/5}k}{n^{1/5}k - 1} \cdot \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (7)$$

where $\hat{y}_i$ is the regression value via *k*-nearest-neighbour's method.

3.2 GA method

Genetic algorithms (GA) are a family of evolutionary computational models inspired by the theory of evolution. These algorithms encode each potential solution of the optimization problem in a simple chromosome-like data structure, and then sift the critical information via some recombination operators that imitate biological evolution processes such as survival of the fittest, crossover and mutation (Whitley, 1994). The basic procedure of GA method adopted in our work is described as follows (Chen, 2007):

1) Representation: Chromosome $\boldsymbol{X}$ is directly represented as a SVM parameter vector $\boldsymbol{X} = [C, \varepsilon, \gamma]$.

2) Fitness: The fitness value evaluating the quality of chromosome $\boldsymbol{X}$ is defined as the mean square error of the 5-fold cross validation (MSE_{CV}) method on the training data with SVM parameters $\boldsymbol{X}$.

3) Initialization and selection: In this study, the initial population is composed of 40 chromosomes randomly generated within the given ranges ov variability of the three parameters to be tuned and the standard roulette wheel method is employed to select survival chromosomes from the current population, in proportion to their fitness values.

4) Crossover and mutation: As the core operation of GA, crossover and mutation play a fundamental role in the progress of searching the best chromosome. In our study, the simulated binary crossover and polynomial mutation methods are chosen to realise the according operations. The probability of crossover p_c and of mutation p_m are respectively set to 0.8 and 0.05.

5) Elitist strategy: The chromosome with the best fitness will skip the crossover and mutation procedure and directly survive until the next generation.

6) Stopping criteria: steps 3-5 are repeated for a predefined number of generations (in our application this is set to 100).

3.3 PSO method

Particle swarm optimization (PSO) is a population-based meta-heuristics that simulates social behaviour such as birds flocking to a promising position (Lin et al., 2008). PSO performs searches through a population (called swarm) of individual solutions (called particles) that update iteratively. Each particle at iteration *t* can be represented by a *D*-dimensional state vector as $\boldsymbol{X}_i^t = \{X_{i1}^t, X_{i2}^t, ..., X_{iD}^t\}$. Then, to obtain the optimal solution, we define *D*-dimensional velocity vectors $\boldsymbol{V}_i^t = \{V_{i1}^t, V_{i2}^t, ..., V_{iD}^t\}$ for each particle and determined by its own best previous experience (*pbest*) and the best experience of all other particles

($gbest$). Particles change velocity according to the $pbest$ and $gbest$ as follows:

$$\boldsymbol{V}_{id}^{t} = \boldsymbol{V}_{id}^{t-1} + c_1 r_1 (pbest_{id}^{t} - \boldsymbol{X}_{id}^{t}) + c_2 r_2 (gbest_{id}^{t} - \boldsymbol{X}_{id}^{t}), \quad d = 1,2,...,D \tag{8}$$

where c_1 , c_2 are the learning factors set to 2 in this study and r_1, r_2 are random numbers distributed uniformly in the range (0, 1), i.e. U(0,1). Then, each particle updates to a new potential solution based on the velocity as:

$$\boldsymbol{X}_{id}^{t+1} = \boldsymbol{X}_{id}^{t} + \boldsymbol{V}_{id}^{t}, \qquad d = 1,2,...,D \tag{9}$$

When the iteration number reaches a pre-determined maximum iteration number, the update process is terminated and the best individual of the last generation is the final solution to the target problem.

4. Experiments results

In this Section, we perform some simulated experiments to investigate the capability of these three methods for optimal searching the SVM parameters. We consider function regression problems which are not directly related to the reliability prediction problem of interest but hold similar characteristics while, on the other hand, being easily implemented and controllable. Through these regression cases, we can systematically compare the prediction performance of the three methods for optimal SVM parameter identification in terms of accuracy, stability and sensitivity to noise. The findings of these experiment studies will guide the choices of the settings of the algorithms for the reliability prediction case of interest.

4.1 Function regression

First, we consider the *sinc* function $f(\boldsymbol{s})$ (Borwein et al., 2010)

$$f(\boldsymbol{s}) = 10\sin(\boldsymbol{s}) / \boldsymbol{s} \qquad \boldsymbol{s} \in [-10,10] \tag{10}$$

The simulated training data are *n* pairs $(\boldsymbol{s}_i, y_i)$, $(i = 1,...,n)$, where $\boldsymbol{s}_i$ are random–uniformly sampled in the pre-defined range and y_i are generated as $y = f(\boldsymbol{s}) + \sigma$. We first consider the case with noise level $\sigma = 2$,and *n*=40. The test data are also random-uniformly sampled in the same range as the training data.

Figure 1, visually shows that all the three parameter selection methods are capable of approximating the target function. GA and PSO methods yield better generalisation performance, at the cost of a much heavier computational burden than the simpler AS method.

To compare in an integrated manner the three methods of SVM parameters tuning, we evaluate the prediction risk, defined as the mean squared error (MSE) between the SVM estimates and the corresponding true values of the target function output for the test input values. For this, and to account for the randomness of the estimation process, we perform the regression seven times for a same target function value. Figure 2 confirms the overall superiority of GA and PSO. One can also notice the fluctuations in GA performance, which has worse stability than the PSO method which consistently gives a high prediction accuracy.

Table 1 gives the results of experiments for different target function types and noise levels. In general the PSO and GA methods perform better than the AS method. Further, the mean value and standard deviation of GA method tend to become large as the noise level increases. This shows the GA method's instability and sensitivity to noise. On the contrary, for all function types and noise level considered, PSO method performs satisfactorily in both mean value and standard deviation, which means a superiority of PSO method both in generalisation performance and stability.

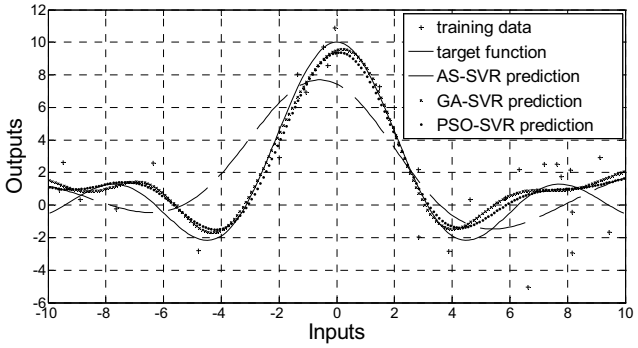

Figure 1: Comparison of SVM estimates for the case of the sinc function with $\sigma = 2$

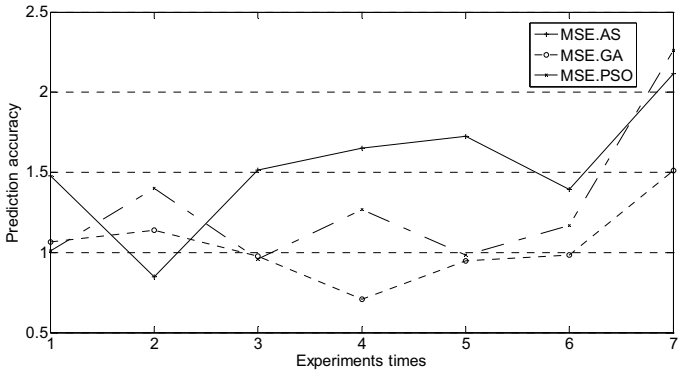

Figure 2: Estimate MSE for seven times for sinc function with $\sigma = 2$

Table 1: MSE for different function types with different noise levels

Noise level	Methods	MSE							Mean	Standard variance
Target function : $y = s$										
$\sigma = 1$	AS	0.2209	1.2387	0.6475	1.3284	0.13303	0.4534	0.1034	0.589333	1.569845
	GA	0.2163	0.9306	0.7536	0.9631	0.1657	0.3780	0.2173	0.5178	0.749019
	PSO	0.1602	0.8115	0.4454	0.7543	0.2547	0.3529	0.2810	0.437143	0.38226
$\sigma = 2$	AS	0.83162	1.9473	2.6628	1.9854	2.5502	2.2687	0.0935	1.762789	5.423183
	GA	4.3132	1.8693	2.5218	0.2298	6.2979	7.1492	2.9968	3.625429	36.25955
	PSO	0.3101	0.1397	1.2607	0.0648	0.4941	0.2232	0.0097	0.357471	1.108788
Target function : $y = s^2 + s + 1$										
$\sigma = 5$	AS	7.6928	49.5629	10.8916	17.3639	12.9373	22.2204	42.8112	23.3543	1611.748
	GA	11.3244	11.6291	4.4968	9.2407	18.4272	12.5531	31.2114	14.1261	443.5563
	PSO	8.5105	7.3955	2.2536	2.7347	14.5867	10.8193	10.6673	8.138229	119.6843
$\sigma = 10$	AS	72.7756	37.8415	23.4862	39.4547	66.1863	129.980	55.4059	60.73289	7362.399
	GA	70.78275	18.6030	12.7345	16.2283	128.013	64.0742	5.4549	45.12724	12049.11
	PSO	81.6381	27.4305	15.6488	11.9962	42.3675	61.2897	8.5596	35.56149	4578.37
Target function : $y = \sin(s)$										
$\sigma = 0.5$	AS	0.3734	0.3047	0.3123	0.3381	0.4152	0.3976	0.3241	0.3522	0.011316
	GA	0.08427	0.08595	0.1429	0.0672	0.1814	0.1289	0.0849	0.110789	0.010236
	PSO	0.1190	0.09044	0.1620	0.08235	0.1557	0.1307	0.0847	0.117841	0.006659
$\sigma = 0.25$	AS	0.3274	0.1761	0.2637	0.4132	0.1814	0.2005	0.2142	0.253786	0.046611
	GA	0.0723	0.0248	0.0125	0.1101	0.0338	0.0308	0.0640	0.049757	0.006977
	PSO	0.0379	0.0168	0.00856	0.0675	0.0490	0.0227	0.0671	0.038509	0.003387

4.2 Reliability prediction

In this Section, a reliability prediction experiment concerning submarine failure data is carried out. The data set contains 70 submarine failure times that increase approximately linearly as time goes by except

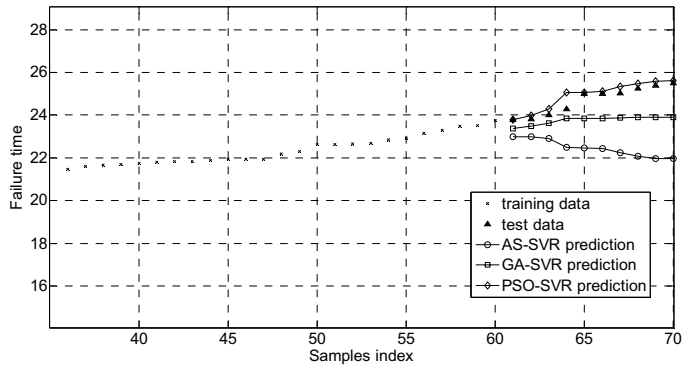

Figure 3: Reliability results for submarine failure data using AS, GA and PSO method.

Table 2: Estimate MSE for reliability predictions in section 4.2

Methods	MSE							Mean	Standard deviation
AS	6.0943	6.0943	6.0943	6.0943	6.0943	6.0943	6.0943	6.0943	0
GA	1.0936	1.2708	5.3472	5.9729	1.2247	0.3772	0.3118	2.2283	2.2059
PSO	0.3118	0.3126	0.3123	0.3124	0.3171	0.3119	0.3118	0.3128	0.0018

for a hopping in correspondence of time index 64. Prediction is done by a one-step ahead strategy for predicting the next (($t+1$)-th) failure time based on the current (t-th) failure time In this experiment, the first 60 time-to-failure data are used as training set and the final 10 data as test set. Because it is difficult to get good estimates of the noise in the training data in practical reliability prediction applications, the AS method, which relies heavily on the noise level estimates, shows bad performance in tracking the trend of t he reliability data. Instead, as Figure 3 shows, the GA and PSO methods are both capable of capturing the trend of the failure data. Even for the "hopping" data, PSO provides a satisfactory prediction performance, whereas GA gives poorer predictions because of weaker generalization ability. In reliability prediction case, The information reported in Table 2 confirm the instability of the GA method and superiority of the PSO method .

5. Conclusion

In this work, we have investigated the AS, GA and PSO parameter methods for selecting the parameters of SVM in regression and prediction tasks. Our experiments results suggest that PSO gives superior performances, whereas AS gives comparatively low accuracy and GA is somewhat unstable.
Although the performance of AS is not comparatively satisfactory, its extremely low computational burden makes it attractive for initializing the parameter values for the GA and PSO methods and optimizing their search evolution process to accelerate it and stabilize it: how to embed this into a dynamic online method is a future research issue.

Reference

Adnan W.,Yaacob M. An integrated neural-fuzzy system of software reliability prediction. Software Testing, Reliability and Quality Assurance, 1994. Conference Proceedings., First International Conference on, 21-22 Dec. 1994 1994. IEEE, 154-158.

Amjady N.,Ehsan M., 1999, Evaluation of power systems reliability by an artificial neural network. Power Systems, IEEE Transactions on, 14, 287-292.

Borwein D., Borwein J.M.,Leonard I.E., 2010, L_p Norms and the Sinc Function. The American Mathematical Monthly, 117, 528-539.

Boser B.E., Guyon I.M.,Vapnik V.N. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory, 1992. ACM, 144-152.

Chen K.Y., 2007, Forecasting systems reliability based on support vector regression with genetic algorithms. Reliability Engineering & System Safety, 92, 423-432.

Cherkassky V.,Ma Y., 2004, Practical selection of SVM parameters and noise estimation for SVM regression. Neural networks, 17, 113-126.

Hong W.C.,Pai P.F., 2006, Predicting engine reliability by support vector machines. The International Journal of Advanced Manufacturing Technology, 28, 154-161.

Lin S.W., Ying K.C., Chen S.C.,Lee Z.J., 2008, Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications, 35, 1817-1824.

Lins I.D., Moura M.C., Zio E.,Droguett E.L., 2011, A particle swarm-optimized support vector machine for reliability prediction. Quality and Reliability Engineering International, 28, 141-158.

Lu H., Kolarik W.J.,Lu S.S., 2001, Real-time performance reliability prediction. Reliability, IEEE Transactions on, 50, 353-357.

Pai P.F.,Hong W.C., 2006, Software reliability forecasting by support vector machines with simulated annealing algorithms. Journal of Systems and Software, 79, 747-755.

Sapankevych N.,Sankar R., 2009, Time series prediction using support vector machines: a survey. Computational Intelligence Magazine, IEEE, 4, 24-38.

Vapnik V. 1995. The nature of statistical learning theory, springer-verlag New York Inc, New York, USA.

Whitley D., 1994, A genetic algorithm tutorial. Statistics and computing, 4, 65-85.

Zio E., 2009, Reliability engineering: Old problems and new challenges. Reliability Engineering & System Safety, 94, 125-141.