Comparison of SVM and LS-SVM for Regression

Haifeng Wang Dejin Hu
School of Mechanical and Power Engineering
Shanghai Jiao Tong University
Shanghai 200030
E-mail: whfdx@sohu.com

Abstract—**Support Vector Machines (SVM) has been widely used in classification and nonlinear function estimation. However, the major drawback of SVM is its higher computational burden for the constrained optimization programming. This disadvantage has been overcome by Least Squares Support Vector Machines (LS-SVM), which solves linear equations instead of a quadratic programming problem. This paper compares LS-SVM with SVM for regression. According to the parallel test results, conclusions can be made that LS-SVM is preferred especially for large scale problem, because its solution procedure is high efficiency and after pruning both sparseness and performance of LS-SVM are comparable with those of SVM.**

I. INTRODUCTION

Support vector machines (SVM) was firstly introduced for classification and nonlinear function estimation [1] and further investigated by many others [2]. The support vector method for regression is formulated in solving a convex optimization problem, more specifically a quadratic programming(QP) problem. This is obtained by employing the Vapnik's ε-insensitive loss function, formulating the approximation problem as an inequality constrained optimization problem and exploiting the Mercer condition in order to relate the nonlinear feature space mapping to the chosen kernel function. Moreover, the model complexity follows from solving this convex optimization problem. SVM models also scale to high dimensional input spaces very well [3][4].

However, the major drawback of SVM is its higher computational burden because of the required constrained optimization programming. Major breakthrough has been obtained at this point with a least squares version of SVM, called LS-SVM. In LS-SVM, one works with equality instead of inequality constraints and a sum squared error (SSE) cost function as it is frequently used in training of classical neural networks. This reformulation greatly simplifies the problem in such a way that the solution is characterized by a linear system, more precisely a Karush-Kuhn-Tucker (KKT) system, which takes a similar form as the linear system that one solves in every iteration step by interior point methods for standard SVM. This linear system can be efficiently solved by iterative methods such as conjugate gradient [5].

The aim of this paper is to compare the performance of LS-SVM with SVM for regression. This paper is organized as follows. In Section 2, the basic idea of SVM for nonlinear function estimation is explained. And some notions of LS-SVM for function estimation are expatiated in Section 3. The comparative tests are carried out and experimental results are discussed in the following Section. Finally, some conclusions are drawn.

II. SVM FOR NONLINEAR FUNCTION ESTIMATION [1][6]

Considering the problem of approximating the dataset

$$D=\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_k,y_k),\ldots,(\mathbf{x}_N,y_N)\},\ \mathbf{x}_k\in R^n, y_k\in R \quad (1)$$

with a nonlinear function

$$f(\mathbf{x})=\langle\boldsymbol{\omega},\varphi(\mathbf{x})\rangle+b \quad (2)$$

where $\langle\cdot,\cdot\rangle$ denotes the dot product; $\boldsymbol{\omega}\in R^{n_h}$ is the weight vector in primal weight space; $\varphi(\cdot)$: $R^n\rightarrow R^{n_h}$ is the nonlinear function that maps the input space to a so-called high dimensional feature space where linear regression is performed; b is the bias term. The dimension n_h of this space is implicitly defined, which means that it can be infinite dimensional.

The optimization problem is given

$$\min\frac{1}{2}\|\boldsymbol{\omega}\|^2+C\sum_{k=1}^{N}(\xi_k+\xi_k^*)$$
$$s.t.\begin{cases}y_k-\langle\boldsymbol{\omega},\varphi(\mathbf{x}_k)\rangle-b\le\varepsilon+\xi_k\\ \langle\boldsymbol{\omega},\varphi(\mathbf{x}_k)\rangle+b-y_k\le\varepsilon+\xi_k^*\\ \xi_k,\xi_k^*\ge 0\end{cases} \quad (3)$$

with the ε-insensitive loss function

$$|y-f(\mathbf{x},\boldsymbol{\omega})|_\varepsilon=\begin{cases}0, & \text{if}\,|y-f(\mathbf{x},\boldsymbol{\omega})|\le\varepsilon\\ |y-f(\mathbf{x},\boldsymbol{\omega})|-\varepsilon, & \text{otherwise}\end{cases} \quad (4)$$

where ε is the approximation accuracy that can be violated by means of the slack variables ξ,ξ^* for the non-feasible case. Constant $C>0$ determines trade-off between flatness of f and the amount up to which deviations larger than ε are tolerated. A smaller value of C tolerated a larger deviation.

The Lagrangian function is given by equation (5), where $\alpha,\alpha^*,\eta,\eta^*\ge 0$ are Lagrange multipliers. To find the saddle point, one obtains the partial derivates of L_{SVM} with respect

0-7803-9422-4/05/$20.00 ©2005 IEEE

to the primal variables ($\boldsymbol{\omega}, b, \xi, \xi^*$) by equation (6).

$$\begin{aligned} L_{SVM} = & \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C\sum_{k=1}^{N}(\xi_k + \xi_k^*) - \\ & \sum_{k=1}^{N}\alpha_k(\varepsilon + \xi_k - y_k + \langle \boldsymbol{\omega}, \varphi(\mathbf{x}_k)\rangle + b\,) - \\ & \sum_{k=1}^{N}\alpha_k^*(\varepsilon + \xi_k^* + y_k - \langle \boldsymbol{\omega}, \varphi(\mathbf{x}_k)\rangle - b) - \\ & \sum_{k=1}^{N}(\eta_k\xi_k + \eta_k^*\xi_k^*) \end{aligned} \tag{5}$$

$$\begin{cases} \dfrac{\partial L_{SVM}}{\partial \boldsymbol{\omega}} = 0 \;\rightarrow\; \boldsymbol{\omega} = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)\varphi(\mathbf{x}_k) \\ \dfrac{\partial L_{SVM}}{\partial b} = 0 \;\rightarrow\; \sum_{k=1}^{N}(\alpha_k^* - \alpha_k) = 0 \\ \dfrac{\partial L_{SVM}}{\partial \xi_k} = 0 \;\rightarrow\; C - \alpha_k - \eta_k = 0 \\ \dfrac{\partial L_{SVM}}{\partial \xi_k^*} = 0 \;\rightarrow\; C - \alpha_k^* - \eta_k^* = 0 \end{cases} \tag{6}$$

The conditions for optimality yield the following dual problem:

$$\begin{aligned} \max_{\alpha,\alpha^*} Q = & -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\langle\varphi(\mathbf{x}_k), \varphi(\mathbf{x}_l)\rangle \\ & -\varepsilon\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \end{aligned} \tag{7}$$

$$\text{s.t.} \begin{cases} \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ \alpha_k, \alpha_k^* \in [0, C] \end{cases}$$

The parameter b can be computed by exploiting the so-called KKT conditions, which state that at the optimal solution the product between dual variables and constrains has to vanish. In the SVM case this means (8) and (9):

$$\begin{cases} \alpha_k[\varepsilon + \xi_k - y_k + \langle\boldsymbol{\omega}, \varphi(\mathbf{x}_k)\rangle + b] = 0 \\ \alpha_k^*[\varepsilon + \xi_k^* + y_k - \langle\boldsymbol{\omega}, \varphi(\mathbf{x}_k)\rangle - b] = 0 \end{cases} \tag{8}$$

$$\begin{cases} \eta_k\xi_k = (C - \alpha_k)\xi_k = 0 \\ \eta_k^*\xi_k^* = (C - \alpha_k^*)\xi_k^* = 0 \end{cases} \tag{9}$$

From (8) it follows that only for $|f(\mathbf{x}_k) - y_k| \geq \varepsilon$ the Lagrange multipliers may be nonzero, or in other words, for all samples inside the ε-tube (i.e. the shaded region in Fig.1) the α_k, α_k^* vanish. So one gets a sparse expansion of $\boldsymbol{\omega}$, and samples that come with non-vanishing coefficients are called *Support Vectors* (SVs).

From equation (8) and (9), some conclusions can be drawn: Firstly only samples $(\mathbf{x}_k, y_k)$ with corresponding $\alpha_k^{(*)} = C$ lie outside the ε-insensitive tube around *f*, which are called *Boundary Support Vectors* (BSVs)[7]. Secondly $\alpha_k\alpha_k^* = 0$, i.e. there can never be a set of dual variables α_k, α_k^* which are both simultaneously nonzero as this would require nonzero slacks in both directions. Finally for support vectors corresponding to $\alpha_k^{(*)} \in (0, C)$, $\xi_k^{(*)} = 0$. And these SVs are called *Normal Support Vectors* (NSVs). Hence b can be computed as follows:

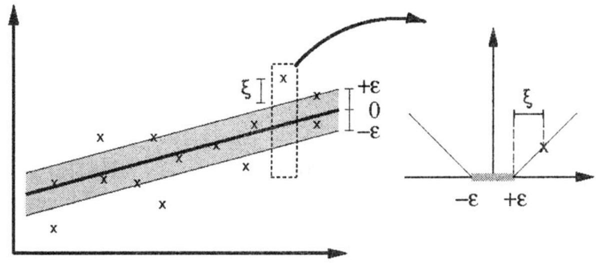

Fig. 1. ε-insensitive loss function and corresponding ε-tube for a linear regression

$$\begin{aligned} b &= y_k - \langle\boldsymbol{\omega}, \varphi(\mathbf{x}_k)\rangle - \varepsilon \quad \text{for } \alpha_k \in (0, C) \\ b &= y_k - \langle\boldsymbol{\omega}, \varphi(\mathbf{x}_k)\rangle + \varepsilon \quad \text{for } \alpha_k^* \in (0, C) \end{aligned} \tag{10}$$

In the end, the resulting SVM for nonlinear function estimation takes the form:

$$f(\mathbf{x}) = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)\langle\varphi(\mathbf{x}), \varphi(\mathbf{x}_k)\rangle + b \tag{11}$$

According to Mercer's condition, the inner product $\langle\varphi(\mathbf{x}), \varphi(\mathbf{x}_k)\rangle$ can be defined through a kernel $K(\mathbf{x}, \mathbf{x}_k)$, so the equation (11) can be expressed as

$$f(\mathbf{x}) = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)K(\mathbf{x}, \mathbf{x}_k) + b \tag{12}$$

Several choices for the kernel $K(\cdot,\cdot)$ are possible:

(1) Linear kernel: $K(\mathbf{x}, \mathbf{x}_k) = \langle\mathbf{x}, \mathbf{x}_k\rangle$

(2) Polynomial kernel:

$$K(\mathbf{x}, \mathbf{x}_k) = (\langle\mathbf{x}, \mathbf{x}_k\rangle + p)^d, \quad d \in N, p > 0$$

(3) Multi-Layer Perceptron kernel:

$$K(\mathbf{x}, \mathbf{x}_k) = \tanh(\phi\langle\mathbf{x}, \mathbf{x}_k\rangle + \theta), \quad \phi > 0, \theta > 0$$

(4) Gaussian Radial Basis Function kernel:

$$K(\mathbf{x}, \mathbf{x}_k) = \exp(-\|\mathbf{x} - \mathbf{x}_k\|^2 / 2\sigma^2)$$

III. LS-SVM FOR NONLINEAR FUNCTION ESTIMATION [3][4][5][8]

In least squares support vector machines for function estimation, the optimization problem is formulated

$$\begin{aligned} & \min_{\boldsymbol{\omega},b,e} J(\boldsymbol{\omega}, e) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{1}{2}\gamma\sum_{k=1}^{N} e_k^2 \\ & \text{s.t.} \;\; y_k = \langle\boldsymbol{\omega}, \varphi(\mathbf{x}_k)\rangle + b + e_k, \;\; k = 1, \ldots, N \end{aligned} \tag{13}$$

where $e_k \in R$ are error variables; $\gamma \geq 0$ is a regularization constant. Smaller γ can avoid overfitting in case of noisy data. Note that there are a SSE fitting error and a regularization term in the cost function, which is also a standard procedure in the training of feedforward neural networks and is related to ridge regression.

The Lagrangian is given by

$$L_{LS-SVM} = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{1}{2}\gamma\sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N}\alpha_k\left\{\langle\boldsymbol{\omega},\varphi(\mathbf{x}_k)\rangle + b + e_k - y_k\right\} \tag{14}$$

with Lagrange multipliers $\alpha_k \in R$. The conditions for optimality are given by

$$\begin{cases} \dfrac{\partial L_{LS-SVM}}{\partial \boldsymbol{\omega}} = 0 \rightarrow \boldsymbol{\omega} = \sum_{k=1}^{N}\alpha_k\varphi(\mathbf{x}_k) \\ \dfrac{\partial L_{LS-SVM}}{\partial b} = 0 \rightarrow \sum_{k=1}^{N}\alpha_k = 0 \\ \dfrac{\partial L_{LS-SVM}}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma\, e_k, \ (k = 1,\ldots,N) \\ \dfrac{\partial L_{LS-SVM}}{\partial \alpha_k} = 0 \rightarrow \langle\boldsymbol{\omega},\varphi(\mathbf{x}_k)\rangle + b + e_k - y_k = 0 \end{cases} \tag{15}$$

These conditions are similar to standard SVM optimality conditions in (6), except for the condition $\alpha_k = \gamma\, e_k$, for which the sparseness property has been lost in LS-SVM (See Fig. 2).

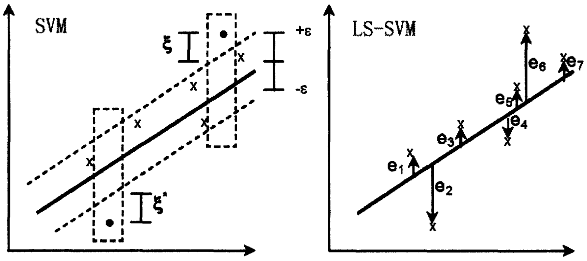

Fig. 2. Sparseness contrast between SVM and LS-SVM for a linear regression

After elimination of $\boldsymbol{\omega}, e$ one obtains the following linear equations

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \boldsymbol{\Omega} + \mathbf{I}/\gamma \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{16}$$

where $y = [y_1;\ldots;y_N]$, $1_v = [1;\ldots;1]$, $\alpha = [\alpha_1;\ldots;\alpha_N]$ and the Mercer condition has been applied again

$$\Omega_{kl} = \langle\varphi(\mathbf{x}_k),\varphi(\mathbf{x}_l)\rangle = K(\mathbf{x}_k,\mathbf{x}_l) \quad k,l = 1,\ldots,N \tag{17}$$

Although the choices of the kernel function $K(\cdot,\cdot)$ in LS-SVM are the same as those in SVM, more emphasis has been put on the powerful RBF kernel.

The resulting LS-SVM model for function estimation becomes

$$f(\mathbf{x}) = \sum_{k=1}^{N}\alpha_k K(\mathbf{x},\mathbf{x}_k) + b \tag{18}$$

where α, b are the solution to (16).

Compare (13) with (3), one can see that LS-SVM is a reformulation of the principles of SVM, which involves equality instead of inequality constraints. Furthermore, LS-SVM uses the least squares loss function instead of the ε-insensitive loss function. In this way, the solution follows from a linear KKT system instead of a computationally hard QP problem. Therefore it is easier to optimize and the computing time is short. At the same time, the dual problem of LS-SVM corresponds to solve a linear KKT system which is a square system with a global and possibly unique solution if the matrix has full rank.

Nevertheless, despite these computationally attractive features, LS-SVM solution also has a major drawback, i.e. lack of sparseness. This point can be illustrated by Fig. 2 with a linear regression. The left figure presents SVM with a ε tube and some slack variables $\xi_k^{(*)}$ which correspond to two support vectors. While in LS-SVM the ε tube and slack variables are replaced by error variables $e_k \in \{e_1,\ldots e_7\}$, which give the distances from each point to the regression function. And all samples in LS-SVM are support vectors, that is to say, all training data are used to produce the result.

Sparseness can also be imposed on LS-SVM with some pruning methods. The simplest way is to plot the spectrum of the sorted $|\alpha_k|$ values, from which the least significant sample for contribution to LS-SVM can be evaluated. After omitting the least important data from the training date set and re-estimating the LS-SVM model, sparseness can also be obtained. The pruning method looks as follows:

(1) Train LS-SVM based on N points.

(2) Remove a small amount of points (e.g. 5% of the set) with smallest values in the $|\alpha_k|$ spectrum.

(3) Re-train the LS-SVM based on the reduced training set.

(4) Go to (2), unless the user-defined performance index degrades. If the performance becomes worse, one checks whether an additional modification of γ, σ might improve the performance.

Note that omitting data points implicitly corresponds to creating an ε-insensitive tube in the underlying cost function which leads to sparseness. Links between standard SVM and LS-SVM can be established for any convex cost function though interior point algorithms, being aware of the fact that in every iteration step one solves a KKT system of the same form as an LS-SVM.

IV. EXPERIMENTS

In the experiments, SVM and LS-SVM methods with the RBF kernel function were firstly used to approximate the following nonlinear sinc function:

$$Y = \sin c(X) \tag{19}$$

Where X was the input taken in the region [-6, 6] and Y was the output. The noise with standard deviation 0.1 was imposed and a training dataset with 240 samples was given.

The RBF kernel parameter and other tuning-parameters were obtained from the 10-fold cross-validation, listed in table 1.

Table 1 Optimized parameters of SVM and LS-SVM

Method	RBF kernel parameter	other tuning-parameters
SVM	σ^2=0.25	C=1, ε=0.1
LS-SVM	σ^2=0.1	γ=100

In Fig.3 a comparison is made between the performance of SVM and LS-SVM. The thin solid line indicates the original nonlinear sinc function. The gray thick solid line shows function estimation by SVM with sparse circled support vectors and dashed ε tube. The dark solid line denotes function estimation by LS-SVM. From Fig.3 one can see that LS-SVM gains an advantage of SVM in nonlinear function estimation with noise pollution. Similar conclusion can be made from performances expressed in Mean Squared Error (MSE): MSE of SVM is 0.0099109, and that of LS-SVM is 0.0098799. The reason is that LS-SVM can be optimized more precisely, because it has a short computing time. Furthermore, LS-SVM uses all samples to find a good approximation model, while SVM only selects some sparse support vectors to model.

Sparseness can be expressed in percentage of support vectors to all training data. For SVM, 24 NSVs and 62 BSVs are obtained, so the sparseness of SVM is 36%. The corresponding alpha spectrum is illustrated in Fig.4, from which one also can see that only a small part of alphas are nonzero. The alphas whose values are between 0 and 1 correspond to NSVs, while those values equal to 1 are BSVs. In contrast to this, the sparseness is lost in the LS-SVM solution and almost all alphas are nonzero (see Fig.5).

With the pruning method introduced in the previous section, the identical support vectors can be obtained in LS-SVM. At this point, the MSE was 0.01082654, and the performance did not become worse observably. The alpha spectrum of pruned LS-SVM is illustrated in Fig.6, where the analogous tube is formed between -20 and 20. This similarity can be comprehended based on the fact that in both cases each sample gets an α value that indicates the relative importance of each training data to the performance of the approximation model. In SVM, an α value of zero means that the sample is not important for the model training and can be removed. Hence, a zero α value in SVM corresponds to a small α value in LS-SVM and a nonzero α value in SVM to a high α value in LS-SVM.

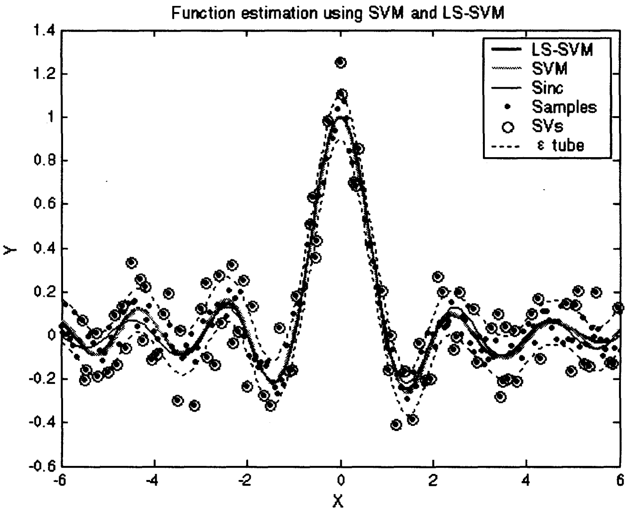

Fig.3. Function estimation with artificial sinc dataset

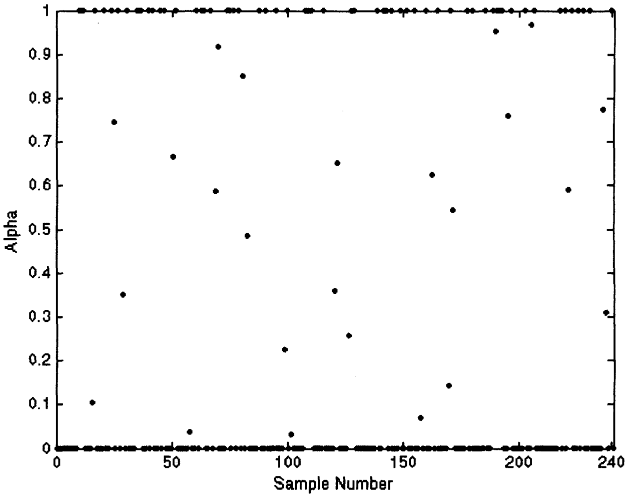

Fig. 4. Alpha spectrum of SVM

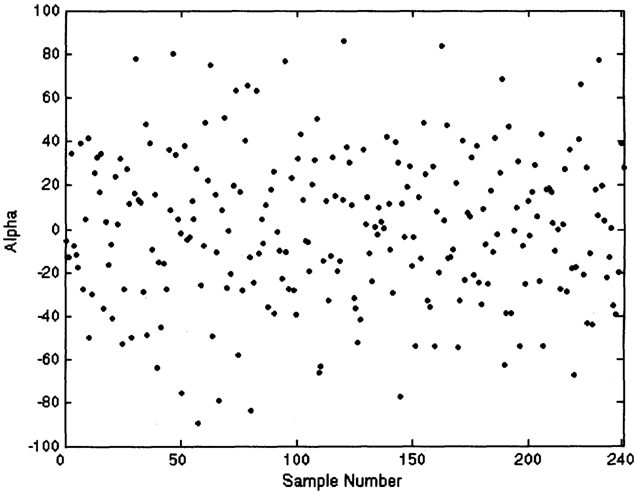

Fig. 5. Alpha spectrum of LS-SVM

Fig.7 illustrates the other comparative result between SVM and LS-SVM based on the motorcycle dataset, a well-

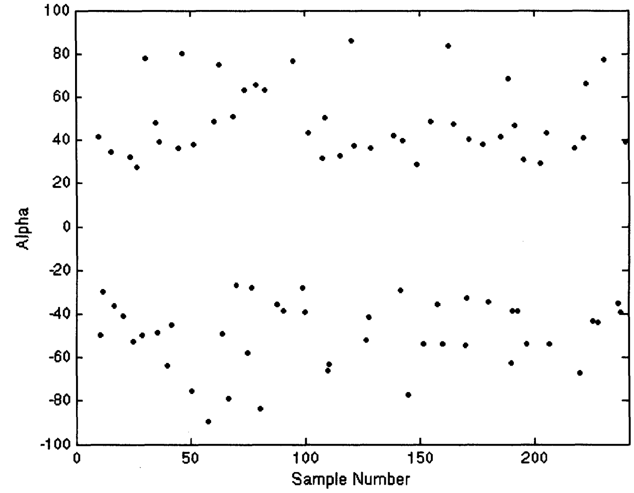

Fig. 6. Alpha spectrum of pruned LS-SVM

known benchmark dataset in statistics [9]. The x values are time measurements in milliseconds after simulated impact and the y values are measurements of head acceleration. The x values are not equidistant and in some cases multiple y observations are present for certain x values. The data are heteroscedastic so that it forms a challenging test case in some sense [5]. Through 10-fold cross-validation, the following tuning parameters were obtained: $\sigma^2=144$, $C=Inf$, $\varepsilon=0.00001$ (SVM) and $\gamma=10$, $\sigma^2=0.5$ (LS-SVM). MSEs of SVM and LS-SVM are 493.002 and 469.932 respectively. In this experiment, SVM suffers more from boundary effects than LS-SVM.

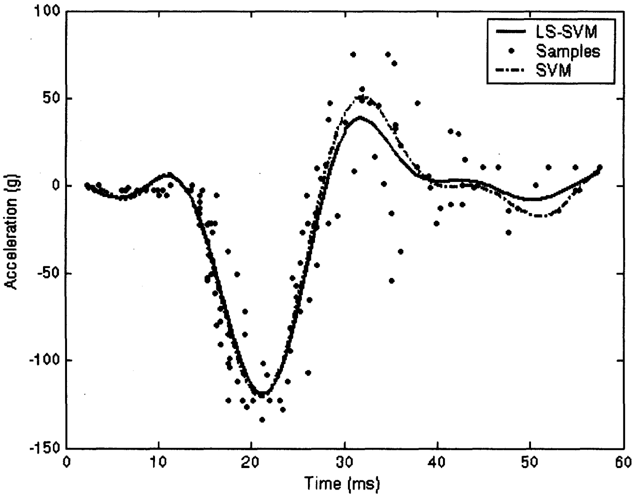

Fig. 7. Function estimation with motorcycle dataset

V. CONCLUSIONS

An attractive property of SVM solution is its sparseness, i.e. many elements in the QP solution vector are equal to zero. However, the application of SVM in large dataset has been limited by the time and memory consumed optimization. This disadvantage has been overcome by LS-SVM, which solves linear equations instead of a QP problem. At the same time, the sparseness is lost. To get a sparse solution, a simplest pruning method is introduced into LS-SVM. The results of parallel tests show that LS-SVM has superiority over SVM. And almost the same performance can be obtained by the pruned LS-SVM. The conclusion can be made that LS-SVM is preferred for large scale regression problems, because its solution procedure is high efficiency and after pruning both sparseness and performance of LS-SVM are comparable with those of SVM.

REFERENCES

[1] V. Vapnik, *The nature of statistical learning theory*. New York: Springer-Verlag, 1995.

[2] N. Cristianini, J. Shawe-Taylor. *An Introduction to Support Vector Machines*, Cambridge: Cambridge University Press, 2000.

[3] J. A. K. Suykens, "Nonlinear modeling and support vector machines", *Proceeding of IEEE Instrumentation and measurement technology*, Budapest, 2001.

[4] J. A. K.Suykens, L. Lukas, J. Vandewalle, "Sparse approximation using least squares support vector machines". *IEEE International symposium on circuits and systems*. Geneva 2000.

[5] J. A. K. Suykens, J. D. Brabanter, L. Lukas, et al., "Weighted least squares support vector machines-robustness and sparse approximation". *Neurocomputing*, 48: 85-105, 2002.

[6] A. J. Smola, B. SchÖlkopf, A tutorial on support vector regression. http: //www.neurocolt.com

[7] R. Xiao, J. C. Wang, F.Y. Zhang, "An approach to incremental svm learning algorithm". *Proceeding of International Conf. on Tools with Artificial Intelligence*, Vancouver,2000.

[8] B. Üstün, "A comparison of support vector machines and partial least squares regression on spectral data". Magisterial dissertation. University of Nijmegen, 2003.

[9] B. W. Silverman, R. L. Parker, J. A. Rice. "Some aspects of the smoothing spline approach to nonparametric curve fitting". *Journal of the Royal Statistical Society (Series B)*, 47: 1- 52, 1985.