Practical Selection of SVM Parameters and Noise Estimation for SVM Regression

Vladimir Cherkassky and Yunqian Ma*
Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA

Abstract

We investigate practical selection of meta-parameters for SVM regression (that is, ε-insensitive zone and regularization parameter C). The proposed methodology advocates analytic parameter selection directly from the training data, rather than resampling approaches commonly used in SVM applications. Good generalization performance of the proposed parameter selection is demonstrated empirically using several low-dimensional and high-dimensional regression problems. Further, we point out the importance of Vapnik's ε-insensitive loss for regression problems with finite samples. To this end, we compare generalization performance of SVM regression (with optimally chosen ε) with regression using 'least-modulus' loss (ε=0). These comparisons indicate superior generalization performance of SVM regression, for finite sample settings.

Keywords: Complexity Control; Parameter Selection; Support Vector Machine; VC theory

1. Introduction

This study is motivated by a growing popularity of support vector machines (SVM) for regression problems [3,6-14]. Their practical successes can be attributed to solid theoretical foundations based on VC-theory [13,14], since SVM generalization performance does not depend on the dimensionality of the input space. However, many SVM regression application studies are performed by 'expert' users having good understanding of SVM methodology. Since the quality of SVM models depends on a proper setting of SVM meta-parameters, the main issue for practitioners trying to apply SVM regression is how to set these parameter values (to ensure good generalization performance) for a given data set. Whereas existing sources on SVM regression [3,6-14]

*Corresponding author.
Email addresses: cherkass@ece.umn.edu (V. Cherkassky), myq@ece.umn.edu (Y. Ma)

give some recommendations on appropriate setting of SVM parameters, there is clearly no consensus and (plenty of) contradictory opinions. Hence, resampling remains the method of choice for many applications. Unfortunately, using resampling for (simultaneously) tuning several SVM regression parameters is very expensive in terms of computational costs and data requirements.

This paper describes simple yet practical analytical approach to SVM regression parameter setting directly from the training data. Proposed approach (to parameter selection) is based on well-known theoretical understanding of SVM regression that provides the basic analytical form of dependencies for parameter selection. Further, we perform empirical tuning of such dependencies using several synthetic data sets. Practical validity of the proposed approach is demonstrated using several low-dimensional and high-dimensional regression problems.

Recently, several researchers [10,13,14] noted similarity between Vapnik's ε-insensitive loss function and Huber's loss in robust statistics. In particular, Vapnik's loss function coincides with a special form of Huber's loss aka least-modulus loss (with ε=0). From the viewpoint of traditional robust statistics, there is well-known correspondence between the noise model and optimal loss function [10]. However, this connection between the noise model and the loss function is based on (asymptotic) maximum likelihood arguments [10]. It can be argued that for finite sample regression problems Vapnik's ε-insensitive loss (with properly chosen ε-parameter) actually would yield better generalization than other loss function (known to be asymptotically optimal for a particular noise density). In order to test this assertion, we compare generalization performance of SVM regression (with optimally chosen ε) with robust regression using least-modulus loss function (ε=0) for several noise densities.

This paper is organized as follows. Section 2 gives a brief introduction to SVM regression and reviews existing methods for SVM parameter setting. Section 3 describes the proposed approach to selecting SVM regression parameters. Section 4 presents empirical comparisons demonstrating the advantages of the proposed approach. Section 5 describes empirical comparisons for regression problems with non-Gaussian noise; these comparisons indicate that SVM regression (with optimally chosen ε) provides better generalization performance than SVM with least-modulus loss. Section 6 describes noise variance estimation for SVM regression. Finally, summary and discussion are given in Section 7.

2. Support Vector Regression and SVM Parameter Selection

In regression formulation, the goal is to estimate an unknown continuous-valued function based on a finite number set of noisy samples $(\mathbf{x}_i, y_i), (i = 1,...,n)$, where d-dimensional input $\mathbf{x} \in R^d$ and the output $y \in R$. Assumed statistical model for data generation has the following form:

$$y = r(\mathbf{x}) + \delta \tag{1}$$

where $r(\mathbf{x})$ is unknown target function (regression), and δ is additive zero mean noise with noise variance σ^2 [3,4].

In SVM regression, the input $\mathbf{x}$ is first mapped onto a m-dimensional feature space using some fixed (nonlinear) mapping, and then a linear model is constructed in this feature space [3,10,13,14]. Using mathematical notation, the linear model (in the feature space) $f(\mathbf{x},\omega)$ is given by

$$f(\mathbf{x},\omega) = \sum_{j=1}^{m} \omega_j g_j(\mathbf{x}) + b \tag{2}$$

where $g_j(\mathbf{x}), j = 1,..., m$ denotes a set of nonlinear transformations, and b is the "bias" term. Often the data are assumed to be zero mean (this can be achieved by preprocessing), so the bias term in (2) is dropped.

The quality of estimation is measured by the loss function $L(y, f(\mathbf{x},\omega))$. SVM regression uses a new type of loss function called ε-insensitive loss function proposed by Vapnik [13,14]:

$$L_\varepsilon(y, f(\mathbf{x},\omega)) = \begin{cases} 0 & if \mid y - f(\mathbf{x},\omega) \mid \leq \varepsilon \\ \mid y - f(\mathbf{x},\omega) \mid -\varepsilon & otherwise \end{cases} \tag{3}$$

The empirical risk is:

$$R_{emp}(\omega) = \frac{1}{n} \sum_{i=1}^{n} L_\varepsilon(y_i, f(\mathbf{x}_i,\omega)) \tag{4}$$

Note that ε-insensitive loss coincides with least-modulus loss and with a special case of Huber's robust loss function [13,14] when ε=0. Hence, we shall compare prediction performance of SVM (with proposed chosen ε) with regression estimates obtained using least-modulus loss(ε=0) for various noise densities.

SVM regression performs linear regression in the high-dimension feature space using ε-insensitive loss and, at the same time, tries to reduce model complexity by minimizing $\| \omega \|^2$. This can be described by introducing (non-negative) slack variables ξ_i, ξ_i^* $i = 1,...n$, to measure the deviation of training samples outside ε-insensitive zone. Thus SVM regression is formulated as minimization of the following functional:

$$\begin{aligned} &\min \quad \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \\ &\text{s.t.} \quad \begin{cases} y_i - f(\mathbf{x}_i,\omega) \leq \varepsilon + \xi_i^* \\ f(\mathbf{x}_i,\omega) - y_i \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, i = 1,..., n \end{cases} \end{aligned} \tag{5}$$

This optimization problem can transformed into the dual problem [13,14], and its solution is given by

$$f(\mathbf{x}) = \sum_{i=1}^{n_{SV}} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) \quad \text{s.t. } 0 \leq \alpha_i^* \leq C, 0 \leq \alpha_i \leq C, \tag{6}$$

where n_{SV} is the number of Support Vectors (SVs) and the kernel function

$$K(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^{m} g_j(\mathbf{x}) g_j(\mathbf{x}_i) \tag{7}$$

It is well known that SVM generalization performance (estimation accuracy) depends on a good setting of meta-parameters parameters C, ε and the kernel parameters. The problem of optimal parameter selection is further complicated by the fact that SVM model complexity (and hence its generalization performance) depends on all three parameters. Existing software implementations of SVM regression usually treat SVM meta-parameters as user-defined inputs. In this paper we focus on the choice of C and ε, rather than on selecting the kernel function. Selecting a particular kernel type and kernel function parameters is usually based on application-domain knowledge and also should reflect distribution of input ($\mathbf{x}$) values of the training data [1,12,13,14]. For example, in this paper we show examples of SVM regression using radial basis function(RBF) kernels where the RBF width parameter should reflect the distribution/range of $\mathbf{x}$-values of the training data.

Parameter C determines the trade off between the model complexity (flatness) and the degree to which deviations larger than ε are tolerated in optimization formulation (5). For example, if C is too large (infinity), then the objective is to minimize the empirical risk (4) only, without regard to model complexity part in the optimization formulation (5).

Parameter ε controls the width of the ε-insensitive zone, used to fit the training data [3,13,14]. The value of ε can affect the number of support vectors used to construct the regression function. The bigger ε, the fewer support vectors are selected. On the other hand, bigger ε-values result in more 'flat' estimates. Hence, both C and ε-values affect model complexity (but in a different way).

Existing practical approaches to the choice of C and ε can be summarized as follows:

- Parameters C and ε are selected by users based on a priori knowledge and/or user expertise [3,12,13,14]. Obviously, this approach is not appropriate for non-expert users. Based on observation that support vectors lie outside the ε-tube and the SVM model complexity strongly depends on the number of support vectors, Schölkopf et al [11] suggest to control another parameter ν (i.e., the fraction of points outside the ε-tube) instead of ε. Under this approach, parameter ν has to be user-defined. Similarly, Mattera and Haykin [7] propose to choose ε- value so that the percentage of support vectors in the SVM regression model is around 50% of the number of samples. However, one can easily show examples when optimal generalization performance is achieved with the number of support vectors larger or smaller than 50%.
- Smola et al [9] and Kwok [6] proposed asymptotically optimal ε- values proportional to noise variance, in agreement with general sources on SVM [3,13,14]. The main practical drawback of such proposals is that they do not reflect sample size. Intuitively, the value of ε should be smaller for larger sample size than for a small sample size (with the same level of noise).
- Selecting parameter C equal to the range of output values [7]. This is a reasonable proposal, but it does not take into account possible effect of outliers in the training data.
- Using cross-validation for parameter choice [3,12]. This is very computation and data-intensive.
- Several recent references present statistical account of SVM regression [10,5] where the ε- parameter is associated with the choice of the loss function (and hence could be optimally tuned to particular noise density) whereas the C parameter is interpreted as a

traditional regularization parameter in formulation (5) that can be estimated for example by cross-validation [5].

As evident from the above, there is no shortage of (conflicting) opinions on optimal setting of SVM regression parameters. Under our approach (described next in Section 3) we propose:

- Analytical selection of C parameter directly from the training data (without resorting to resampling);
- Analytical selection of ε - parameter based on (known or estimated) level of noise in the training data.

Further ample empirical evidence presented in this paper suggests the importance of ε-insensitive loss, in the sense that SVM regression (with proposed parameter selection) consistently achieves superior prediction performance vs other (robust) loss functions, for different noise densities.

3. Proposed Approach for Parameter Selection

Selection of parameter C. Optimal choice of regularization parameter C can be derived from standard parameterization of SVM solution given by expression (6):

$$\begin{aligned} |f(\mathbf{x})| &\le \left|\sum_{i=1}^{n_{SV}} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x})\right| \\ &\le \sum_{i=1}^{n_{SV}} |(\alpha_i - \alpha_i^*)| \cdot |K(\mathbf{x}_i, \mathbf{x})| \\ &\le \sum_{i=1}^{n_{SV}} C \cdot |K(\mathbf{x}_i, \mathbf{x})| \end{aligned} \tag{8}$$

Further we use kernel functions bounded in the input domain. To simplify presentation, assume RBF kernel function

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2p^2}\right) \tag{9}$$

so that $K(\mathbf{x}_i, \mathbf{x}) \le 1$. Hence we obtain the following upper bound on SVM regression function:

$$\left|f(\mathbf{x})\right| \le C \cdot n_{SV} \tag{10}$$

Expression (10) is conceptually important, as it relates regularization parameter C and the number of support vectors, 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) n_{sv}, one can robustly let $C \ge \left|f(\mathbf{x})\right|$ for all training samples, which leads to setting C equal to the range of response values of training data [7]. However, such a setting is quite sensitive to the possible presence of outliers, so we propose to use instead the following prescription for regularization parameter:

$$C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) \tag{11}$$

where $\bar{y}$ is the mean of the training responses (outputs), and σ_y is the standard deviation of the training response values. Prescription (11) can effectively handle outliers in the training data. In practice, the response values of training data are often scaled so that $\bar{y}$ =0; then the proposed C is $3\sigma_y$.

Selection of ε . It is well-known that the value of ε should be proportional to the input noise level, that is $\varepsilon \propto \sigma$ [3,6,9,13]. Here we assume that the standard deviation of noise σ is known or can be estimated from data (practical approaches to noise estimation are discussed in Section 6). However, the choice of ε should also depend on the number of training samples. From standard statistical theory, the variance of observations about the trend line (for linear regression) is:

$$\sigma^2_{y/x} \propto \frac{\sigma^2}{n} \tag{12}$$

This suggests the following prescription for choosing ε :

$$\varepsilon \propto \frac{\sigma}{\sqrt{n}} \tag{13}$$

Based on a number of empirical comparisons, we found that (13) works well when the number of samples is small, however for large values of n prescription (13) yields ε -values that are too small. Hence we propose the following (empirical) dependency:

$$\varepsilon = \tau\sigma\sqrt{\frac{\ln n}{n}} \tag{14}$$

Based on empirical tuning, the constant value $\tau = 3$ gives good performance for various data set sizes, noise levels and target functions for SVM regression. Thus expression (14) is used in all empirical comparisons presented in Sections 4 and 5.

4. Experimental Results for Gaussian Noise

First we describe experimental procedure used for comparisons, and then present empirical results.

Training data: simulated training data $(\mathbf{x}_i, y_i), (i = 1,...n)$ where **x**-values are sampled on uniformly-spaced grid in the input space, and *y*-values are generated according to $y = r(\mathbf{x}) + \delta$. Different types of the target functions $r(\mathbf{x})$ are used. The *y*-values of training data are corrupted by additive noise. We used Gaussian noise (results described in this section) and several non-Gaussian additive symmetric noise densities (discussed in Section 5). Since SVM approach is not sensitive to a particular noise distribution, we expect to show good generalization performance with different types of noise, as long as an optimal value of ε (reflecting standard deviation of noise σ) has been used.

Test data: the test inputs are sampled randomly according to uniform distribution in **x**-space.

Kernel function: RBF kernel functions (9) are used in all experiments, and the kernel width parameter *p* is appropriately selected to reflect the input range of the training/test data. Namely, the RBF width parameter is set to *p ~ (0.2-0.5)* range (x)*. For higher *d*-dimensional problems the RBF width parameter is set so that p^d *~ (0.2-0.5)* where all *d*

input variables are pre-scaled to [0,1] range. Such values yield good SVM performance for various regression data sets.

Performance metric: since the goal is optimal selection of SVM parameters in the sense of generalization, the main performance metric is prediction risk

$$R_{pred}(\omega) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_{t\arg et} - f(\mathbf{x}_{test}, \omega))^2 \tag{15}$$

defined as MSE between SVM estimates and true values of the target function for test inputs.

The first set of results show how SVM generalization performance depends on a proper choice of SVM parameters for univariate *sinc* target function:

$$r(x) = a\frac{\sin(x)}{x} \qquad x \in [-10,10] \tag{16}$$

The following values of a were used $1,10,0.1,-10,-0.1$ to generate five data sets using small sample size (n=30) with additive Gaussian noise (with different noise levels σ as shown in Table 1) . For these data sets, we used RBF kernels with width parameter p=4.

Table 1 shows:

(a) Parameter values C and ε (using expressions proposed in Section 3) for different training sets.

(b) Prediction risk and percentage of support vectors (%SV) obtained by SVM regression with proposed parameter values.

(c) Prediction risk and percentage of support vectors (%SV) obtained using least-modulus loss function (ε =0).

We can see that the proposed method for choosing ε is better than least-modulus loss function, as it yields lower prediction risk and better (more sparse) representation.

Table 1
Results for univariate *sinc* function (small size): Data Set 1- Data Set5

Data Set	a	Noise Level(σ)	C-selection	ε -selection	Prediction Risk	%SV
1	1	0.2	1.58	ε =0	0.0129	100%
				ε =0.2 (prop.)	0.0065	43.3%
2	10	2	15	ε =0	1.3043	100%
				ε =2.0 (prop.)	0.7053	36.7%
3	0.1	0.02	0.16	ε =0	1.03e-04	100%
				ε =0.02 (prop.)	8.05e-05	40.0%
4	-10	0.2	14.9	ε =0	0.0317	100%
				ε =0.2 (prop.)	0.0265	50.0%
5	-0.1	0.02	0.17	ε =0	1.44e-04	100%
				ε =0.02 (prop.)	1.01e-04	46.7%

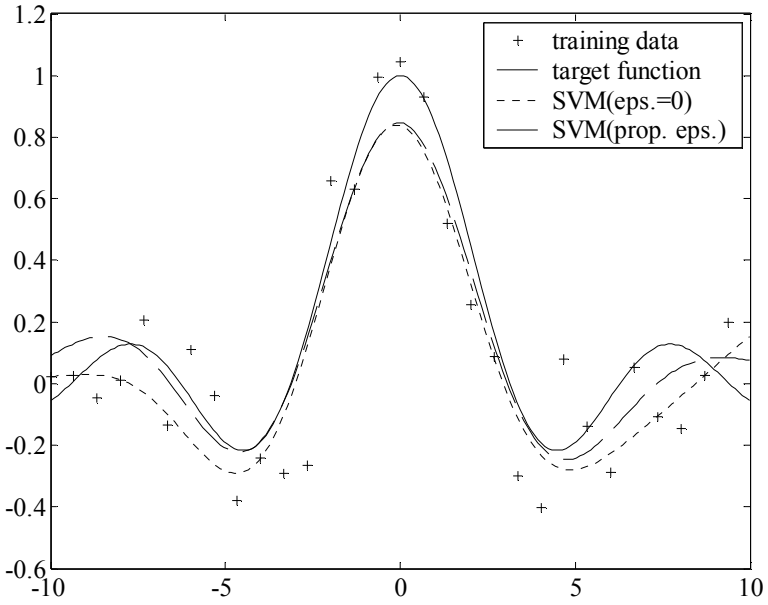


Fig. 1. For Data Set 1, SVM estimate using proposed parameter selection vs using least-modulus loss.

Visual comparisons (for univariate *sinc* Data Set 1) between SVM estimates using proposed parameter selection and using least-modulus loss are shown in Fig.1, where the solid line is the target function, the '+' denote training data, the dotted line is an estimate using least-modulus loss and the dashed line is the SVM estimate function using our method.

The accuracy of expression (14) for selecting 'optimal' ε as a function of n (the number of training samples is demonstrated in Fig. 2. Results in Fig.2 show that proposed ε -values vs optimal ε -values (obtained by exhaustive search in terms of prediction risk) for Data Set 1 (see Table 1) for different number of training samples.

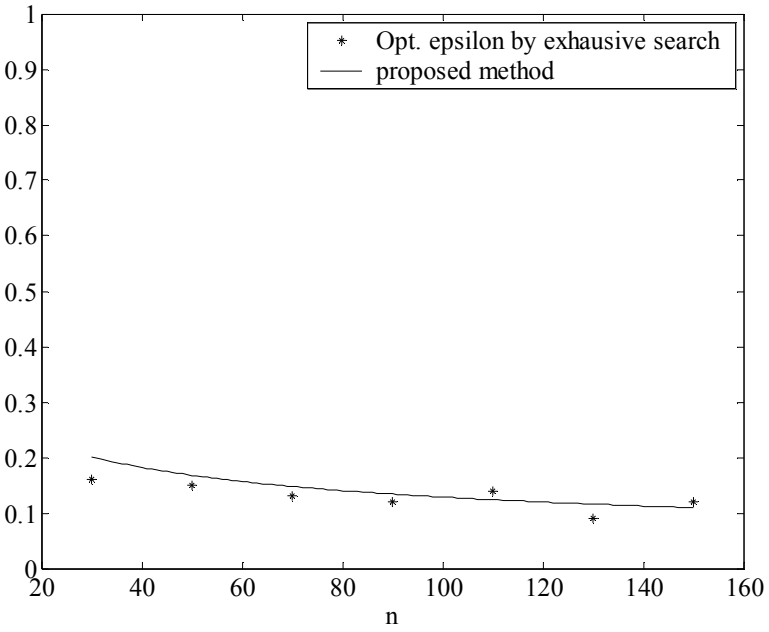
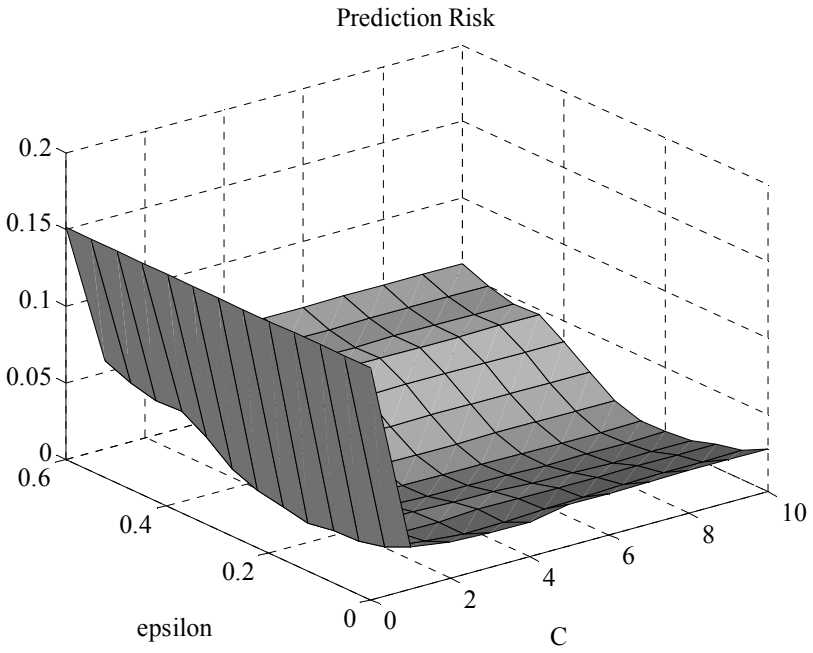


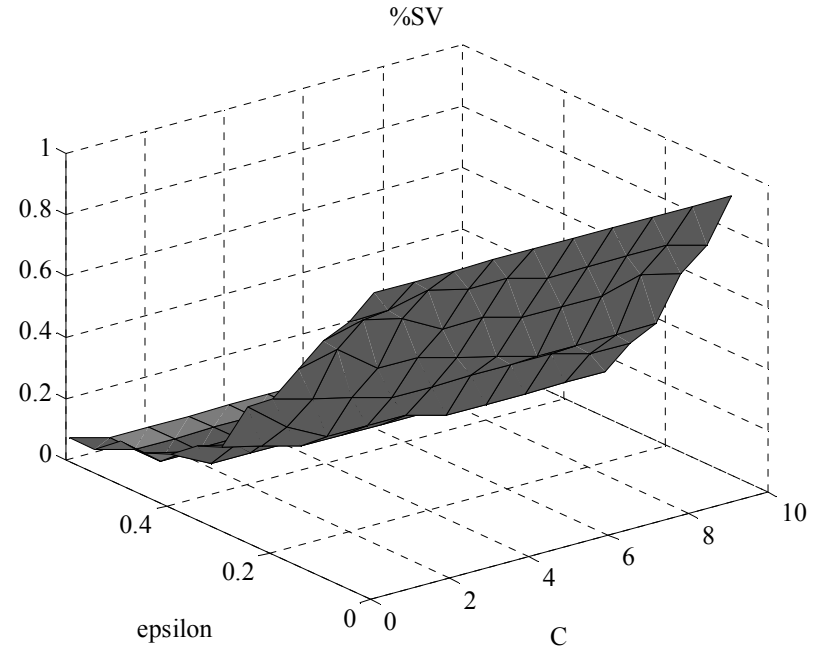
Fig. 2. Proposed ε -values vs optimal ε -values (obtained by exhaustive search in terms of prediction risk) for Data set 1 for different number of training data (n=30, 50, … , 150).

Dependence of prediction risk as a function of chosen C and ε -values for Data Set 1 (i.e., *sinc* target function, 30 training samples) in shown in Fig. 3a. Fig. 3b shows the percentage of support vectors (%SV) selected by SVM regression, which is an important factor affecting generalization performance. Visual inspection of results in Fig.3a indicates that proposed choice of ε , C gives good/ near optimal performance in terms of prediction risk. Also, one can clearly see that C-values above certain threshold have only minor effect on the prediction risk. Our method guarantees that the proposed chosen C-values result in SVM solutions in flat regions of prediction risk. Using three dimensional Fig.3b, we can see that small ε -values correspond to higher percentage of support vectors, whereas parameter C has negligible effect on the percentage of SV selected by SVM method.

Fig. 4 shows prediction risk as a function of chosen C and ε -values for *sinc* target function for Data Set 2 and Data Set 3. We can see that the proposed choice of C yields optimal and robust C-value corresponding to SVM solutions in flat regions of prediction risk.

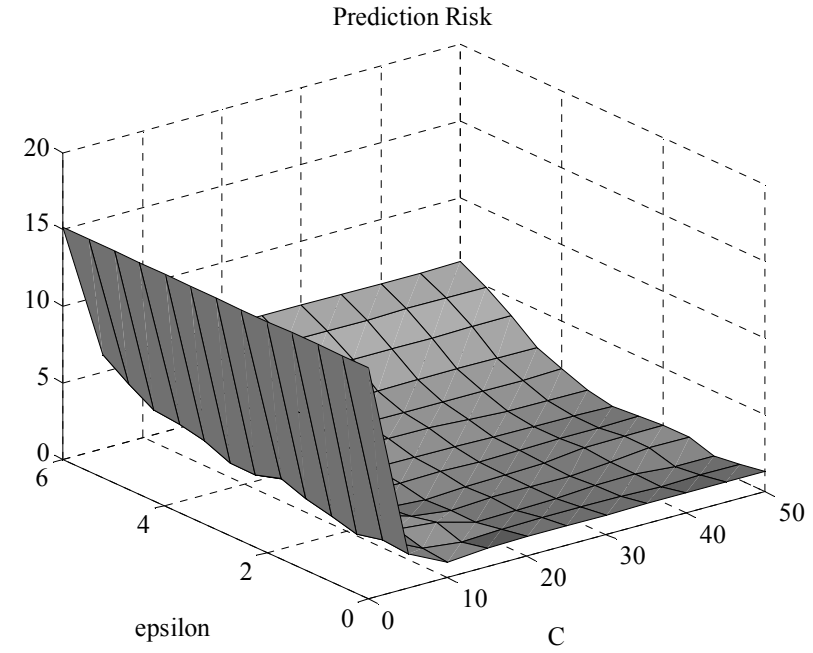


(a)

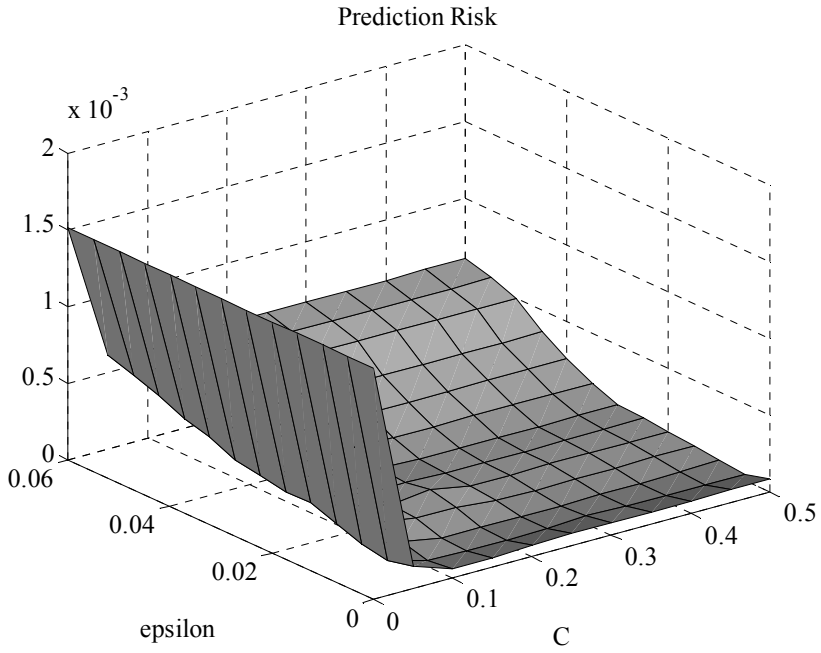


(b)

Fig. 3. Results for small sample size, *sinc* target function: Set 1 (a) Prediction risk (b) The number of SV as a fraction of training data



(a)



(b)

Fig. 4. Results for small sample size, *sinc* target function: (a) Prediction Risk for Data Set 2 (b) Prediction Risk for Data Set 3

In order to investigate the effect of the sample size (on selection of ε -value), we generate 200 training samples using univariate *sinc* target function (as in Data Set 1) with

Gaussian noise (σ=0.2). Fig.5 shows the dependence of prediction risk on SVM parameters for this data set (large sample size). According to proposed expression (14) and (11), proposed ε is 0.1, proposed C is 1.58, which is consistent with results in Fig.5. Also, the prediction risk is 0.0019, which compares favorably with SVM using least-modulus loss (ε=0) where the prediction risk is 0.0038. Similarly, the proposed method compares favorably with selection $\varepsilon = 0.8485\ \sigma$ proposed by Kwok [6]. For this data set, Kwok's method yields $\varepsilon = 0.17$ and the prediction risk is 0.0033. The reason that our approach to ε-selection gives better results is that previous methods for selecting ε-value [6,9] do not depend on sample size.

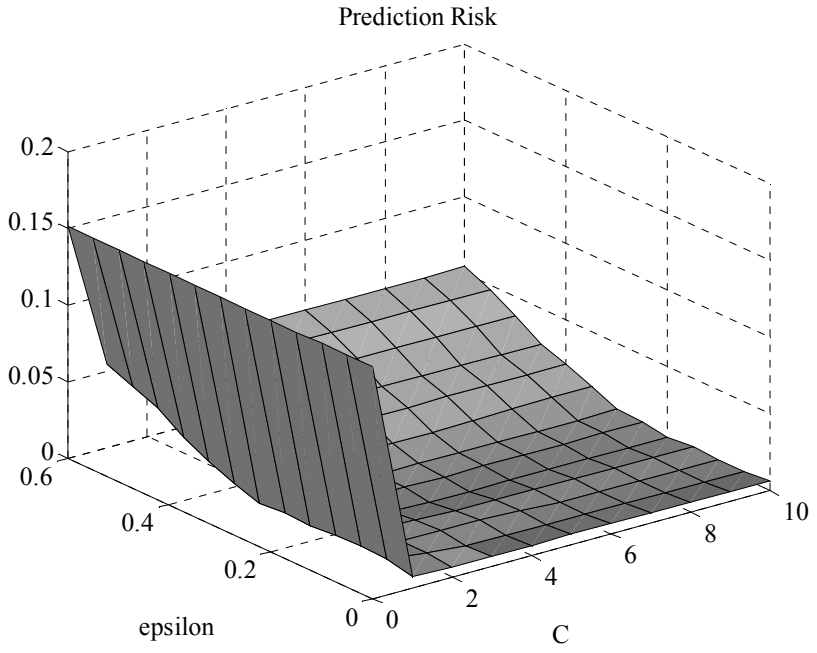
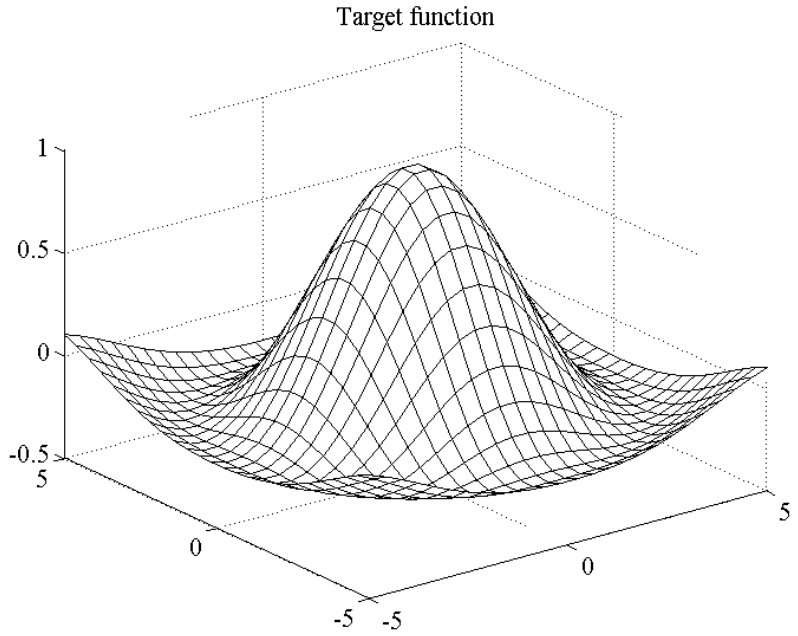


Fig. 5. Result for large sample size *sinc* function (Data Set 1): Prediction risk

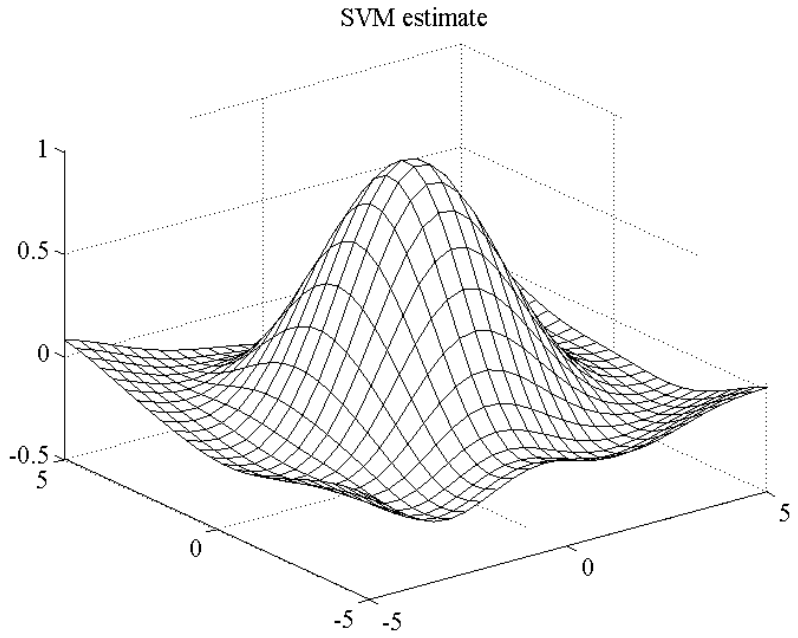
Next we show results of SVM parameter selection for high-dimensional problems. The first data set is generated using two-dimensional sinc target function [13,14]

$$r(\mathbf{x}) = \frac{\sin\sqrt{x_1^2 + x_2^2}}{\sqrt{x_1^2 + x_2^2}} \tag{17}$$

defined on a uniform square lattice [-5,5]*[-5,5] , with response values corrupted with Gaussian noise (σ=0.1 and σ =0.4 respectively). The number of training samples is 169, and the number of test samples is 676. The RBF kernel width parameter p=2 is used. Fig. 6a shows the target function and Fig. 6b shows the SVM estimate obtained using proposed parameter selection for σ =0.1. The proposed C =1.16 and ε =0.05 (for σ =0.1) and ε =0.21 (for σ =0.4). Table 2 compares the proposed parameter selection with estimates obtained using least-modulus loss, in terms of prediction risk and the percentage of SV chosen by each method.



(a)



(b)

Fig. 6. (a) 2D *sinc* target function (b) SVM regression estimate using proposed method for for σ =0.1

Table 2
Comparison of the proposed method for ε with least-modulus loss (ε =0) for two-dimensional *sinc* target function data sets.

Noise Level	ε -selection	Prediction Risk	%SV
σ=0.1	ε =0	0.0080	100%
	ε (Proposed)	0.0020	62.7%
σ=0.4	ε =0	0.0369	100%
	ε (Proposed)	0.0229	60.9%

Next we show results of SVM parameter selection for higher dimensional additive target function

$$r(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \quad (18)$$

where $\mathbf{x}$ -values are distributed in hypercube $[0,1]^5$. Output (response) values of training samples are corrupted by additive Gaussian noise is (with σ=0.1 and σ=0.2). Training data size is n=243 samples (i.e., 3 points per each input dimension). The test size is 1024. The RBF kernel width parameter p=0.8 is used for this data set.. The optimal value of C is 34 and the optimal ε=0.045 for σ=0.1 and ε=0.09 for σ=0.2. Comparison results between the proposed methods for parameter selection with the method using least-modulus loss function are shown in Table 3. Clearly, the proposed approach gives better performance in terms of prediction risk and robustness.

Table 3
Comparison of proposed method for ε parameter selection with least-modulus loss (ε =0) for high-dimensional additive target function

Noise Level	ε -selection	Prediction Risk	%SV
σ=0.1	ε =0	0.0443	100%
	ε (Proposed)	0.0387	86.7%
σ=0.2	ε =0	0.1071	100%
	ε (Proposed)	0.0918	90.5%

5. Experimental Results for Non-Gaussian Noise

This section describes empirical results for regression problems with non-Gaussian additive symmetric noise in the statistical model (1). The main motivation is to demonstrate practical advantages of Vapnik's ε -insensitive loss vs other loss functions. Whereas practical advantages of SVM regression are well-known, there is a popular opinion [10] that one should use a particular loss function for a given noise density. Hence, we perform empirical comparisons between SVM regression (with proposed parameter selection) vs SVM regression using least-modulus loss, for several finite-sample regression problems.

First, consider Student's *t*-distribution for noise. Univariate *sinc* target function is used for comparisons: $r(x) = 10\sin(x)/x \quad x \in [-10,10]$. Training data consists of n=30 samples. RBF kernels with width parameter p=4 are used for this data set. Several experiments have been performed using various degrees of freedom (5, 10, 20, 30, 40) for generating *t*-distribution. Empirical results indicate superior performance of the proposed method for parameter selection. Table 4 shows comparison results with least-modulus loss for Student's noise with 5 degrees of freedom (when σ of noise is 1.3). According to proposed expressions (14) and (11), proposed ε is 1.3 and C is 16.

Table 4
Comparison of proposed method for ε with least-modulus loss (ε =0) for *t*-distribution noise

ε -selection	Prediction Risk	%SV
ε =0	0.9583	100%
ε (Proposed)	0.6950	40%

Next, we show comparison results for Laplacian noise density:

$$p(\delta) = \frac{1}{2}\exp(-|\delta|) \tag{19}$$

Smola et al [10] suggest that for this noise density model, the least-modulus loss should be used. Whereas this suggestion might work in an asymptotical setting, it does not guarantee superior performance with finite samples. We compare the proposed approach for choosing ε with the least-modulus loss method in noise density model (19). This experiment uses the same *sinc* target function as in Table 4 (with sample size n=30). The σ of noise for Laplacian noise model (19) is 1.41 (precisely $\sqrt{2}$). Using our proposed approach, ε =1.41 and C is 16. Table 5 shows comparison results. Visual comparison of results in Table 5 is also shown in Fig. 7, where the solid line is the target function, the '+' denote training data, the dotted line is an estimate found using least-modulus loss and the dashed line is an estimate found using SVM method with proposed parameter selection.

Table 5
Comparison of the proposed method for ε with least-modulus loss (ε =0) for Laplacian noise.

ε -selection	Prediction Risk	%SV
ε =0	0.8217	100%
ε (Proposed)	0.5913	46.7%

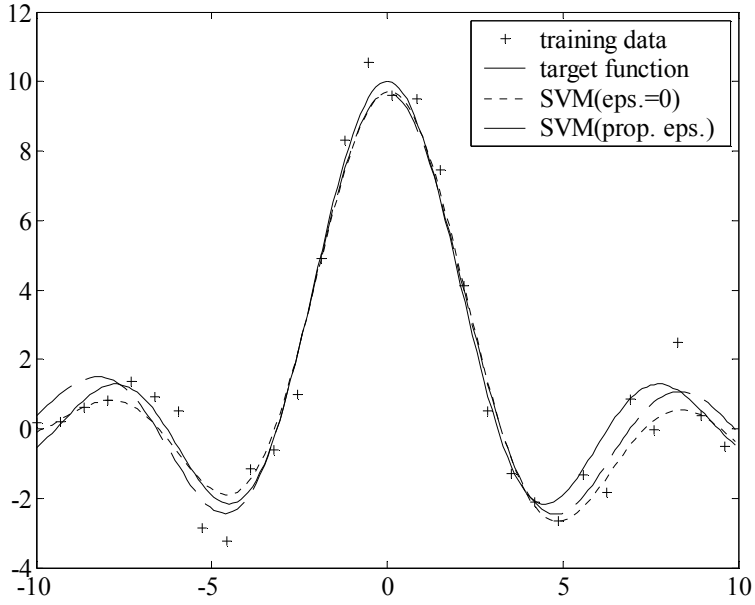


Fig. 7. SVM estimate using proposed parameter selection vs using least-modulus loss

Finally, consider uniform distribution for the additive noise. Univariate *sinc* target function is used for comparisons: $r(x) = \sin(x)/x \qquad x \in [-10,10]$. Several experiments have been performed using different noise level σ. Training sample size n=30 is used in the experiments. According to proposed expressions (14) and (11), C is 1.6, ε is 0.1(for σ=0.1), ε is 0.2(for σ=0.2), ε is 0.3(for σ=0.3). Table 6 shows comparison results.

Table 6
Comparison of proposed method for ε with least-modulus loss (ε =0) for uniformly distributed noise

Noise Level	ε -selection	Prediction Risk	%SV
σ=0.1	ε =0	0.0080	100%
	ε (Proposed)	0.0036	60%
σ=0.2	ε =0	0.0169	100%
	ε (Proposed)	0.0107	43.3%
σ=0.3	ε =0	0.0281	100%
	ε (Proposed)	0.0197	50%

6. Noise Variance Estimation

The proposed method for selecting ε relies on the knowledge of the standard deviation of noise σ. The problem, of course, is that the noise variance is not known a priori, and it needs to be estimated from training data $(\mathbf{x}_i, y_i), (i = 1,...n)$.

In practice, the noise variance can be readily estimated from the squared sum of residuals (fitting error) of the training data. Namely, the well-known approach of estimating noise variance (for linear models) is by fitting the data using low bias (high-complexity) model (say high-order polynomial) and applying the following formula to estimate noise [3,4]

$$\hat{\sigma}^2 = \frac{n}{n-d} \cdot \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{20}$$

where d is the 'degrees of freedom' (DOF) of the high-complexity estimator and n is the number of training samples. Note that for linear estimators (i.e., polynomial regression) DOF is simply the number of free parameters (polynomial degree); whereas the notion of DOF is not well defined for other types of estimators [3].

We used expression (20) for estimating noise variance using higher-order algebraic polynomials (for univariate regression problems) and *k*-nearest-neighbors regression. Both approaches yield very accurate estimates of the noise variance; however, we only show the results of noise estimation using *k*-nearest-neighbors regression. In *k*-nearest-neighbors method, the function is estimated by taking a local average of the training data. Locality is defined in terms of the *k* data points nearest the estimation point. The model complexity (DOF) of the *k*-nearest neighbors method can be estimated as:

$$d = \frac{n}{k} \tag{21}$$

Even though the accuracy of estimating DOF for *k*-nearest-neighbors regression via (21) may be questionable, it provides rather accurate noise estimates when used in conjunction with (20).

Combining expressions (20) and (21), we obtain the following prescription for noise variance estimation via *k*-nearest neighbor's method:

$$\begin{aligned} \hat{\sigma}^2 &= \frac{n}{n-d} \cdot \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \frac{n}{n - \frac{n}{k}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \frac{k}{k-1} \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \end{aligned} \tag{22}$$

Typically, small values of *k* (in the 2-6 range) corresponding to low-bias/high variance estimators should be used in formula (22). In order to illustrate the effect of different *k*-values on the accuracy of noise variance estimation, we use three-dimension figure showing estimated noise as a function of *k* and n (number of training samples). Fig. 8 shows noise estimation results for univariate *sinc* target function corrupted by Gaussian

noise with σ=0.6 (noise variance is 0.36). It is evident from Fig. 8 that *k*-nearest neighbor method provides robust and accurate noise estimates with *k*-values chosen in a (2-6) range.

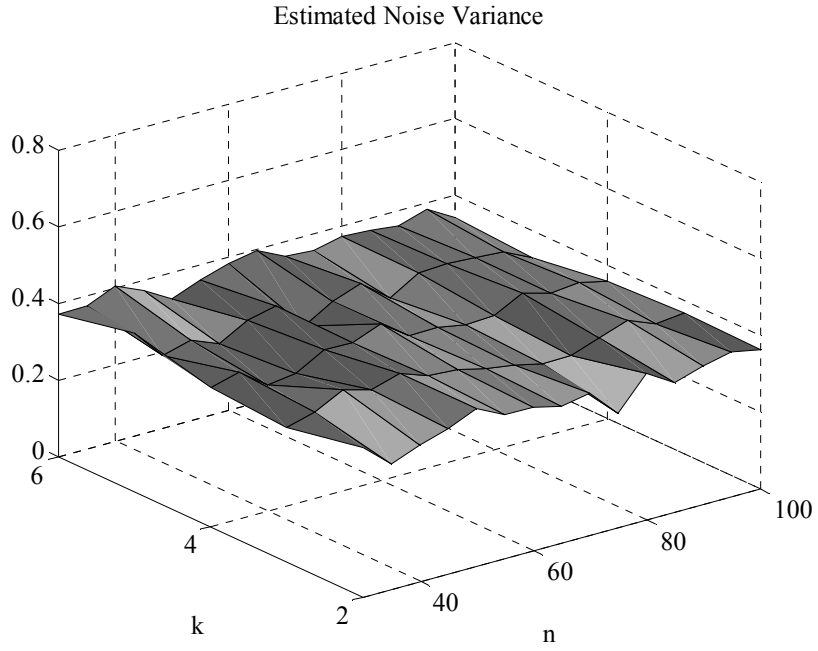


Fig.8. Using *k*-nearest neighbor method for estimating noise variance for univariate *sinc* function with different *k* and *n* values when the true noise variance =0.36

Since accurate estimation of noise variance does not seem to be affected much by specific *k*-value, we use *k* nearest neighbor method (with *k*=3). With *k*=3 expression (22) becomes

$$\hat{\sigma}^2 = 1.5\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \ . \qquad (23)$$

We performed noise estimation experiments using *k*-nearest neighbor method (with *k*=3) with different target functions, different sample size and different noise levels. In all cases, we obtained accurate noise estimates. Here, we only show noise estimation results for the univariate *sinc* target function for different true noise levels σ=0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 (true noise variance is 0.01, 0.04, 0.09, 0.16, 0.25, 0.36, 0.49, 0.64 accordingly). Fig. 9 shows the scatter plot of noise level estimates obtained via (23) for 10 independently generated data sets (for each true noise level). Results in Fig.9 correspond to the least favorable experimental set-up for noise estimation (that is, small number of samples n=30 and large noise levels).

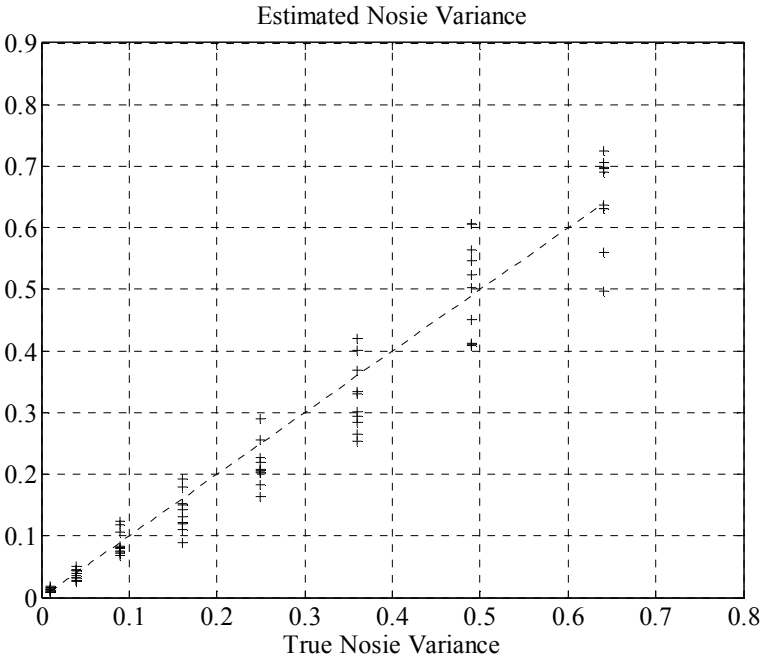


Fig. 9. Scatter plot of noise estimates obtained using *k*-nearest neighbors method (*k*=3) for univariate *sinc* function for different noise level (n=30)

Empirical results presented in this section show how to estimate (accurately) the noise level from available training data. Hence, this underscores practical applicability of proposed expression (14) for ε -selection. In fact, empirical results (not shown here due to space constraints) indicate that SVM estimates obtained using estimated noise level for ε -selection yield similar prediction accuracy (within 5%) to SVM estimates obtained using known noise level, for data sets in Section 4 and 5.

7. Summary and Discussion

This paper describes practical recommendations for setting meta-parameters for SVM regression. Namely the values of ε and *C* parameters are obtained directly from the training data and (estimated) noise level. Extensive empirical comparisons suggest that the proposed parameter selection yields good generalization performance of SVM estimates under different noise levels, types of noise, target functions and sample sizes. Hence proposed approach for SVM parameter selection can be immediately used by practitioners interested in applying SVM to various application domains.

Our empirical results suggest that with proposed choice of ε , the value of regularization parameter *C* has only negligible effect on the generalization performance (as long as *C* is larger than a certain threshold analytically determined from the training data). The proposed value of *C*-parameter is derived for RBF kernels; however the same approach can be applied to other kernels bounded in the input domain. For example, we successfully applied proposed parameter selection for SVM regression with polynomial

kernel defined in $[0,1]$ (or $[-1,1]$) input domain. Future related research may be concerned with investigating optimal selection of parameters C and ε for different kernel types, as well as optimal selection of kernel parameters (for these types of kernels). In this paper (using RBF kernels), we used fairly straightforward procedure for a 'good' setting of RBF width parameter independent of C and ε selection, thereby conceptually separating kernel parameter selection from SVM meta-parameter selection. However, it is not clear whether such a separation is possible with other kernel types.

The second contribution of this paper is demonstrating the importance of ε- insensitive loss function on the generalization performance. Several recent sources [10,5] assert that an optimal choice of the loss function (i.e. least-modulus loss, Huber's loss, quadratic loss etc.) should match a particular type of noise density (assumed to be known). However, these assertions are based on proofs asymptotic in nature. So we performed a number of empirical comparisons between SVM regression (with optimally chosen parameter values) and 'least-modulus' regression (with ε=0). All empirical comparisons show that SVM regression with ε- insensitive loss provide better prediction performance that regression with least-modulus loss, even in the case of Laplacian noise (for which least-modulus regression is known to be statistically 'optimal'). Likewise, recent study [2] shows that SVM loss (with proposed ε) outperforms other commonly used loss functions (squared loss, least-modulus loss) for linear regression with finite samples. Intuitively, superior performance of ε-insensitive loss for finite-sample problems can be explained by noting that noisy data samples very close to the true target function should not contribute to the empirical risk. This idea is formally reflected in Vapnik's loss function, whereas Huber's loss function assigns squared loss to samples with accurate (close to the truth) response values. Conceptually, our findings suggest that for finite-sample regression problems we only need the knowledge of noise level (for optimal setting of ε), instead of the knowledge of noise density. In other words, optimal generalization performance of regression estimates depends mainly on the noise variance rather than noise distribution. The noise variance itself can be estimated directly from the training data, i.e. by fitting very flexible (high-variance) estimator to the data. Alternatively, one can first apply least-modulus regression to the data, in order to estimate noise level.

Further research in this direction may be needed, to gain better understanding of the relationship between optimal loss function, noise distribution and the number of training samples. In particular, an interesting research issue is to find the minimum number of samples beyond which a theoretically optimal loss function (for a given noise density) indeed provides superior generalization performance.

Acknowledgements

The authors thank Dr. V. Vapnik for many useful discussions. We also acknowledge several useful suggestions from anonymous reviewers. This work was supported, in part, by NSF grant ECS-0099906.

References

[1] O. Chapelle and V. Vapnik, Model Selection for Support Vector Machines. In Advances in Neural Information Processing Systems, Vol 12, (1999)

[2] V. Cherkassky and Y. Ma, Selecting of the Loss Function for Robust Linear Regression, Neural computation, under review (2002)

[3] V. Cherkassky and F. Mulier, Learning from Data: Concepts, Theory, and Methods. (John Wiley & Sons, 1998)

[4] V. Cherkassky, X. Shao, F.Mulier and V. Vapnik, Model Complexity Control for Regression Using VC Generalization Bounds. IEEE Transaction on Neural Networks, Vol 10, No 5 (1999) 1075-1089

[5] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning. Data Mining, Inference and Prediction, (Springer, 2001)

[6] J.T. Kwok, Linear Dependency between ε and the Input Noise in ε –Support Vector Regression, in: G. Dorffner, H. Bishof, and K. Hornik (Eds): ICANN 2001, LNCS 2130 (2001) 405-410

[7] D. Mattera and S. Haykin, Support Vector Machines for Dynamic Reconstruction of a Chaotic System, in: B. Schölkopf, J. Burges, A. Smola, ed., Advances in Kernel Methods: Support Vector Machine, MIT Press, (1999)

[8] K. Muller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, V. Vapnik, Using Support Vector Machines for Time Series Prediction, in: B. Scholkopf, J. Burges, A. Smola, ed., Advances in Kernel Methods: Support Vector Machine, MIT Press, (1999)

[9] A. Smola, N. Murata, B. Schölkopf and K. Muller, Asymptotically optimal choice of ε –loss for support vector machines, Proc. ICANN, (1998)

[10] A. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998

[11] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Support Vector regression with automatic accuracy control, in L. Niklasson, M. Bodén, and T. Ziemke, ed., Proceedings of ICANN'98, Perspectives in Neural Computing, (Springer, Berlin, 1998) 111-116

[12] B. Scholkopf, J. Burges, A. Smola, Advances in Kernel Methods: Support Vector Machine. (MIT Press, 1999)

[13] V. Vapnik. The Nature of Statistical Learning Theory (2nd ed.). (Springer, 1999)

[14] V. Vapnik. Statistical Learning Theory, (Wiley, New York, 1998)

Vladimir Cherkassky is with Electrical and Computer Engineering at the University of Minnesota. He received Ph.D. in Electrical Engineering from University of Texas at Austin in 1985. His current research is on methods for predictive learning from data, and he has co-authored a monograph *Learning From Data* published by Wiley in 1998. He has served on editorial boards of IEEE Transactions on Neural Networks, the Neural Networks Journal, and the Neural Processing Letters. He served on the program committee of major international conferences on Artificial Neural Networks, including International Joint Conference on Neural Networks (IJCNN), and World Congress on Neural Networks (WCNN). He was Director of NATO Advanced Study Institute (ASI) From Statistics to Neural Networks: Theory and Pattern Recognition Applications held in

France, in 1993. He presented numerous tutorials and invited lectures on neural network and statistical methods for learning from data.

Yunqian Ma is PhD candidate in Department of Electrical Engineering at the University of Minnesota. He received M.S. in Pattern Recognition and Intelligent System at Tsinghua University, P.R.China in 2000. His current research interests include support vector machines, neural network, model selection, multiple model estimation, and motion analysis.