



ELSEVIER

AVAILABLE AT

[www.ElsevierComputerScience.com](http://www.ElsevierComputerScience.com)

POWERED BY SCIENCE @ DIRECT

Neural Networks 17 (2004) 113–126

Neural  
Networks

[www.elsevier.com/locate/neunet](http://www.elsevier.com/locate/neunet)

# Practical selection of SVM parameters and noise estimation for SVM regression

Vladimir Cherkassky\*, Yunqian Ma

*Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA*

Received 6 January 2003; accepted 15 May 2003

## Abstract

We investigate practical selection of hyper-parameters for support vector machines (SVM) regression (that is,  $\varepsilon$ -insensitive zone and regularization parameter  $C$ ). The proposed methodology advocates analytic parameter selection directly from the training data, rather than re-sampling approaches commonly used in SVM applications. In particular, we describe a new analytical prescription for setting the value of insensitive zone  $\varepsilon$ , as a function of training sample size. Good generalization performance of the proposed parameter selection is demonstrated empirically using several low- and high-dimensional regression problems. Further, we point out the importance of Vapnik's  $\varepsilon$ -insensitive loss for regression problems with finite samples. To this end, we compare generalization performance of SVM regression (using proposed selection of  $\varepsilon$ -values) with regression using 'least-modulus' loss ( $\varepsilon = 0$ ) and standard squared loss. These comparisons indicate superior generalization performance of SVM regression under sparse sample settings, for various types of additive noise.

© 2003 Elsevier Ltd. All rights reserved.

*Keywords:* Complexity control; Loss function; Parameter selection; Prediction accuracy; Support vector machine regression; VC theory

## 1. Introduction

This study is motivated by a growing popularity of support vector machines (SVM) for regression problems (Cherkassky & Mulier, 1998; Drucker, Burges, Kaufman, Smola, & Vapnik, 1997; Kwok, 2001; Mattera & Haykin, 1999; Muller et al., 1999; Schölkopf, Bartlett, Smola, & Williamson, 1998; Schölkopf, Burges, & Smola, 1999; Schölkopf & Smola, 2002; Smola, Murata, Schölkopf, & Muller, 1998; Smola & Schölkopf, 1998; Vapnik, 1998, 1999). Their practical success can be attributed to solid theoretical foundations based on VC-theory (Vapnik, 1998, 1999), 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. Since the quality of SVM models depends on a proper setting of SVM hyper-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 (Cherkassky & Mulier, 1998; Kwok, 2001; Mattera & Haykin, 1999; Muller et al., 1999; Schölkopf et al., 1998, 1999; Smola et al., 1998; Smola & Schölkopf, 1998; Vapnik, 1998, 1999) give some recommendations on appropriate setting of SVM parameters, there is no general consensus and many contradictory opinions. Hence, re-sampling remains the method of choice for many applications. Unfortunately, using re-sampling 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 proposed (analytical) prescriptions for parameter selection. Further, we perform empirical tuning of these analytical dependencies using synthetic data sets. Practical validity of the proposed approach is demonstrated using several low- and high-dimensional regression problems.

Recently, several researchers (Smola & Schölkopf, 1998; Vapnik, 1998, 1999) noted the similarity between Vapnik's  $\varepsilon$ -insensitive loss function and Huber's loss in robust

\* Corresponding author.

*E-mail addresses:* cherkass@ece.umn.edu (V. Cherkassky); myq@ece.umn.edu (Y. Ma).

statistics (Huber, 1964). In particular, Vapnik's loss function coincides with a special form of Huber's loss aka least-modulus (LM) loss (with  $\varepsilon = 0$ ). From the viewpoint of traditional robust statistics, there is a well-known correspondence between the noise model and optimal loss function (Schölkopf & Smola, 2002; Smola & Schölkopf, 1998). However, this connection between the noise model and the loss function is based on (asymptotic) maximum likelihood arguments (Smola & Schölkopf, 1998). It can be argued that for finite-sample regression problems Vapnik's  $\varepsilon$ -insensitive loss (with properly chosen  $\varepsilon$ -value) may yield better generalization than other loss functions (known to be asymptotically optimal for a particular noise density). In order to test this assertion, we compare generalization performance of SVM linear regression (with optimally chosen  $\varepsilon$ ) with robust regression using LM loss function ( $\varepsilon = 0$ ) and also with optimal least squares regression, 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 selection. Section 3 describes the proposed approach for selecting SVM parameters. Section 4 presents empirical comparisons. These comparisons include regression data sets with non-linear target functions, corrupted with Gaussian noise, as well as non-Gaussian noise. Section 5 presents extensive empirical comparisons for higher dimensional linear regression problems under different settings and noise models. 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

We consider standard regression formulation under general setting for predictive learning (Cherkassky & Mulier, 1998; Hastie, Tibshirani, & Friedman, 2001; Vapnik, 1999). The goal is to estimate unknown real-valued function in the relationship:

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

where  $\delta$  is independent and identically distributed (i.i.d.) zero mean random error (noise),  $\mathbf{x}$  is a multivariate input and  $y$  is a scalar output. The estimation is made based on a finite number of samples (training data):  $(\mathbf{x}_i, y_i)$ ,  $(i = 1, \dots, n)$ . The training data are i.i.d. samples generated according to some (unknown) joint probability density function (pdf)

$$p(\mathbf{x}, y) = p(\mathbf{x})p(y|\mathbf{x}) \quad (2)$$

The unknown function in Eq. (1) is the mean of the output conditional probability (aka regression function)

$$r(\mathbf{x}) = \int yp(y|\mathbf{x})dy \quad (3)$$

A learning method (or estimation procedure) selects the 'best' model  $f(\mathbf{x}, \omega_0)$  from a set of approximating functions (or possible models)  $f(\mathbf{x}, \omega)$  parameterized by a set of parameters  $\omega \in \Omega$ . The quality of an approximation is measured by the loss or discrepancy measure  $L(y, f(\mathbf{x}, \omega))$ , and the goal of learning is to select the best model minimizing (unknown) prediction risk:

$$R(\omega) = \int L(y, f(\mathbf{x}, \omega))p(\mathbf{x}, y)d\mathbf{x} dy \quad (4)$$

It is known that the regression function (3) is the one minimizing prediction risk (4) with the squared loss function loss:

$$L(y, f(\mathbf{x}, \omega)) = (y - f(\mathbf{x}, \omega))^2 \quad (5)$$

Note that the set of functions  $f(\mathbf{x}, \omega)$ ,  $\omega \in \Omega$  supported by a learning method may or may not contain the regression function (3). Thus, the problem of regression estimation is the problem of finding the function  $f(\mathbf{x}, \omega_0)$  (regressor) that minimizes the prediction risk functional

$$R(\omega) = \int (y - f(\mathbf{x}, \omega))^2 p(\mathbf{x}, y)d\mathbf{x} dy \quad (6)$$

using only the training data. This risk functional measures the accuracy of the learning method's *predictions* of unknown target function  $r(\mathbf{x})$ .

In SVM regression, the input  $\mathbf{x}$  is first mapped onto an  $m$ -dimensional feature space using some fixed (non-linear) mapping, and then a linear model is constructed in this feature space (Cherkassky & Mulier, 1998; Smola & Schölkopf, 1998; Vapnik, 1998, 1999). Using mathematical notation, the linear model (in the feature space)  $f(\mathbf{x}, \omega)$  is given by

$$f(\mathbf{x}, \omega) = \sum_{j=1}^m \omega_j g_j(\mathbf{x}) + b \quad (7)$$

where  $g_j(\mathbf{x})$ ,  $j = 1, \dots, m$  denotes a set of non-linear transformations, and  $b$  is the 'bias' term.

Regression estimates can be obtained by minimization of the empirical risk on the training data. Typical loss functions used for minimization of empirical risk include squared error and absolute value error. SVM regression uses a new type of loss function called  $\varepsilon$ -insensitive loss proposed by Vapnik (1998, 1999):

$$L_\varepsilon(y, f(\mathbf{x}, \omega)) = \begin{cases} 0 & \text{if } |y - f(\mathbf{x}, \omega)| \leq \varepsilon \\ |y - f(\mathbf{x}, \omega)| - \varepsilon & \text{otherwise} \end{cases} \quad (8)$$

The empirical risk is:

$$R_{\text{emp}}(\omega) = \frac{1}{n} \sum_{i=1}^n L_\varepsilon(y_i, f(\mathbf{x}_i, \omega)) \quad (9)$$

Note that  $\varepsilon$ -insensitive loss coincides with LM loss and with a special case of Huber's robust loss function (Huber, 1964)

when  $\varepsilon = 0$  (Vapnik, 1998). Hence, we shall compare prediction performance of SVM (with proposed  $\varepsilon$ -value) with regression estimates obtained using LM loss ( $\varepsilon = 0$ ), for various noise densities.

SVM regression performs linear regression in the high-dimensional feature space using  $\varepsilon$ -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  $\xi_i, \xi_i^*$   $i = 1, \dots, n$ , to measure the deviation of training samples outside  $\varepsilon$ -insensitive zone. Thus, SVM regression is formulated as minimization of the following functional:

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \text{subject to } \begin{cases} y_i - f(\mathbf{x}_i, \omega) - b \leq \varepsilon + \xi_i^* \\ f(\mathbf{x}_i, \omega) + b - y_i \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (10)$$

where  $C$  is a positive constant (regularization parameter). This optimization formulation can be transformed into the dual problem (Vapnik, 1998, 1999), and its solution is given by

$$f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) + b \quad (11)$$

where the dual variables are subject to constraints  $0 \leq \alpha_i, \alpha_i^* \leq C$ , and the kernel function  $K(\mathbf{x}, \mathbf{x}')$  is a symmetric function satisfying Mercer's conditions (Vapnik, 1998, 1999). The sample points that appear with non-zero coefficients in Eq. (11) are called support vectors (SVs).

It is well known that SVM generalization performance (estimation accuracy) depends on a good setting of hyper-parameters  $C, \varepsilon$  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 hyper-parameters as user-defined inputs. In this paper we focus on the choice of  $C$  and  $\varepsilon$ , rather than on selecting the kernel function. Selecting a particular kernel type and kernel function parameters is usually based on application-domain knowledge and may reflect distribution of input ( $\mathbf{x}$ ) values of the training data (Chapelle & Vapnik, 1999; Schölkopf et al., 1999; Vapnik, 1998, 1999). For example, in this paper we show examples of SVM regression using radial basis function (RBF) kernels where the RBF width parameter reflects the distribution/range of  $\mathbf{x}$ -values of training data.

Parameter  $C$  determines the trade off between the model complexity (flatness) and the degree to which deviations larger than  $\varepsilon$  are tolerated in optimization formulation (10). For example, if  $C$  is too large (infinity), then the objective is to minimize the empirical risk (9) only, without regard to model complexity part in the optimization formulation (10).

Parameter  $\varepsilon$  controls the width of the  $\varepsilon$ -insensitive zone, used to fit the training data (Cherkassky & Mulier, 1998; Vapnik, 1998, 1999). The value of  $\varepsilon$  can affect the number of SVs used to construct the regression function. Larger  $\varepsilon$ -value result in fewer SVs selected, and result in more 'flat' (less complex) regression estimates. Hence, both  $C$  and  $\varepsilon$ -values affect model complexity (but in a different way).

Existing practical approaches to the choice of  $C$  and  $\varepsilon$  can be summarized as follows:

- Parameters  $C$  and  $\varepsilon$  are selected by users based on a priori knowledge and/or user expertise (Cherkassky & Mulier, 1998; Schölkopf et al., 1999; Vapnik, 1998, 1999). Obviously, this approach is not appropriate for non-expert users. Based on the observation that SVs lie outside the  $\varepsilon$ -tube and the SVM model complexity strongly depends on the number of SVs, Schölkopf et al. (1998) suggested that another parameter  $\nu$  (i.e. the fraction of points outside the  $\varepsilon$ -tube) should be controlled instead of  $\varepsilon$ . Under this approach, parameter  $\nu$  has to be user-defined. Similarly, Mattera and Haykin (1999) propose to choose  $\varepsilon$ -value so that the percentage of SVs in the SVM regression model is around 50% of the number of samples. However, one can easily show examples where optimal generalization performances are achieved with the number of SVs more or less than 50%.
- Kwok (2001) and Smola et al. (1998) proposed asymptotically optimal  $\varepsilon$  values which are proportional to noise variance, in agreement with general sources on SVM (Cherkassky & Mulier, 1998; Vapnik, 1998, 1999). The main practical drawback of such a proposal is that it does not reflect sample size. Intuitively, the value of  $\varepsilon$  should be smaller for larger sample sizes (when the data has the same level of noise).
- Selecting parameter  $C$  equal to the range of output values (Mattera & Haykin, 1999). 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 selection (Cherkassky & Mulier, 1998; Schölkopf et al., 1999). This approach is very computation and data-intensive.
- Several researchers have recently presented a statistical interpretation of SVM regression (Smola & Schölkopf, 1998; Hastie et al., 2001) where the loss function used for empirical risk (9) is related to particular type of additive noise in regression formulation (1). Under this approach, the value of  $\varepsilon$ -parameter can be optimally tuned for particular noise density, whereas the  $C$  parameter is interpreted as a traditional regularization parameter in formulation (10), which is usually estimated by cross-validation (Hastie et al., 2001).

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 re-sampling);
- Analytical selection of  $\varepsilon$ -parameter based on (known or estimated) level of noise in the training data, and on the (known) number of training samples.

In addition, empirical evidence presented later in this paper suggests the importance of  $\varepsilon$ -insensitive loss for finite-sample estimation, in the sense that SVM regression (with proposed parameter selection) achieves superior prediction performance compared to other (robust) loss functions, for different noise densities.

### 3. Proposed approach for parameter selection

*Selection of parameter  $C$ .* Following [Mattera and Haykin \(1999\)](#), consider standard parameterization of SVM solution given by Eq. (11), assuming that the  $\varepsilon$ -insensitive zone parameter has been (somehow) chosen. Also suppose, without loss of generality, that the SVM kernel function is bounded in the input domain. For example, RBF kernels (used in empirical comparisons presented later in Section 4) satisfy this assumption:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2p^2}\right) \quad (12)$$

where  $p$  is the width parameter.

Under these assumptions, one can relate the value of  $C$  to the range on response values of the training data. Specifically, referring to Eq. (11), note that the regularization parameter  $C$  defines the range of values  $0 \leq \alpha_i, \alpha_i^* \leq C$  assumed by dual variables used as linear coefficients in SVM solution (11). Hence, a ‘good’ value for  $C$  can be chosen equal to the range of output (response) values of training data ([Mattera & Haykin, 1999](#)). However, such a selection of  $C$  is quite sensitive to possible outliers (in the training data), so we propose instead the following prescription for regularization parameter:

$$C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) \quad (13)$$

where  $\bar{y}$  and  $\sigma_y$  are the mean and the standard deviation of the  $y$  values of training data. Proposed selection of  $C$  given by Eq. (13) coincides with prescription suggested by [Mattera and Haykin \(1999\)](#) when the data has no outliers, but yields better  $C$ -values (in our experience) when the data contains outliers.

*Selection of  $\varepsilon$ .* It is well-known that the value of  $\varepsilon$  should be proportional to the input noise level, that is  $\varepsilon \propto \sigma$  ([Cherkassky & Mulier, 1998](#); [Kwok, 2001](#); [Smola et al., 1998](#); [Vapnik, 1999](#)). Here we assume that the standard deviation of noise  $\sigma$  is known or can be estimated from data

(practical approaches to noise estimation are discussed later in Section 6). However, the choice of  $\varepsilon$  should also depend on the number of training samples: intuitively, larger sample sizes should yield smaller  $\varepsilon$ -values. Precise nature of such a dependency can be derived using a combination of simple statistical arguments followed by empirical tuning/verification, as discussed next. First, let us try to relate the value of  $\varepsilon$  to an empirical distribution of ‘errors’  $\delta_i = \hat{y}_i - y_i$ , ( $i = 1, \dots, n$ ) observed for a given training data set of size  $n$ . Consider the sample mean of these errors:

$$\hat{\delta} = \frac{1}{n}(\delta_1 + \delta_2 + \dots + \delta_n) \quad (14)$$

Random variable  $\hat{\delta}$  can be interpreted as empirical estimate of noise observed (or derived) from available training data set of size  $n$ . Hence, the choice of  $\varepsilon$  should depend on the variance of  $\hat{\delta}$ . In order to estimate the variance of  $\hat{\delta}$ , recall that component errors  $\delta_i$  in expression (14) all have zero mean and variance  $\sigma^2$  (where  $\sigma^2$  is the variance of additive noise in regression formulation (1)). According to the Central Limit Theorem, the sample mean (14) is (approximately) Gaussian with zero mean and variance  $\sigma^2/n$ . Hence, it seems reasonable to set the value of  $\varepsilon$  proportional to the ‘width’ of the distribution of  $\hat{\delta}$ :

$$\varepsilon \sim \frac{\sigma}{\sqrt{n}} \quad (15)$$

Based on a number of empirical comparisons, we found that Eq. (15) works well when the number of samples is small, however, for large values of  $n$  prescription (15) yields  $\varepsilon$ -values that are too small (practically zero). Hence, we propose the following (empirical) dependency:

$$\varepsilon \sim \sigma \sqrt{\frac{\ln n}{n}} \quad (16)$$

We do not have specific theoretical justification for factor  $\ln n$  in the above expression, other than this factor typically appears in analytical bounds used in VC theory ([Vapnik, 2001](#)). Based on the empirical tuning, we found the following practical prescription for  $\varepsilon$ :

$$\varepsilon = 3\sigma \sqrt{\frac{\ln n}{n}} \quad (17)$$

This expression provides good performance for various data set sizes, noise levels and target functions for SVM regression. Expression (17) will be used in all empirical comparisons presented in Sections 4 and 5.

### 4. Experimental results for non-linear target functions

This section presents empirical comparisons for non-linear regression, first with Gaussian noise, and then with non-Gaussian noise.

4.1. Results for Gaussian noise

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

**Training data.** Simulated training data  $(\mathbf{x}_i, y_i)$ ,  $(i = 1, \dots, n)$ , where  $\mathbf{x}$ -values are sampled on uniformly spaced grid in the input space, and  $y$ -values are generated according to statistical model (1), i.e.  $y = r(\mathbf{x}) + \delta$ . Different types of the target functions  $r(\mathbf{x})$  are used. The  $y$ -values of training data are corrupted by additive noise  $\delta$  with zero mean and standard deviation  $\sigma$ . We used Gaussian noise (for comparisons presented in this section) and several non-Gaussian symmetric noise densities (for results presented in Section 4.2). Since SVM approach is not sensitive to a particular noise distribution, we expect to observe good generalization performance with different types of noise, as long as an optimal value of  $\epsilon$  (reflecting standard deviation of noise  $\sigma$ ) has been used.

**Test data.** The test inputs are sampled randomly according to uniform distribution in  $\mathbf{x}$ -space.

**Kernel function.** RBF kernel functions (12) 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, for univariate problems, RBF width parameter is set to  $p \sim (0.1-0.5) * \text{range}(x)$ . For multivariate  $d$ -dimensional problems the RBF width parameter is set so that  $p^d \sim (0.1-0.5)$  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.** Prediction risk is defined as the mean squared error (MSE) between SVM estimates and the true values of the target function for test inputs.

Note that regression estimates themselves are random, since they are obtained using random (finite) training data. Our initial comparisons (in this section) are made for such random estimates obtained using a single random realization of training data. This is done mainly for illustration purposes (i.e. visual comparison of regression estimates obtained by different methods for the same training data set). Later, more representative comparisons (in Sections

Table 1  
Results for univariate *sinc* function (small sample size): Data Set 1–Data Set 5

Data Set	$a$	Noise level ( $\sigma$ )	C-selection	$\epsilon$ -selection	Prediction risk	%SV
1	1	0.2	1.58	0	0.0129	100
				0.2	0.0065	43.3
2	10	2	15	0	1.3043	100
				2.0	0.7053	36.7
3	0.1	0.02	0.16	0	$1.03 \times 10^{-4}$	100
				0.02	$8.05 \times 10^{-5}$	40.0
4	-10	0.2	14.9	0	0.0317	100
				0.2	0.0265	50.0
5	-0.1	0.02	0.17	0	$1.44 \times 10^{-4}$	100
				0.02	$1.01 \times 10^{-4}$	46.7

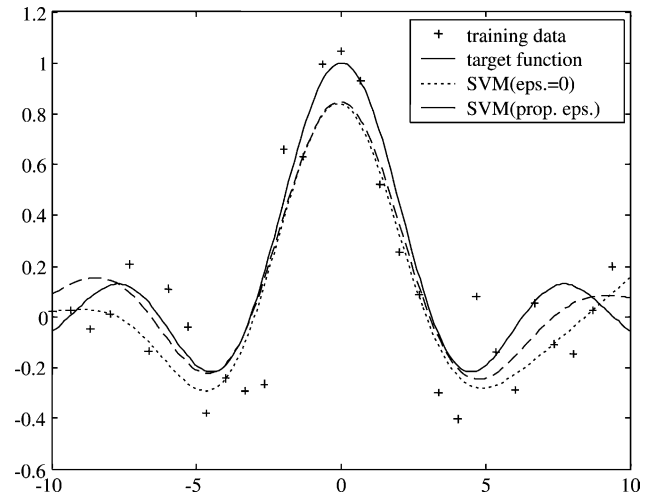


Fig. 1. Comparison of SVM estimate using proposed parameter selection versus using least-modulus loss, for Data Set 1 (*sinc* target function, 30 samples).

4.2 and 5) use regression estimates obtained using many (100) realizations of training data with the same statistical characteristics (i.e. number of samples, noise level, noise distribution, etc.). Such comparisons are presented in tables showing prediction risk (MSE) averaged over 100 realizations of random training data.

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

$$r(x) = a \frac{\sin(x)}{x} \quad x \in [-10, 10] \quad (18)$$

The following values of  $a$ : 1, 10, 0.1, -10, -0.1, were used to generate five data sets using small sample size ( $n = 30$ ) with additive Gaussian noise (with different noise levels  $\sigma$  shown in Table 1). For these data sets, we used RBF kernels with width parameter  $p = 3$ . Table 1 shows:

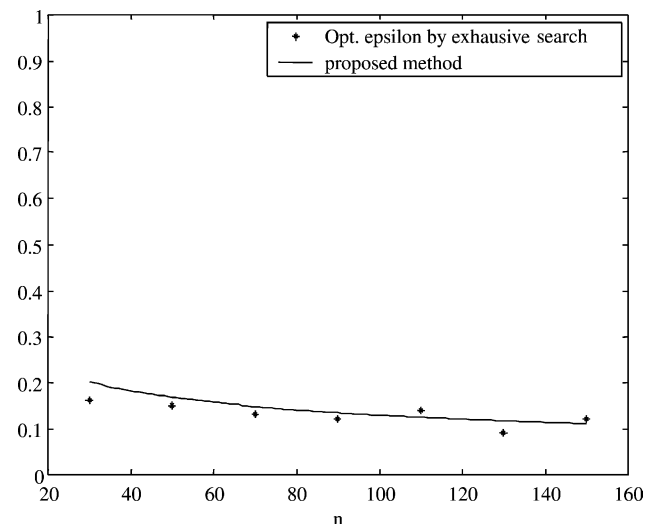


Fig. 2. Proposed  $\epsilon$ -values versus optimal  $\epsilon$ -values (providing smallest prediction risk) for Data Set 1 for different number of training samples ( $n = 30, 50, \dots, 150$ ).

- (a) Parameter values  $C$  and  $\varepsilon$  (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 %SV obtained using LM loss function ( $\varepsilon = 0$ ).

We can see that the proposed method for choosing  $\varepsilon$  is better than LM loss function, as it yields lower prediction risk and better (more sparse) representation.

Visual comparisons (for univariate *sinc* function, Data Set 1) between SVM estimates using proposed parameter selection and using LM loss are shown in Fig. 1, where the solid line is the target function, the ‘+’ denotes training data, the dotted line is an estimate using LM loss and the dashed line is the SVM estimate using proposed parameter settings.

The accuracy of expression (17) for selecting the value of  $\varepsilon$  as a function of the number of training samples ( $n$ ) is demonstrated in Fig. 2. Fig. 2 shows the proposed  $\varepsilon$ -values versus optimal  $\varepsilon$ -values (obtained by exhaustive search) for

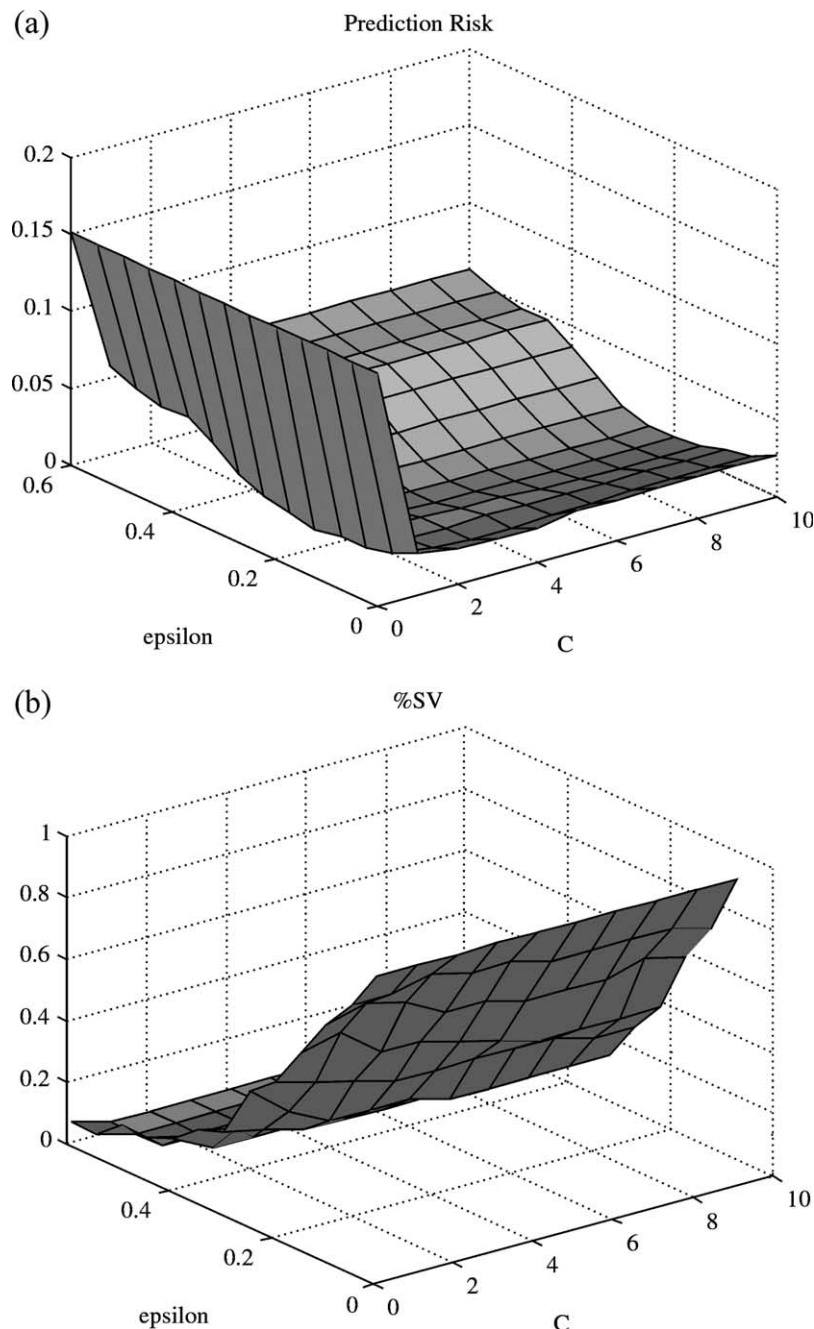


Fig. 3. Prediction risk as a function of SVM parameters. Results obtained using Data Set 1 (small sample size, *sinc* target function): (a) prediction risk and (b) percentage of SVs as a fraction of training data.

Data Set 1 with noise level  $\sigma = 0.2$  (see Table 1), for different number of training samples.

Dependence of prediction risk as a function of chosen  $C$  and  $\epsilon$ -values for Data Set 1 (i.e. *sinc* target function, 30 training samples) is shown in Fig. 3(a). Fig. 3(b) shows the %SV selected by SVM regression, which is an important factor affecting generalization performance. Visual inspection of results in Fig. 3(a) indicates that the proposed choice of  $\epsilon$ ,  $C$  yields 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 (see Fig. 3(a)). As evident from Fig. 3(b),

small  $\epsilon$ -values correspond to higher percentage of support vectors, whereas parameter  $C$  has rather negligible effect on the percentage of SV selected by SVM method.

Fig. 4 shows prediction risk as a function of chosen  $C$  and  $\epsilon$ -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  $C$ -values corresponding to SVM solutions in flat regions of prediction risk.

In order to investigate the effect of the sample size (on selection of  $\epsilon$ -value), we generated 200 training samples using univariate *sinc* target function (as in Data Set 1) with Gaussian noise ( $\sigma = 0.2$ ). Fig. 5 shows the prediction risk as

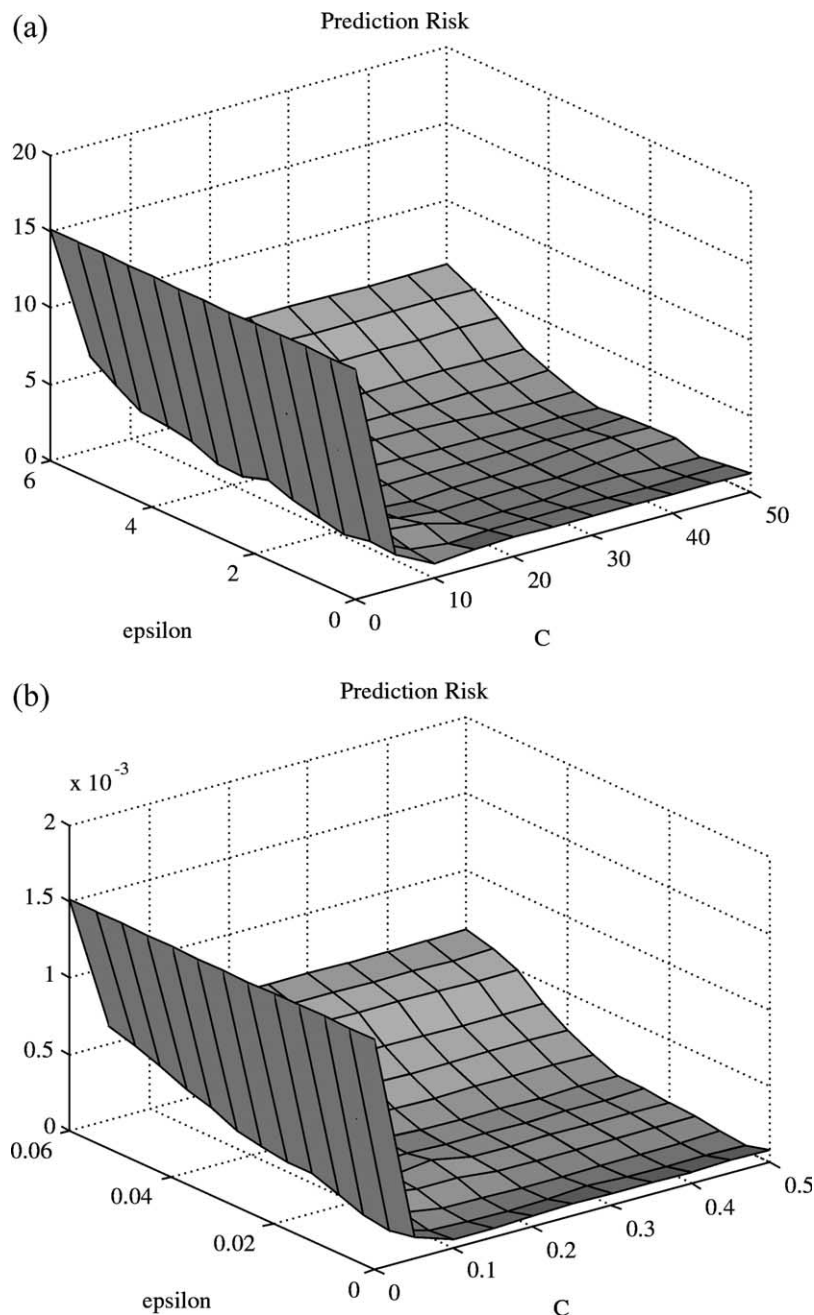


Fig. 4. Prediction risk as a function of SVM parameters (small sample size): (a) results obtained using Data Set 2 and (b) results obtained using Data Set 3.

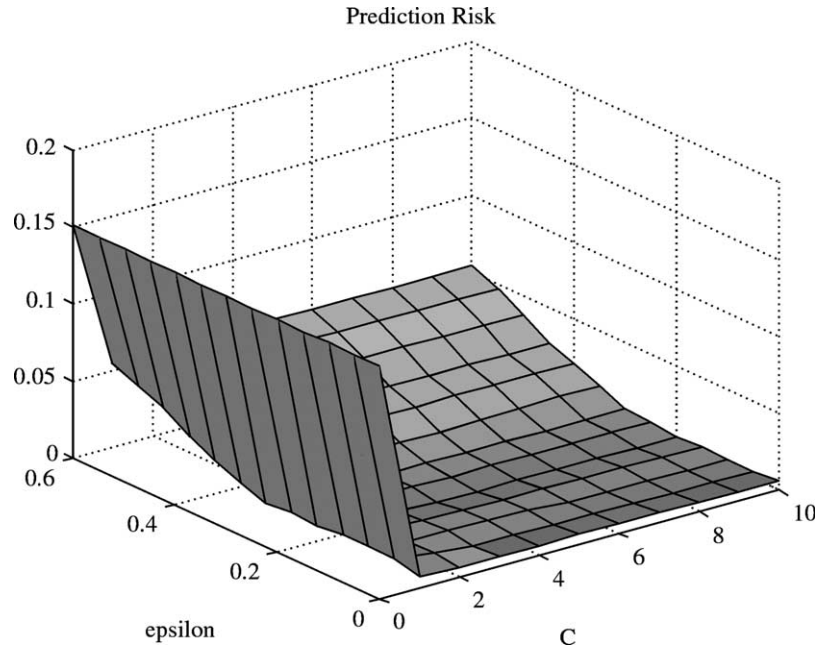


Fig. 5. Prediction risk as a function of SVM parameters (for Data Set 1: *sinc* target function, large sample size).

a function of SVM parameters for this data set (large sample size). According to proposed expression (13) and (17), the value of  $\varepsilon$  is 0.1, and  $C$  is 1.58, which is consistent with the results shown in Fig. 5. For these values of  $\varepsilon$  and  $C$ , the prediction risk is 0.0019, which compares favorably with SVM using LM loss ( $\varepsilon = 0$ ) where the prediction risk is 0.0038. Similarly, the proposed method compares favorably with selection  $\varepsilon = 0.8485\sigma$  proposed by Kwok (2001). For this data set, Kwok's method yields  $\varepsilon = 0.17$  and the prediction risk is 0.0033. According to Schölkopf and Smola (2002) asymptotically optimal  $\varepsilon = 0.612\sigma$ , which yields  $\varepsilon = 0.12$  and the prediction risk 0.0022 (for this data set). The reason that our approach to  $\varepsilon$ -selection gives better results is that all previously proposed methods for selecting  $\varepsilon$ -value (Kwok, 2001; Schölkopf & Smola, 2002; Smola et al., 1998) do not depend on sample size.

Next we show results of SVM parameter selection for multivariate regression problems. The first data set is generated using two-dimensional *sinc* target function.

$$r(\mathbf{x}) = \frac{\sin\sqrt{x_1^2 + x_2^2}}{\sqrt{x_1^2 + x_2^2}} \quad (19)$$

defined on a uniform square lattice  $[-5,5]^2$ , with response values corrupted with Gaussian noise ( $\sigma = 0.1$  and  $\sigma = 0.4$ , 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. The proposed approach selects the following values  $C = 1.16$  and  $\varepsilon = 0.05$  (for  $\sigma = 0.1$ ) and  $\varepsilon = 0.21$  (for  $\sigma = 0.4$ ). Table 2 compares SVM estimates (with proposed parameter selection) and estimates obtained using LM loss, in terms of prediction risk and the percentage of SV chosen by each method.

Finally, consider higher dimensional additive target function

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

where  $\mathbf{x}$ -values are distributed in hypercube  $[0,1]^5$ . Output (response) values of training samples are corrupted by additive Gaussian noise (with  $\sigma = 0.1$  and  $\sigma = 0.2$ ). 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 proposed method yields the value of  $C = 34$  and the value of  $\varepsilon = 0.045$  for  $\sigma = 0.1$  and  $\varepsilon = 0.09$  for  $\sigma = 0.2$ . Comparison results between the proposed methods for parameter selection with the method using LM loss function are shown in Table 3. Clearly, the proposed approach gives better performance in terms of prediction risk and robustness.

#### 4.2. Results for non-Gaussian noise

Next we present empirical results for regression problems with non-Gaussian additive symmetric noise in the statistical model (1). The main motivation is to

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

Noise level	$\varepsilon$ -selection	Prediction risk	%SV
$\sigma = 0.1$	0	0.0080	100
	Proposed	0.0020	62.7
$\sigma = 0.4$	0	0.0369	100
	Proposed	0.0229	60.9



Table 3  
Comparison of the proposed method for  $\varepsilon$ -selection with least-modulus loss ( $\varepsilon = 0$ ) for high-dimensional additive target function

Noise level	$\varepsilon$ -selection	Prediction risk	%SV
$\sigma = 0.1$	0	0.0443	100
	Proposed	0.0387	86.7
$\sigma = 0.2$	0	0.1071	100
	Proposed	0.0918	90.5

demonstrate practical advantages of Vapnik's  $\varepsilon$ -insensitive loss versus other (robust) loss functions. Specifically, we perform empirical comparisons between SVM regression (with proposed parameter selection) versus SVM regression using LM loss ( $\varepsilon = 0$ ), for several finite-sample regression problems.

We consider three types of non-Gaussian noise

- Student's  $t$ -distribution noise
- Uniform distributed noise
- Laplacian noise.

Univariate *sinc* target function is used for comparisons:

$$r(x) = \sin(x)/x \quad x \in [-10, 10]$$

Training sample size  $n = 30$ . The  $x$  values are sampled on a uniformly spaced grid in the input space. RBF kernels with width parameter  $p = 3$  are used for this data set. According to proposed expressions (13) and (17),  $C = 1.6$ ,  $\varepsilon = 0.1$  (for  $\sigma = 0.1$ ),  $\varepsilon = 0.2$  (for  $\sigma = 0.2$ ),  $\varepsilon = 0.3$  (for  $\sigma = 0.3$ ). The comparison results show prediction risk obtained using SVM regression and using LM loss, on the same data sets. In order to perform more meaningful comparisons, all comparison results are averaged using 100 random realizations of the training data.

First, consider Student's  $t$ -distribution for noise. Several experiments have been performed using various degrees of freedom (DOF) (40, 50, 100) for generating  $t$ -distribution. Empirical results indicate superior performance of the proposed method for SVM parameter selection, in comparison with LM loss regression. Table 4 shows comparisons with regression estimates obtained using LM loss for Student's noise (with 100 DOF) for different noise levels  $\sigma$ .

Table 4  
Comparison results (mean of 100 realizations) of SVM with proposed  $\varepsilon$ -selection versus least-modulus loss ( $\varepsilon = 0$ ) for  $t$ -distribution of noise (with 100 degrees of freedom)

Noise level	$\varepsilon$ -selection	Prediction risk
$\sigma = 0.1$	0	0.003
	Proposed	0.003
$\sigma = 0.2$	0	0.015
	Proposed	0.014
$\sigma = 0.3$	0	0.031
	Proposed	0.029

Table 5  
Comparison results (mean of 100 realizations) of SVM with proposed  $\varepsilon$ -selection versus least-modulus loss ( $\varepsilon = 0$ ) for uniform noise

Noise level	$\varepsilon$ -selection	Prediction risk
$\sigma = 0.1$	0	0.005
	Proposed	0.004
$\sigma = 0.2$	0	0.020
	Proposed	0.013
$\sigma = 0.3$	0	0.042
	Proposed	0.022

Second, consider uniform distribution for the additive noise. Table 5 shows comparison results for different noise levels  $\sigma$ . These results indicate superior performance of SVM method with proposed selection of  $\varepsilon$ .

Finally, we show comparison results for Laplacian noise density. Smola et al. (1998) suggest that for this noise density model, the LM loss should be used. We compare the proposed approach for choosing  $\varepsilon$  with the LM loss method. Empirical results in Table 6 indicate that for this data set, the LM loss ( $\varepsilon = 0$ ) yields better prediction accuracy than SVM loss with proposed parameter selection, in agreement with Smola et al. (1998).

## 5. Empirical results for linear regression

In this section we present empirical comparisons for several linear regression estimators using three representative loss functions: squared loss, LM and  $\varepsilon$ -insensitive loss with selection of  $\varepsilon$  given by Eq. (17). Our goal is to investigate the effect of a loss function on the prediction accuracy of linear regression with finite samples. Even though SVM regression has been extensively used for regression applications (Schölkopf et al., 1999), its success is mainly due to remarkable ability of SVM models to handle *non-linear* high-dimensional problems. However, there is little consensus and understanding of the importance of  $\varepsilon$ -insensitive loss itself for standard *linear regression* estimation. The only existing study (Drucker et al., 1997) showing empirical comparisons between SVM and ordinary least squares (OLS) for linear regression makes rather indefinite conclusions. This study applies SVM and OLS to a linear regression problem with 30 input variables, where

Table 6  
Comparison results (mean of 100 realizations) of SVM with proposed  $\varepsilon$ -selection versus least-modulus loss ( $\varepsilon = 0$ ) for Laplacian noise

Noise level	$\varepsilon$ -selection	Prediction risk
$\sigma = 0.1$	0	0.003
	Proposed	0.004
$\sigma = 0.2$	0	0.010
	Proposed	0.015
$\sigma = 0.3$	0	0.019
	Proposed	0.030

regression estimates are obtained from 60 noisy training samples, and concludes that at high noise levels SVM is better than OLS, but at low noise levels OLS is better than SVM. This study is rather sketchy since it uses a single data set for regression comparisons, and does not describe any systematic procedure for selecting the value of  $\varepsilon$ .

This section presents comparisons between three different methods, SVM, LM regression and OLS, for linear regression with finite samples. To make such comparisons ‘fair’, we use (in this section) SVM regression implementation with large (infinite)  $C$ -values in formulation (10). Hence, such SVM formulation becomes equivalent to minimization of  $\varepsilon$ -insensitive loss for the training data, without penalization (regularization) term. This enables meaningful comparisons between SVM and other formulations/loss functions (e.g. least squares) which do not use the regularization term.

All comparisons for different methods are shown for three representative unimodal noise densities: Gaussian, Laplacian and Uniform. The goal (of comparisons) is to gain better understanding of relative advantages/limitations of different methods for linear regression: optimal least squares (OLS), LM and SVM regression. Note that SVM method has a tunable parameter  $\varepsilon$  selected via analytical prescription (17) for all comparisons presented in this paper. Alternatively, optimal selection of  $\varepsilon$  can be done using re-sampling methods. We empirically compared the re-sampling approach (via cross-validation) and analytical approach for selecting the value of  $\varepsilon$ , and found no significant difference in terms of prediction accuracy of SVM estimates.

*Training data.* Simulated training data  $(\mathbf{x}_i, y_i)$ , ( $i = 1, \dots, n$ ) with random  $\mathbf{x}$ -values uniformly distributed in the input space, and  $y$ -values generated according to Eq. (1). *Target function* is high-dimensional

$$r(\mathbf{x}) = 4x_1 + 4x_2 + 3x_3 + 3x_4 + 2x_5 + x_6 + \dots + x_{20},$$

$$\mathbf{x} \in [0, 1]^{20} \quad (21)$$

*Sample size.* Various training sample sizes ( $n = 30, 40, 50$ ) are used to contrast relative performance of different methods under large sample settings and sparse sample settings. The distinction can be quantified using the ratio of the number of samples (sample size) to the number of input variables.

*Additive noise.* The following types of noise were used: Gaussian noise, uniform noise and Laplacian noise. Notice that squared loss is (asymptotically) optimal for Gaussian noise and LM loss is (asymptotically) optimal for Laplacian noise density. We also varied the noise level (as indicated by different signal-to-noise ratio (SNR) values) for high-dimensional data, in order to understand the effect of noise level on methods’ performance. SNR is defined as the ratio of the standard deviation of the true (target function) output values over the standard deviation of the additive noise.

*Experimental protocol.* For a given training sample with specified statistical properties (sample size, noise level/type,

etc. as defined above) we estimate parameters of regression via minimization of the empirical risk using three different loss functions, i.e. standard square loss, modulus loss and  $\varepsilon$ -insensitive loss (with proposed selection of  $\varepsilon$ -value). The quality of each model is evaluated as its prediction accuracy, or MSE. This quantity is measured using large number of independent test samples uniformly distributed in the input space. Specifically, 2000 test samples were used to estimate the prediction risk. Since the model itself depends on a particular (random) realization of training sample (of fixed size), its (measured) prediction accuracy is also a random variable. Hence, we repeat the experimental procedure (described above) with many different realizations of training data (100 runs) and show average prediction accuracy (risk) for methods’ comparison. Graphical presentation of prediction accuracy (risk) for three estimation methods uses the following labels: OLS (for ordinary least squares method), LM (for least-modulus method) and SVM (for SVM with  $\varepsilon$ -insensitive loss using proposed optimal selection of  $\varepsilon$ ). Notice that LM method is a special case of SVM with  $\varepsilon$ -insensitive loss (with  $\varepsilon = 0$ ).

Next we show comparisons for high-dimensional target function (21). Results shown in Fig. 6 are intended to illustrate how methods’ prediction performance depends on the sparseness of training data. This is accomplished by comparing prediction risk (MSE) for data sets with different sample sizes ( $n = 30, 40$  and  $50$ ) under the same  $\text{SNR} = 2$ . Results in Fig. 6 indicate that SVM method consistently (for all types of noise) outperforms other methods under sparse settings, i.e. for 30 samples when the ratio  $n/d$  is smaller than 2. However, for 50 samples, when this ratio is larger than 2, we approach large-sample settings, and the methods’ performance becomes similar. The distinction between sparse setting and large-sample setting is not very clear cut as it also depends on the noise level. That is why comparisons in Fig. 6 are shown for a given (fixed) SNR value for all data sets. Next we show comparisons for the same high-dimensional target function (21) under sparse setting ( $n = 30$  samples) for different noise levels ( $\text{SNR} = 1, 3, 5, 7$ ) in order to understand the effect of noise level on methods’ performance (shown in Fig. 7).

Results in Fig. 7 clearly show superiority of SVM method for large noise levels; however, for small noise levels SVM does not provide any advantages over OLS. Note that MSE results in Fig. 7 are shown on a logarithmic scale, so that the difference in prediction performance (MSE) for different methods at high noise levels ( $\text{SNR} = 1$ ) is quite significant (i.e. of the order of 100% or more).

## 6. Noise variance estimation

The proposed method for selecting  $\varepsilon$  relies on the knowledge of the standard deviation of noise  $\sigma$ . The problem, of course, is that the noise variance is not known

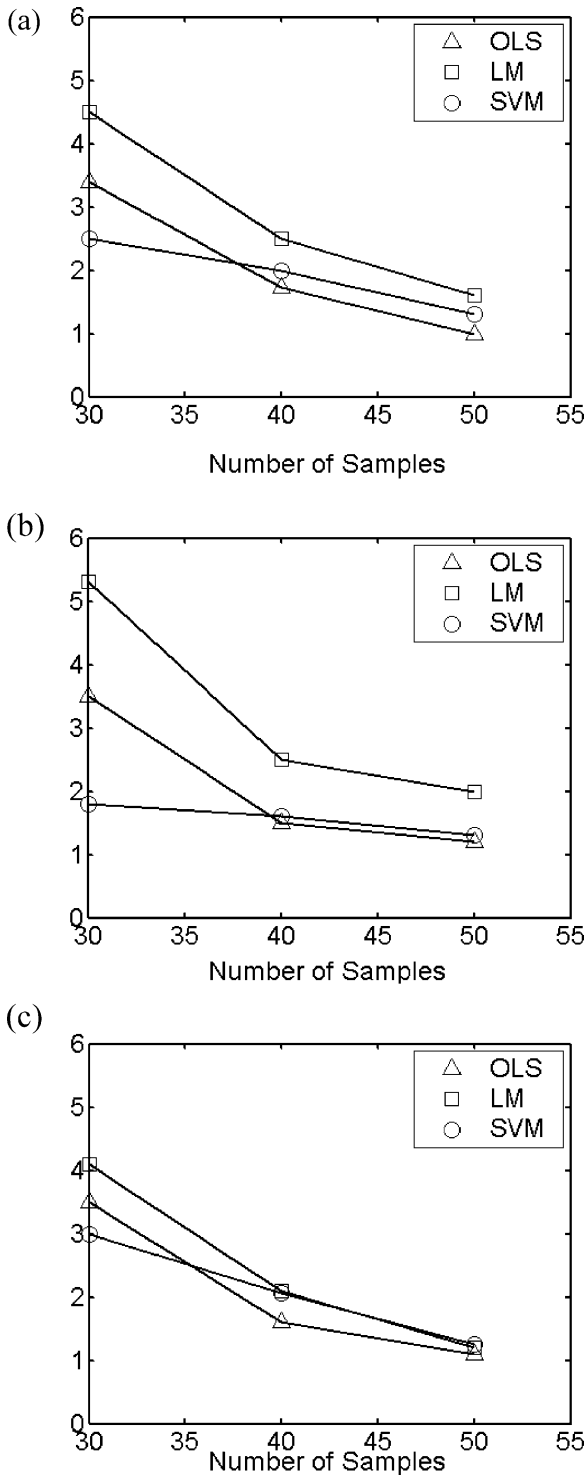


Fig. 6. Prediction accuracy versus sample size  $n = 30, 40, 50$  for high-dimensional linear regression,  $\text{SNR} = 2$  (a) Gaussian noise, (b) Uniform noise, and (c) Laplacian noise.

a priori, and it needs to be estimated from training data  $(\mathbf{x}_i, y_i), (i = 1, \dots, n)$ .

In practice, the noise variance can be readily estimated from the squared sum of residuals (fitting error) of

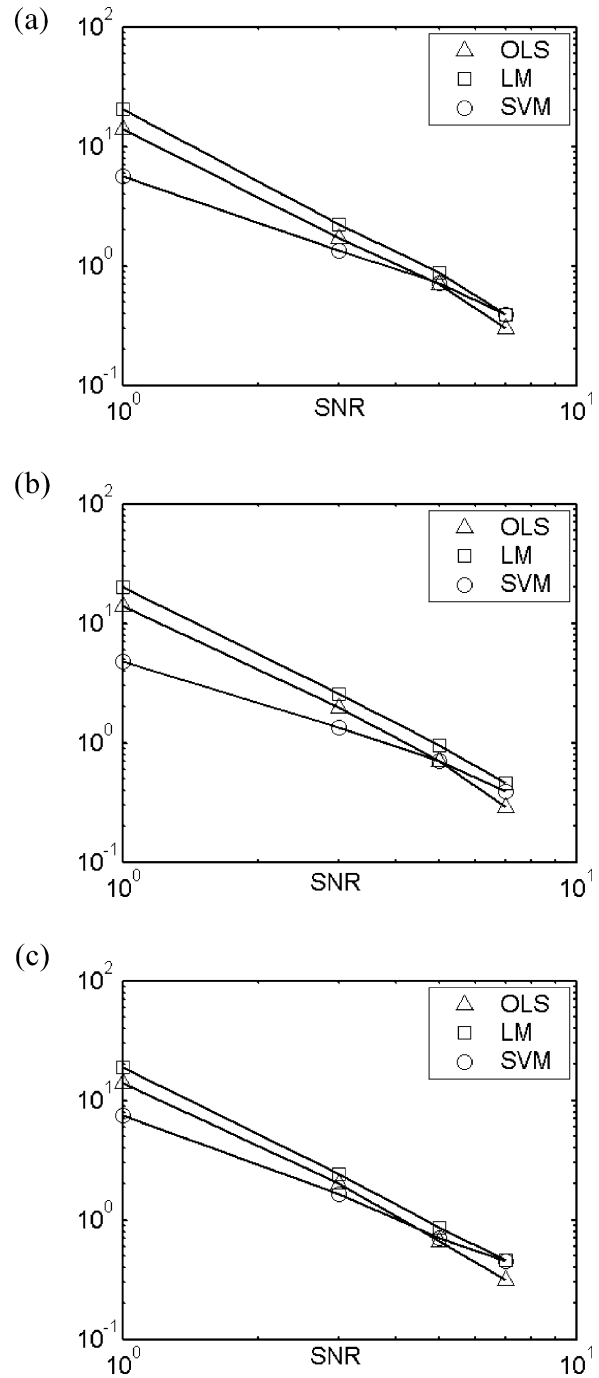


Fig. 7. Prediction accuracy versus SNR (1, 3, 5, and 7) for high-dimensional linear regression,  $n = 30$ , (a) Gaussian noise, (b) Uniform noise, and (c) Laplacian noise.

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 (Cherkassky & Mulier, 1998; Cherkassky, Shao, Mulier, & Vapnik, 1999; Hastie et al.,

2001).

$$\hat{\sigma}^2 = \frac{n}{n-d} \cdot \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (22)$$

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 (Cherkassky & Mulier, 1998).

We used expression (22) for estimating noise variance using higher-order algebraic polynomials (for univariate regression problems) and  $k$ -nearest-neighbors regression (for multivariate problems). 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. Accurate estimates of the model complexity (DOF) for  $k$ -nearest neighbors are not known, even though an estimate  $d = n/k$  is commonly used (Hastie et al., 2001). Cherkassky and Ma (2003) recently introduced new (more accurate) estimate of model complexity:

$$d = n/(n^{1/5}k) \quad (23)$$

This estimate of DOF for  $k$ -nearest-neighbors regression provides rather accurate noise estimates when used in conjunction with Eq. (22). Combining expressions (22) and (23), we obtain the following prescription for noise variance estimation via  $k$ -nearest-neighbor’s method:

$$\hat{\sigma}^2 = \frac{n^{1/5}k}{n^{1/5}k-1} \cdot \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (24)$$

Typically, small values of  $k$  (in the 2–6 range) corresponding to low-bias/high variance estimators should be used in formula (24). In order to illustrate the effect of different  $k$ -values on the accuracy of noise variance estimation, we use three-dimensional 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 noise variance  $\sigma^2 = 0.36$ . For example, for  $n = 30$ ,  $k = 3$ , the noise variance estimate is  $\hat{\sigma}^2 = 0.34$ . 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.

Since accurate estimation of noise variance does not seem to be affected much by specific  $k$ -value, 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. However, here we only show noise estimation results obtained using the univariate *sinc* target function for different levels of true noise variance 0.01, 0.04, 0.09, 0.16, 0.25, 0.36, 0.49, 0.64. Fig. 9 shows the scatter plot of noise level estimates obtained via Eq. (24) 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).

Empirical results presented in this section show how to estimate (accurately) the noise level from available training data. This underscores practical applicability of the proposed expression (17) for  $\varepsilon$ -selection. In fact, empirical results (not shown here due to space constraints) indicate that SVM estimates obtained using estimated noise level for  $\varepsilon$ -selection yield similar prediction accuracy (within 5%) to

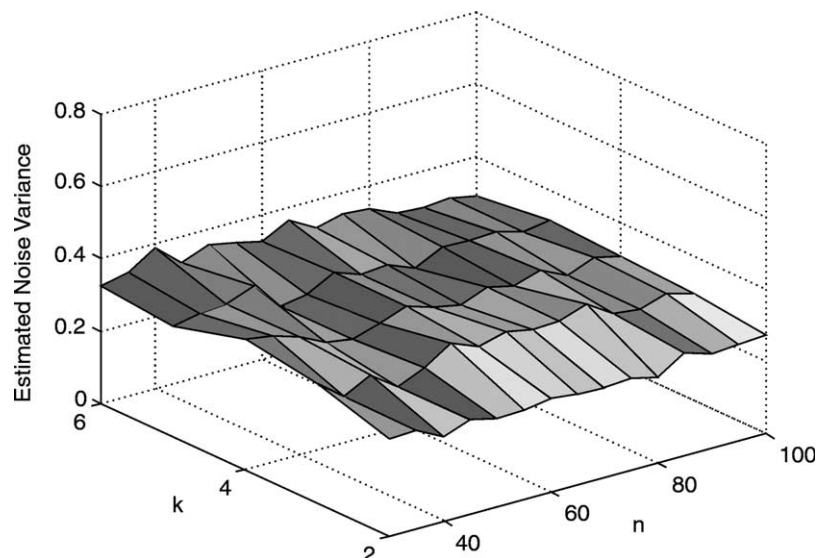


Fig. 8. Using  $k$ -nearest-neighbors method for estimating noise variance for univariate *sinc* function with different  $k$  and  $n$  values when the true noise variance is 0.36.

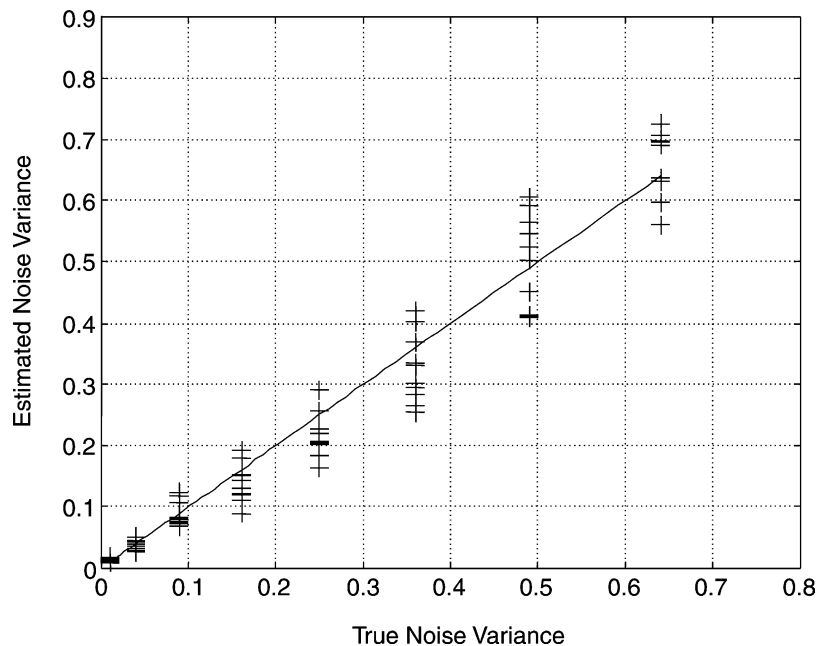


Fig. 9. Scatter plot of noise estimates obtained using  $k$ -nearest-neighbors method ( $k = 3$ ) for univariate *sinc* function for different noise levels. Results are obtained using 10 independent data sets with  $n = 30$  samples, for each noise level.

SVM estimates obtained using known noise level, for data sets in Sections 4 and 5.

## 7. Summary and discussion

This paper describes practical recommendations for setting meta-parameters for SVM regression. Namely the values of  $\varepsilon$  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, the 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 the proposed choice of  $\varepsilon$ , the value of regularization parameter  $C$  has only negligible effect on the generalization performance (as long as  $C$  is larger than a certain threshold determined analytically 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  $\varepsilon$  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

$\varepsilon$  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 types of kernels.

Another contribution of this paper is demonstrating the importance of  $\varepsilon$ -insensitive loss function for generalization performance. Several recent sources (Hastie et al., 2001; Smola & Schölkopf, 1998) assert that an optimal choice of the loss function (i.e. LM 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 asymptotic proofs. Our empirical comparisons suggest that SVM loss (with proposed  $\varepsilon$ ) outperforms other commonly used loss functions (squared loss, LM loss) for linear regression under sparse sample settings. These findings seemingly contradict an opinion that a given loss function is statistically 'optimal' for particular noise density (Hastie et al., 2001; Smola & Schölkopf, 1998). This contradiction can be explained by noting that statistical optimality proofs are based on asymptotic arguments. Indeed, our experimental results in Figs. 6 and 7 show that under large sample settings (low noise, large sample size), a given loss function clearly favors a particular noise density (according to statistical theory); however, for finite (small) samples SVM loss gives better results. Intuitively, superior performance of  $\varepsilon$ -insensitive loss for finite-sample problems can be explained by noting that noisy data samples which are 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  $\varepsilon$ ), 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 LM 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) would indeed provide superior generalization performance.

### Acknowledgements

The authors thank Dr V. Vapnik for many useful discussions. This work was supported, in part, by NSF grant ECS-0099906.

### References

- Chapelle, O., & Vapnik, V. (1999). Model selection for support vector machines (Vol. 12). *Advances in neural information processing systems*.
- Cherkassky, V., & Ma, Y. (2003). Comparison of model selection for regression. *Neural Computation*, 15 (7), 1691–1714.
- Cherkassky, V., & Mulier, F. (1998). *Learning from data: Concepts, theory, and methods*. New York: Wiley.
- Cherkassky, V., Shao, X., Mulier, F., & Vapnik, V. (1999). Model complexity control for regression using VC generalization bounds. *IEEE Transaction on Neural Networks*, 10(5), 1075–1089.
- Drucker, H., Burges, C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. In M. Moser, J. Jordan, & T. Petsche (Eds.), (Vol. 9) (pp. 155–161). *Neural information processing systems*, Cambridge, MA: MIT Press.
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*. Berlin: Springer.
- Huber, P. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35, 73–101.
- Kwok, J. T. (2001). Linear dependency between  $\varepsilon$  and the input noise in  $\varepsilon$ -support vector regression. In G. Dorffner, H. Bishof, & K. Hornik (Eds.), *ICANN 2001* (pp. 405–410). *LNCSS 2130*.
- Mattera, D., & Haykin, S. (1999). Support vector machines for dynamic reconstruction of a chaotic system. In B. Schölkopf, J. Burges, & A. Smola (Eds.), *Advances in kernel methods: Support vector machine*. Cambridge, MA: MIT Press.
- Muller, K., Smola, A., Ratsch, G., Schölkopf, B., Kohlmorgen, J., & Vapnik, V. (1999). Using support vector machines for time series prediction. In B. Schölkopf, J. Burges, & A. Smola (Eds.), *Advances in kernel methods: Support vector machine*. Cambridge, MA: MIT Press.
- Schölkopf, B., Bartlett, P., Smola, A., & Williamson, R. (1998). Support vector regression with automatic accuracy control. In L. Niklasson, M. Bodén, & T. Ziemke (Eds.), *Proceedings of ICANN'98* (pp. 111–116). *Perspectives in neural computing*, Berlin: Springer.
- Schölkopf, B., Burges, J., & Smola, A. (1999). *Advances in kernel methods: Support vector machine*. Cambridge, MA: MIT Press.
- Schölkopf, B., & Smola, A. (2002). *Learning with kernels: Support vector machines, regularization, and beyond*. Cambridge, MA: MIT Press.
- Smola, A., Murata, N., Schölkopf, B., & Muller, K. (1998). Asymptotically optimal choice of  $\varepsilon$ -loss for support vector machines. *Proceedings of ICANN 1998*.
- Smola, A., & Schölkopf, B. (1998). *A tutorial on support vector regression*. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK.
- Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.
- Vapnik, V. (1999). *The nature of statistical learning theory* (2nd ed). Berlin: Springer.
- Vapnik, V. (2001). Personal communication.