Applying the Bayesian Evidence Framework to ν-Support Vector Regression

Martin H. Law and James T. Kwok

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay
Hong Kong
{martin,jamesk}@cs.ust.hk

Abstract. Following previous successes on applying the Bayesian evidence framework to support vector classifiers and the ϵ-support vector regression algorithm, in this paper we extend the evidence framework also to the ν-support vector regression (ν-SVR) algorithm. We show that ν-SVR training implies a prior on the size of the ϵ-tube that is dependent on the number of training patterns. Besides, this prior has properties that are in line with the error-regulating behavior of ν. Under the evidence framework, standard ν-SVR training can then be regarded as performing level one inference, while levels two and three allow automatic adjustments of the regularization and kernel parameters respectively, without the need of a validation set. Furthermore, this Bayesian extension allows computation of the prediction intervals, taking uncertainties of both the weight parameter and the ϵ-tube width into account. Performance of this method is illustrated on both synthetic and real-world data sets.

1 Introduction

Recently, there is growing interest in using the support vector machines (SVMs) for various classification and regression problems. SVMs are motivated by results from statistical learning theory [8], and have shown superior performance over traditional techniques in many practical applications. The two commonly used SVM methods for regression problems are the ϵ-support vector regression (ϵ-SVR) algorithm [7] and the ν-support vector regression (ν-SVR) algorithm [5]. In particular, ν-SVR, which will be the focus in this paper, has the advantage of being able to automatically adjust the width of the ϵ-tube.

However, to obtain a high level of performance, some parameters in the ν-SVR algorithm still have to be tuned. These include 1) a kernel parameter that helps to define the feature space; and 2) two regularization parameters that determine the tradeoff between the value of ϵ, the training accuracy and model complexity. Data-resampling techniques such as cross-validation can be used, but they are usually very expensive in terms of computation and/or data.

In this paper, we address this issue by adopting the Bayesian approach. In general, the Bayesian approach is attractive in being logically consistent,

simple and flexible. Recently, various Bayesian techniques have been applied to SVMs for classification problems. Here, we follow [2, 3] in applying the evidence framework [4] to ν-SVR. The evidence framework is divided into three levels of inference, and is computationally equivalent to the type II maximum likelihood method in Bayesian statistics. Its use in feedforward neural networks has allowed the automatic selection of regularization parameters and network architectures, without the need of a validation set [4].

The rest of this paper is organized as follows. Brief overviews of the ν-SVR and the evidence framework will be given in Sections 2 and 3 respectively. Application of the evidence framework to ν-SVR will be discussed in Section 4. Simulation results are presented in Section 5, and the last section gives some concluding remarks.

2 ν-Support Vector Regression

In this section, we first introduce ϵ-SVR and then present ν-SVR as an improvement. Interested readers may consult [7, 8] for details.

Let the training data D be $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with input $\mathbf{x}_i$ and output $y_i \in \Re$. In ϵ-SVR, $\mathbf{x}$ is first mapped to $\mathbf{z} = \phi(\mathbf{x})$ in feature space $\mathcal{F}$, then a linear function $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{z} + b$ is constructed in $\mathcal{F}$ such that it deviates least from the training data according to the ϵ-insensitive loss function

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

while at the same time is as "flat" as possible (i.e., $\|\mathbf{w}\|$ is as small as possible). Mathematically, this means minimizing $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^N(\xi_i + \xi_i^*)$, subject to

$$y_i - f_i \le \epsilon + \xi_i^*, \quad f_i - y_i \le \epsilon + \xi_i, \quad \xi_i, \xi_i^* \ge 0, \tag{1}$$

for $1 \le i \le N$, where $f_i = f(\mathbf{x}_i, \mathbf{w})$ and C is a user-defined constant. Notice that after training, those nonzero ξ_i's and ξ_i^*'s will be exactly equal to the difference between the corresponding y_i and f_i.

A drawback of ϵ-SVR is that ϵ can be difficult to tune. ν-SVR alleviates this problem by trading off ϵ against model complexity and training error using a parameter $\nu > 0$. Mathematically, the problem now becomes

$$\underset{\mathbf{w},\epsilon,\xi_i,\xi_i^*}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\nu\epsilon + \frac{1}{N}\sum_{i=1}^N(\xi_i + \xi_i^*)\right), \tag{2}$$

subject to (1) and $\epsilon \ge 0$. Schölkopf *et al.* [5] showed that asymptotically, with probability 1, ν equals both the fraction of data points that become support vectors and the fraction of data points lying outside the ϵ-tube[1]. This suggests that in situations where some prior knowledge on these fractions is available, ν may be easier to adjust than ϵ.

[1] Notice that data points lying outside the ϵ-tube are also the error patterns.

3 The Evidence Framework

A model $\mathcal{H}$, with a k-dimensional parameter vector $\mathbf{w}$, consists of its functional form f, the distribution $p(D|\mathbf{w}, \beta, \mathcal{H})$ that the model makes about the data, and a prior parameter distribution $p(\mathbf{w}|\alpha, \mathcal{H})$. Here, α and β are usually called the hyper-parameters. Moreover, $p(D|\mathbf{w}, \beta, \mathcal{H})$ and $p(\mathbf{w}|\alpha, \mathcal{H})$ are often written as:

$$p(D|\mathbf{w}, \beta, \mathcal{H}) \propto \exp(-\beta E_D(D|\mathbf{w}, \mathcal{H})), \tag{3}$$

$$p(\mathbf{w}|\alpha, \mathcal{H}) \propto \exp(-\alpha E_W(\mathbf{w}|\mathcal{H})). \tag{4}$$

3.1 Level 1 Inference

The evidence framework is divided into three levels of inference. For given values of α and β, the first level infers the posterior distribution of $\mathbf{w}$ by the Bayes rule: $p(\mathbf{w}|D, \alpha, \beta, \mathcal{H}) \propto p(D|\mathbf{w}, \beta, \mathcal{H})p(\mathbf{w}|\alpha, \mathcal{H})$. Substituting in (3) and (4), finding the *maximum a posteriori* (MAP) estimate $\mathbf{w}_{MP}$ of $\mathbf{w}$ is then the same as minimizing $M(\mathbf{w}) \equiv \alpha E_W + \beta E_D$.

3.2 Level 2 Inference

The second level of inference determines α and β by maximizing $p(\alpha, \beta|D, \mathcal{H})$. When $p(\alpha, \beta|\mathcal{H})$ is flat, the *evidence* $p(D|\alpha, \beta, \mathcal{H})$ can be used instead. By approximating $p(\mathbf{w}|D, \alpha, \beta, \mathcal{H})$ with a Gaussian at $\mathbf{w}_{MP}$, it can be shown that

$$\log p(D|\alpha, \beta, \mathcal{H}) = -\alpha E_W^{MP} - \beta E_D^{MP} - \frac{1}{2}\log\det\mathbf{A} + \frac{k}{2}\log\alpha + \frac{N}{2}\log\beta - \frac{N}{2}\log 2\pi,$$

where $\mathbf{A} = \frac{\partial^2 M}{\partial \mathbf{w}^2}$ is the hessian matrix, E_W^{MP} and E_D^{MP} are the values of E_W and E_D evaluated at $\mathbf{w}_{MP}$. Optimal values of α and β (denoted by α_{MP} and β_{MP} respectively) are then obtained by iterating the process of finding $\mathbf{w}_{MP}$ from the level 1 inference and estimating α, β by maximizing $p(D|\alpha, \beta, \mathcal{H})$ above.

3.3 Level 3 Inference

The third level of inference ranks different models by $p(\mathcal{H}|D)$. Assuming a flat $p(\mathcal{H})$ for all models, different models are then rated by $p(D|\mathcal{H})$. This can be computed by again taking a Gaussian approximation for $p(D|\alpha, \beta, \mathcal{H})$ at $(\alpha_{MP}, \beta_{MP})$.

4 Applying the Evidence Framework to ν-SVR

4.1 Probabilistic Model and Level 1 Inference

The ϵ-insensitive cost function corresponds to the following noise model [7]:

$$p(y_i|\mathbf{x}_i, \mathbf{w}, \epsilon, \beta, \mathcal{H}) = \frac{\beta}{2(1+\epsilon\beta)}\exp(-\beta|y_i - f_i|_\epsilon). \tag{5}$$

Notice that [7] does not have the factor β, but is added here to play the important role of controlling the noise variance. For $\mathbf{w}$, we adopt the commonly-used weight decay prior

$$p(\mathbf{w}|\alpha, \mathcal{H}) = (\frac{\alpha}{2\pi})^{\frac{k}{2}} \exp(-\frac{\alpha}{2}\|\mathbf{w}\|^2). \tag{6}$$

Comparing (5), (6) with (3), (4), we see that $E_D(D|\mathbf{w}, \mathcal{H}) = \sum_{i=1}^{N} |y_i - f_i|_\epsilon$, and $E_W(\mathbf{w}|\mathcal{H}) = \frac{1}{2}\|\mathbf{w}\|^2$. For ϵ, we adopt the following prior

$$p(\epsilon|\beta, \nu, \mathcal{H}) = c(\beta, \nu)(1 + \epsilon\beta)^N \exp(-N\beta\nu\epsilon), \tag{7}$$

where $c(\beta, \nu) = (N\nu)^{N+1}\beta/(\exp(N\nu)\Gamma(N+1, N\nu))$ is the normalization factor. Here, $\Gamma(a, x) = \int_x^\infty t^{a-1} \exp(-t)\, dt$ is related to the incomplete gamma function. Justifications for this prior will be discussed in Section 4.2.

Assuming that the patterns are i.i.d., then by using the Bayes rule, finding the MAP estimates of $\mathbf{w}$ and ϵ is equivalent to

$$\underset{\mathbf{w},\epsilon}{\text{maximize}} \quad \log p(\mathbf{w}|\alpha, \mathcal{H}) + \log p(\epsilon|\beta, \nu, \mathcal{H}) + \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i, \mathbf{w}, \epsilon, \beta, \mathcal{H}).$$

Plugging in (5), (6) and (7), this becomes

$$\underset{\mathbf{w},\epsilon}{\text{minimize}} \quad \frac{\alpha}{2}\|\mathbf{w}\|^2 + N\beta\left(\nu\epsilon + \frac{1}{N}\sum_{i=1}^{N} |y_i - f_i|_\epsilon\right).$$

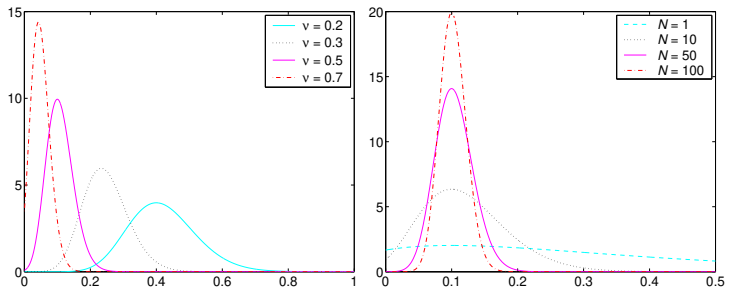
Setting $C = N\beta/\alpha$, we obtain the same optimization problem as in (2). Thus, ν-SVR training can be regarded as performing level 1 inference under this probabilistic model.

4.2 Properties of the Prior $p(\epsilon|\beta, \nu, \mathcal{H})$

Assuming that the model is unbiased and the true noise really follows (5), the probability that a particular $\mathbf{x}$ will have its corresponding y lying outside the ϵ-tube (and thus considered as an error) is equal to

$$1 - \int_{-\epsilon}^{\epsilon} \frac{\beta}{2(1 + \epsilon\beta)} \exp(-\beta \cdot 0) d\delta = \frac{1}{1 + \epsilon\beta}.$$

Now, at the peak of $p(\epsilon|\beta, \nu, \mathcal{H})$, $d \log p(\epsilon|\beta, \nu, \mathcal{H})/d\epsilon = 0$, which gives $\nu = 1/(1 + \epsilon\beta)$. Thus, the prior prefers the value of ϵ such that ν equals the fraction of errors. This is in line with the observation that ν equals the fraction of errors asymptotically with probability one [5]. Moreover, as ν increases, the prior distribution shifts towards zero (Figure 1a). If $\nu = 1$, then $p(\epsilon|\beta, \nu, \mathcal{H})$ has maximum at $\epsilon = 0$. When ν increases beyond 1, $p(\epsilon|\beta, \nu, \mathcal{H})$ becomes strictly decreasing with ϵ. These observations are also in line with [5] in that when $\nu > 1$, the value of ϵ obtained after training will be zero.



(a) different ν's ($N = 25$) (b) different N's ($\nu = 0.5$)

Fig. 1. Variations of $p(\epsilon|\beta, \nu, \mathcal{H})$ with ν and N ($\beta = 10$).

Moreover, notice that ϵ appears as a product with β at the peak of $p(\epsilon|\beta, \nu, \mathcal{H})$. Hence, the preferred value of ϵ by the prior is inversely proportional to β. Or, in other words, this preferred ϵ has a linear dependency with the scale parameter $1/\beta$ of the noise model in (5). In fact, it can be shown that the mean value of ϵ under this prior distribution also exhibits the same trend. These are thus in line with the observation in [6] that there is a linear scaling between the optimal ϵ and the noise in the data.

Besides, unlike traditional Bayesian priors, this prior depends on the size of training data and becomes more concentrated around the peak with more training data (Figure 1b). Notice that while the effect of a traditional Bayesian prior diminishes with the arrival of more data, the logarithm of the prior in (7) scales linearly with N just like the log likelihood. Hence, our prior belief, which embodies the idea that ν governs the fraction of errors, will not be overwhelmed with the arrival of more data.

4.3 Level 2 inference

The second level of inference determines the hyperparameters α, β and ν by maximizing $p(\alpha, \beta, \nu|D, \mathcal{H}) \propto p(D|\alpha, \beta, \nu, \mathcal{H})p(\alpha, \beta, \nu|\mathcal{H})$. When $p(\alpha, \beta, \nu|\mathcal{H})$ is a flat prior, the evidence for α, β and ν, $p(D|\alpha, \beta, \nu, \mathcal{H})$, can be used instead. Now, $\mathcal{E} \equiv \log p(D|\alpha, \beta, \nu, \mathcal{H})$ can be obtained by integrating out $\mathbf{w}$ and ϵ, as:

$$\mathcal{E} = \log \int P(D|\mathbf{w}, \epsilon, \beta, \mathcal{H}) P(\mathbf{w}|\alpha, \mathcal{H}) P(\epsilon|\beta, \nu, \mathcal{H}) \, d\mathbf{w} \, d\epsilon. \tag{8}$$

As in [4], we approximate the joint posterior distribution of $\mathbf{w}$ and ϵ using a single Gaussian at their MAP values ($\mathbf{w}_{MP}$ and ϵ_{MP} respectively). This entails computation of the hessian matrix $\mathbf{A} = \frac{\partial^2}{\partial[\mathbf{w}\epsilon]^2}(\alpha E_W + \beta E_D + N\beta\nu\epsilon)$, and consequently derivatives of E_D. However, E_D is not differentiable due to the

non-smooth nature of the underlying ϵ-insensitive loss function. Hence, similar to [3], we approximate this loss function by a smooth function

$$V(u) = \varsigma_\eta(u-\epsilon) + \varsigma_\eta(-u-\epsilon), \tag{9}$$

where $\varsigma_\eta(u) = \frac{1}{\eta}\log\big(1+\exp(\eta u)\big), \eta > 0$.

Define $r(u) \equiv d^2\varsigma_\eta(u)/du^2 = \eta/((1+e^{\eta u})(1+e^{-\eta u}))$, $r_i \equiv r(y_i - f_i - \epsilon) + r(f_i - y_i - \epsilon)$ and $s_i \equiv r(y_i - f_i - \epsilon) - r(f_i - y_i - \epsilon)$. To obtain $\mathcal{E}$, we can first solve the eigen system[2] for the $N \times N$ matrix $\mathbf{K}$ with entries $r_i K(\mathbf{x}_i, \mathbf{x}_j)$, where K is the kernel function. Denote the resultant n non-zero eigenvalues and the corresponding eigenvectors by ρ_j and $\mathbf{u}_j$ respectively. Then,

$$\begin{aligned}\mathcal{E} = &-\alpha E_W^{MP} - \beta E_D^{MP} - N\beta\nu\epsilon_{MP} + \frac{n}{2}\log\alpha - \frac{1}{2}\sum_{j=1}^{n}\log(\alpha+\beta\rho_j) \\ &+(N+1)\log\beta - N\nu + (N+1)\log(N\nu) - \log\Gamma(N+1, N\nu) - \frac{1}{2}\log\psi \\ &+\log\left(1+\mathrm{erf}(\sqrt{\frac{\psi}{2}}\epsilon_{MP})\right) - (N+1)\log 2 + \frac{1}{2}\log(2\pi),\end{aligned} \tag{10}$$

where

$$\psi = \beta\sum_{i=1}^{N} r_i - \frac{\beta^2}{\alpha}\Big(\sum_{i=1,j=1}^{N} s_i s_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{j=1}^{n}\frac{\beta\rho_j}{\alpha+\beta\rho_j}\big(\sum_{l=1}^{N} u_{jl}\sum_{i=1}^{N} s_i K(\mathbf{x}_i, \mathbf{x}_l)\big)^2\Big), \tag{11}$$

and $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)\,dt$ is the error function.

Iteration formula for α, β and ν to maximize $\mathcal{E}$ can be devised. However, they will not be described here because of the lack of space.

4.4 Level 3 inference

The third level of inference rates different models by their $p(D|\mathcal{H})$. This, again, is obtained by integrating out α, β and ν, as:

$$p(D|\mathcal{H}) = \int p(D|\alpha,\beta,\nu,\mathcal{H})p(\alpha|\mathcal{H})p(\beta|\mathcal{H})p(\nu|\mathcal{H})d\alpha\; d\beta\; d\nu.$$

As α and β are scale parameters, we take uniform priors over $\log\alpha$ and $\log\beta$, whereas for ν we take a uniform prior over $[0,1]$. As in the level 2 inference, we also adopt the Gaussian approximation for $p(D|\alpha,\beta,\nu,\mathcal{H})$. It can be shown that $\log p(D|\mathcal{H})$ is given by:

$$\begin{aligned}\log p(D|\mathcal{H}) = &\log p(D|\alpha_{MP},\beta_{MP},\nu_{MP},\mathcal{H}) - \frac{1}{2}\log(\alpha_{MP}E_W^{MP}) - \frac{1}{2}\log\kappa \\ &+\log\mathrm{erf}\Big(\sqrt{\frac{\kappa}{2\beta_{MP}(E_D^{MP}+N\nu_{MP}\epsilon_{MP})}}\Big) + \text{constant},\end{aligned} \tag{12}$$

[2] Solving this eigen system takes $O(N^3)$ time. However, notice that r_i is significant only when $|y_i - f_i| - \epsilon$ is small. Hence, we need only include into $\tilde{\mathbf{K}}$ those patterns that are lying close to the edges of the ϵ-tube.

where $\kappa = \beta_{MP}(E_D^{MP} + N\nu_{MP}\epsilon_{MP})(\frac{N+1}{\nu_{MP}^2} - \triangle(\triangle + \frac{N(1-\nu_{MP})}{\nu_{MP}})) - (N\nu_{MP}\beta_{MP})^2$,

$$\triangle = N\exp(-\log\Gamma(N+1, N\nu) + N\log(N\nu) - N\nu). \tag{13}$$

4.5 Posterior Predictive Distribution and Error Bars

To compute the posterior predictive distribution $p(y|\mathbf{x}, D, \mathcal{H})$ at a particular $\mathbf{x}$, the Bayesian approach again requires integrating out the posterior uncertainty of $\mathbf{w}$ and ϵ from $p(y|\mathbf{x}, D, \mathbf{w}, \epsilon, \mathcal{H})$. As mentioned in Section 4.3, the joint posterior distribution of $\mathbf{w}$ and ϵ is approximated by a single Gaussian at their MAP values. This joint distribution can be decomposed into the product of $p(\mathbf{w}|D, \epsilon, \mathcal{H})$ and $p(\epsilon|D, \mathcal{H})$. It can be shown that $p(\mathbf{w}|D, \epsilon, \mathcal{H})$ is a Gaussian distribution with mean $\mathbf{w}_{MP} + \mathbf{A}_{11}^{-1}\mathbf{a}_{12}(\epsilon - \epsilon_{MP})$, and $p(\epsilon|D, \mathcal{H})$ is a normal distribution with peak at ϵ_{MP} and truncated at $\epsilon = 0$ (because $\epsilon \geq 0$). Here $\mathbf{A}_{11}$ and $\mathbf{a}_{12}$ are sub-matrices of the hessian matrix $\mathbf{A}$.

The mean of the posterior predictive distribution can be computed as:

$$\begin{aligned}
&\int yp(y|\mathbf{x}, D, \mathbf{w}, \epsilon, \mathcal{H})p(\mathbf{w}, \epsilon|D, \mathcal{H})d\mathbf{w}\, d\epsilon\, dy \\
&= \int\left(\int yp(y|\mathbf{x}, D, \mathbf{w}, \epsilon, \mathcal{H})p(\mathbf{w}|D, \epsilon, \mathcal{H})d\mathbf{w}dy\right)p(\epsilon|D, \mathcal{H})d\epsilon \\
&= \int(\mathbf{z}^T\left(\mathbf{w}_{MP} + \mathbf{A}_{11}^{-1}\mathbf{a}_{12}(\epsilon - \epsilon_{MP})\right) + b)p(\epsilon|D, \mathcal{H})d\epsilon \\
&= f(\mathbf{x}, \mathbf{w}_{MP}) + \mathbf{z}^T\mathbf{A}_{11}^{-1}\mathbf{a}_{12}(E[\epsilon] - \epsilon_{MP}),
\end{aligned}$$

where $E[\epsilon] = \epsilon_{MP} + \int \epsilon p(\epsilon|D, \mathcal{H})d\epsilon = p(\epsilon = 0|D, \mathcal{H})/\psi$ (recall that ψ is given by (11)), while

$$\mathbf{z}^T\mathbf{A}_{11}^{-1}\mathbf{a}_{12} = \beta\sum_{i=1} s_i\left(\frac{1}{\alpha}K(\mathbf{x}\text{'}\mathbf{x}_i) - \frac{1}{\alpha}\sum_{j=1}^{n}\frac{\beta\rho_j}{\alpha + \beta\rho_j}(\sum_{l=1}^{N}u_{jl}K(\mathbf{x}, \mathbf{x}_l))(\sum_{l=1}^{N}u_{jl}K(\mathbf{x}_i, \mathbf{x}_l))\right).$$

Similarly, the variance $\sigma_y^2(\mathbf{x})$ of the predictive distribution can be shown to be:

$$\begin{aligned}
&\int y^2p(y|\mathbf{x}, D, \mathbf{w})p(\mathbf{w}, \epsilon|D, \mathcal{H})d\mathbf{w}\, d\epsilon\, dy - \left(\int yp(y|\mathbf{x}, D, \mathbf{w})p(\mathbf{w}, \epsilon|D, \mathcal{H})d\mathbf{w}\, d\epsilon\, dy\right)^2 \\
&= \mathbf{z}^T\mathbf{A}_{11}^{-1}\mathbf{z} + (\mathbf{z}^T\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^2\mathrm{VAR}[\epsilon] \\
&\quad + \frac{1}{3\beta^2}\Big(\beta^2E[\epsilon^2]^2 + 2\beta E[\epsilon] + 4 + \int_{-\sqrt{\psi}\epsilon_{MP}}^{\infty}\frac{2c(2\pi)^{-1/2}}{(\frac{\beta}{\sqrt{\psi}}t + \beta\epsilon_{MP} + 1)}\exp(-\frac{1}{2}t^2)dt,\Big)
\end{aligned}$$

where $c = 1/\Phi(-\epsilon_{MP}\sqrt{\psi})$ with $\Phi(z)$ being the area under the Gaussian distribution $N(0, 1)$ from z to ∞, $\mathrm{VAR}[\epsilon^2] = \epsilon_{MP}(E[\epsilon] - \epsilon_{MP}) + \frac{1}{\psi}$, and the last integral can be computed by numerical integration.

5 Result

In this section, we report results on applying the evidence framework to ν-SVR, using the Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\omega\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. Three data sets are used in the experiments. Two of them are real-world data from the UCI machine learning repository [1].

The first one is the toy data $\mathrm{sinc}(x) = \sin(x)/x$, with x uniformly distributed over $[-12, 12]$. In one set of experiments, we add Gaussian noise $\mathrm{N}(0, 0.05^2)$ to the outputs; while in another set, we add noise following the noise model in (5), with $\epsilon = 0.05$ and $\beta = 50$. The test set has 10,000 patterns. The experiment is repeated using 10 independent training sets, each of size 80.

The second one is the abalone data, and the task is to predict the ages of the abalones based on 8 input attributes. We used 256 patterns for training and the remaining 3921 patterns for testing. The experiment is repeated 25 times.

The third data set is the Boston housing data, and the task is to predict housing values in the Boston suburbs using 13 input attributes. We used 128 patterns for training and the remaining 378 patterns for testing. The experiment is repeated 10 times.

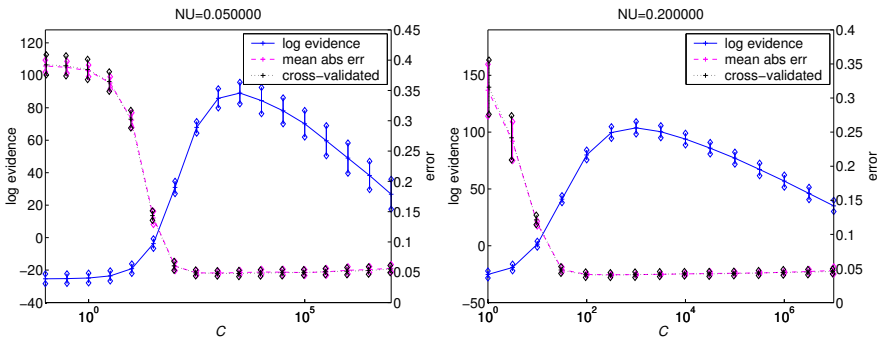
5.1 Choosing the Regularization and Kernel Parameters

Figure 2 illustrates the choice of $C = N\beta/\alpha$ for a particular ν by using level 2 inference. Figure 3 illustrates the choice of the kernel parameter ω by using level 3 inference. As can be seen from these figures, evidence from the level 2 and level 3 inference follows the testing error closely.

For comparison, 10-fold cross-validation has also been performed, and the average validation set errors are reported in Figures 2 and 3. As can be seen, cross-validation also yields a close match to the testing error. However, this procedure is very computationally demanding[3]. For example, in preparing the curve in Figure 2 for the sinc data, we experimented with 17 values of C and 11 values of ν. Each C, ν combination requires 10 SVM's to be trained (for 10-fold cross-validation), leading to a total of $17 \times 11 \times 10 = 1870$ SVM's for each independent training set. It becomes even worse in choosing the kernel parameter. In the cross-validation experiments, we experimented with 9 values of ω, giving a total of $1870 \times 9 = 16830$ SVM's to be trained.

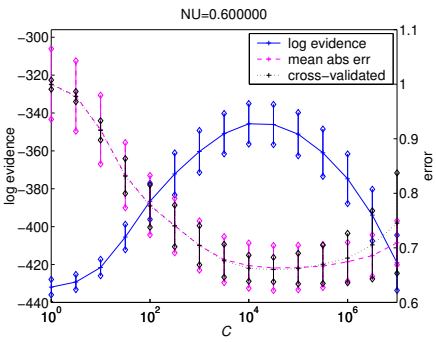
On the contrary, our proposed method is iterative and, for a fixed ω, this usually takes 10 to 50 iterations to find a good enough C and ν. The computational demand in each iteration comes mainly from the SVM training and the eigen decomposition of $\tilde{\mathbf{K}}$. As $\tilde{\mathbf{K}}$ is dominated by points with large r_i's (which lie close to the edges of the ϵ-tube and are thus usually small in number), we can discard those data points with small r_i's and thus result in a much smaller eigen problem. Hence, in practice, most of the time is spent on the SVM training. In comparison with cross-validation, we required just around 10 to 50 SVM's in

[3] In fact, cross-validation is so computationally demanding that experiments on choosing the kernel parameters for the two real datasets have not been performed.

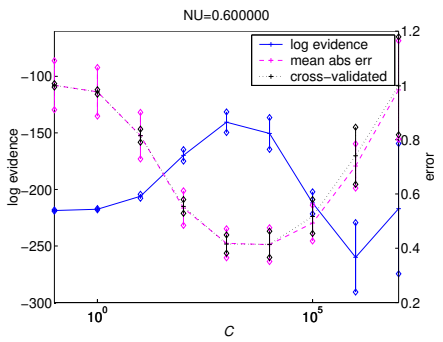


(a) sinc, Gaussian noise

(b) sinc, ϵ-insensitive noise

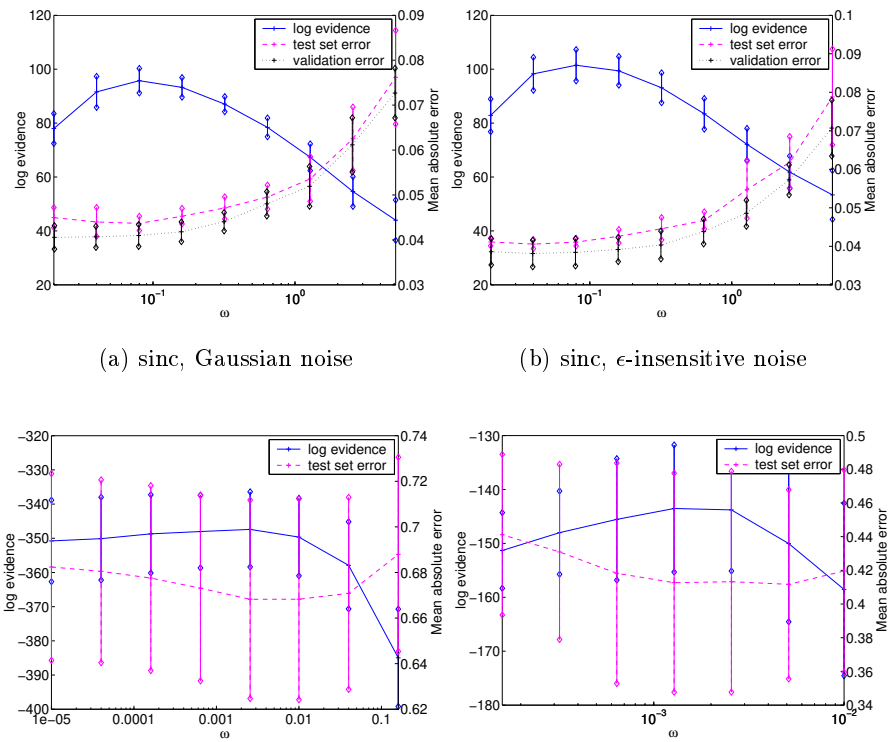


(c) abalone



(d) Boston

Fig. 2. Plots showing the logarithm of the evidence $p(D|\alpha, \beta, \nu, \mathcal{H})$, the testing errors and the average validation set errors at different values of C's. The error bars correspond to the ± 1 standard deviations based on the 10 repetitions.



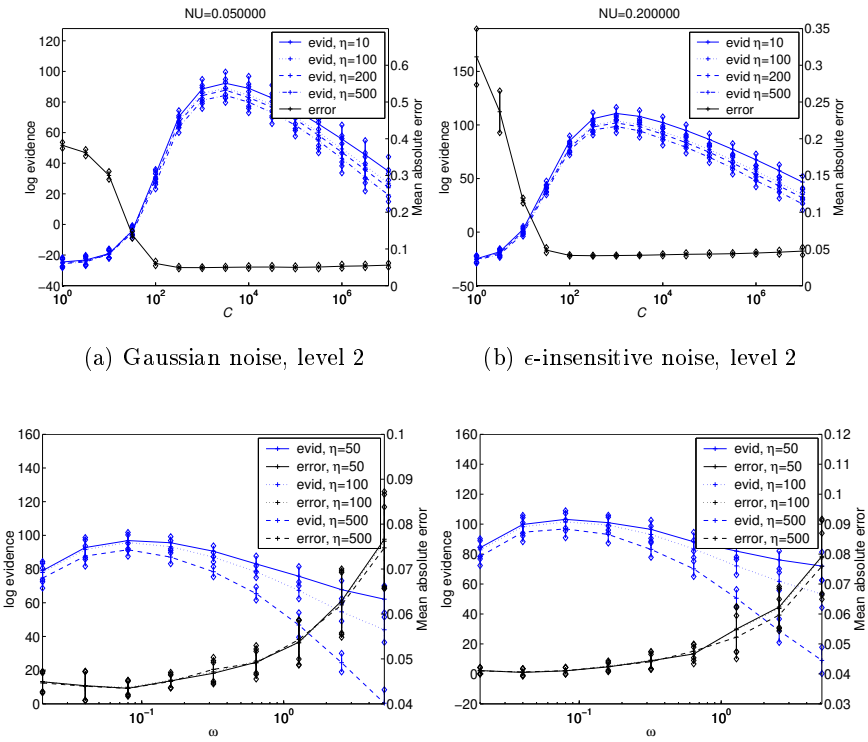
(a) sinc, Gaussian noise (b) sinc, ϵ-insensitive noise

(c) abalone (d) Boston

Fig. 3. Plots showing the logarithm of the evidence $p(D|\mathcal{H})$, the test set error (with C and ν selected by the proposed method), and the validation set error (with the best C and ν). The error bars correspond to the ± 1 standard deviations based on the 10 repetitions.

preparing the Figure 2 curve for the sinc data, and on average $30 \times 9 = 270$ SVM's in preparing the one in Figure 3.

As on the question of how the choice of η affects the inference results, Figure 4 shows the level 2 and level 3 results for the sinc data set at different η's. One can see that the evidence curves are all similar in shape as η varies. In particular, their peaks are all located at the same position.



(a) Gaussian noise, level 2

(b) ϵ-insensitive noise, level 2

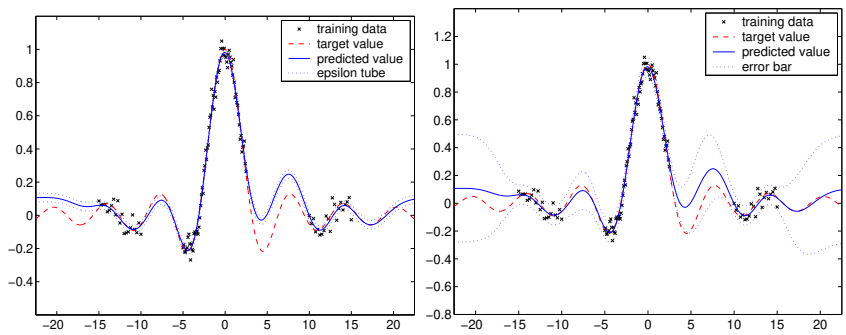
(c) Gaussian noise, level 3

(d) ϵ-insensitive noise, level 3

Fig. 4. Plots showing the effect of η on level 2 and level 3 inference with the sinc dataset.

5.2 Error Bars

Figure 5 shows the output predictions on a variation of the sinc problem, with the training patterns distributed unevenly in the input space. As expected, $\sigma_y^2(\mathbf{x})$ increases in regions where the data are sparse. For distant $\mathbf{x}$'s, the uncertainty reduces to that of the prior uncertainty.



(a) with the ϵ-tube. (b) with $\pm 1\sigma_y(\mathbf{x})$ error bars.

Fig. 5. Uncertainties in output predictions on the sinc dataset ($\nu = 0.56$).

6 Conclusion

In this paper, we extend the application of the evidence framework to ν-SVR. This allows automatic adjustment of the regularization and kernel parameters to their near-optimal values, without the need to set data aside in a validation set. Moreover, posterior predictive distribution and error bars can also be computed, which can then be used as a measure of uncertainty associated with ν-SVR's output predictions.

References

1. C. Blake, E. Keogh, and C.J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html University of California, Irvine, Department of Information and Computer Sciences.
2. J.T. Kwok. Moderating the outputs of support vector machine classifiers. *IEEE Transactions on Neural Networks*, 10:1018–1031, 1999.
3. J.T. Kwok. The evidence framework applied to support vector machines. *IEEE Transactions on Neural Networks*, 11(5):1162–1173, 2000.
4. D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, May 1992.
5. B. Schölkopf and A.J. Smola. New support vector algorithms. NeuroCOLT2 Technical Report NC2-TR-1998-031, GMD FIRST, 1998.
6. A.J. Smola, N. Murata, B. Schölkopf, and K.-R. Müller. Asymptotically optimal choice of ϵ-loss for support vector machines. In *Proceedings of the International Conference on Artificial Neural Networks*, 1998.
7. A.J. Smola and B. Schölkopf. A tutorial on support vector regression. NeuroCOLT2 Technical Report NC2-TR-1998-030, Royal Holloway College, 1998.
8. V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.