Journal of Korean
Data & Information Science Society
2004, Vol. 15, No. 2, pp. 477~483

Estimating Regression Function with ε-Insensitive Supervised Learning Algorithm

Changha Hwang[1)]

Abstract

One of the major paradigms for supervised learning in neural network community is back-propagation learning. The standard implementations of back-propagation learning are optimal under the assumptions of identical and independent Gaussian noise. In this paper, for regression function estimation, we introduce ε-insensitive back-propagation learning algorithm, which corresponds to minimizing the least absolute error. We compare this algorithm with support vector machine(SVM), which is another ε-insensitive supervised learning algorithm and has been very successful in pattern recognition and function estimation problems. For comparison, we consider a more realistic model would allow the noise variance itself to depend on the input variables.

Keywords : Back-propagation learning, ε-insensitive, neural network, regression, support vector machine

1. Introduction

In regression problems it is important not only to predict the output variables but also to have some estimate of the error bars associated with those predictions. An important contribution to the prediction and estimation arises from the intrinsic noise on the data. In most conventional treatments of regression, it is assumed that the data have no outliers and the noise can be modeled by a Gaussian distribution with a constant variance. However, in many applications the data have outliers. Furthermore, it will be more realistic to allow the noise variance itself to depend on the input variables. The outliers existing in data imply that the regression methods need to be robust. In the literature, there are many robust

1) Professor, Dept. of Statistical Information, Catholic University of Daegu.
E-mail : chhwang@cu.ac.kr

estimators. The ε-insensitive loss function(ILF) introduced by Vapnik(1995) provides the robust estimator. This is similar to loss functions used in the field of robust statistics. The ε-ILF enjoy robustness properties with respect to rather general classes of distributions. The details are illustrated in Vapnik(1998), and Smola and Scholkopf(1998).

Support vector machine(SVM) regression is a learning technique where the goodness of fit is measured not by the usual quadratic loss function, but by a different loss function called the ε-ILF. In its present form, SVM was developed by Vapnik and co-workers. In regression and time series prediction applications, SVM showed excellent performances.

Recently, Fyfe and Gabrys(1999) have proposed an ε-insensitive unsupervised learning rule called as ε-insensitive Hebbian learning, and showed that this rule is capable of performing principal component analysis(PCA) type learning in a linear network under a variety of conditions. One of the major paradigms for supervised learning in neural network(NN) is back-propagation(BP) learning. Bishop and Qazaz(1997), and Goldberg and Williams(1998) treated regression problem where there is the variance of noise depends on the inputs. In this paper, we derive ε-insensitive BP algorithm for regression problem. We then compare this with SVM based on ε-ILF in the case where the noise variance itself depends on the input variables.

2. The Noise Model for ε-ILF

In this section, we study the use of the ILF can be justified under the assumption that the noise affecting the data is additive and Gaussian, where the variance and mean are random variables whose probability distributions can be explicitly computed. For the details see Pontil et al.(2000). Data terms of the type $V(y_i - f(\boldsymbol{x}_i))$ can be interpreted in probabilistic terms as non-Gaussian noise models. Recently, Pontil et al.(2000) derived a noise model corresponding to Vapnik's ε-ILF, which is defined as follows:

$$V(x) \equiv |x|_\varepsilon = \begin{cases} 0 & \text{if} \quad |x| \le \varepsilon \\ |x| - \varepsilon & \text{otherwise} \end{cases}. \tag{1}$$

This loss function can be approximated as in Seok et al.(2002) by a smooth function

$$V(x) = g(x-\varepsilon) + g(-x-\varepsilon), \tag{2}$$

where $g(x) = \frac{1}{u}\log(1+\exp(ux))$, $u > 0$. The ε-ILF is similar to some of the functions used in robust statistics, which are known to provide robustness against outliers. However, the loss function (1) is not only a robust cost function,

because of its linear outside the interval $[-\varepsilon, \varepsilon]$, but also assigning zero cost to errors smaller than ε. It turns out that the underlying noise model consists of the superposition of Gaussian processes with different variances and means, that is:

$$\exp(-|x|_\varepsilon) = \int_{-\infty}^{\infty} \int_0^{\infty} \lambda_\varepsilon(t)\sqrt{\beta}\exp(-\beta(x-t)^2)\,d\beta\,dt$$

with

$$\lambda_\varepsilon(t) = \frac{1}{2(\varepsilon+1)}\left(\chi_{[-\varepsilon,\varepsilon]} + \delta(t-\varepsilon) + \delta(t+\varepsilon)\right),$$
$$P(\beta) = \frac{C}{\beta^2}\exp\left(-\frac{1}{4\beta}\right).$$

The probability distribution function $P_\varepsilon(\beta, t)$ has the form $P_\varepsilon(\beta, t) = P(\beta)\lambda_\varepsilon(t)$. Here, $\chi_{[-\varepsilon,\varepsilon]}(t)$ is 1 for $t \in [-\varepsilon, \varepsilon]$, 0 otherwise. For the details of $\lambda_\varepsilon(t)$, $P(\beta)$ and $P_\varepsilon(\beta, t)$, see Pontil et al.(2000). The above model has a simple interpretation: using the ILF is equivalent to assuming that the noise affecting the data is Gaussian. However, the variance and the mean of the Gaussian noise are random variables: the variance ($\sigma^2 = 1/(2\beta)$) has a unimodal distribution that does not depend on ε, and the mean has a distribution which is uniform in the interval $[-\varepsilon, \varepsilon]$. The distribution of the mean is consistent with the current understanding of the ILF: errors smaller than ε do not count because they may be due entirely to the bias of the Gaussian noise.

3. ε-Insensitive Supervised Learning Algorithms

In this section we derive ε-insensitive BP algorithm based on ε-ILF. We also review SVM regression based on ε-ILF. Suppose we are given a training data set $\{(\boldsymbol{x}_i, y_i) \in R^d \times R\}_{i=1}^n$, obtained by sampling, with noise, some unknown function $f(\boldsymbol{x})$ and we asked to recover the function f from the data.

3.1 ε-Insensitive Back-propagation Algorithm

We now illustrate how to derive ε-insensitive BP algorithm for multi-layer neural network(NN) with one hidden layer. First of all, we need to review how the outputs in the previous layer are transferred to the successive layer. Let w_{ji} denote a weight in the input layer, going from input i to hidden unit j, and θ_j denote the bias for hidden unit j. Let v_{jk} denote a weight in the hidden layer,

going from hidden unit j to output unit k, and θ_k denote the bias for output unit k. Let g_j, g_k denote activation functions of hidden unit j and output unit k, respectively. We often use continuous sigmoidal and linear functions as activation functions in hidden and output layers, respectively. Then, we have the following relations:

$$a_{pj} = \sum_i w_{ji} x_{pi},\ h_{pj} = g_j(a_{pj}),\ b_{pk} = \sum_j v_{kj} h_{pj},\ \hat{y}_{pk} = g_k(b_{pk}),$$

where the subscript p corresponds to the pth observation value.

We now illustrate learning procedures based on two loss functions (1) and (2), which are then minimized with respect to the weights and biases in the network. See Haykin(1999) for the derivation of conventional BP algorithm for NNs. The proposed BP learning can be summarized as follows:

(1) For units in output layer,

$$\Delta_p v_{kj} = \alpha \begin{cases} 0, & |y_{pk} - \hat{y}_{pk}| < \varepsilon \\ \text{sign}(y_{pk} - \hat{y}_{pk}) h_{pj}, & |y_{pk} - \hat{y}_{pk}| \geq \varepsilon \end{cases} \quad \text{(loss (1))}$$

$$= \alpha \left[-\frac{\partial g(y_{pk} - \hat{y}_{pk} - \varepsilon)}{\partial \hat{y}_{pk}} - \frac{\partial g(-y_{pk} + \hat{y}_{pk} - \varepsilon)}{\partial \hat{y}_{pk}} \right] h_{pj} \quad \text{(loss (2))}$$

$$\equiv \alpha \delta_{pk} h_{pj}$$

$$\Delta_p \theta_k = \beta \delta_{pk}$$

(2) For units in hidden layer,

$$\Delta_p w_{ji} = \alpha \sum_k \delta_{pk} v_{kj} \hat{y}_{pi}$$

$$\equiv \alpha \delta_{pj} \hat{y}_{pi},$$

$$\Delta_p \theta_j = \beta \delta_{pj},$$

where α, β are learning rates.

3.2 Support Vector Machine

For given training data set $\{(\boldsymbol{x}_i, y_i) \in R^d \times R\}_{i=1}^n$, we consider the following model

$$f(\boldsymbol{x}) = \boldsymbol{w}^t \phi(\boldsymbol{x}) + b,$$

where the input data are projected to a higher dimensional feature space. In empirical risk minimization we optimize the cost function based on Vapnik's ε-ILF. This leads to the optimization problem

$$\text{minimize} \quad \frac{1}{2} \| \boldsymbol{w} \|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*),$$

$$\text{subject to} \quad \begin{cases} y_i - \boldsymbol{w}^t \phi(\boldsymbol{x}_i) - b \le \varepsilon + \xi_i \\ \boldsymbol{w}^t \phi(\boldsymbol{x}_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \ \xi_i^* \ge 0 \end{cases}$$

Here ε is the accuracy that we demand for the approximation, which can be violated by means of the slack variables ξ, ξ_i^*. The conditions for optimality yield the following dual problem in the Lagrange multipliers α, α^*:

$$\text{maximize} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i^*),$$

$$\text{subject to} \quad \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \quad \text{and} \quad \alpha_i, \ \alpha_i^* \in [0, C]$$

The kernel trick $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^t \phi(\boldsymbol{x}_j)$ is again applied within the formulation of this quadratic programming problem. Finally, the SVM for nonlinear function estimation takes the form:

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

The well used kernels for regression problem are given below.

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^t \boldsymbol{y} + 1)^p, \quad K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}}.$$

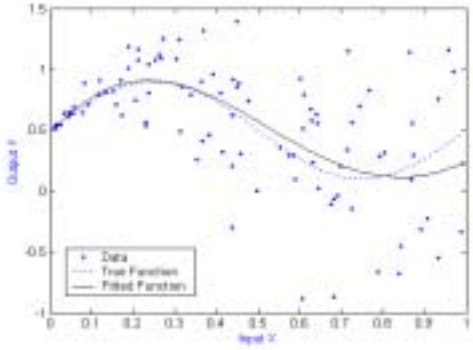
Here, p and σ^2 are kernel parameters.

4. Empirical Studies

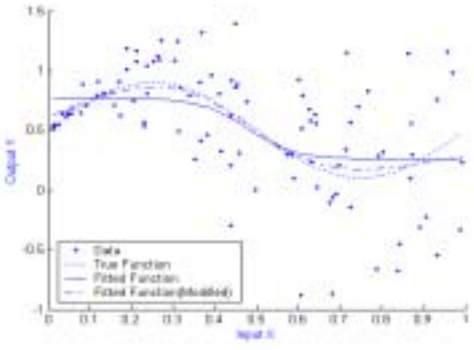
We illustrate the performance of the proposed learning procedures through one simulated example. The simulated data set consists of 100 input values(x) generated from a uniform distribution $U(0,1)$ and 100 output values(y) generated from a normal distribution $N(0.5 + 0.4 \times \sin(2\pi x), x^2)$, which is rather realistic model would allow the noise variance itself to depend on the input variables. Such model is said to heteroscedastic in the statistics literature. In much of the work on regression problems in the statistical and neural networks literatures, it is assumed there is a global noise level, independent of input vector $\boldsymbol{x}$. Bishop and Qazaz(1997) examined the case of input-dependent noise for neural networks, and Goldberg and Williams(1998) developed the treatment of input-dependent noise model for Gaussian process regression. problem where there is the variance of noise depends on the inputs.

In the simulation study, we used SVM with ε=0.01, C=500 and kernel

parameter σ=0.8. These values were determined by 10-fold cross validation. Fig. 1 shows regression estimation results by SVM. We used a multi-layer NN with 10 hidden units. We also used ε=0.01 for ε-insensitive BP of NN. These numbers were also determined by 10-fold cross validation. The NN was iterated to get solutions until either MSE<0.01 or 30,000 iteration was satisfied first. The estimation results by NN are reported in Fig. 2. We conducted experiments for NN by using two ε-insensitive BP algorithms.



<Fig. 1> Function Estimation by SVM



<Fig. 2> Function Estimation by NN

In Fig. 1, the dotted and the solid lines represent the true function and the fitted regression function, respectively. In Fig. 2, the dotted line represents the true function. Te dash-dotted and the solid lines represent the regression functions fitted by two ε-insensitive BP algorithms based on error functions (1) and (2), respectively. In both Figs, the dots represent the observed values. From Fig. 1, we recognize SVM based on ε-ILF works quite well for the input-dependent noise model. Fig.2 shows that the BP algorithm based on the ε-ILF works bad, but the algorithm based on the approximated ε-ILF works quite well like SVM. We observed the similar patterns, although we did not reported here about preliminary simulation studies for the input-dependent noise model.

To conclude, we recommend using SVM, since NNs based on two ε-insensitive BP algorithms are much more expensive computationally than SVM. The main formulation of SVM results in solving a simple convex optimization problem. Hence, this is not a computationally expensive way.

References

1. Bishop, C., Qazaz, C. (1997). Regression with input-dependent noise: A Bayesian treatment, *Advances in Neural Information Processing Systems*, 9, 347-353.
2. Fyfe, C., Gabrys, B. (1999). ε-insensitive unsupervised learning, Proceedings of International Conference on Neural Networks and Artificial Intelligence, 10-18.
3. Goldberg, P., Williams, C. (1998). Regression with input-dependent noise: A Gaussian process treatment, *Advances in Neural Information Processing Systems*, 10, 493-499.
4. Haykin, S. (1999). Neural networks: A comprehensive foundation, Prentice Hall.
5. Pontil, M., Mukherjee, S., Girosi, F. (2000). On the noise model of upport vector machines regression, *Lecture Notes in Artificial Intelligence*, 1968, 316-324.
6. Seok, K., Hwang, C., Cho, D. (2002). Prediction intervals for support vector machine regression, *Communications in Statistics: Theory and Methods,* 31, 1887-1898.
7. Smola, A., Scholkopf, B. (1998). A tutorial on support vector regression, Technical Report, NeuroCOLT.
8. Vapnik, V. (1995). The Nature of Statistical Learning Theory, Springer, New York.
9. Vapnik, V. (1998). Statistical Learning Theory, John Wiley & Sons, New York.

[received date : Dec. 2003 , accepted date : May. 2004]