Model Evaluation

The most common error estimate for regression functions is the *mean squared error*.

We define a loss function called $\mathcal{L}_2$ that computes the squared residual at an observation $(\overline{x}, y)$ given a model $\hat{f}$,

$$\mathcal{L}_2(y, \hat{f}(\overline{x})) = \left(y - \hat{f}(\overline{x})\right)^2 .$$

Now, given a regression training set,

$$D = \{(\overline{x}_1, y_1), (\overline{x}_2, y_2), \ldots, (\overline{x}_l, y_l)\} \subseteq \mathbb{R}^n \times \mathbb{R},$$

we define the mean squared error computed on D as,

$$\text{mse}_D \left[\hat{f}_D[k, \lambda, \varepsilon, C]\right] = \frac{1}{l} \sum_{i=1}^{l} \mathcal{L}_2 \left(y_i, \hat{f}_D[k, \lambda, \varepsilon, C](\overline{x}_i)\right),$$

with $(\overline{x}_i, y_i) \in D$ for some appropriate model $\hat{f}_D[k, \lambda, \varepsilon, C] : \ \mathbb{R}^n \to \mathbb{R}$

As before, our error metric is the average loss over of model $\hat{f}_D$ over the data set D.

Model Evaluation

In this case the error mse_D represents the training error and we can find the optimal training error by optimizing over the model parameters,

$$\min_{k,\lambda,\varepsilon,C} \text{mse}_D \left[\hat{f}_D[k,\lambda,\varepsilon,C]\right].$$

As we know from our work in classification, the training error tends to be overly optimistic. Therefore we use other testing techniques such as the hold-out method or cross-validation. The hold-out method applies to regression as follows. We start by splitting the set D into two non-overlapping partitions P and Q such that,

$$D = P \cup Q,$$

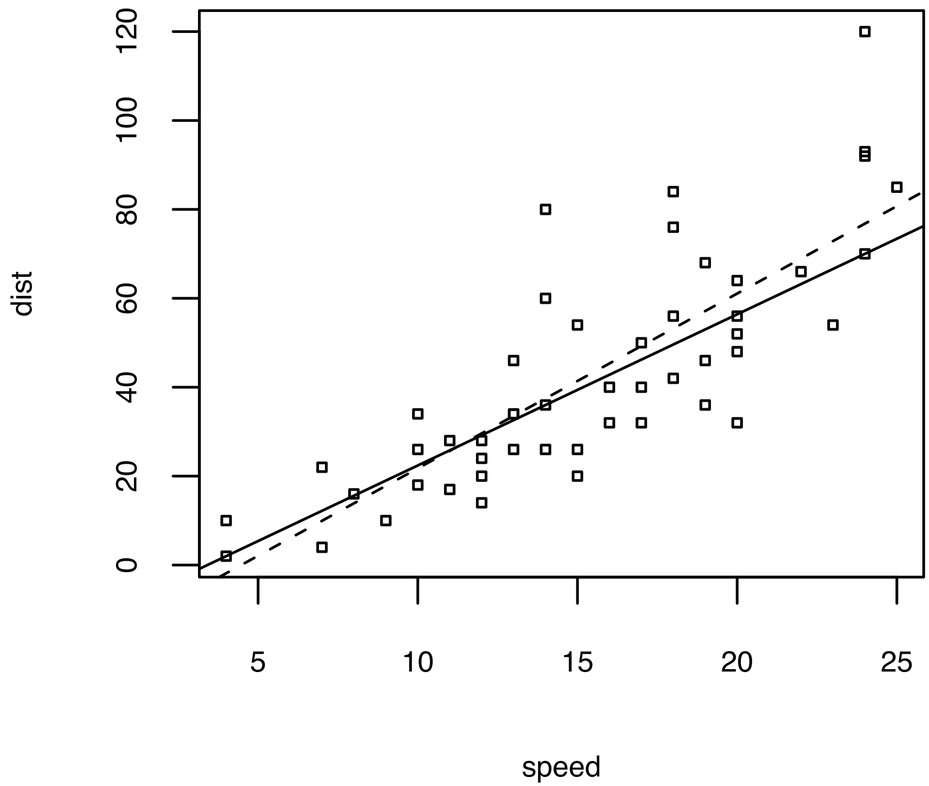
where we use P as a training set and Q as a test set. The test error can then be computed as,

$$\text{mse}_Q \left[\hat{f}_P[k,\lambda,\varepsilon,C]\right] = \frac{1}{|Q|} \sum_{(\overline{x}_i, y_i) \in Q} \mathcal{L}_2 \left(y_i, \hat{f}_P[k,\lambda,\varepsilon,C](\overline{x}_i)\right).$$

We can use the test error to find the optimal model $\hat{f}^*$,

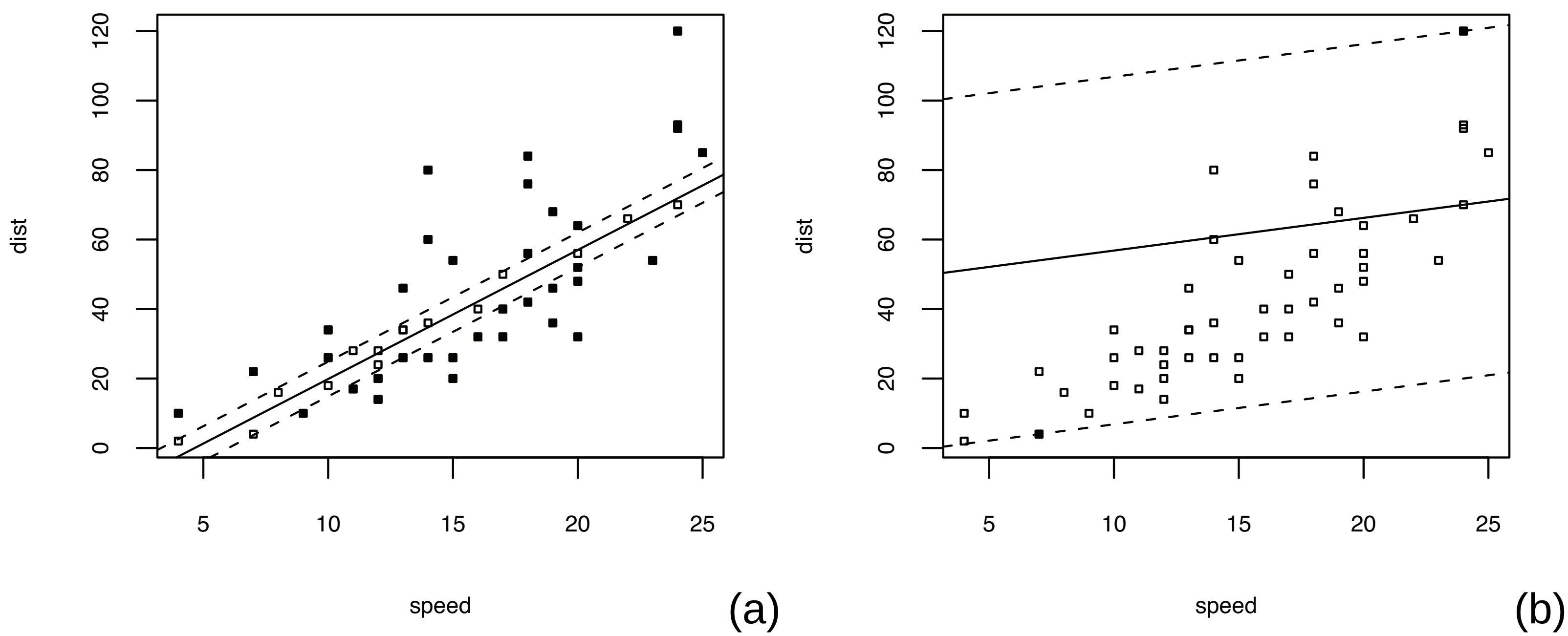
$$\hat{f}^* = \operatorname*{argmin}_{k,\lambda,\varepsilon,C} \text{mse}_Q \left[\hat{f}_P[k,\lambda,\varepsilon,C]\right].$$

Examples



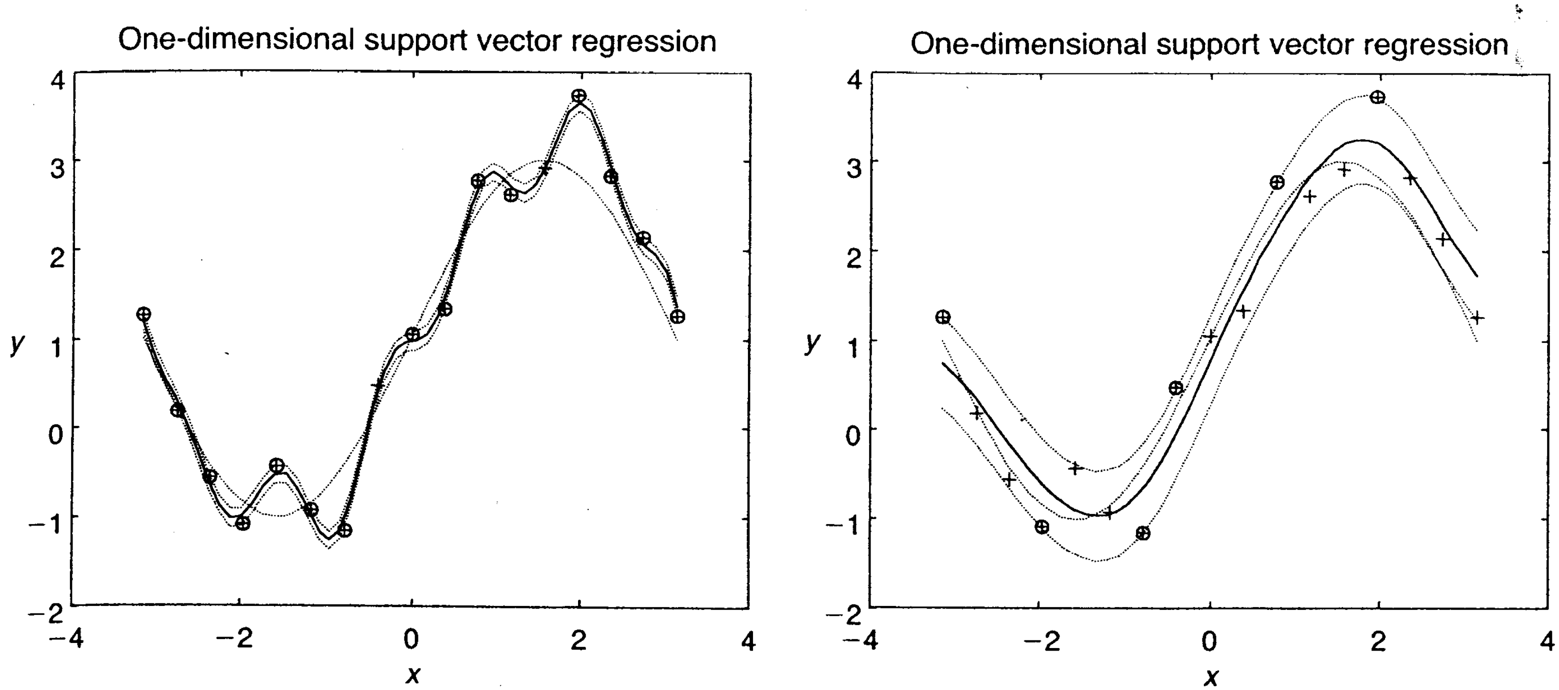
Comparing a simple linear regression model (dashed line) for the ‘cars’ data set with a support vector regression model (solid line).

Examples



Linear support vector regression model of the ‘cars’ data set with (a) $\varepsilon = 5$ and (b) $\varepsilon = 50$.

Non-Linear Regression



Influence of an insensitivity zone e on modeling quality. A nonlinear SVM creates a regression function with Gaussian kernels and models a highly polluted (25% noise) sine function (dashed). Seventeen measured training data points (plus signs) are used. *Left*, $\varepsilon = 0.1$, fifteen SV are chosen (encircled plus signs). *Right*, $\varepsilon = 0.5$, six chosen SVs produced a much better regressing function.

(Source: Learning and Soft Computing, V. Kecman, MIT Press, 2001)