Principles of Statistical Mining 04/20/2010

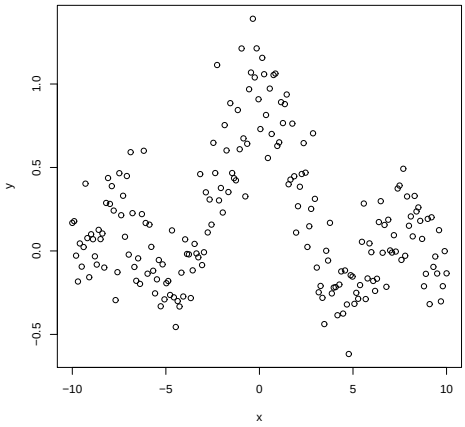
Notes on [SVM]: Intro to Support Vector Regression

Professor: Ernest Fokoué **Note:** Draft

And which of you by being anxious can add a cubit unto the measure of his life? *Luke 12:25*

1 Motivating Example

Let $\{(\mathbf{x}_i, \mathrm{y}_i),\, i = 1, \cdots, n\}$ be a random sample of pairs where $\mathbf{x}_i \in \mathbb{R}^p$ is the vector of explanatory variables, and $\mathrm{y}_i \in \mathbb{R}$ is the real-valued response. For instance, the following scatter plot is made up of noisy data points, and we seek to recover the underlying function. In other words, we assume that there is an underlying function, f, such that each response



variable Y_i can be expressed as

$$Y_i = f(\mathbf{x}_i) + \eta_i$$

where η_i is the noise term. Now, in traditional linear regression, we assumed that the true function f was represented as $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, that the noise was normally distributed with constant variance σ^2, and we found that using the loss function $\ell(\mathrm{y}, f(\mathbf{x})) = (\mathrm{y} - f(\mathbf{x})^2$ (squared error loss) allowed us to fit a function to the data. The function underlying the above scatter is clearly not linear, and we need a different approach. Now, three typical questions arise, namely:

- ☐ What class of functions can we resort to to capture the nonlinear structure underlying the above scatter?
- ☐ What loss function can we use to isolate the best function in that class?
- ☐ What technique can be used to best isolate the best function in the class?

1.1 Brief description of Support Vector Regression

Borrowing from the paradigm of large margin learning used in Support Vector Classification, Support Vector Regression uses the so-called ε-insensitive loss function as the criterion for measuring the goodness of fit of the chosen function to the data. In other words, for some positive ε, define

$$\ell(u) \equiv |u|_\varepsilon \equiv \begin{cases} 0, & |u| < \varepsilon \\ |u| - \varepsilon & \texttt{otherwise} \end{cases}$$

The idea of the ε-insensitive loss function (below) is clearly similar the idea of margin encountered earlier in support vector classification.

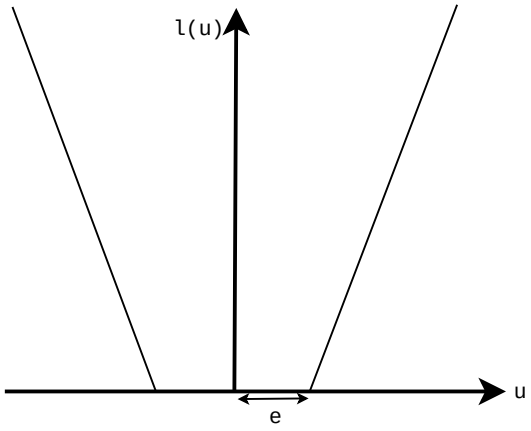


Figure 1: *ε-insensitive loss function. The loss is zero within the ε margin, and linear outside of the margin on both sides.*

For the so-called ε-SVR or ε-Support Vector Regression, the support vector machine regression function $\hat{f}$ is obtained by solving a constrained minimization problem with objective function

$$E(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^{n} (\xi_i + \xi_i^*),$$

subject to

$$\begin{aligned} \mathrm{y}_i - \mathbf{w}^\top \Phi(\mathbf{x}_i) - b &\leq \varepsilon + \xi_i, \\ \mathbf{w}^\top \Phi(\mathbf{x}_i) + b - \mathrm{y}_i &\leq \varepsilon + \xi_i^* \end{aligned}$$

where $\xi_i, \xi_i^* \geq 0$, and $C > 0$ and $\varepsilon > 0$ are positive constants to be specified by the experimenter. Finally, $\Phi(\cdot)$ is the feature space mapping implied in the kernel $K(\cdot, \cdot)$ such that

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{|\mathbf{s}|} \hat{\alpha}_{s_j} K(\mathbf{x}_{s_j}, \mathbf{x}) + \hat{b}$$

where $s_j \in \{1, 2, \cdots, n\}$, $\mathbf{s}^\top = \{s_1, s_2, \cdots, s_{|\mathbf{s}|}\}$ and $|\mathbf{s}| \lll n$.

1.2 Support Vector Regression on the Motivating Example

Just like with Support Vector Machine Classification, the kernel plays a central role in support vector regression. Some of the most commonly used kernel functions have already been encountered in classification and include:

- ☐ The polynomial kernel of Vladimir Vapnik. (Professor Vladimir Vapnik is the co-inventor of the Support Vector Machine paradigm)

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + \delta)^d$$

- ☐ The Gaussian Radial Basis function kernel

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

- ☐ The Laplace kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|\right)$$

The package kernlab can be used for Support Vector Regression. Simple run `library(kernlab)` and `help(ksvm)` to get the following description.

```
ksvm(x, y = NULL, scaled = TRUE, type = NULL, kernel ="rbfdot",
          kpar = "automatic", C = 1, nu = 0.2, epsilon = 0.1,
          prob.model = FALSE, class.weights = NULL, cross = 0, fit = TRUE,
          cache = 40, tol = 0.001, shrinking = TRUE, ...,
          subset, na.action = na.omit)
```

Note: For the above motivating example, run the R script `simple-svr.R`

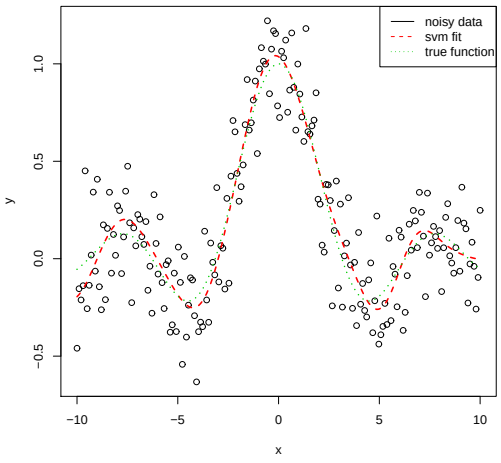


Figure 2: *Support Vector Regression on the sinc function with the RBF kernel.*