STAT562 Lecture 15 Kernel SVM

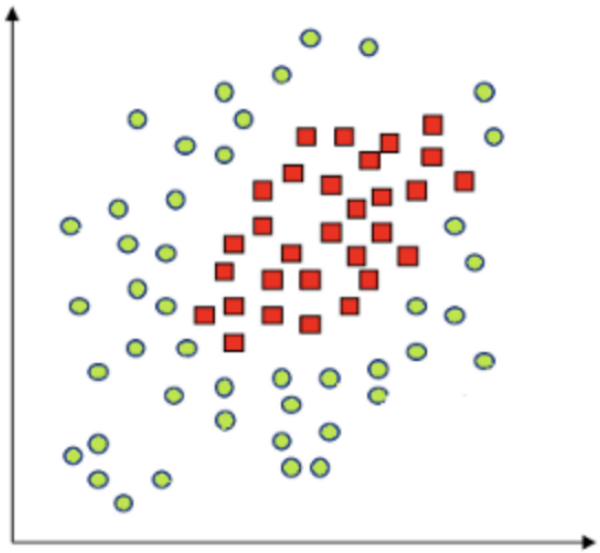
Beidi Qiang

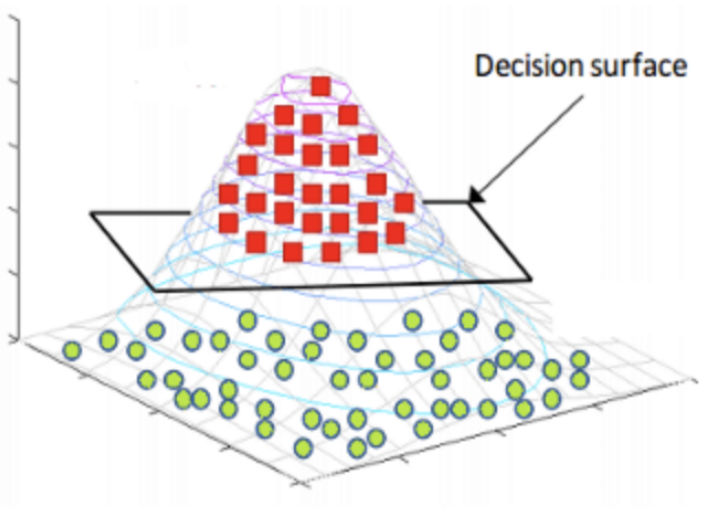
SIUE

Kernel SVM

Kernel SVMs are a class of Support Vector Machines (SVMs) that use kernel functions to classify data.
It is a generalization technique of the linear SVM on nonlinear data.
Kernel SVM offers more flexibility when dealing with a linearly inseparable classification task.

Example: Non-linear Boundaries and Kernel Transformation





The Kernel Trick

Let's consider a simple example. Suppose we have a dataset dimensional features $x_1 = (1, 2)$ and $x_2 = (3, 4)$.

- we want to map our vectors to higher-dimensional (say a 4-d) space by the 2nd order terms, i.e.

$$\phi(x_1) = (1, 2, 2, 4), \phi(x_2) = (9, 12, 12, 16)$$

- The distance between the two new vectors in high dimensions can be calculated by taking the inner product.

$$< \phi(x_1), \phi(x_2) >= \phi(x_1) \cdot \phi(x_2) = 1 \times 9 + 2 \times 12 + 2 \times 12 + 4 \times 16 = 121$$

- The kernel Trick: you get the same answer by

$$(< x_1, x_2 >)^2 = (1 \times 3 + 2 \times 4)^2 = 11^2 = 121$$

Kernel

The support vector machine enlarges the feature space using "kernel trick", in order to accommodate a non-linear boundary.

- The inner product of two vectors is defined as

$$< \mathbf{x_1}, \mathbf{x_2} >= \sum_{j=1}^{p} x_{1j} x_{2j}.$$

- The kernel is a generalization of the inner product, denoted by $K(\mathbf{x_1}, \mathbf{x_2})$.
- For example, a d-degree polynomial kernel has the form

$$K(\mathbf{x_1}, \mathbf{x_2}) = (1 + \sum_{j=1}^{p} x_{1j} x_{2j})^d.$$

Some More insights on using kernels with SVM

Recall in SVM we have the following optimization problem: Chooses $\beta_0, \cdots, \beta_p$ to maximize M, subject to

$$\sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \text{ for all i.}$$

- The solution to the SVM optimization problem can be simplified to involves only the inner products of the observations.
- The classifier (hyperplane) can be expressed in terms of the inner product between the new observation and each of the training points, as $f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i < \mathbf{x}, \mathbf{x_i} >$.
- It turns out if a training observation is not a support vector, then its α_i equals zero, i.e. $f(x) = \beta_0 + \sum_{i \in S} \alpha_i < \mathbf{x}, \mathbf{x_i} >$, where S is the collection of indices of the support vectors.

Popular Kernels

- d-degree polynomial kernel :

$$K(\mathbf{x_i}, \mathbf{x_{i'}}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d.$$

- radial basis function (RBF) kernel:

$$K(\mathbf{x_i}, \mathbf{x_{i'}}) = exp[-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2].$$

- Gaussian kernel (Gaussian RBF):

$$K(\mathbf{x_i}, \mathbf{x_{i'}}) = exp[-\frac{1}{2\sigma^2} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2].$$

Example: polynomial and radial kernel

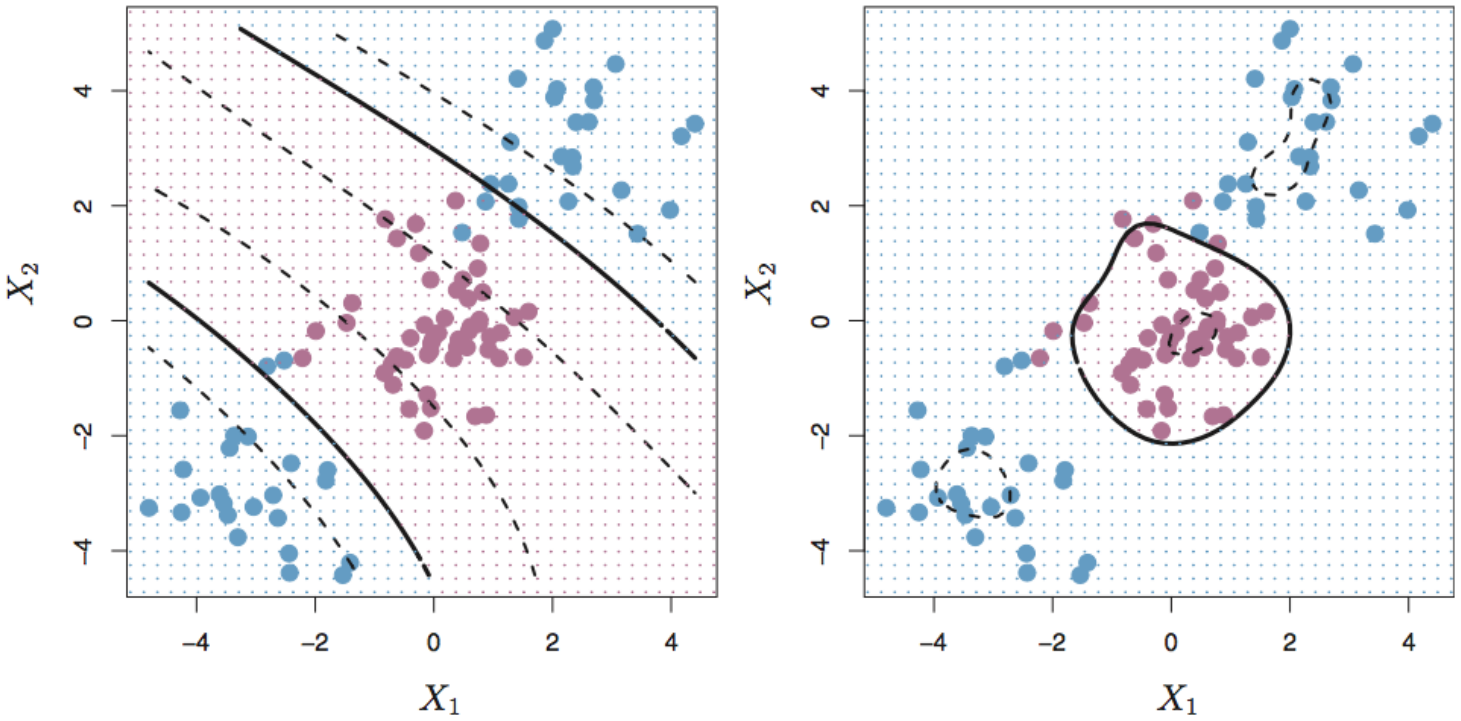


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

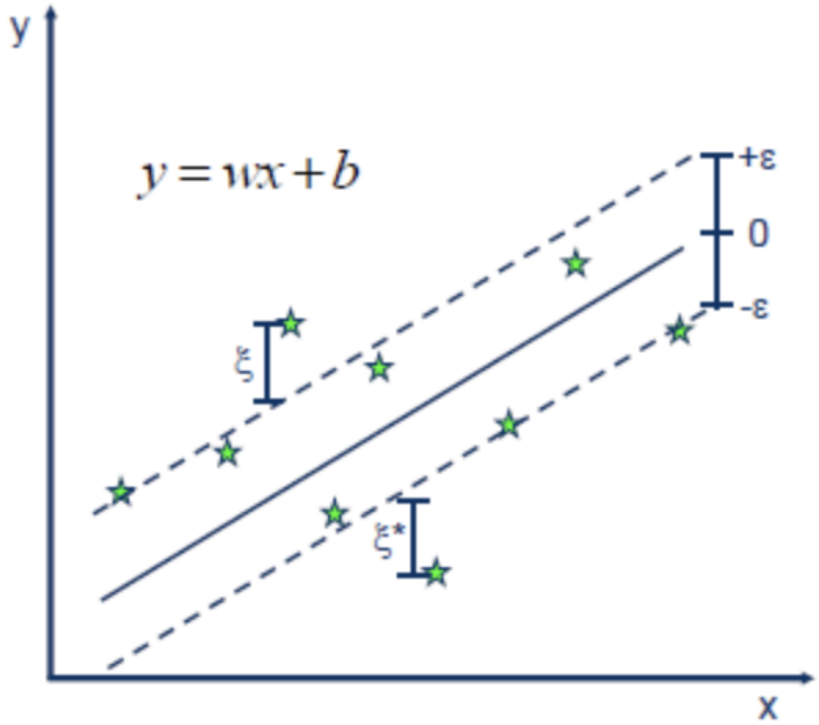
Extending to Multi-classes

We extend SVMs to the more general case where we have some arbitrary number of classes. Suppose we have K classes. There are two popular approaches:

- **One-Versus-One Classification:** Construct $\binom{K}{2}$ SVMs, each compares a pair of classes.The final classification is performed by assigning the test observation to the class to which it was most frequently assigned.
- **One-Versus-All Classification:** We fit K SVMs, each time comparing one of all the K classes to the remaining $K-1$ classes. Let $\beta_{0k}, \cdots, \beta_{pk}$ denote the parameters that result from fitting an SVM comparing the kth class to the others. We assign an observation to the class for which $\beta_{0k} + \beta_{1k}x_1 + \cdots + \beta_{pk}x_p$ is largest.

Support Vector Regression (FYI)

Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit hyperplane, that has the maximum number of points within its margin (ϵ).



- Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)$$

- Constraints:

$$y_i - wx_i - b \le \varepsilon + \xi_i$$

$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$