Support Vector Machines

Thomas Lonon

Financial Engineering/Financial Analytics
Stevens Institute of Technology

August 23, 2018

Three different concepts

- Maximal Margin Classifier
- Support Vector Classifier
- Support Vector Machine

Hyperplanes

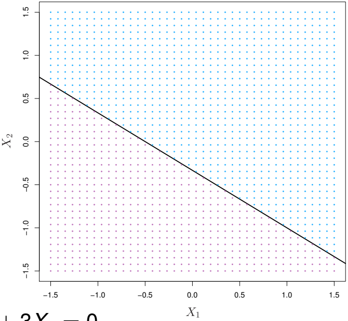
Our training data consists of N pairs $(x_1, y_y), (x_2, y_2), \ldots, (x_N, y_N)$, with $x_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$. Define a hyperplane by

$$\{x : f(x) = x^T\beta + \beta_0 = 0\}$$

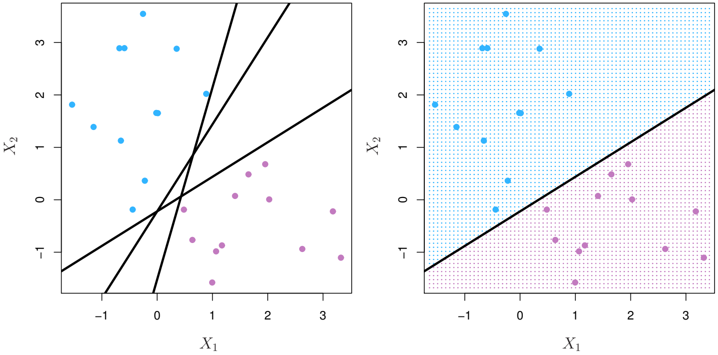
where β is a unit vector.

If $\beta_0 \neq 0$, we call this **affine**

[1]



hyperplane for $1 + 2X_1 + 3X_2 = 0$



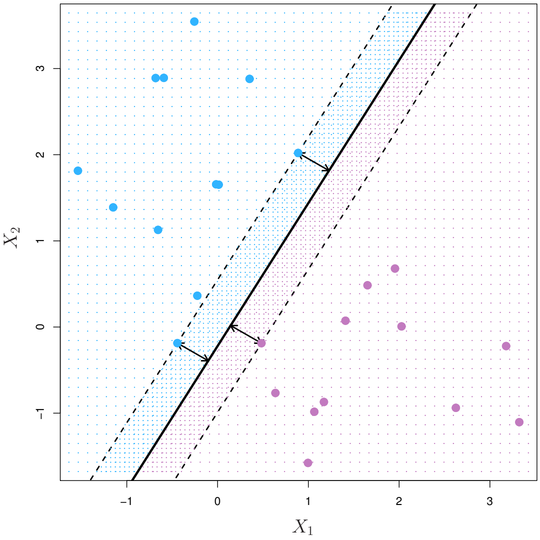
[1]

Maximum Margin Optimization

$$\max_{\beta_0,\beta_1,\ldots,\beta_p} M$$

subject to
$$\|\beta\| = 1$$
$$y_i(x_i^T\beta + \beta_0) \geq M,$$
$$i = 1, \ldots, N$$

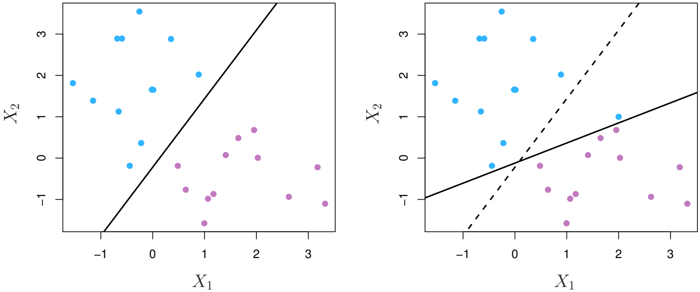
[1]



The main issue with the Maximal Margin Classifier, is overfitting

This results in great sensitivity to individual data points

[1]



Dealing with Overlap

Define the slack variables $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$. We then modify our previous constraints in one of two ways:

$$y_i(x_i^T\beta + \beta_0) \geq M - \xi_i$$

or

$$y_i(x_i^T\beta + \beta_0) \geq M(1 - \xi_i)$$

$\forall i, \xi_i \geq 0, \sum_{i=1}^{N} \xi_i \leq$ constant

With these overlapping datasets, our optimization for the margins becomes:

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,\xi_1,\ldots,\xi_N} M$$

subject to:

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

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} \geq M(1 - \xi_i)$$

$$\xi_i \geq 0$$

$$\sum_{i=1}^{N} \xi_i \leq \text{ a constant, } C$$

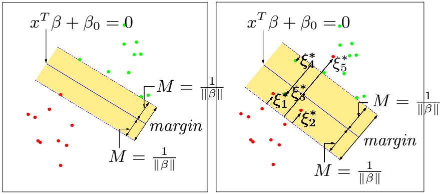


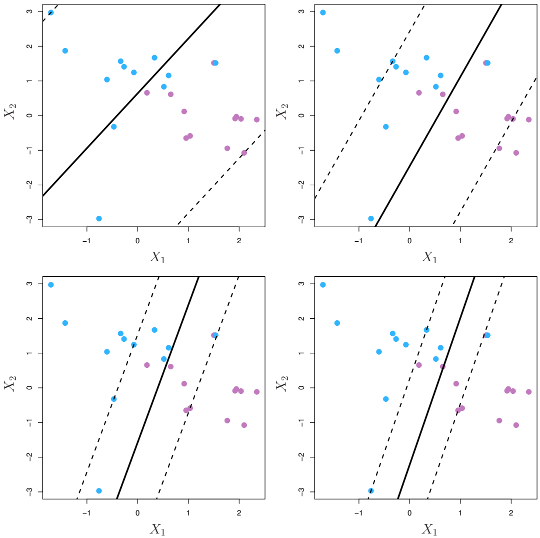
FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$. *The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong side of their margin by an amount* $\xi_j^* = M\xi_j$; *points on the correct side have* $\xi_j^* = 0$. *The margin is maximized subject to a total budget* $\sum \xi_i \leq$ *constant. Hence* $\sum \xi_j^*$ *is the total distance of points on the wrong side of their margin.*

Support Vectors

These points that either lie directly on the margin, or violate the margin are called *Support Vectors*

The number of which will be defined by C

[1]



Example of Non-linear Boundary

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,\xi_1,\ldots,\xi_N} M$$

subject to:

$$\sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1$$

$$y_i(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2 \geq M(1-\xi_i)$$

$$\xi_i \geq 0, \sum_{i=1}^{N}\xi_i \leq \text{ a constant, } C$$

Inner Product

The inner product of two observations, x_i and x_k is given by:

$$\langle x_i, x_k \rangle = \sum_{j=1}^{p} x_{ij} x_{kj}$$

The linear support vector classifier can be expressed as:

$$f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i \langle x, x_i \rangle$$

Kernels

We can represent the relationship between two variables as a *kernel*

Popular choices of K in SVM are

- **Linear:** $K(x, x') = \langle x_i, x_{i'} \rangle$
- d^{th}**-Degree Polynomial:** $K(x, x') = (1 + \langle x, x' \rangle)^d$
- **Radial basis:** $K(x, x') = e^{-\gamma \|x - x'\|^2}$
- **Neural Network:** $K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2)$

Support Vector Machines

If the kernel is non-linear, the classifier

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i) + \beta_0$$

is referred to as a *support vector machine*

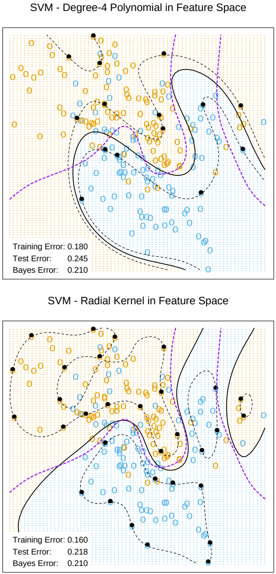


FIGURE 12.3. *Two nonlinear SVMs for the mixture data. The upper plot uses a 4th degree polynomial*

SVM for Regression

The linear regression model has the form:

$$f(x) = x^T \beta + \beta_0$$

Where β is estimated by minimizing

$$H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2} \|\beta\|^2$$

where

$$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon \\ |r| - \epsilon & \text{otherwise} \end{cases}$$

We can compare this error measure V_ϵ to more robust measures used in statistics, such as the Huber

$$V_H(r) = \begin{cases} r^2/2 & \text{if } |r| \leq c \\ c|r| - c^2/2 & |r| > c \end{cases}$$

This reduces from quadratic to linear the contributions of observations with absolute value greater than c, which makes fitting less sensitive to outliers.

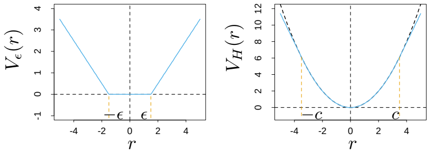


FIGURE 12.8. *The left panel shows the ϵ-insensitive error function used by the support vector regression machine. The right panel shows the error function used in Huber's robust regression (blue curve). Beyond $|c|$, the function changes from quadratic to linear.*

If $\hat{\beta}, \hat{\beta}_0$ are the minimizer of H, we have the solution

$$\hat{\beta} = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) x_i$$

$$\hat{f}(x) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) \langle x, x_i \rangle + \beta_0$$

where $\hat{\alpha}_i^*, \hat{\alpha}_i$ are positive and solve the quadratic programming problem

$$\min_{\alpha, \alpha_i^*} \epsilon \sum_{i=1}^{N} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{N} y_i (\alpha_i^* - \alpha_i) + \frac{1}{2} \sum_{i,i'=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_{i'}^* - \alpha_{i'}) \langle x_i, x_{i'} \rangle$$

subject to constraints

$$0 \leq \alpha_i, \alpha_i^* \leq 1/\lambda$$

$$\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0$$

$$\alpha_i \alpha_{i'} = 0$$

Regression and Kernels

For a set of basis functions $\{h_m(x)\}, m = 1, 2, \ldots, M$

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x) + \beta_0$$

To estimate β and β_0 we minimize

$$H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2} \sum \beta_m^2$$

The solution has the form:

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{a}_i K(x, x_i)$$

with $K(x, y) = \sum_{m=1}^{M} h_m(x) h_m(y)$

[1] Trevor Hastie Gareth James, Daniela Witten and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Number v. 6. Springer, 2013.