Statistical Methods in Machine Learning

Support vector machine

April, 8, 2019

Support Vector Machines

- Here we approach the two-class classification problem in a direct way:
- We try and find a plane that separates the classes in feature space.
- If we cannot, we get creative in two ways:
  - We soften what we mean by "separates", and
  - We enrich and enlarge the feature space so that separation is possible.

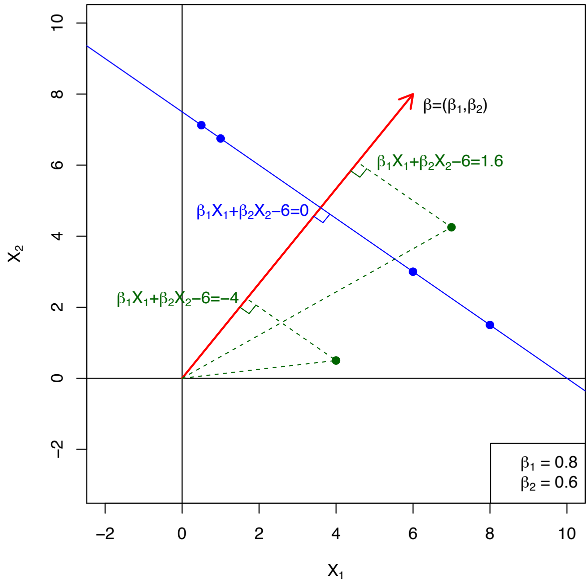
What is a Hyperplane?

- A hyperplane in p dimensions is a flat affine subspace of dimension $p-1$
- In general the equation for a hyperplane has the form

$$\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p = 0$$

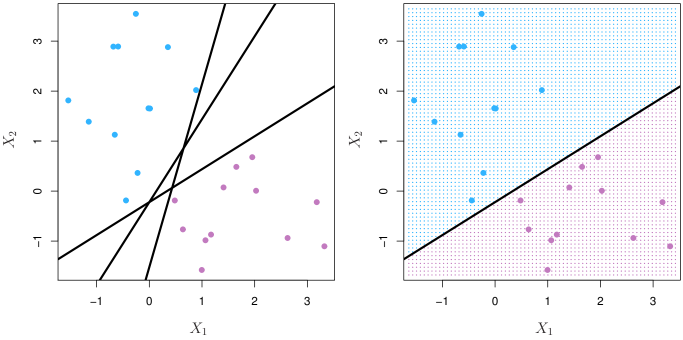
- If $p = 2$, a hyper plane is a line.
- The vector $\beta = (\beta_1, \ldots, \beta_p)$ is called the normal vector of the plane — it points in a direction orthogonal to the surface of a hyperplane.

Hyperplane in 2 Dimensions



Separating Hyperplanes

- If $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_P$, then $f(X) > 0$ for points on one side of the hyperplane, and $f(X) < 0$ for points on the other side.
- We code the colored points as $Y_i = +1$ for blue, say, and $Y_i = -1$ for red,
- if $Y_i f(X_i) > 0$ for all i, then $f(X) = 0$ defines a separating hyperplane.



Maximal Margin Classifier

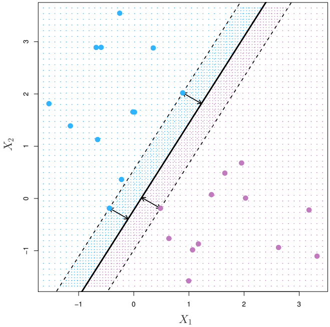
- Among all separating hyperplanes, find the one that makes the biggest gap or margin between the two classes.
- Constrained optimization problem

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

subject to $\|\beta\|_2^2 = \sum_{j=1}^p \beta_j^2 = 1$ and

$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M, \quad i = 1, .., n$$

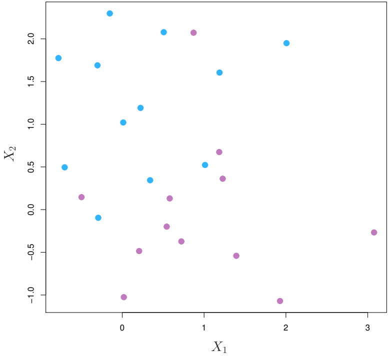
-



This can be rephrased as a convex quadratic program, and solved efficiently. The function `svm()` in package `e1071` solves this problem efficiently

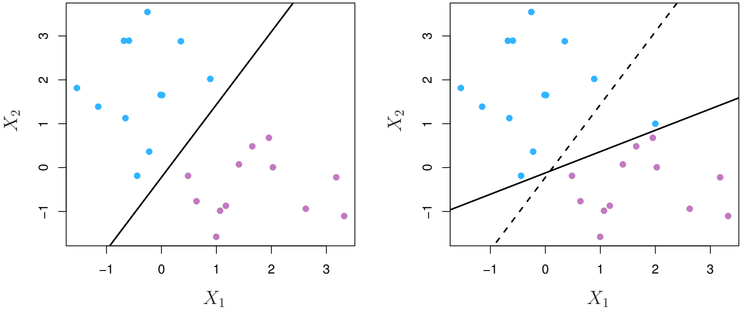
Non-separable Data

- The data on the left are not separable by a linear boundary. This is often the case unless $n < p$.



Noisy Data

- Sometimes the data are separable, but noisy. This can lead to a poor solution for the maximal-margin classifier.



- The support vector classifier maximizes a soft margin.

Support Vector Classifier

- The Support Vector classifier is defined as

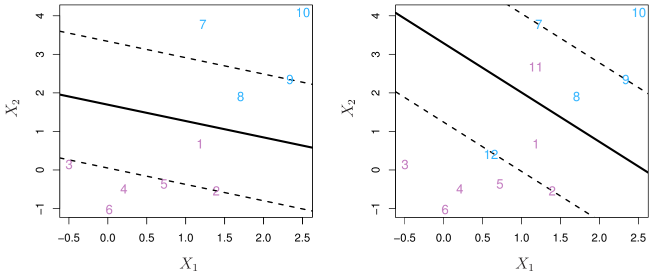
$$\max_{\beta_0,\ldots,\beta_p,\xi_1,\ldots,\xi_n} M$$

  subject to $\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2 = 1$ and

$$y_i(\beta_0 + x_i^T \beta) \geq M(1 - \xi_i), \quad i = 1, .., n$$

  and $\xi_i \geq 0$, $\sum_{i=1}^{n} \xi_i \leq C$
- the term $x_i^T \beta = \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}$
- The value ξ_i in the constraint $y_i(\beta_0 + x_i^T \beta) \geq M(1 - \xi_i)$ is the proportional amount by which the prediction is on the wrong side of its margin.
- by bounding the $\sum_{i=1}^{n} \xi_i$, we bound the total proportional amount by which predictions fall on the wrong side of their margin.
- Misclassification occurs when $\xi_i > 1$

Support Vector Classifier (cont.)



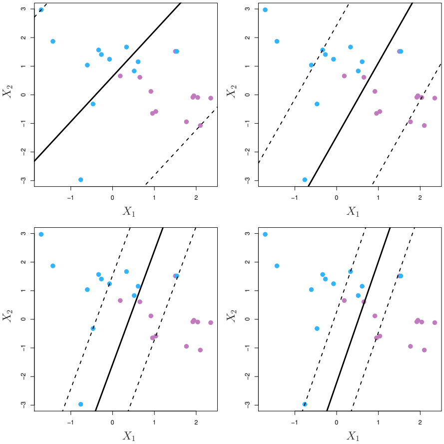
- The problem is quadratic with linear inequality constraints, hence it is a convex optimization problem.
- We describe a quadratic programming solution using Lagrange multipliers

$$\min_{\beta_0,\dots,\beta_p} \frac{1}{2}\|\beta\|_2^2 + \lambda \sum_{i=1}^{n} \xi_i$$

subjet to $\xi_i \geq 0$ and $y_i(\beta_0 + x_i^T \beta) \geq 1 - \xi_i$

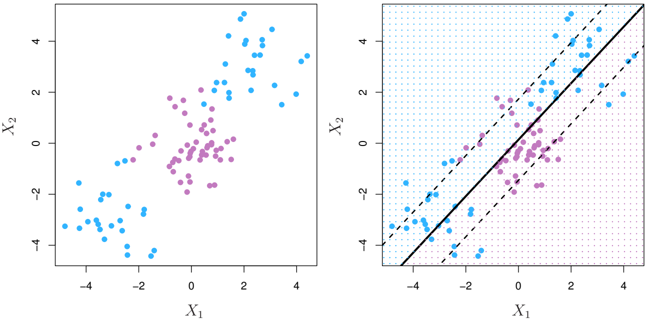
λ is a regularization parameter

- There is a one-one correspondence between C and λ
-



Linear boundary can fail

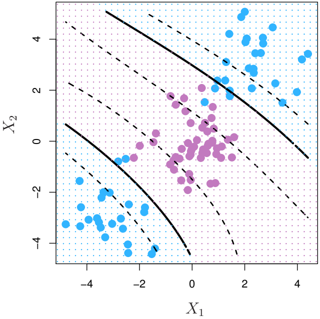
- Sometime a linear boundary simply won't work, no matter what value of C.
-



- we can make the procedure more flexible by enlarging the feature space using basis expansions such as polynomials or splines

Feature Expansion

- Enlarge the space of features by including transformations
- Fit a support-vector classifier in the enlarged space
- This results in non-linear decision boundaries in the original space.
- Example:
  - Cubic Polynomials of two variables
  - $f(x) =$
    $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \beta_6 x_1^3 + \beta_7 x_2^3 + \beta_8 x_1 x_2^2 + \beta_9 x_1^2 x_2$



Nonlinearities and Kernels

- Polynomials (especially high-dimensional ones) get wild rather fast.
- There is a more elegant and controlled way to introduce nonlinearities in support-vector classifiers — through the use of kernels.
- Before we discuss these, let's talk about inner products
- Inner product between vectors $\langle x, y \rangle = \sum_{j=1}^{p} x_j y_j$
- The linear support vector classifier can be represented as

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

- To estimate the parameters $\alpha_1, \ldots, \alpha_n$ and β_0, all we need are the n inner products $\langle x, x_i \rangle$ between all pairs of training observations.

Kernels and Support Vector Machines

- If we can compute inner-products between observations, we can fit a SV classifier.
- Suppose we have transformed features $h(x) = (h_1(x_i), \ldots, h_M(x_i))$, resulting a nonlinear function $f(x) = \hat{\beta}_0 + h(x)^T \hat{\beta}$
- We know that $\hat{\beta}$ is in the form of

$$\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i h(x_i)$$

so that

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

- As in the linear SVM, it turns out that most of the α_i can be zero:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{A}} \hat{\alpha}_i \langle h(x), h(x_i) \rangle$$

where $\mathcal{S}$ is the active set of indices i such that $\hat{\alpha}_i > 0$

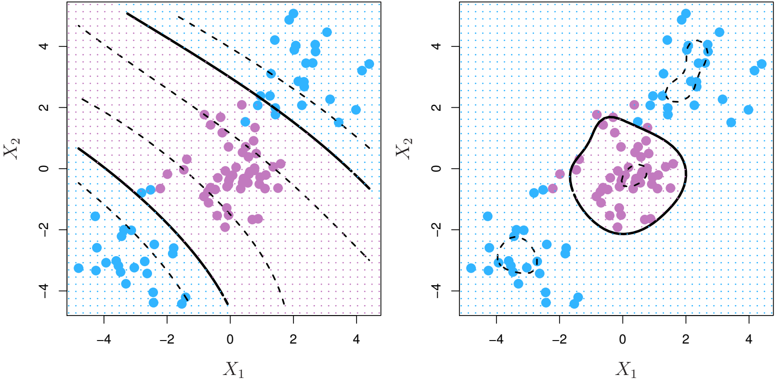
Kernels

- For high-degree polynomial, $h(x)$ could be very "long"
- Fortunately, if we choose the transformation $h(\cdot)$ wisely,

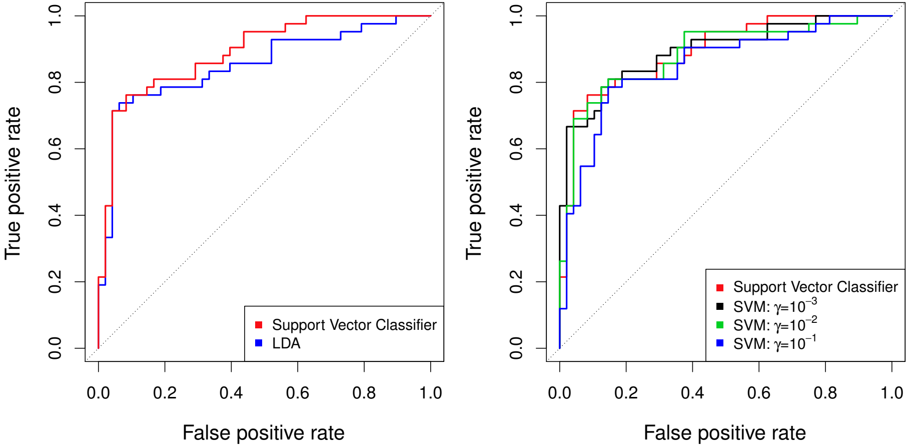
$$\langle h(x), h(x_i) \rangle = (1 + \langle x, x_i \rangle)^d$$

- $K(x, x') = \langle h(x), h(x') \rangle$ is called a kernel
- In fact, we need not specify the transformation $h(x)$ at all, but require only knowledge of the kernel function
- Three popular choices are
  - d-degree polynomial: $K(x, x') = (1 + \langle x, x' \rangle)^d$
  - radial kernel: $K(x, x') = \exp(-\gamma \langle x, x' \rangle)$
  - neural network: $K(x, x') = \tanh(\alpha \langle x, x' \rangle + \beta)$

Example: Radial Kernel



Example: Heart Data



ROC curve is obtained by changing the threshold 0 to threshold t in $\hat{f}(x) > t$, and recording false positive and true positive rates as t varies. Here we see ROC curves on test data.

SVMs: more than 2 classes?

- The SVM as defined works for K = 2 classes. What do we do if we have $K > 2$ classes?

  OVA One versus All. Fit K different 2-class SVM classifiers $\hat{f}_k(x)$, $k = 1, \ldots, K$; each class versus the rest. Classify x_0 to the class for which $\hat{f}_k(x^*)$ is largest

  OVO One versus One. Fit all $\binom{K}{2}$ pairwise classifiers $\hat{f}_{kl}(x)$. Classify x_0 to the class that wins the most pairwise competitions.

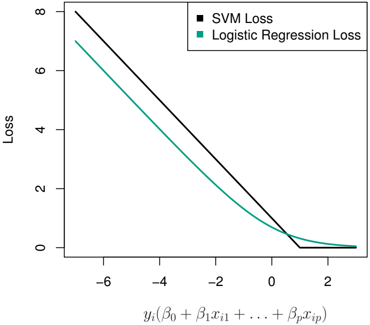
- If K is not too large, use OVO.

Support Vector versus Logistic Regression?

- For the linear SVM, one can rewrite the optimization problem as

$$\min_{\beta}\left\{\sum_{i=1}^{n} l(y_i, f(x_i)) + \lambda\|\beta\|^2\right\}$$

  where $l(y, f(x)) = \max[0, 1 - yf(x)]$
- The loss is known as the hinge loss.



Very similar to negative log-likelihood in logistic regression

Which to use: SVM or Logistic Regression

- When classes are (nearly) separable, SVM does better than LR. So does LDA.
- When not, LR (with ridge penalty) and SVM very similar.
- If you wish to estimate probabilities, LR is the choice.
- For nonlinear boundaries, kernel SVMs are popular. Can use kernels with LR and LDA as well, but computations are more expensive.