Support Vector Machines

October 16, 2018

General information

- support vector machine (SVM) is an approach for classification that was developed in the computer science community in the 1990's
- SVMs have been shown to perform well in a variety of settings, and are often considered one of the best “out of the box” classifiers
- The support vector machine is a generalization of a simple and intuitive classifier called the maximal margin classifier
- Support vector machines are intended for the binary classification setting in which there are two classes
- There are extensions of support vector machines to the case of more than two classes.
- There are close connections between support vector machines and logistic regression.

Some simpler approaches

- Maximal margin classifier is elegant and simple but unfortunately cannot be applied to most data sets, since it requires that the classes are separable by a linear boundary
- Support vector classifier is an extension of the maximal margin classifier that can be applied in a broader range of cases.
- The maximal margin classifier, the support vector classifier, and the support vector machine are often described as “support vector machines”.

What is a hyperplane?

- In a p-dimensional space, a hyperplane is a flat affine subspace of dimension $p - 1$.
- For instance, in two dimensions, a hyperplane is a flat one dimensional subspace - in other words, a line.
- In three dimensions, a hyperplane is a flat two-dimensional subspace – that is, a plane.
- In $p > 3$ dimensions, it can be hard to visualize a hyperplane, but the notion of a $(p - 1)$-dimensional flat subspace still applies.
- Mathematically it is simple. The equation

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

  defines a p-dimensional hyperplane, again in the sense that if a point $X = (X_1, X_2, ..., X_p)$ in the p-dimensional space satisfies the equation, then X lies on the hyperplane.

Hyperplane as a border

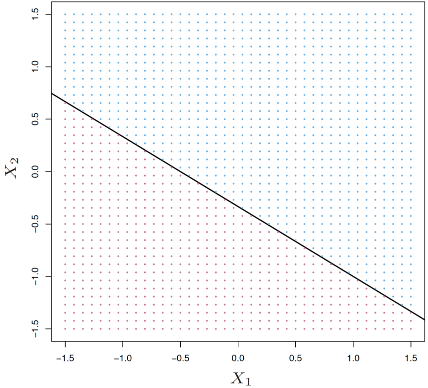
- Hyperplane can be viewed as a plane that divides space into two classes:
- Class in the direction of β:

$$\beta_0 + \beta_1 X_1 + \dots \beta_p X_p > 0$$

- Class in the opposite direction than the one pointed by β:

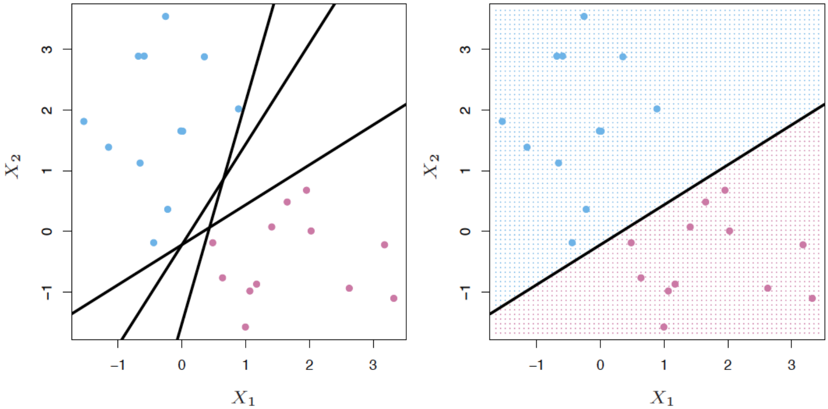
$$\beta_0 + \beta_1 X_1 + \dots \beta_p X_p < 0$$

Example



- The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown.
- The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$
- The purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.

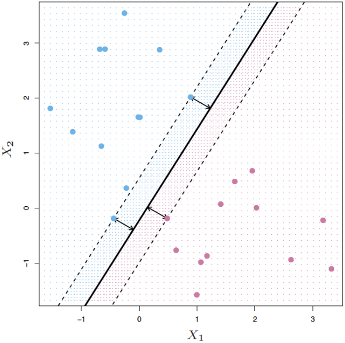
Separating data by hyperplane



- If one can separate data by a hyperplane it can be done in infinitely many ways.
- Which one to choose?

Maximal margin

- A natural choice is the maximal margin hyperplane, which is the separating hyperplane that is hyperplane farthest from the training observations.
- The hyperplane that has the farthest minimum distance to the training observations



- There is the unique solution, with the vectors that are closest to the line named **support vectors** (three of them are seen in the graph)
- Change of the location of other vectors does not change the solution as long as they do not enter the strip that separates the closest observations.

How to construct the maximal margin classifier?

- A set of **training observations** $x_1, ..., x_n \in \mathbb{R}^p$
- Associated class labels $y_1, ..., y_n \in \{-1, 1\}$.
- Solve the following problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

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

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \;\; \forall\, i = 1, \ldots, n.$$

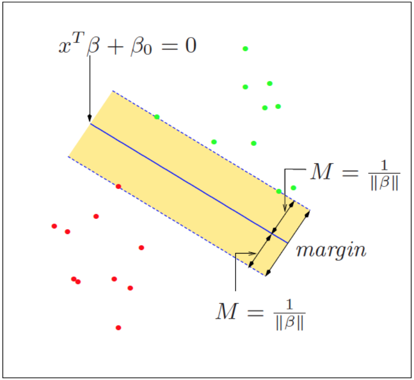
- Equivalently

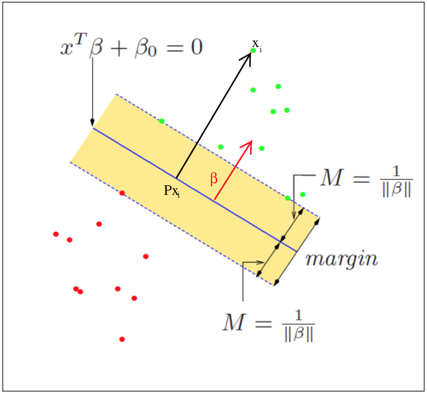
$$\min_{\beta,\beta_0} \|\beta\|$$

$$\text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1,\; i = 1, \ldots, N,$$

- Why are there equivalent? Take $M = 1/\|\beta\|$.

Graphical interpretation



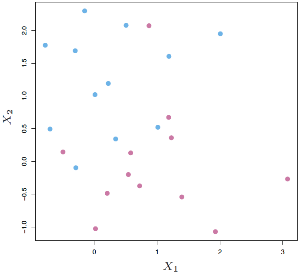


If $\|\beta\| = 1$, then $x^T\beta + \beta_0$ is the distance of x_i from the hyperplane, thus maximizing M is maximizing the margin of the distance from the plane.

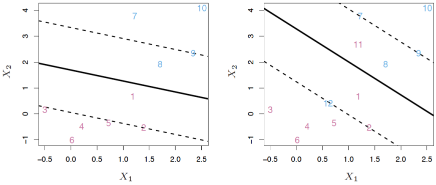
What if there is no margin?

The non-separable case:
If there is no hyperplane to separated two sets than the idea fails.
One needs a modification of the method.

No separation hyperplane



How to separate? Example:



Relaxing separation constrains

- The problem can be solved by introduction of *slack variables*: $\epsilon_1, \ldots, \epsilon_n$.
- These variables allow individual variables to be on the wrong side of the margin
  - If $\epsilon_i = 0$ then the *i*th observation is on the correct side of the margin
  - If $\epsilon_i > 0$ then the *i*th observation is on the wrong side of the margin.
  - If $\epsilon_i > 1$ then the *i*th observation is on the wrong side of the hyperplane.

Graphical interpretation

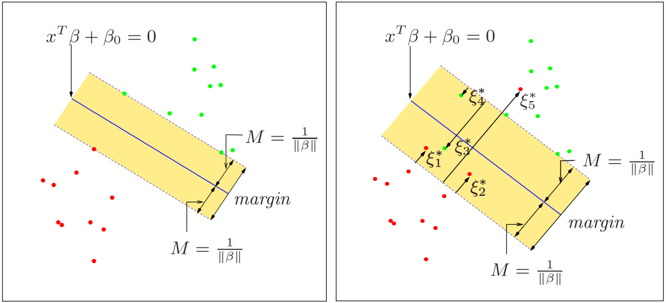


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width $2M = 2/\|\beta\|$. The right panel shows the nonseparable (overlap) case. The points labeled ξ_j^* are on the wrong 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.*

On the graph $1 > \xi_1 > 0$, $1 > \xi_2 > 0$, $1 > \xi_3 > 0$, $1 > \xi_4 > 0$, $\xi_5 > 1$.

Modified optimization problem

- The problem reduces to solving the following optimization problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

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

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

$$\epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq C,$$

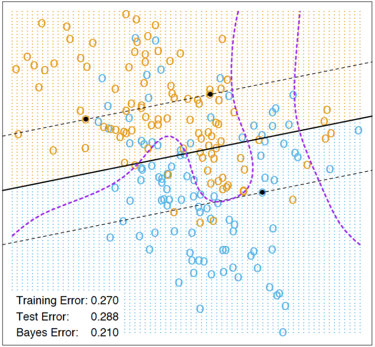
- The parameter $C > 0$ plays the role of tuning parameter that describes the size of the margin around boundary that allows for being on the wrong side of a hyperplane

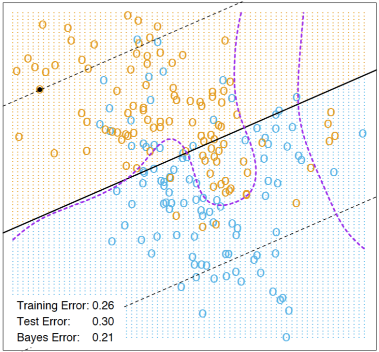
The role of tuning

- C bounds the sum of the ϵ_i's, and so it determines the number and severity of the violations to the margin (and to the hyperplane) that will be tolerated.
- C is a budget for the amount that the margin can be violated by the n observations.
    - If $C = 0$ then there is no budget for violations to the margin
    - For $C > 0$ no more than C observations can be on the wrong side of the hyperplane, because if an observation is on the wrong side of the hyperplane then $\epsilon_i > 1$, and $\sum_{i=1}^{n} \epsilon_i \leq C$.
    - As the budget C increases, there is more tolerance of violations to the margin, and so the margin will widen. Conversely, as C decreases, we become less tolerant of violations to the margin and so the margin narrows.

Example of Gaussian mixtures

- The vector support classifier for $C = 0.00001$ (left) and $C = 100$ (right)





Summary of properties

- Only observations that either lie on the margin or that violate the margin will affect the hyperplane–an observation that lies strictly on the correct side of the margin does not affect the support vector classifier!
- Observations that lie directly on the margin, or on the wrong side of the margin for their class, are known as support vectors. These observations do affect the support vector classifier.
- When the tuning parameter C is large, then the margin is wide, many observations violate the margin, and so there are many support vectors. In this case, many observations are involved in determining the hyperplane – the classifier has low variance but potentially high bias (non-smooth).
- In contrast, if C is small, then there will be fewer support vectors and hence the resulting classifier will have low bias but high variance (smooth).
- The decision rule is based only on a potentially small subset of the training observations (the support vectors). It is quite robust to the behavior of observations that are far away from the hyperplane – distinct from other classification methods such as linear discriminant analysis.

Handling non-linearity

- Vector support classifiers are limited in the way that the boundaries are linear
- Support vector machines are extensions of the previous methods to non-linear boundaries
- Simple way can be made by adding ‘higher order’ variables: we could fit a support vector classifier using $2p$ features

$$X_1, X_1^2, X_2, X_2^2, ..., X_p, X_p^2$$

Solve

$$\underset{\beta_0,\beta_{11},\beta_{12}\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M \qquad\qquad ($$

$$\text{subject to } y_i\left(\beta_0+\sum_{j=1}^{p}\beta_{j1}x_{ij}+\sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right)\geq M(1-\epsilon_i),$$

$$\sum_{i=1}^{n}\epsilon_i\leq C,\quad \epsilon_i\geq 0,\quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2=1.$$

Which non-linear functions?

- There are many way of introducing non-linear variables.
- Support vector machines are using original structure of the method
- It can be shown that in the original linear problem the data entered the computation only through the inner products:

$$\langle x, x' \rangle = \sum_{i=1}^{p} x_i x_i' = |x||x'| \cos \alpha$$

  Only angles and length are used!
- Generalization to a non-linear case is replacing in the computations $\langle x, x' \rangle$ by a non-linear kernel function

$$K(x, x')$$

The classifier

- The classical classifier can be written as

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

- If a data point x_i is outside of the margin (x_i is not a support vector), then the corresponding α_i is vanishing so that

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle$$

  where S are indices of the support vectors.

- In non-linear approach, we replace the inner product in the procedure by a non-linear kernel function so that the final classifier takes the form

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i)$$

Kernels

Several classes of kernels are popular in application of the method

- Polynomial kernel of degree *d*:

$$K(x, x') = \left(1 + \sum_{j=1}^{p} x_j x_j'\right)^d$$

- Radial kernel

$$K(x, x') = \exp\left(-\gamma \sum_{j=1}^{p} \left(x_j - x_j'\right)^2\right)$$

- In non-linear approach, we replace the inner product in the procedure by a non-linear kernel function so that the final classifier takes the form

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i)$$

Illustration

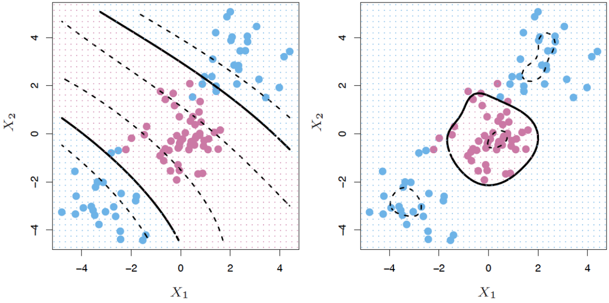
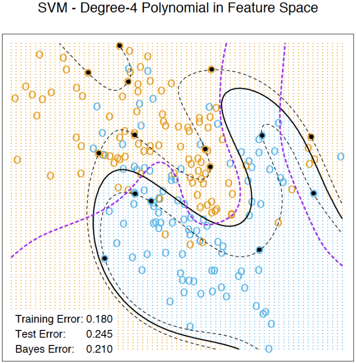
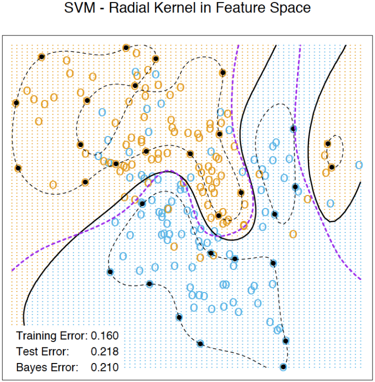


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

Gaussian mixture data


SVM - Degree-4 Polynomial in Feature Space


SVM - Radial Kernel in Feature Space

Benefit over additional features approach

- Instead of using unspecified number of additional functions to add features that may result in complicated computation one needs only provide the matrix

  $$K(x_i, x_j), i, j = 1, \ldots, n$$

  There are only $\binom{n}{2}$ distinct numerical evaluations.
- Suitable for complex problems

Extension to more than two classes

- It turns out that the concept of separating hyperplanes upon which SVMs are based does not lend itself naturally to more than two classes. The two most popular approaches are:
    - **one-versus-one** – $\binom{K}{2}$ one-versus-one SVMs, each of which compares a pair of classes, then tally the number of times that the test observation is assigned to each of the K classes. The final classification is performed by assigning the test observation to the class to which it was most frequently assigned in these pairwise classifications.
    - **one-versus-all** – comparing one of the K classes to the remaining $K - 1$ classes. We assign the observation x^* to the class for which $\beta_{0k} + \beta_{1k}x_1^* + \beta_{2k}x_2^* + ... + \beta_{pk}x_p^*$ is largest, as this amounts to a high level of confidence that the test observation belongs to the kth class rather than to any of the other classes.

Computational tools

- One popular choice is the `e1071` library in **R**.
- Another option is the `LiblineaR` library, which is useful for very large linear problems.
- The `e1071` library contains implementations for a number of statistical learning methods. In particular, the `svm()` function can be used to fit a support vector classifier when the argument kernel=”linear” is used.
- This function uses a slightly different formulation. A cost argument allows us to specify the cost of a violation to the margin. When the cost argument is small, then the margins will be wide and many support vectors will be on the margin or will violate the margin. When the cost argument is large, then the margins will be narrow and there will be few support vectors on the margin or violating the margin.