

Classification: Support Vector Machine (SVM)

Prof. (Dr.) Honey Sharma

What Is a Hyperplane?

In a p -dimensional space, a hyperplane is a flat affine subspace of hyperplane 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.

The mathematical definition of a hyperplane is quite simple.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p = 0$$

defines a p -dimensional hyperplane, again in the sense that if a point $X = (X_1, X_2, \dots, X_p)'$ in p -dimensional space (i.e. a vector of length p) satisfies above equation, then X lies on the hyperplane.

Now, suppose that X does not satisfy hyperplane; rather,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p > 0$$

Then this tells us that X lies to one side of the hyperplane. On the other hand, if

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p < 0$$

then X lies on the other side of the hyperplane. So we can think of the hyperplane as dividing p -dimensional space into two halves. One can easily determine on which side of the hyperplane a point lies by simply calculating the sign of the equation.

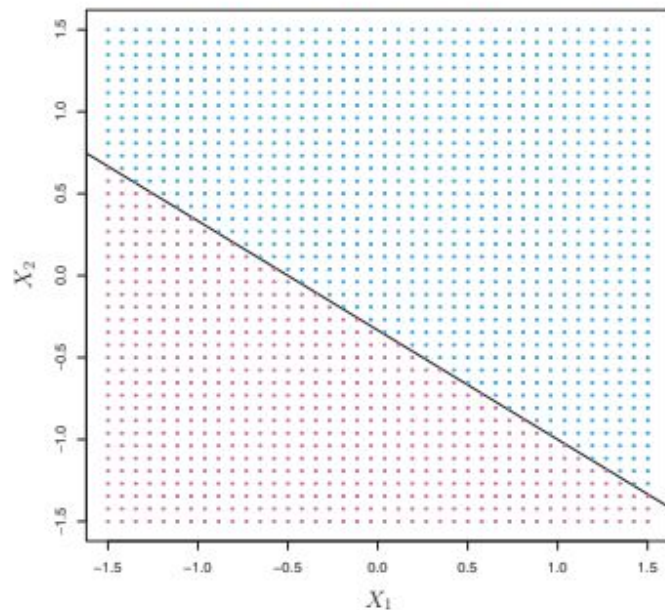


FIGURE 9.1. The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.

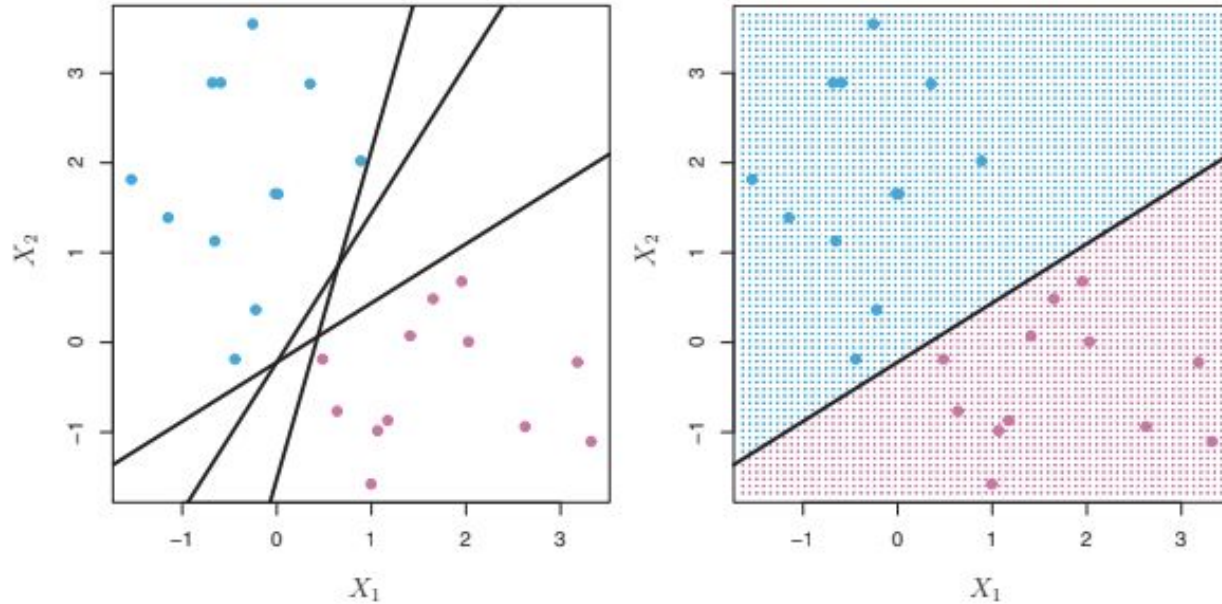
Classification Using a Separating Hyperplane

Now suppose that we have a $n \times p$ data matrix X that consists of n training observations in p -dimensional space,

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \dots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix},$$

and that these observations fall into two classes—that is, $y_1, \dots, y_n \in \{-1, 1\}$ where -1 represents one class and 1 the other class. We also have a test observation, a p -vector of observed features $x^* = \{x_1^*, \dots, x_p^*\}$

Our goal is to develop a classifier based on the training data that will correctly classify the test observation using its feature measurements.



Left: There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black. Right: A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.

We can label the observations from the blue class as $y_i = 1$ and those from the purple class as $y_i = -1$. Then a separating hyperplane has the property that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} > 0 \text{ if } y_i = 1,$$

and

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} < 0 \text{ if } y_i = -1.$$

Equivalently, a separating hyperplane has the property that

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) > 0$$

for all $i = 1, \dots, n$.

If a separating hyperplane exists, we can use it to construct a very natural classifier: a test observation is assigned a class depending on which side of the hyperplane it is located.

Maximal Margin Classifier

In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.

This is because a given separating hyperplane can usually be shifted a tiny bit up or down, or rotated, without coming into contact with any of the observations.

In order to construct a classifier based upon a separating hyperplane, we must have a reasonable way to decide which of the infinite possible separating hyperplanes to use.

A natural choice is the **maximal margin hyperplane (also known as the optimal separating hyperplane)**, which is the separating hyperplane that is farthest from the training observations.

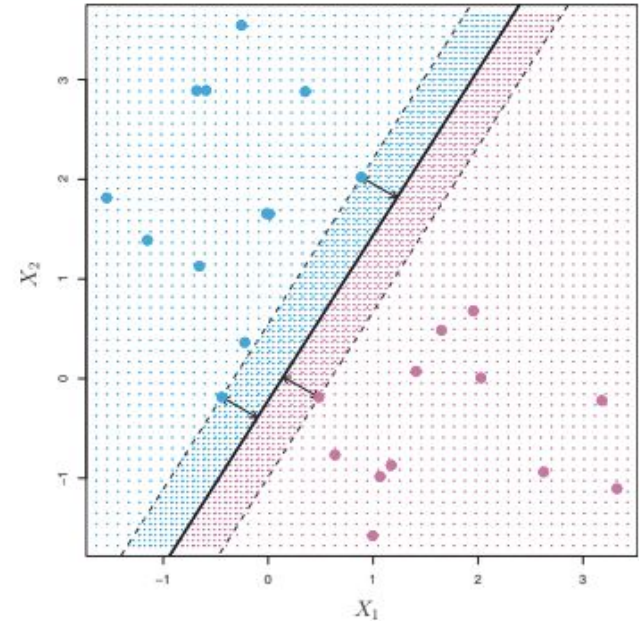
That is, we can compute the (perpendicular) distance from each training observation to a given separating hyperplane; the smallest such distance is the minimal distance from the observations to the hyperplane, and is known as the **margin**.

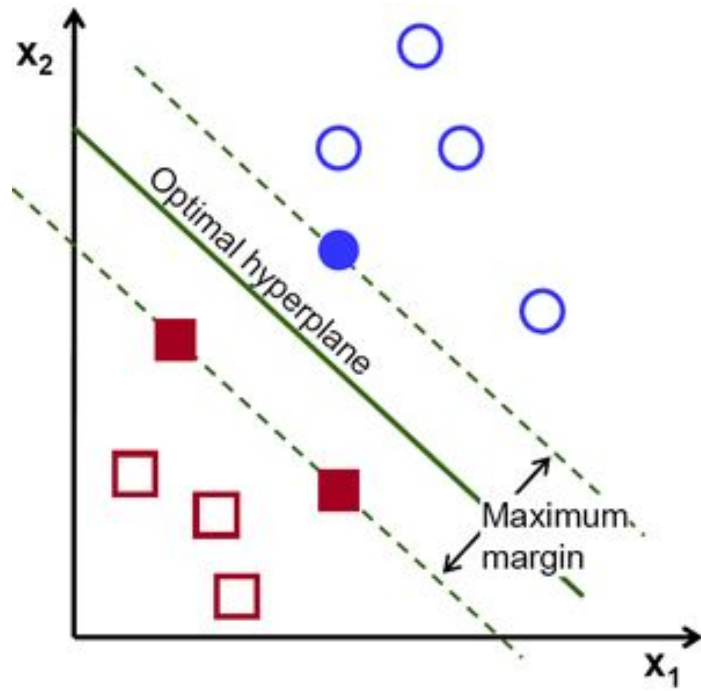
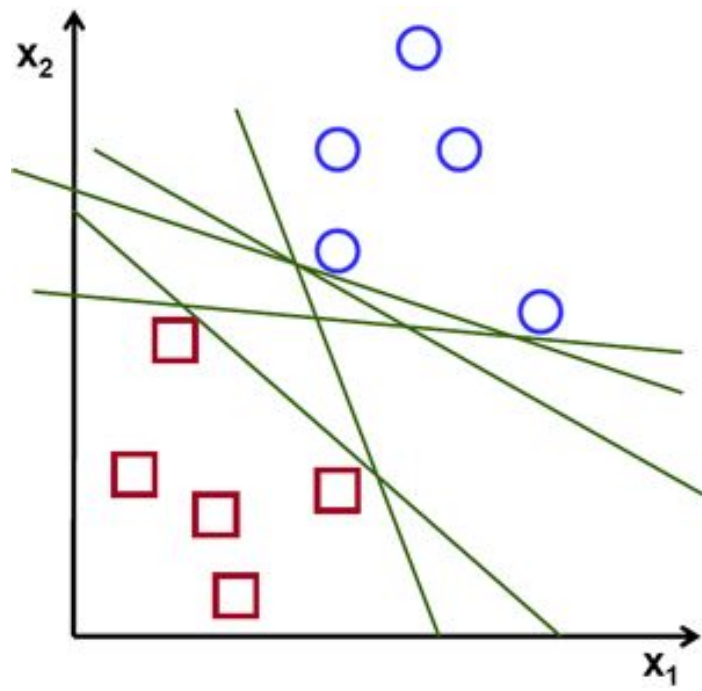
The **maximal margin hyperplane** is the separating hyperplane for which the margin is largest—that is, it is the hyperplane that has the farthest minimum distance to the training observations.

We can then classify a test observation based on which side of the maximal margin hyperplane it lies. This is known as the maximal margin classifier. We hope that a classifier that has a large margin on the training data will also have a large margin on the test data, and hence will classify the test observations correctly. Although the maximal margin classifier is often successful, it can also lead to overfitting.

There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. **The two blue points and the purple point that lie on the dashed lines are the support vectors**, they are vectors in 2-dimensional space and they “support” the maximal margin hyperplane in the sense vector that if these points were moved slightly then the maximal margin hyperplane would move as well.

Interestingly, the **maximal margin hyperplane depends directly on the support vectors, but not on the other observations**: a movement to any of the other observations would not affect the separating hyperplane, provided that the observation’s movement does not cause it to cross the boundary set by the margin.





Construction of the Maximal Margin Classifier

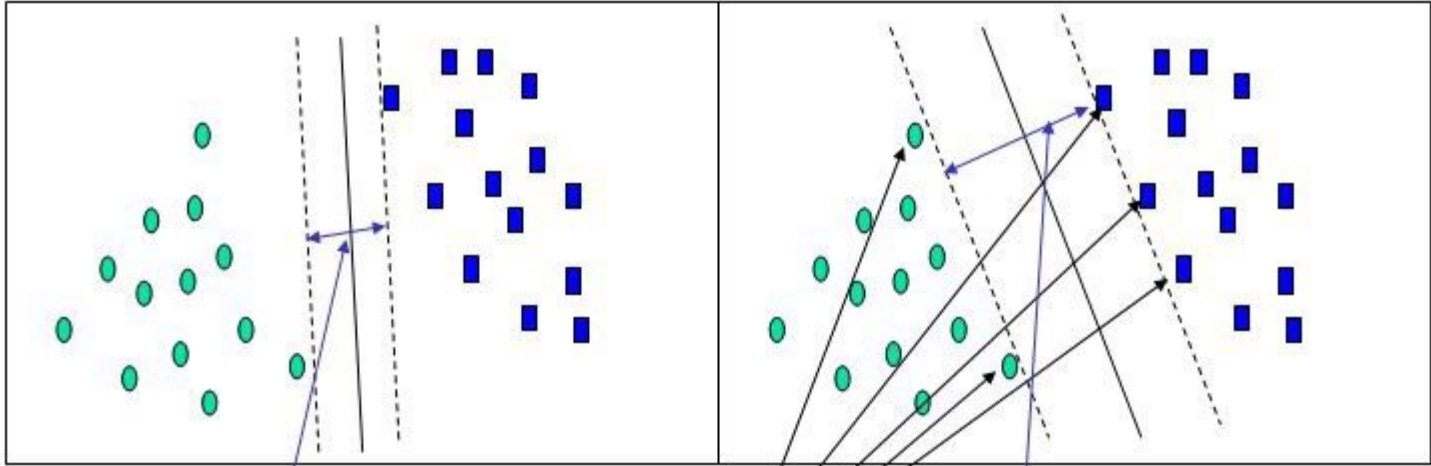
We now consider the task of constructing the maximal margin hyperplane based on a set of n training observations $x_1, \dots, x_n \in \mathbb{R}^p$ and associated class labels $y_1, \dots, y_n \in \{-1, 1\}$. Briefly, the maximal margin hyperplane is the solution to the optimization problem.

$$\begin{aligned} & \underset{\beta_0, \beta_1, \dots, \beta_p}{\text{maximize}} \quad M \\ & \text{subject to} \quad \sum_{j=1}^p \beta_j^2 = 1, \\ & \quad y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M \quad \forall i = 1, \dots, n. \end{aligned}$$

First of all, the constraint $y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M \forall i = 1, \dots, n$

guarantees that each observation will be on the correct side of the hyperplane, provided that M is positive. (Actually, for each observation to be on the correct side of the hyperplane we would simply need LHS of the constraint should be > 0 , so the **constraint in fact requires that each observation be on the correct side of the hyperplane, with some cushion, provided that M is positive.**)

Hence, M represents the margin of our hyperplane, and the optimization problem chooses $\beta_0, \beta_1, \dots, \beta_p$ to maximize M . This is exactly the definition of the maximal margin hyperplane!. **M can be seen as a measure of our confidence that the observation was correctly classified.**



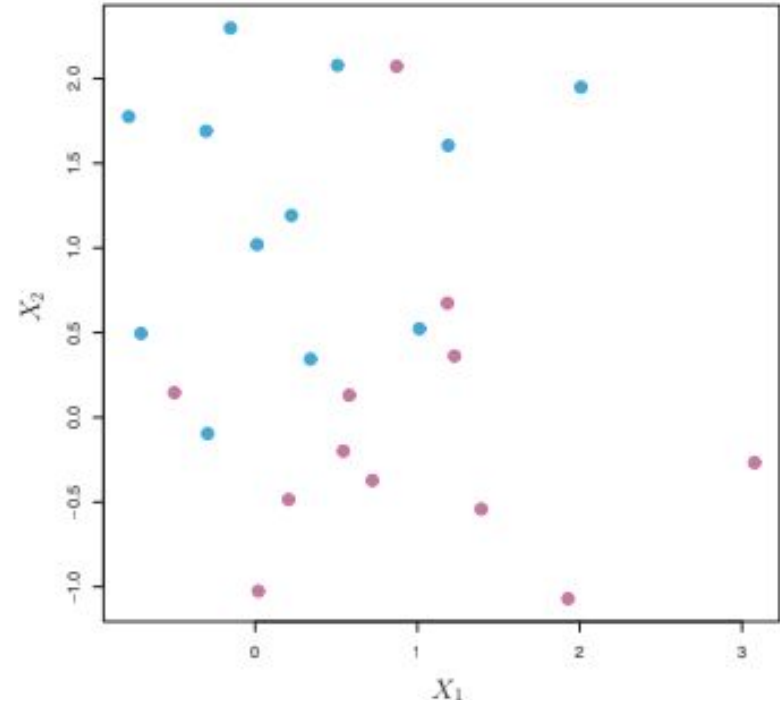
Small Margin

Large Margin

Support Vectors

The Non-separable Case

The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists. However, as we have hinted, in many cases no separating hyperplane exists, and so there is no maximal margin classifier. In this case, we cannot exactly separate the two classes. However, as we will see in the next slide, we can extend the concept of a separating hyperplane in order to develop a hyperplane that almost separates the classes, using a so-called soft margin. The generalization of the maximal margin classifier to the non-separable case is known as the support vector classifier.



Support Vector Classifiers

We seen that observations that belong to two classes are not necessarily separable by a hyperplane.

A classifier based on a separating hyperplane will necessarily perfectly classify all of the training observations; this can lead to sensitivity to individual observations. Even the addition of a single observation may leads to a dramatic change in the maximal margin hyperplane.

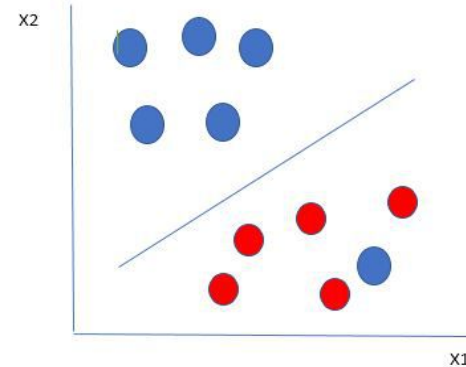
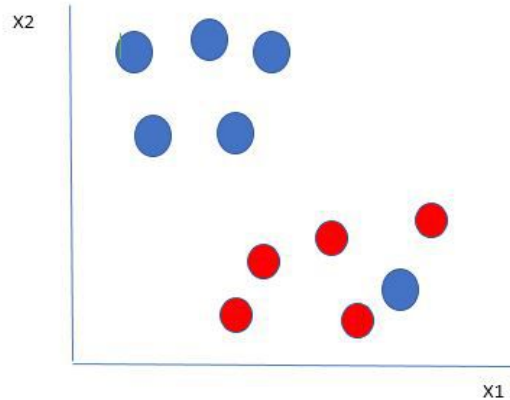
The fact that the maximal margin hyperplane is extremely sensitive to a change in a single observation suggests that it may have overfit the training data.

We might be willing to consider a classifier based on a hyperplane that does not perfectly separate the two classes, in the interest of

- Greater robustness to individual observations, and
- Better classification of most of the training observations.

That is, it could be worthwhile to misclassify a few training observations in order to do a better job in classifying the remaining observations.

The support vector classifier, sometimes called a **soft margin classifier**, support does exactly this. Rather than seeking the largest possible margin so that every observation is not only on the correct side of the hyperplane but also on the correct side of the margin, we instead allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.



The support vector classifier is the solution to the optimization problem

$$\begin{aligned} & \underset{\beta_0, \beta_1, \dots, \beta_p, \epsilon_1, \dots, \epsilon_n}{\text{maximize}} && M \\ & \text{subject to} && \sum_{j=1}^p \beta_j^2 = 1, \\ & && y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \\ & && \epsilon_i \geq 0, \quad \sum_{i=1}^n \epsilon_i \leq C, \end{aligned}$$

where

- C is a nonnegative tuning parameter;
- M is the width of the margin; we seek to make this quantity as large as possible.
- $\epsilon_1, \dots, \epsilon_n$ are slack variables that allow individual observations to be on slack the wrong side of the margin or the hyperplane.

First of all, the slack variable ϵ_i tells us where the i th observation is located, relative to the hyperplane and relative to the margin. If

- $\epsilon_i = 0$ then the i th observation is on the correct side of the margin
- $\epsilon_i > 0$ then the i th observation is on the wrong side of the margin,
- $\epsilon_i > 1$ then it is on the wrong side of the hyperplane.

We now consider the role of the tuning parameter C .

C bounds the sum of the ξ_i 's, and so it determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate. We can think of C as 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, and it must be the case that $\xi_1 = \dots = \xi_n = 0$, in which case it simply amounts to the maximal margin hyperplane and of course, a maximal margin hyperplane exists only if the two classes are separable.

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 $\xi_i > 1$.

As the budget C increases, we become more tolerant 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.

In practice, C is treated as a tuning parameter that is generally chosen via cross-validation. C controls the bias-variance trade-off of the statistical learning technique.

When C is small, we seek narrow margins that are rarely violated, fewer support vectors; this amounts to a classifier that is highly fit to the data, which may have low bias but high variance.

On the other hand, when C is larger, the margin is wider and we allow more violations to it so there are many support vectors; this amounts to fitting the data less hard and obtaining a classifier that is potentially more biased but may have lower variance.

Classification with Non-linear Decision Boundaries

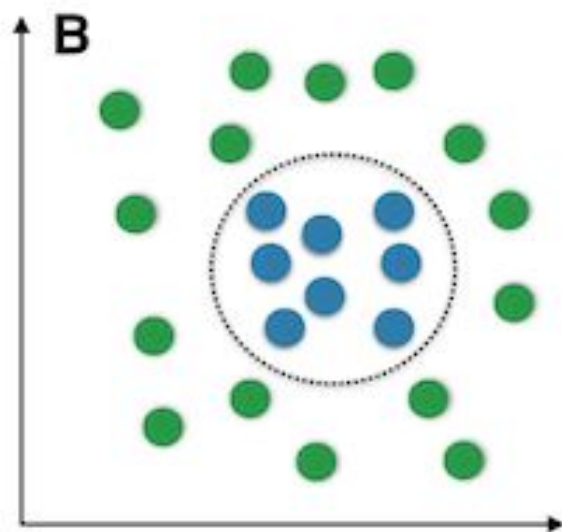
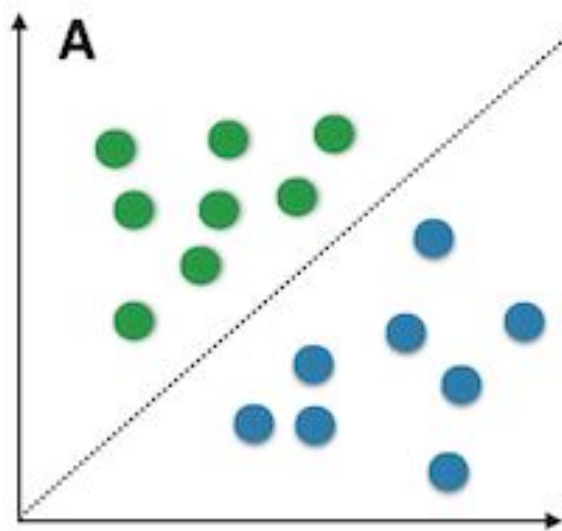
In practice we are sometimes faced with non-linear class boundaries. In the case of the support vector classifier, we could address the problem of possibly non-linear boundaries between classes in a similar way, by enlarging the feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors. For instance, rather than fitting a support vector classifier using p features

$$X_1, X_2, \dots, X_p$$

we could instead fit a support vector classifier using $2p$ features

$$X_1, X_1^2, X_2, X_2^2, \dots, X_p, X_p^2$$

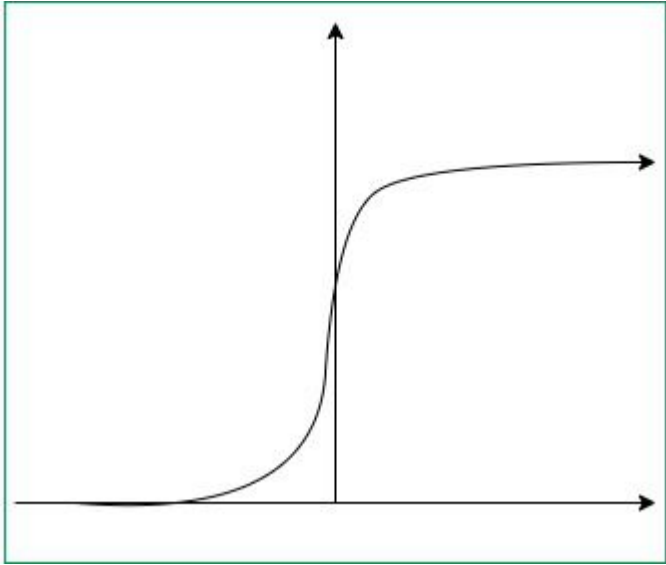
Linear vs. nonlinear problems



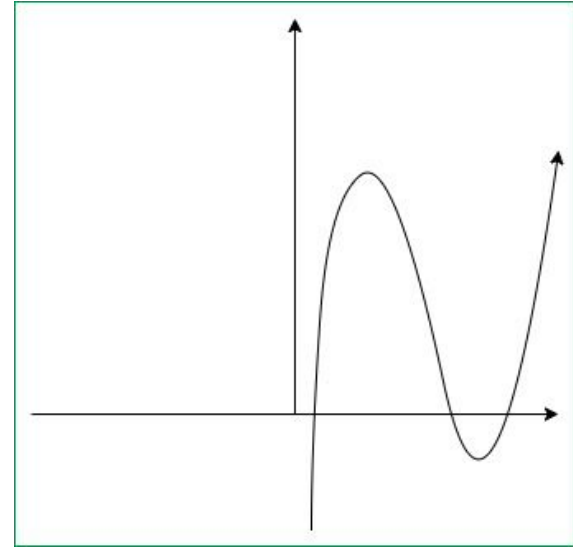
Support Vector Machine (SVM)

The support vector machine (SVM) is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using kernels. we may want to enlarge our feature space in order to accommodate a non-linear boundary between the classes.

Popular kernels are: Polynomial Kernel, Gaussian Kernel, Radial Basis Function (RBF), Laplace RBF Kernel, Sigmoid Kernel, Anove RBF Kernel



Sigmoid Kernel



Polynomial Kernel