Support Vector Machines

1. **Separable Case:** Maximal Margin Classifier

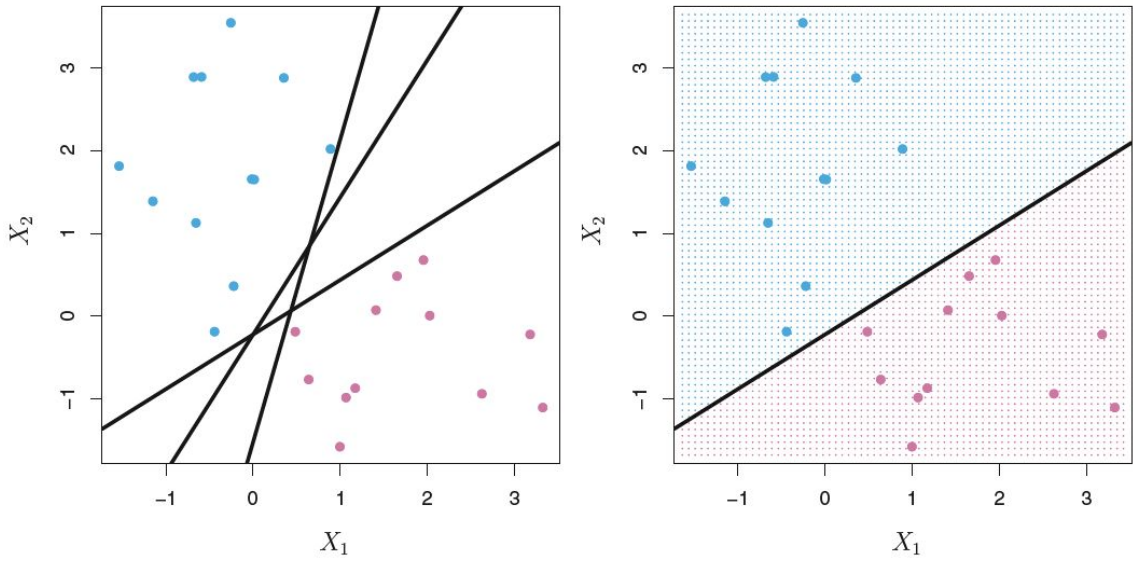
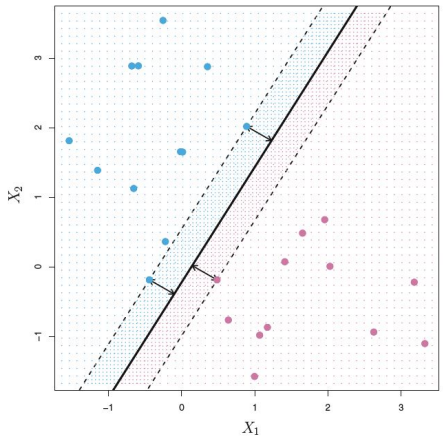


Image Courtesy: Fig 9.2 from Ref 1

Infinite number of separating hyperplanes possible. In the above case, it is 2D hyperplane, that is a line. If a line is slightly tilted, it will still separate the training sets into same group, but its performance on points not seen before will be different. Hence we choose the one which has farthest from all the training sets to be the one. Minimum distance (taken by drawing perpendicular from point to the line) taken over all the points is called “margin”. The maximal margin hyperplane is the separating hyperplane for which the margin is largest. It has tendency to “overfit”.

9. Support Vector Machines



The points shown with lines drawn are at minimum margins. They are called “support vectors”, as they support the separating hyperplane. If these are moved slightly, then separating line would move as well.

Some details:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{maximize}} \; M \tag{9.9}$$

$$\text{subject to} \; \sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.10}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \;\; \forall \, i = 1, \ldots, n. \tag{9.11}$$

M is the margin. Higher M, the better. It depends on points.

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

Guarantees that each observation is on the correct side of the line (M = 0+ also will do, M > 0 ensures certain margin)

2. The non-Separable Case: Support Vector Classifier
Many cases, no clear separation and hence no hyperplane. We can have so called soft margin. We allow some points to be on the wrong side of hyperplane. SV Classifier is a generalization of Maximal Margin Classifier explained above.

9.2 Support Vector Classifier:
Even if a separating hyperplane exists it is possible that a classifier based on a separating hyperplane might not be desirable. Reason is it might have very small margin, it may be sensitive to small change in training data. It might have overfit to the training data.
It may be worthwhile to allow some points on the wrong side of hyperplane or even wrong side of margin to reduce the overfit. Shown in figure.

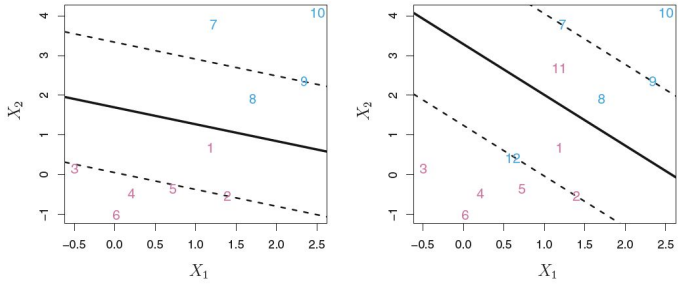


FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation* 1 *is on the wrong side of the margin.* Blue observations: *Observations* 7 *and* 10 *are on the correct side of the margin, observation* 9 *is on the margin, and observation* 8 *is on the wrong side of the margin. No observations are on the wrong side of the hyperplane.* Right: *Same as left panel with two additional points,* 11 *and* 12. *These two observations are on the wrong side of the hyperplane and the wrong side of the margin.*

Modified definition

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} \quad M \tag{9.12}$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.13}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{9.14}$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C, \tag{9.15}$$

C is non-negative obviously.
Note that sum of epsilons are less than C, but epsilon might be greater than 1, which can make right hand side negative. This makes classification on the wrong side for those points.

greater detail momentarily. Once we have solved (9.12)–(9.15), we classify a test observation x^* as before, by simply determining on which side of the hyperplane it lies. That is, we classify the test observation based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \ldots + \beta_p x_p^*$.

If epsilon is 0, then ith point is on the correct size of margin and hyperplane.

If $\epsilon_i > 0$ then the ith observation is on the wrong side of the margin, and we say that the ith observation has *violated* the margin. If $\epsilon_i > 1$ then it is on the wrong side of the hyperplane.

C determines the number of violations and severity we can tolerate in our classifier.

If C = 0, then it is the maximal margin classifier that we studied in the beginning.

For any C > 0, no more than C observations can be on the wrong side of the hyperplane. Why? Because that implies epsilon for i > 1, say it is 1+, then sum of all those will be C+, and not C.

As C increases, we are more tolerant to violations and margin widens.

C is small - overfit (margin is small), C wide - bias

The equations have an interesting property.
The points that lie on the margin or within the margin decide the hyperplane boundary. The points that are strictly on the correct side dont affect the boundary at all.
The fact that the support vector classifier's decision rule is based only on a potentially small subset of the training observations (the support vectors) means that it is quite robust to the

behavior of observations that are far away from the hyperplane. LDA is sensitive to all points. In contrast, logistic regression, unlike LDA, has very low sensitivity to observations far from the decision boundary. In fact we will see in Section 9.5 that the support vector classifier and logistic regression are closely related.

3. Non linear boundaries case: Support Vector Machine
Done with kernels.

9.3.2 The Support Vector Machine

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 will now discuss this extension, the details of which are

We want to enlarge the feature space to accommodate a non-linear boundary between classes.
In the support vector classifier

It can be shown that

- The linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle, \qquad (9.18)$$

where there are n parameters α_i, $i = 1, \ldots, n$, one per training observation.

If a training observation is not a support vector, (ie. it is not in margin or on the margin line) corresponding alpha is 0. To compute alphas, we need k*(k-1)/2 inner product pairs, where k is the number of support vectors.
Now generalize 9.18 from inner product to some kernel function K
Linear kernel is what we already know and it is also pearson correlation (inner product is nothing but that)

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}, \qquad (9.21)$$

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d. \qquad (9.22)$$

- Polynomial kernel of degree d.

It essentially amounts to fitting a support vector classifier in a higher-dimensional space involving polynomials of degree d, rather than in the original feature space. When the support vector classifier is combined with a non-linear kernel such as (9.22), the resulting classifier is known as a support vector machine.
Another option is radial kernel: gamma is positive.

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2). \qquad (9.24)$$

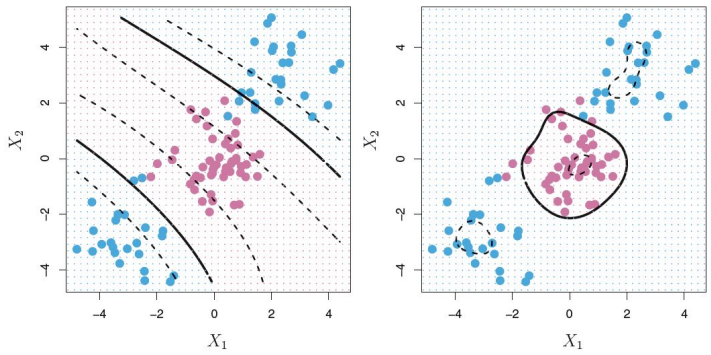


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

The striking thing above is either of these is capable of making the right decision.

How does radial kernel work?

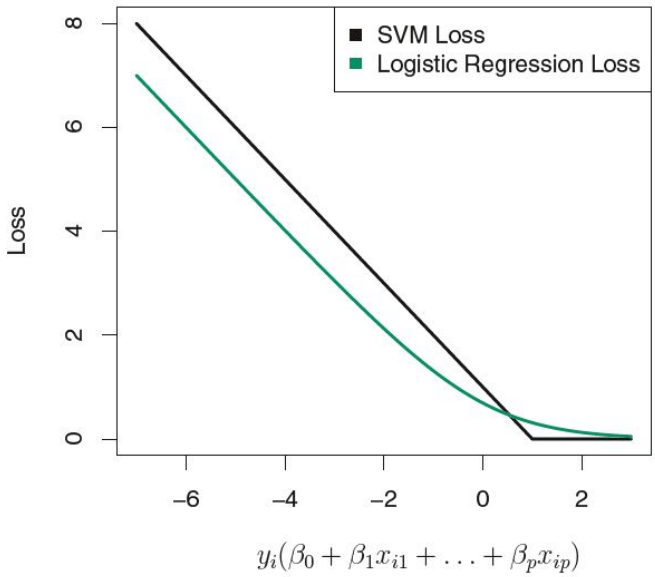
When distance is large, that observation's K is small.

will play virtually no role in $f(x^*)$. Recall that the predicted class label for the test observation x^* is based on the sign of $f(x^*)$. In other words, training observations that are far from x^* will play essentially no role in the predicted class label for x^*. This means that the radial kernel has very *local* behavior, in the sense that only nearby training observations have an

"For some kernels, such as the radial kernel (9.24), the feature space is implicit and infinite-dimensional, so we could never do the computations there anyway!"

Relation to Logreg

When the classes are well separated, SVMs tend to behave better than logistic regression; in more overlapping regimes, logistic regression is often preferred. Details in the book



Extension to Multiclass:
SVM is not easy to extend to multiclass. Common practice is training k one vs Rest classifier for a k class classification. Also fitting k*(k-1)/2 one-one classifier is possible