Support Vector Machines

Objectives

- Classification of binary outcomes based on a metric feature space using hyper-boundaries.

Background

- The underlying idea comes from using a *maximal margin classifier*, which however is too restrictive in that it requires perfect separation between the classes in the feature space.
- The *support vector classifier* allows controlling for sampling variability and relaxes the assumption of perfectly separable classes.
- The *support vector machine* is an extension of the *support vector classifier*. It also allows for overlapping classes and non-linear hyper-boundaries between two classes.
- The support vector machine and logistic regression are quite similar for linear predictors (see Gareth James et al. pp 356-359).

Comments

- SVM is a distance-based procedure. Consequently, all problems associated with scaling, transformations, and redundancy among the features are relevant.
- While SVM are primarily designed for binary classification problems, they can also be adopted to a multiple classification problem and predictions of metric target variables (see Boehmke, section 14.3.2).

Maximal Margin Classifier

Hyperplane

- In the p-dimension feature space there exists an affine $(p-1)$-dimension hyperplane that splits the feature space.
- Example in $p = 2$ for
  - $\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 > 0$ observations are on one side of the feature space, for
  - $\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 < 0$ observations are on the other side of the feature space, and for
  - $\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 = 0$ observations are on the separating hyperplane.

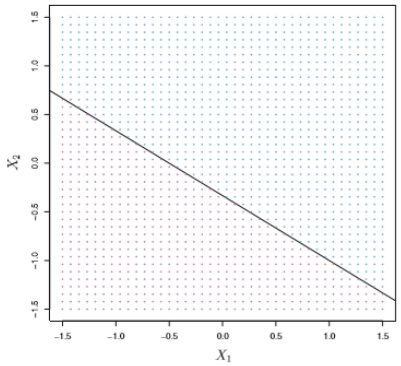


FIGURE 9.1. *The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.*

- An affine equation can be rewritten as $X_1 = -\frac{\beta_0}{\beta_2} - \frac{\beta_1}{\beta_2} \cdot X_2$. Thus, scaling of the coefficients β_0, β_1 and β_2 is arbitrary and $c \cdot \beta_0, c \cdot \beta_1$ and $c \cdot \beta_2$ leads to an identical hyperplane.

That's why we need to do scaling

Classification Rule based on Hyperplanes

- If the outcomes in the feature space can be perfectly separated by a hyperplane, then there exists an infinite number of hyperplanes:

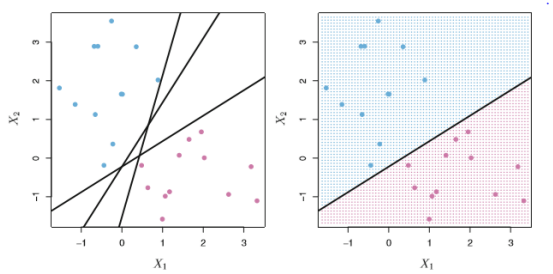


FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision 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.*

- Assuming that the binary outcome variable is code

$$y_i = \begin{cases} 1 & \text{for the first class} \\ -1 & \text{for the second class} \end{cases}$$

- Then a decision rule in terms of the hyperspace can be formulated as
  - $\beta_0 + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2} > 0$ for $y_i = 1$
  - $\beta_0 + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2} < 0$ for $y_i = -1$, or equivalently combining both rules
  - $y_i \cdot (\beta_0 + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2}) > 0$.
- Give a set of feature values $(x_{i1}, x_{i2})^T$ for the i^{th} observation the decision rule is evaluated and dependent of the sign of the hyperplane equations. Consequently, an object i is assigned to either the "negative" or "positive" class.

Maximal Margin Classifier

- The objective is to find the best hyperplane which is the furthest distance away from the target outcomes in the feature space.

- The maximal margin classifier which tries to *maximize smallest object distance* in both classes to the hyperplane
- This establishes a cushion or *margin* M around the hyperplane free of any objects from either class.

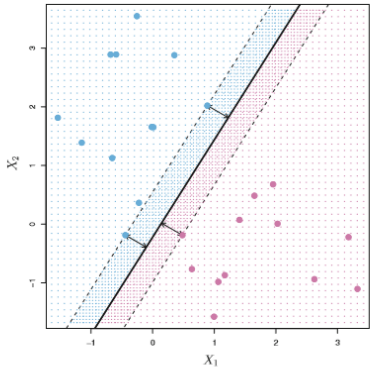


FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the hyperplane is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.*

- In general, just $p+1$ feature *support vectors* determine the $p-1$ dimensional maximal margin hyperplane.
  This is a problem: Sampling variability may shift the small set of $p+1$ *support vectors*. Consequently, the established hyperplane experiences a high dree of variability.
- Technically the maximal margin classifier is the solution of an optimization problem under constraints

$$\max_{\beta_0,\beta_1,\dots,\beta_p,M} M \quad \text{subject to} \quad \boxed{\sum_{j=1}^{p} \beta_j^2} = 1 \text{ with}$$

Quadratic Optimization

$$y_i \cdot (\beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_p \cdot x_{ip}) \geq M \quad \forall\, i = 1,2,\dots,n$$

why?

why?

- The equality constraint $\sum_{j=1}^{p} \beta_j^2 = 1$ just ensures that the *perpendicular distance* of any object i to the hyperplane is given by $y_i \cdot (\beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_p \cdot x_{ip})$.
- If the class are not perfectly separable then no solution for a positive margin $M > 0$ of the optimization problem exisits.

Support Vector Classifier

- Support vector classifier overcome the problem of perfectly separable classes.
- Additionally, it allows incorporating more observations into the determination of the decision rule, which makes it more robust against sample variations.
- It provides a trade-off of potentially misclassifying a few observations against the benefit of classifying the remaining observations well by introducing a *soft margin*:
  - Objects that are on the correct side of the hyperplane but within the margin.
  - Objects which are on the wrong side of the hyperplane and therefore misclassified.

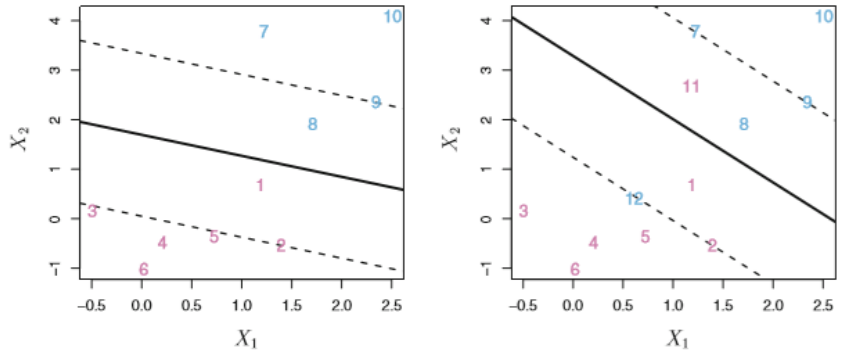


FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation* 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.*

- The optimization problem now becomes:

$$\max_{\beta_0,\beta_1,\dots,\beta_p,\epsilon_1,\dots,\epsilon_n,M} M \text{ subject to } \sum_{j=1}^{p} \beta_j^2 = 1 \text{ with}$$

$$y_i \cdot (\beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_p \cdot x_{ip}) \geq M \cdot (1 - \epsilon_i) \;\; \forall\, i = 1,2,\dots,n$$

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

- The *slack variable* ϵ_i determines whether an object i is

$$\begin{cases} \epsilon_i = 0 & \text{correctly classified not inside the margin } M \\ 0 < \epsilon_i < 1 & \text{correctly classified but inside the margin } M \\ 1 \leq \epsilon_i & \text{incorrectly classified} \end{cases}$$

- The hyper-parameter C denotes a budget constraint allowing a specific degree of inside the margin and incorrectly classified.
  The larger C the more tolerant the model becomes by allowing violations.
- Only observations inside the margin or wrongly classified determine the specification of the hyperplane.
- The hyper-parameter C controls the *bias-variance* trade-off:
  - large C incorporates more objects and thus reduces the variance
  - but may increase the bias.

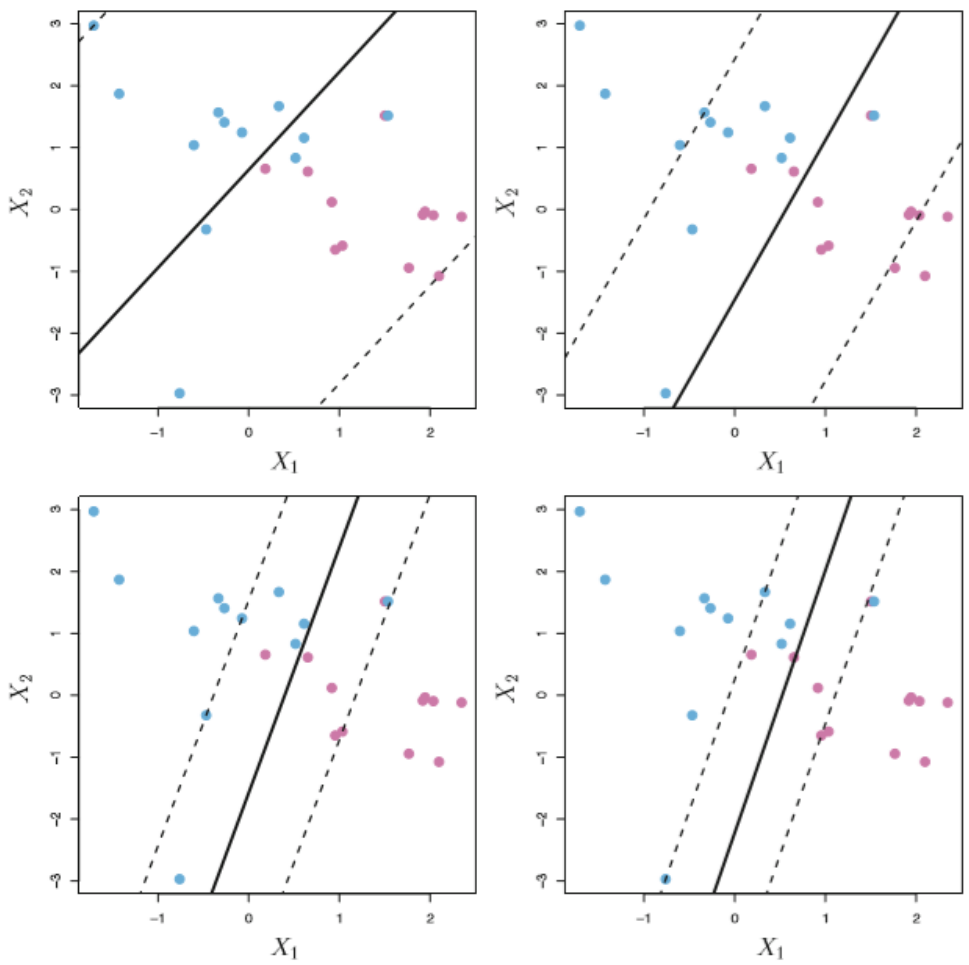


FIGURE 9.7. *A support vector classifier was fit using four different values of the tuning parameter C in (9.12)–(9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance for observations being on the wrong side of the margin, and so the margin will be large. As C decreases, the tolerance for observations being on the wrong side of the margin decreases, and the margin narrows.*

Support Vector Machines

- Support Vector Machines have the capability to automatically deal with non-linear decision boundaries.

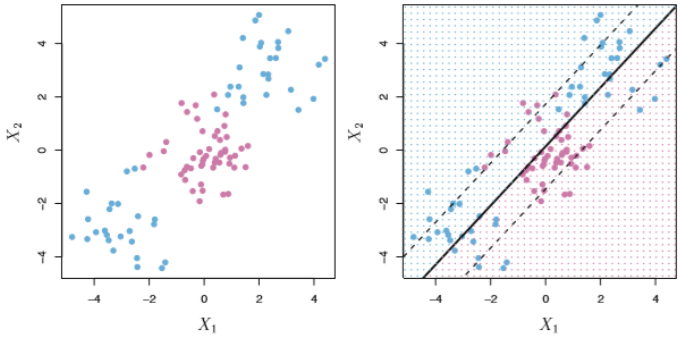
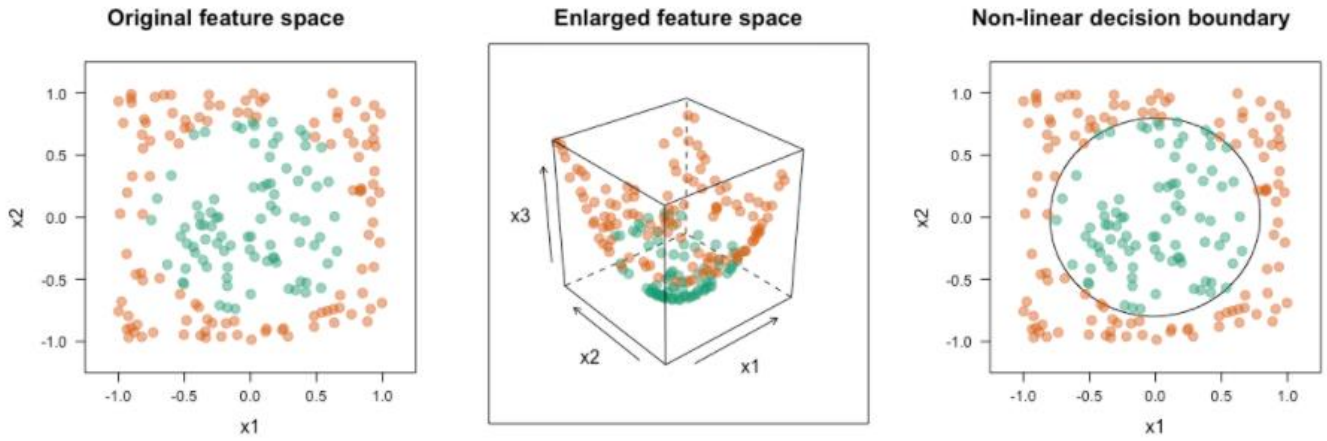


FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

- An approach of dealing with this non-linear boundary is by augmenting the feature space with polynomial terms of the original features.



Here the polynomial augmented feature variable $X_3 = X_1^2 + X_2^2$ separates both target classes.

- In this augmented feature space, the decision boundary remains linear.
- An alternative approach is to use a kernel density predictor based on a feature vector $\mathbf{x}$ to the support points $\mathbf{x}_i$.
- One can show that the linear support vector classifier can be expressed as

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

  with the inner product $\langle \mathbf{x}, \mathbf{x}_i \rangle = \sum_{j=1}^{p} x_j \cdot x_{ij}$.
- To estimate the unknown coefficients $\beta_0, \alpha_1, \ldots, \alpha_n$ all $n \cdot (n-1)/2$ pairs of the inner products among the training set feature vectors $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ need to be evaluated.
- Only those coefficients $\alpha_1, \ldots, \alpha_n$ associated with support vector observations (within the soft margin or wrongly classified) are non-zero. Therefore, one can rewrite

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

- Generalizations of the inner product $\langle \mathbf{x}, \mathbf{x}_i \rangle$ are kernel functions, which can be used instead:

$$K(\mathbf{x}, \mathbf{x}_i) = \begin{cases} \sum_{j=1}^{p} x_j \cdot x_{ij} & \text{correlation kernel} \\ \left(1 + \sum_{j=1}^{p} x_j \cdot x_{ij}\right)^d & \text{polynomial kernel} \\ \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_j)^2\right) & \text{radial kernel (similar to the Gaussian density)} \end{cases}$$

linear classifier

- The unknown hyper-parameters d and γ must be estimated via cross-evaluation from the training data.
- For the radial kernel $K(\mathbf{x}^*, \mathbf{x}_i)$ if an object $\mathbf{x}^*$ is far away from the training objects $\mathbf{x}_i$ its kernel will be very small, thus it will not contribute to the decision function $f(\mathbf{x}^*)$.

- The advantage of using kernels for an augmented feature space are mainly computational and numerical stability.

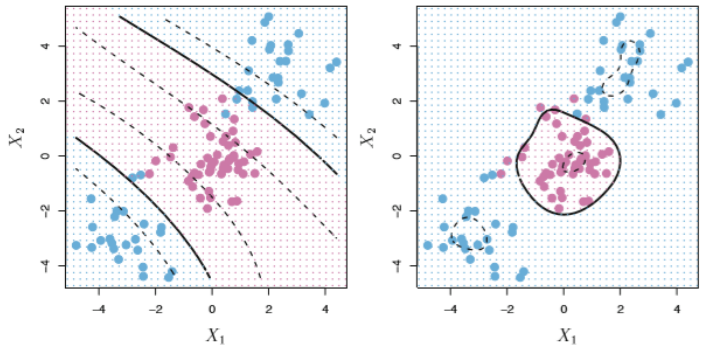
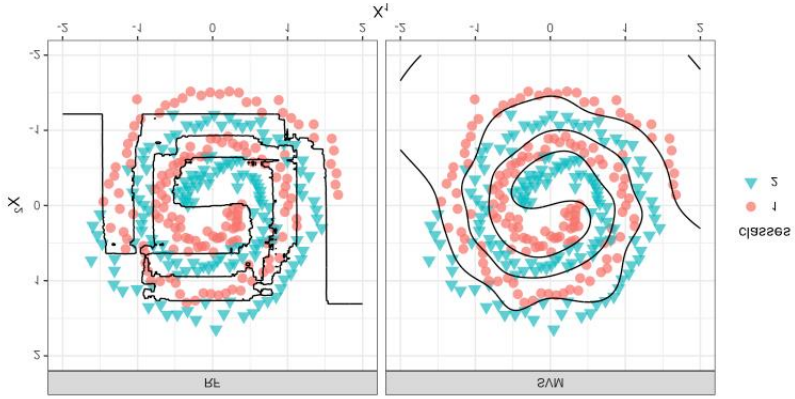


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

- Increasing the feature space runs the risk of building an overfitting model which will not perform well for the test dataset.
- Example: Comparison of the random forest algorithm against the radial basis kernel of a SVM for the spiral dataset.



but SVM is more restricted(metric variables).

SVM with more than two classes

- ***One-versus-one*** approach calibrates all $K \cdot (K-1)/2$ pairwise classifiers and the assigns the prediction of an object $\mathbf{x}$ to that class, which won most frequently among the $K \cdot (K-1)/2$ predictions.
- The ***one-versus-all*** approach calibrates K classifiers with one class against the remaining classes. The object $\mathbf{x}$ is assigned to that class for which the distance $|\beta_0 + \beta_1 \cdot X_1 + \cdots + \beta_P \cdot X_P|$ from the hyperplane is the largest.