NYC DATA SCIENCE ACADEMY

Support Vector Machines

Data Science with R: Machine Learning

Outline

PART 1

Maximal Margin Classifier

Hyperplanes

- In order to begin talking about support vector machine classifier, we must first understand the idea of a separating surface.

- In general, a hyperplane is a subspace of one dimension less than its ambient space:
  - In 2-dimensional space, hyperplanes are 1-dimensional lines.
  - In 3-dimensional space, hyperplanes are 2-dimensional planes.
  - In p-dimensional space, hyperplanes are (p - 1)-dimensional objects.

- Hyperplanes are said to be flat and affine:
  - They preserve parallel relationships.
  - They need not pass through the origin.

Hyperplanes: Mathematically

- The equation of a hyperplane has the following form:

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

- For any point vector X in the space, there are two possibilities:
  - X satisfies the equation above and thus itself falls on the hyperplane.
  - X does not satisfy the equation above and falls on one side of the hyperplane.

- If X does not satisfy the equation, then one of the following must be true:

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

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

Hyperplanes: Mathematically

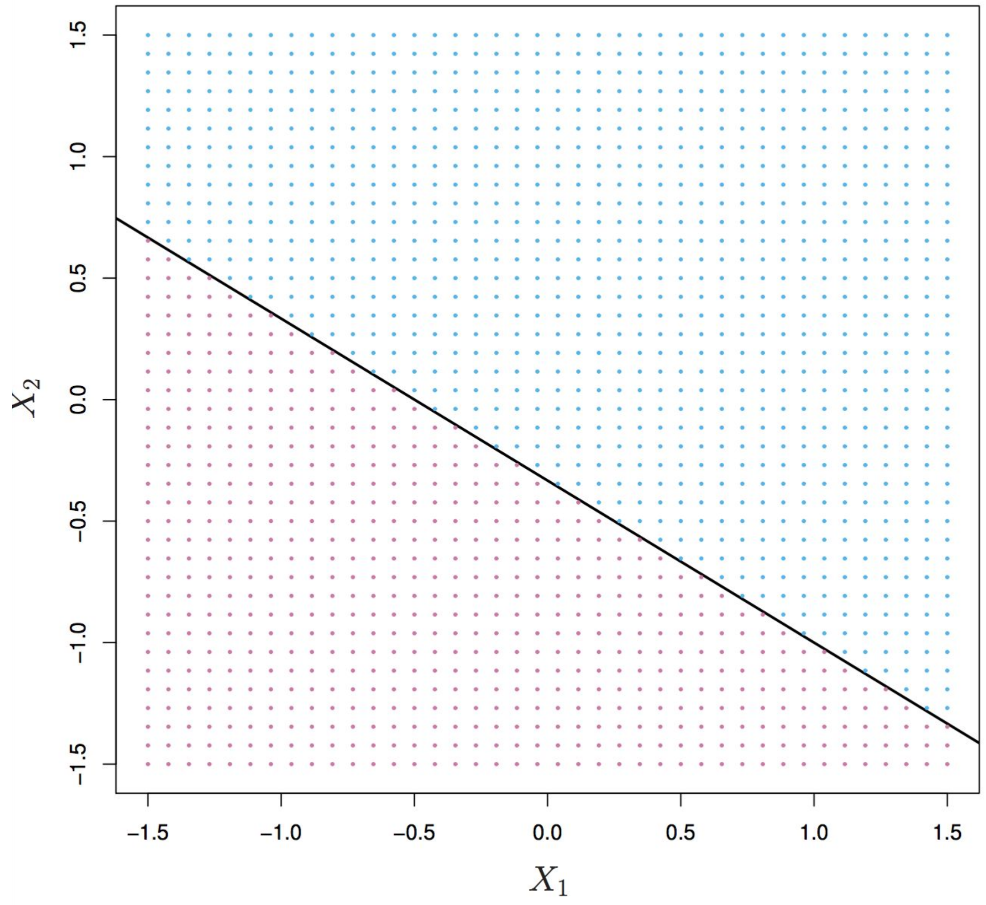
- Extracting the β coefficients from this equation (not including the intercept) yields what is called a normal vector:
  - This vector points in a direction orthogonal to the surface of the hyperplane and essentially defines its orientation.

- For any given point in the feature space, we can project onto the normal vector of the hyperplane.
  - Based on the sign of the resulting value, we can determine on which side of the hyperplane the point falls.

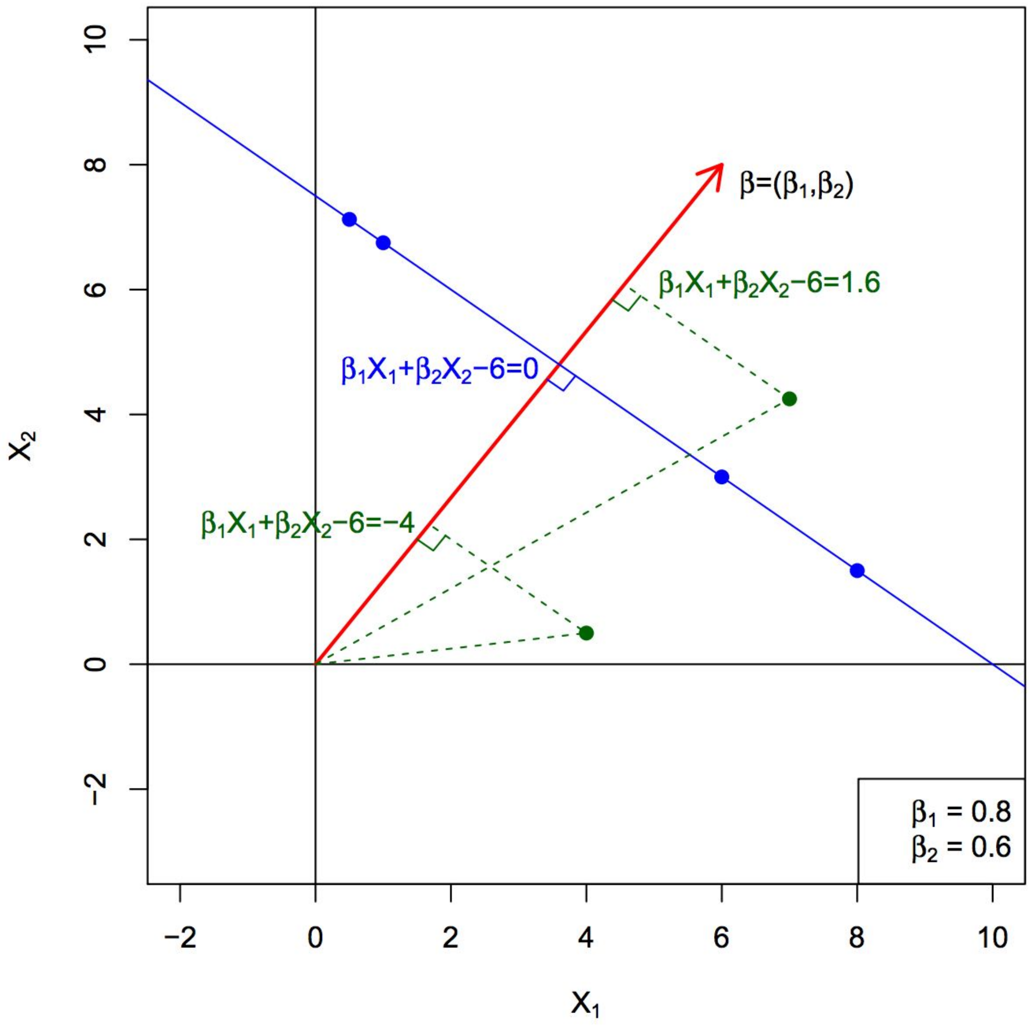
- When the normal vector is of unit length, the value of the hyperplane function defines the Euclidean distance from the point to the hyperplane.

Hyperplanes: Visually

- The following is the 2-dimensional hyperplane $1 + 2X_1 + 3X_2 = 0$:



Hyperplanes: Visually



Separating Hyperplanes: Mathematically

- Our main goal is to develop a classifier that will help us predict into which category a new observation will fall.
  - Suppose our observations fall into one of two classes which we can label as {-1, 1} without loss of generality.
  - Also suppose that it is possible to construct a hyperplane that perfectly separates the observations based on these class labels.
  - This hyperplane would then have the following properties for all $i = 1, \ldots, n$:

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

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

Separating Hyperplanes: Mathematically

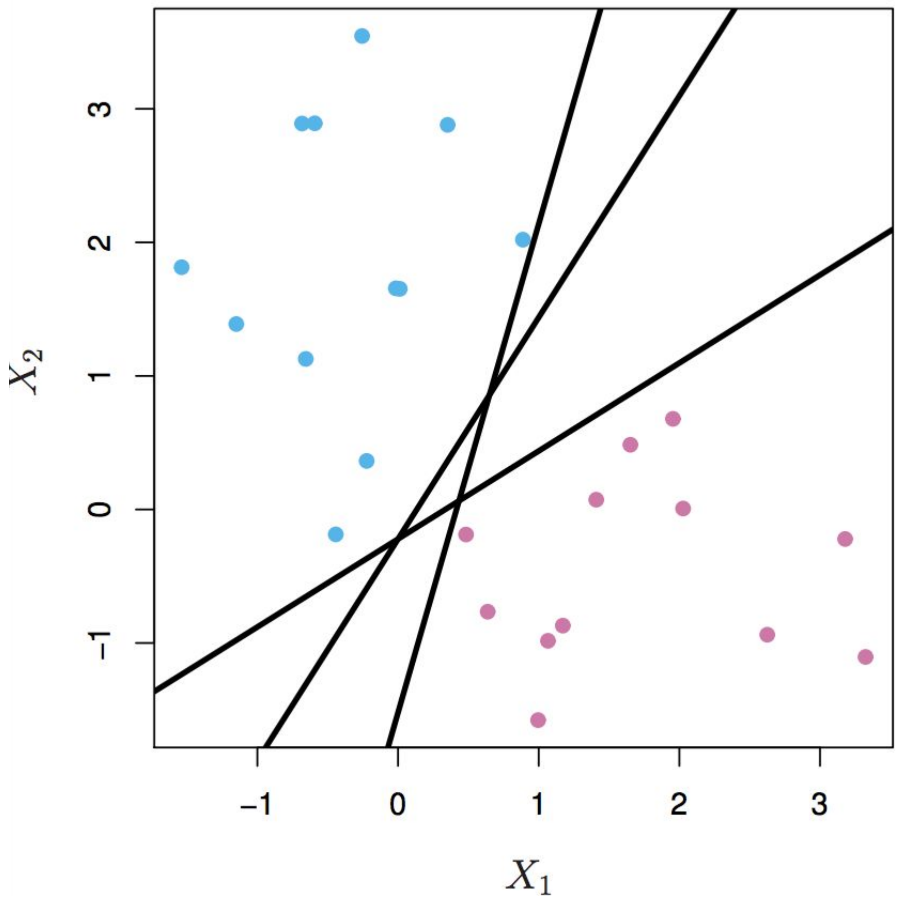
- Equivalently, the separating hyperplane would have the following property:

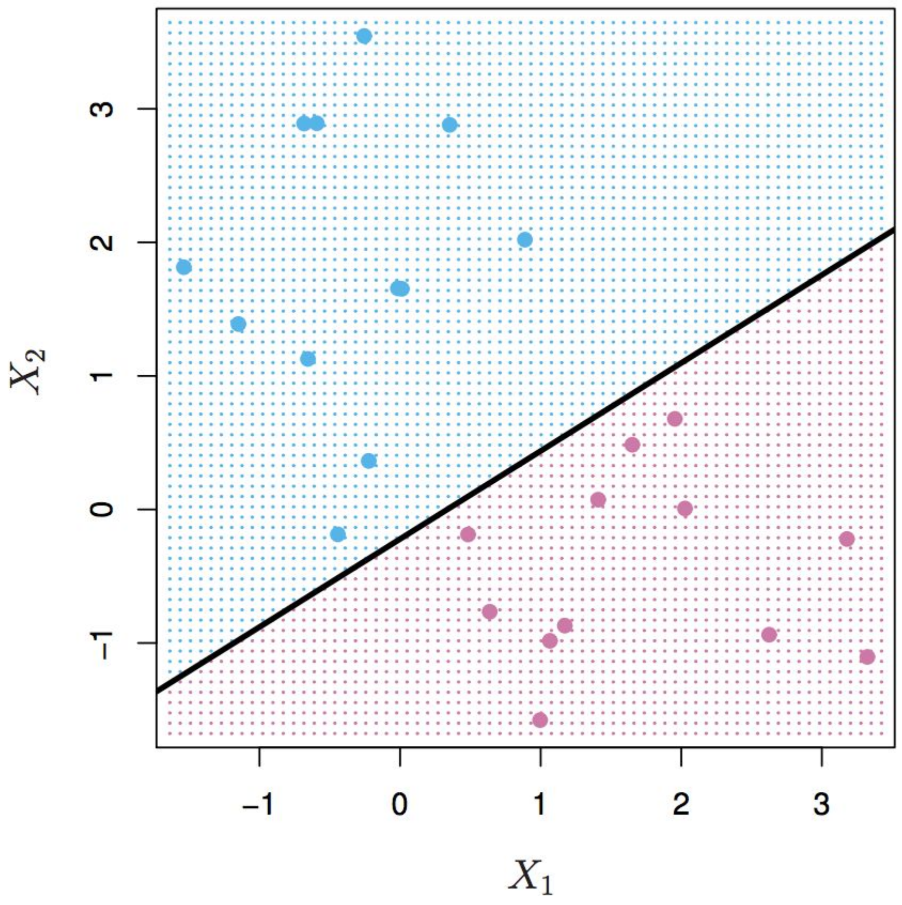
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0$$

- By simply evaluating the value of the hyperplane function given an observation, we can determine on which side of the hyperplane the observation falls. To classify this observation:
  - If the value is positive, classify the observation into group 1.
  - If the value is negative, classify the observation into group -1.

- The magnitude of the evaluation also yields information regarding the confidence of our classification prediction:
  - Large values imply the observation is far from the hyperplane (high conf.).
  - Small values imply the observation is close to the hyperplane (low conf.).

Separating Hyperplanes: Visually





The Maximal Margin Classifier

- If the data at hand can be separated perfectly with a hyperplane in the first place, then there are infinite separating hyperplanes in the feature space!
  - How do we determine which of these hyperplanes is the best?

- First compute the distance from each training observation to a separating hyperplane. Of these distances, the smallest distance is called the margin.

- Let’s try finding the maximal margin hyperplane, which:
  - Is the separating hyperplane that is farthest from the training observations.
  - Creates the biggest gap/margin between the two classes.

- Hopefully, if the maximal margin hyperplane has a large margin on the training data, it will also have a large margin on the test data.

The Maximal Margin Classifier: Mathematically

- The construction of the maximal margin classifier is the solution to the following optimization problem:

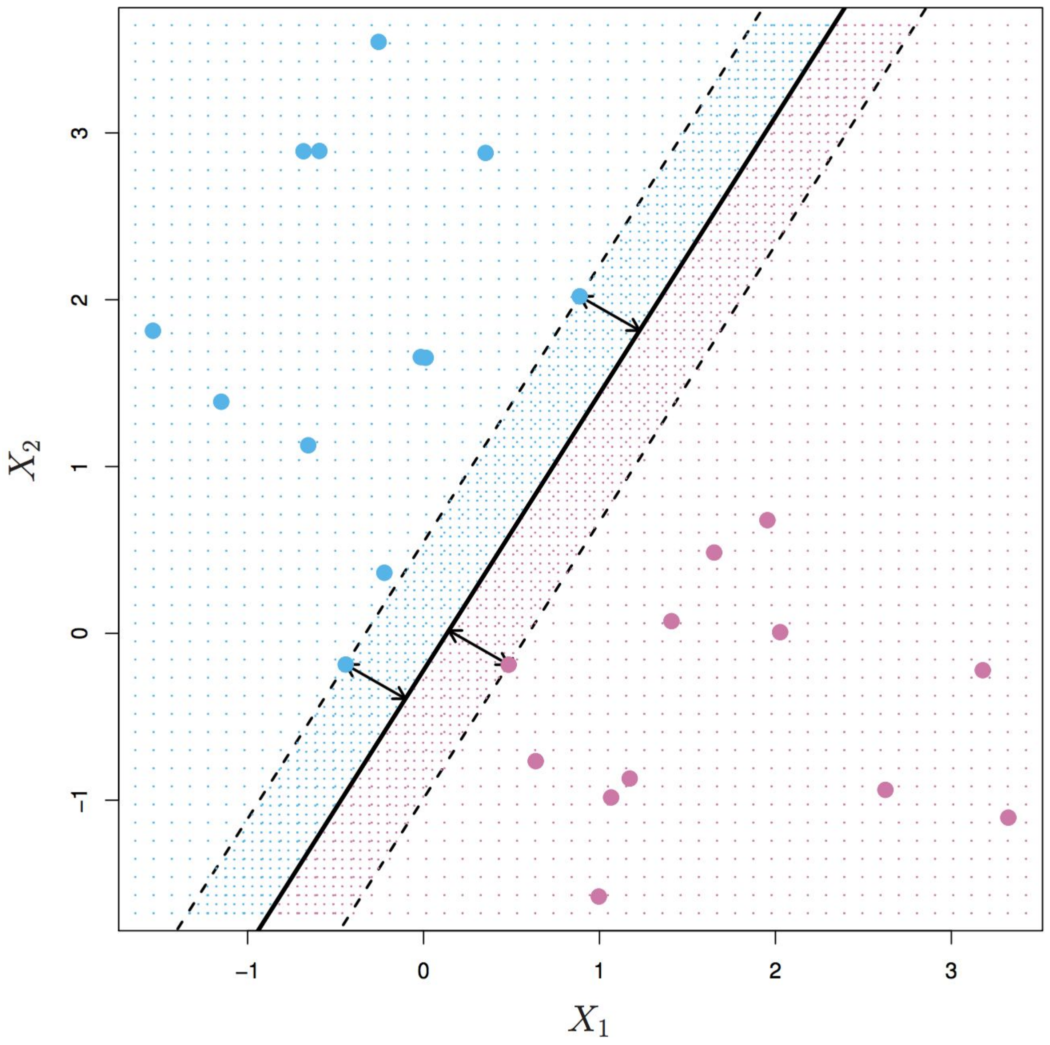
$$\max_{\beta_0,\beta_1,\ldots,\beta_p} \quad M$$

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

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

- Simply put:
  - ➢ Maximize the margin M.
  - ➢ Ensure the normal vector is of unit length (not actually a constraint! Why?).
  - ➢ Guarantee that each observation is on the correct side of the hyperplane.

The Maximal Margin Classifier: Visually



PART 2

Support Vector Classifier

Limitations of the Maximal Margin Classifier

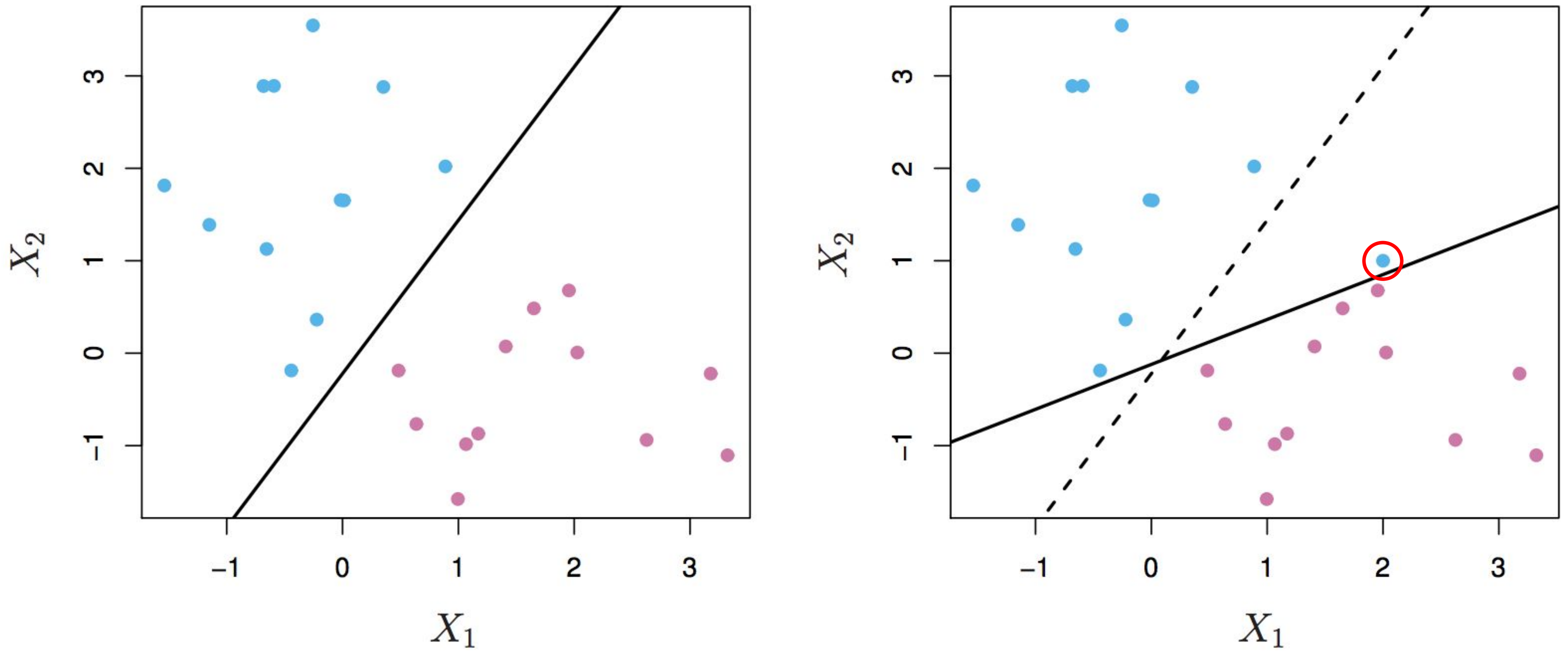
- The observations that fall closest to the separating hyperplane (equidistant) define the width of the margin. These observations are known as the support vectors because the hyperplane depends on their location.
  - If these observations were to move around in the feature space, the maximal margin hyperplane would also move.

- The maximal margin hyperplane directly depends only on the support vectors -- not the remaining observations!
  - The definition of the classifier can be very sensitive to outliers or a single change in the data.
  - High sensitivity to a small change suggests that we have overfit the classifier.

- What if no separating hyperplane exists?
  - There would be no solution to the optimization problem with $M > 0$.

Limitations of the Maximal Margin Classifier

- The maximal margin classifier can potentially be a poor solution if the data are noisy:



The Support Vector Classifier

- The support vector classifier is an extension of the maximal margin classifier that makes some compromises in an effort to improve upon the aforementioned limitations. The support vector classifier:
  - May not perfectly separate the classes.
  - Provides greater robustness to outliers and thus a lower sensitivity to individual observation shifts.
  - Helps better classify most of the training observations.

- By giving up the ability to have a perfect classifier on the training data we:
  - Take a penalty by possibly misclassifying some observations.
  - Do a better job classifying the remaining observations more confidently.
  - May have better predictive power for future observations.

The Soft Margin

- The main difference between the maximal margin classifier and the support vector classifier is that the latter implements what is called a soft margin.

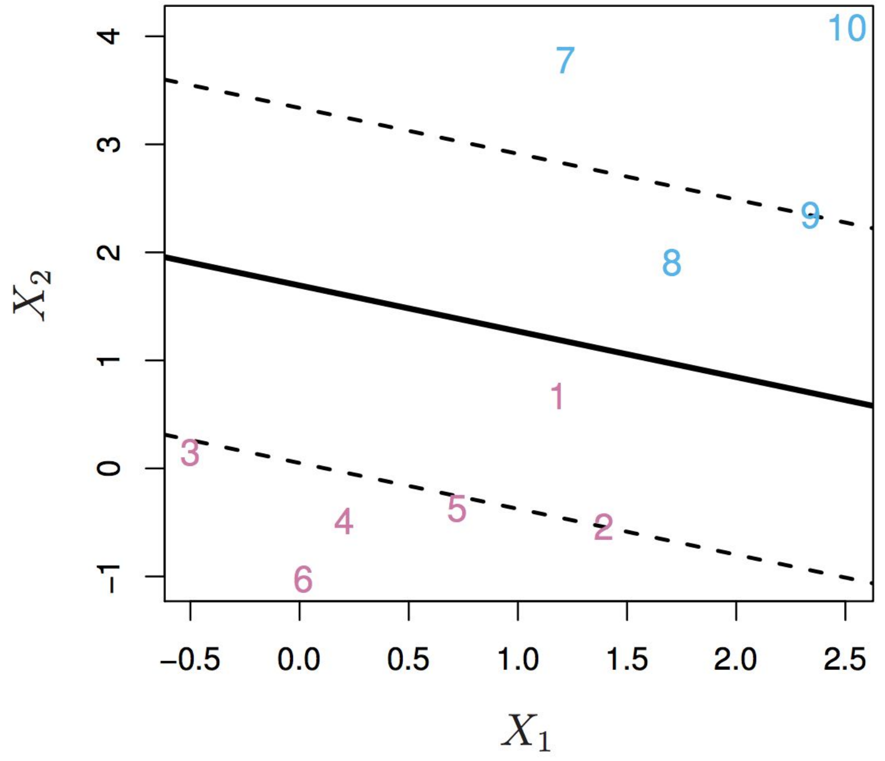
- The maximal margin classifier tries to identify the largest possible margin such that each observation is both:
  - On the correct side of the hyperplane.
  - On the correct side of the margin.

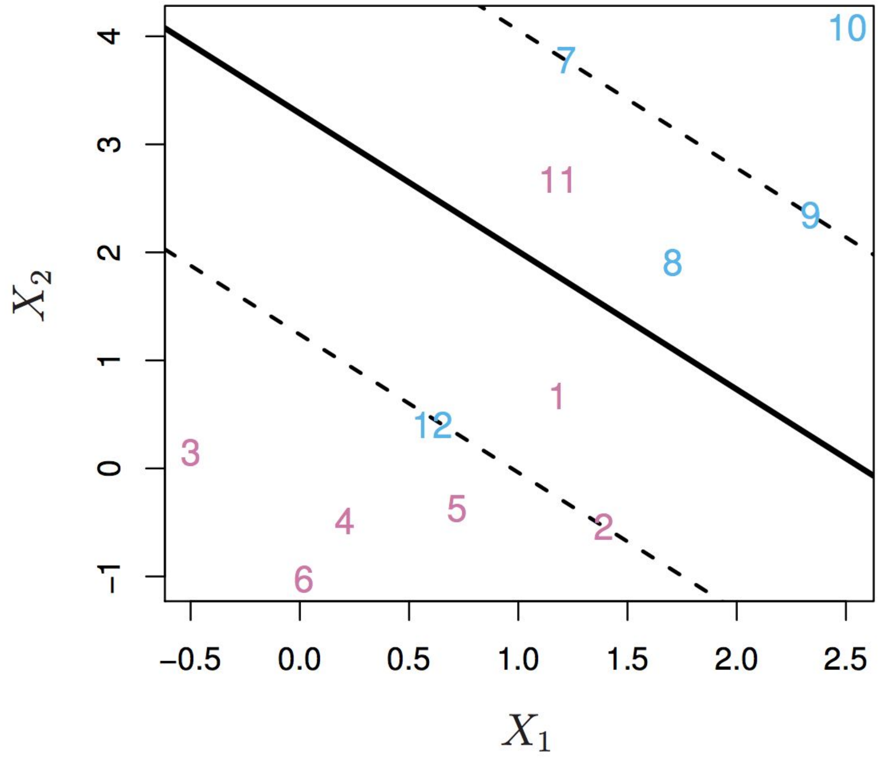
- The support vector classifier relaxes these constraints by allowing some observations to be on the incorrect side of either the margin or the hyperplane.

- In the case where the data is not separable, there would exist no maximal margin classifier, but there would exist a support vector classifier.

The Soft Margin: Visually

- The soft margin allows some observations to fall on the wrong side of the margin or the hyperplane itself:





The Support Vector Classifier: Mathematically

- The construction of the support vector classifier is the solution to the following optimization problem:

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\epsilon_2,\ldots,\epsilon_n} M$$

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

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

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

- We still desire to:
  - Maximize the margin M.
  - Ensure the normal vector is of unit length.

The Support Vector Classifier: Mathematically

- What are these new ϵ terms?
  - These terms are called slack variables.
  - They allow individual observations to potentially fall on the wrong side of the margin or hyperplane.

- The slack variables tell us where the i^{th} observation is located relative to the margin and hyperplane:
  - If $\epsilon_i = 0$, then the i^{th} observation is on the correct side of both the margin and the hyperplane.
  - If $\epsilon_i > 0$, then the i^{th} observation violates the margin.
  - If $\epsilon_i > 1$, then the i^{th} observation violates the hyperplane.

- The magnitude of the slack variables is proportional to the distance from each observation to the margin.

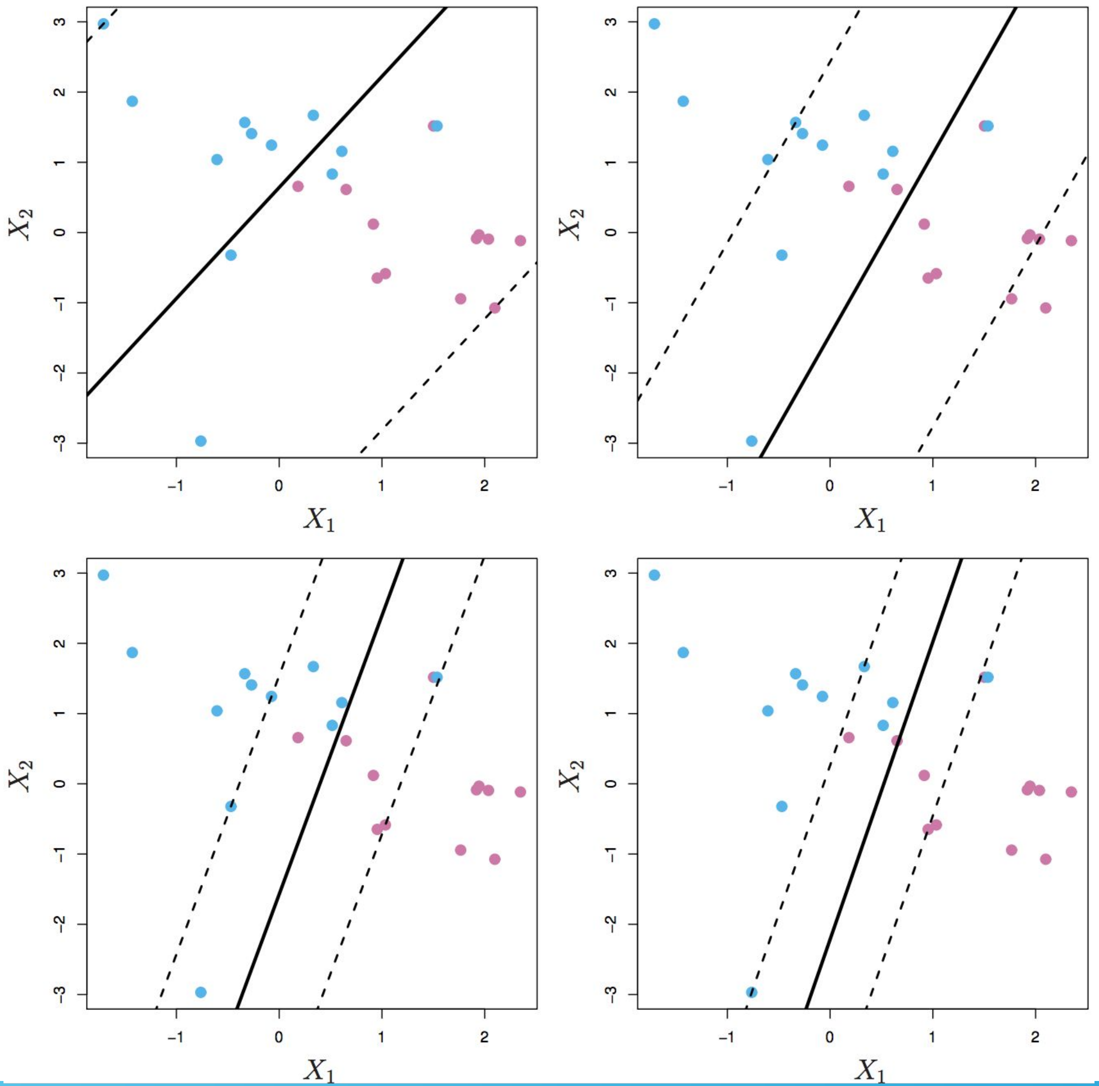
The Support Vector Classifier: Mathematically

- What is the new C term?
  - C is a tuning parameter that helps determine the threshold of tolerable violations to the margin and hyperplane.

- C is often thought of as a budget for the slack variables:
  - If $C = 0$, then there is no budget for the slack variables. For every i, $\epsilon_i = 0$, and the problem reduces to the maximal margin classifier.
  - As C increases, there is more budget for violations; the classifier becomes more tolerant so the margin will widen (low variance, high bias).
  - As C decreases, there is less budget for violations; the classifier becomes less tolerant so the margin will narrow (high variance, low bias).

- Why is it the case that no more than C observations can be on the wrong side of the hyperplane?

The Support Vector Classifier: Visually



Image

The Support Vector Classifier

- Similar to the maximal margin classifier, not all observations in the feature space directly affect the orientation of the hyperplane.
  - Only observations that either fall on the margin or violate the margin affect the solution to the optimization problem.
  - These observations are again called the support vectors.

- Observations that fall on the correct side of the margin have no direct bearing on the ultimate classifier.
  - If these observations were shifted around in the feature space, the hyperplane would remain unchanged (as long as they did not end up crossing over the margin).

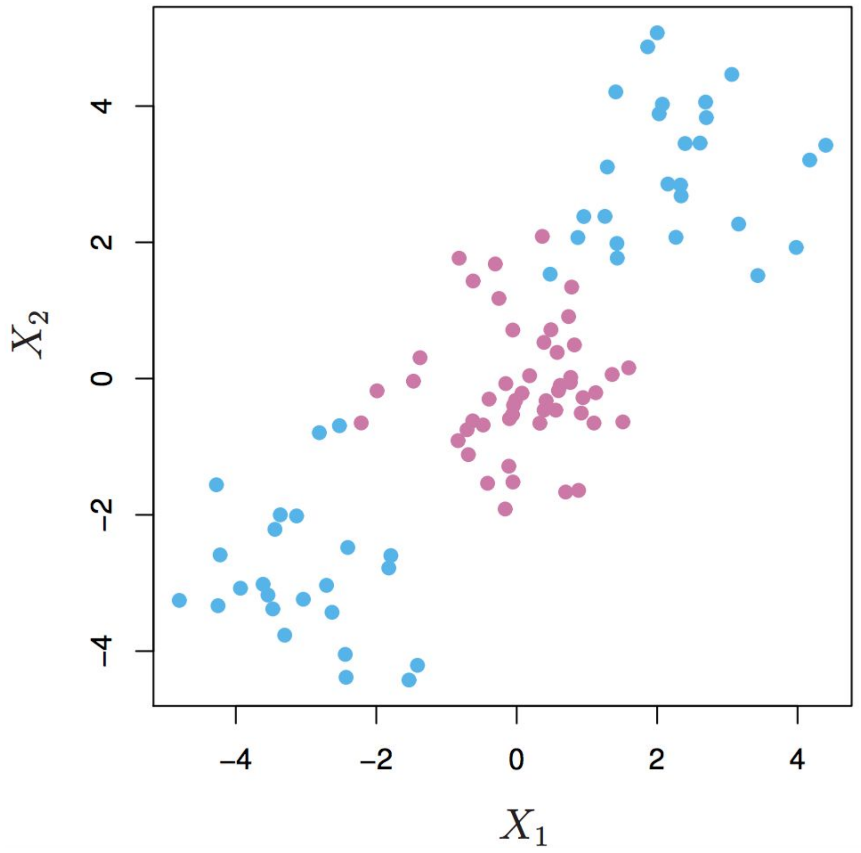
- Ultimately, the support vector classifier is more robust than the maximal margin classifier.

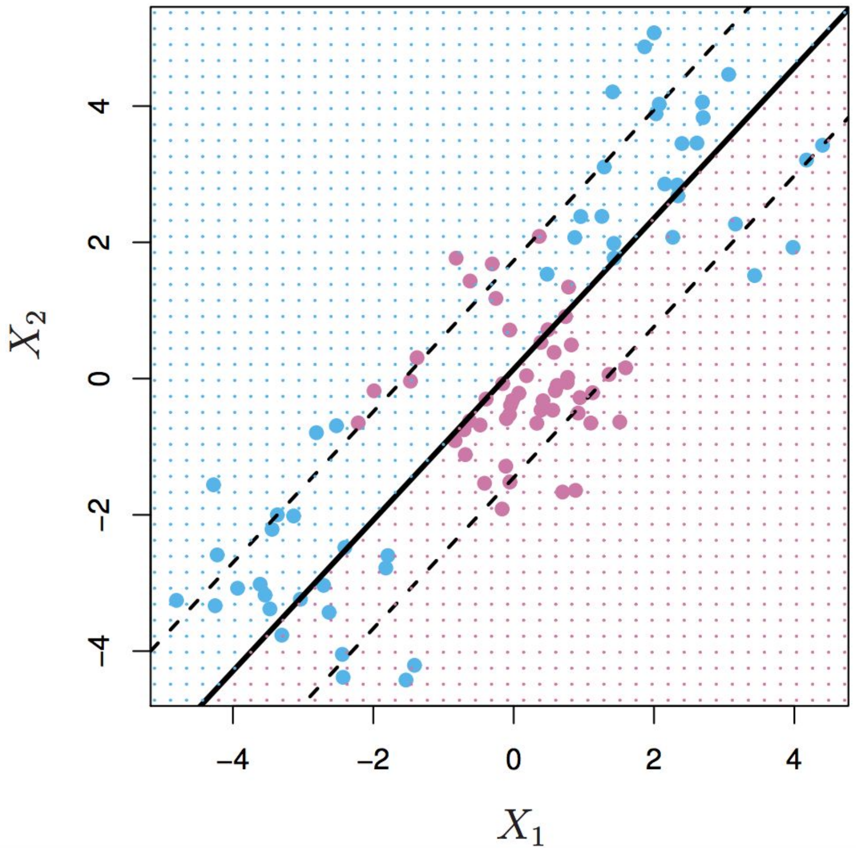
PART 3

Support Vector Machines

Limitations of the Support Vector Classifier

- The support vector classifier assumes that the boundary between classes is roughly linear; however, this process fails when the boundary is non-linear:





NYC DATA SCIENCE ACADEMY

Image

Feature Expansion

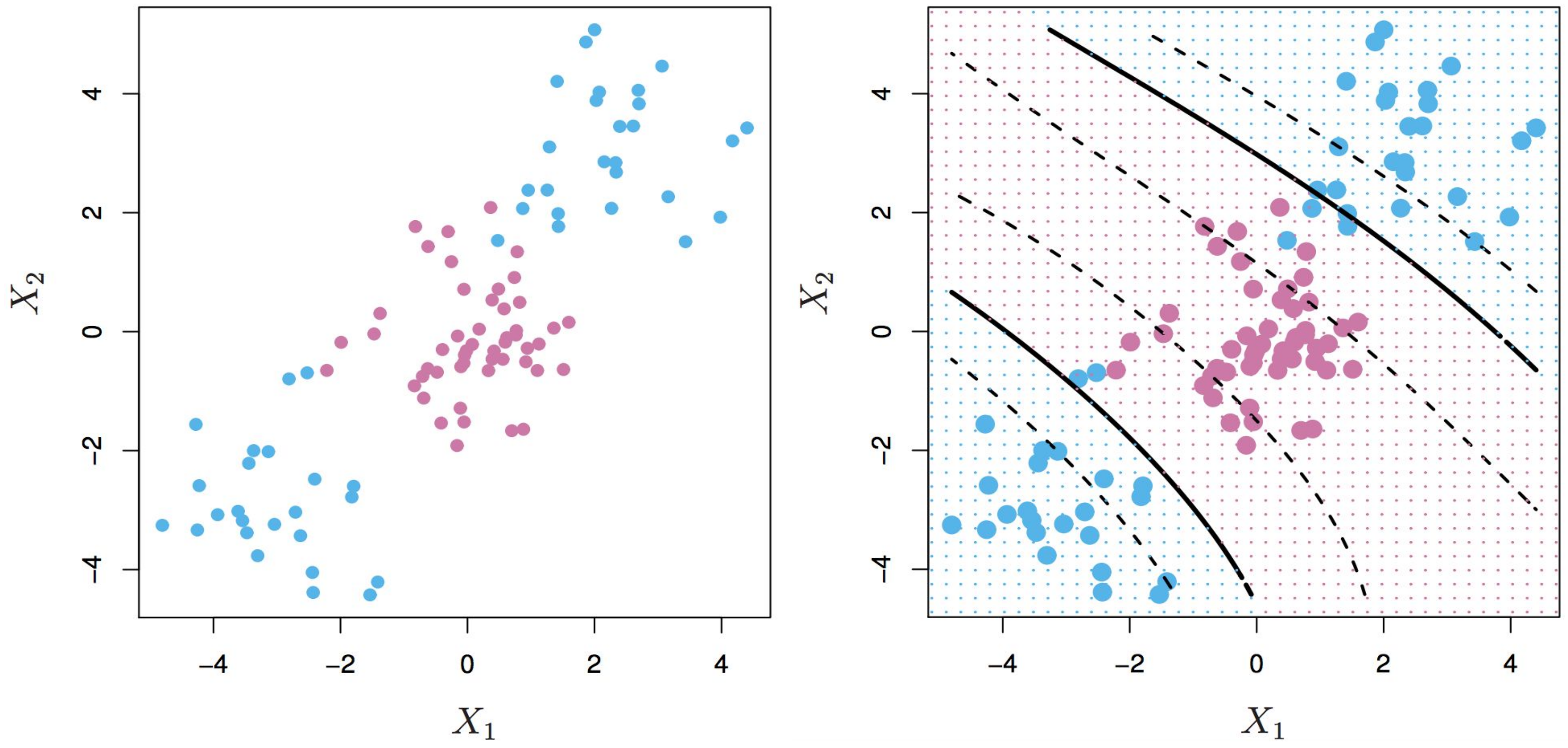
- Similar to adding polynomial terms in the linear regression setting, we can address the non-linearity by considering an enlargement of the feature space of our original dataset.
  - ➢ Implement functions of the predictors themselves by using higher-order polynomial functions of the predictors.

- By fitting a support vector classifier in the enlarged feature space, the decision boundaries become non-linear in the original feature space.
  - ➢ Suppose we only have X_1 and X_2 in our original dataset. Instead, use variables: X_1, X_2, X_1^2, X_2^2, and X_1X_2.
  - ➢ In the enlarged feature space, the following decision boundary is linear:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0$$

Feature Expansion

- Implementing feature expansion in the support vector classifier can help solve problems with data that are not linearly separable. How can we do this efficiently?



The Support Vector Machine

- The support vector machine is an extension of the support vector classifier that results from enlarging the feature space using kernels.
  - Using kernels is more efficient way of implementing feature expansion from a computational standpoint.

- The solution to the support vector classifier problem can be reduced in such a way that only involves the inner products of the observations instead of the actual observations themselves.
  - How can we do this?

The Support Vector Machine

- It can be shown that the linear support vector classifier can be represented as:

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

  - Here there are n parameters, one α per training observation.

- In order to estimate the α parameters and the intercept β_0, all that is needed are the inner products between all pairs of observations.

- To evaluate the function, we need to compute the inner product between the new observation and each of the training observations.

The Support Vector Machine

- Computationally, it turns out that the α parameters are only nonzero if the corresponding observation is itself a support vector!
  - If a training observation is not a support vector, then its corresponding α is necessarily 0; so most of the terms in the original equation disappear.

- Thus, if S is the collection of indices of the support vectors, we have:

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

- This formulation of the problem typically involves far fewer calculations than the original optimization described for support vector classifiers.

- How can we gain more flexibility with the support vector machine?

Kernels: Linear

- A kernel is a function that quantifies the similarity of two observations; just as we have seen there are many measures of similarity in terms of distance, so too there are many different types of kernels.

- We are already familiar with the idea of the inner product used to improve upon the calculations in the support vector classifier. Represented as a kernel:

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

- This is known as a linear kernel because the resulting support vector classifier is linear in the features.

Kernels: Polynomial

- An extension of the linear kernel is the polynomial kernel of degree d, defined as follows:

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

- Using a polynomial kernel with $d > 1$ is analogous to fitting a support vector classifier using feature expansion based on polynomials of degree d rather than the original feature space.
  - The decision boundary now appears to be much more flexible.

- When the support vector classifier is combined with a non-linear kernel, the resulting classifier is called a support vector machine.

Kernels: Radial

- The radial kernel is also another popular choice, defined as follows:

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

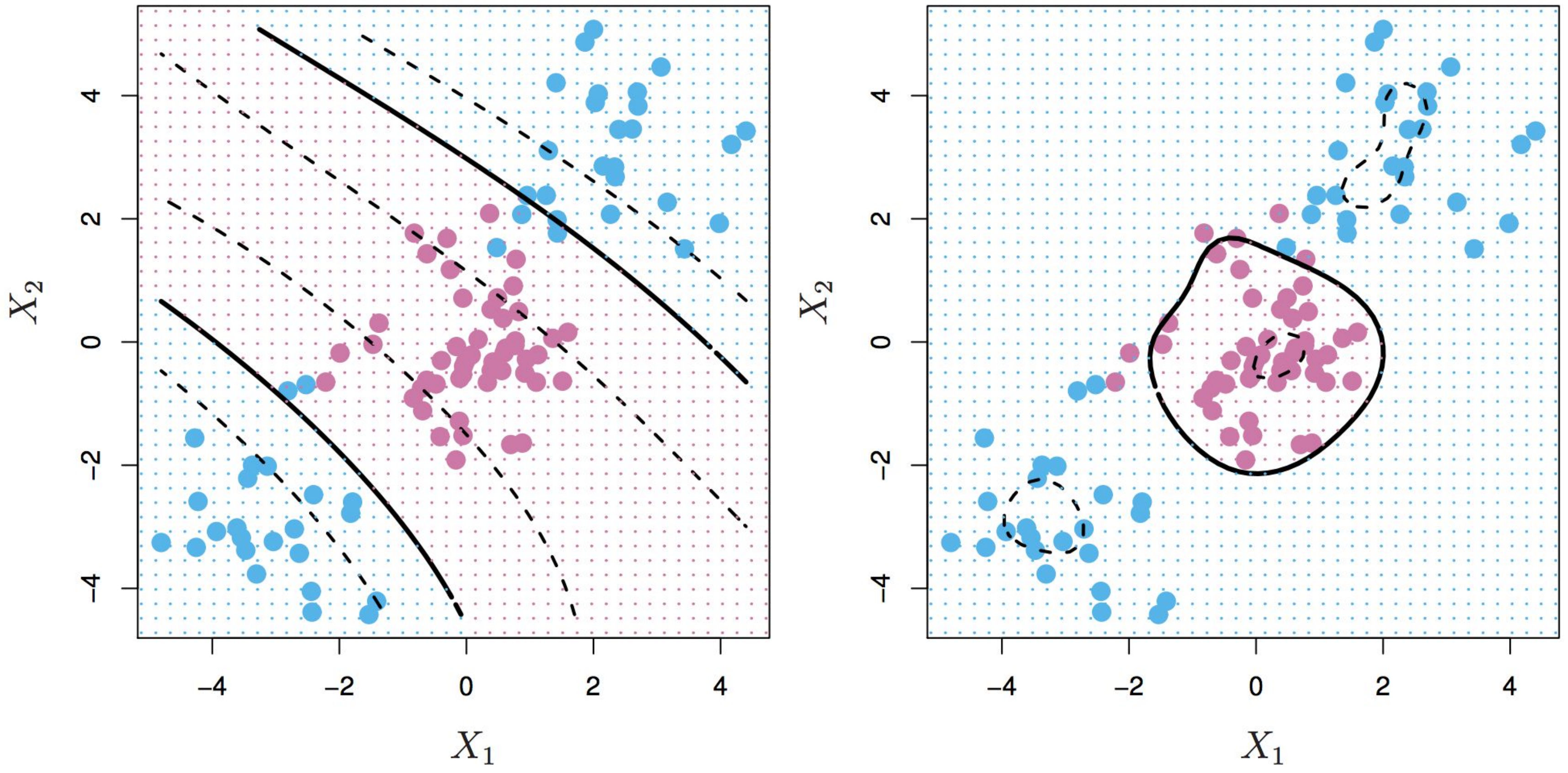
  - Here, the γ term is a positive constant and another tuning parameter.

- For the radial kernel, suppose we have a test observation.
  - If it is far from a training observation, the Euclidean distance will be large, but the value of the radial kernel will be small.
  - If it is close to a training observation, the Euclidean distance will be small, but the value of the radial kernel will be large.

- The radial kernel exhibits local behavior since only nearby training observations have an abundant effect on the class label of a test observation.

Kernels: Visually

- Polynomial and radial kernel solutions to the data exhibiting non-linear decision boundaries:



Kernels

- In order to implement any of the kernels, we simply define the kernel we would like to use in the support vector classifier as follows:

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

- Ultimately, using kernels is much more computationally efficient because we only need to compute the kernel for distinct pairs of observations in our dataset.
  - Furthermore, we need not actually work in the enlarged feature space (in the case of the radial kernel, we wouldn't be able to in the first place because the feature space is infinite!).

PART 4

Multi-Class SVMs

Multi-Class SVMs

- So far we’ve only considered implementing support vector machines in respect to two categories. Unfortunately, the SVM procedure does not directly lend itself to applications beyond binary classification; however, there are some ways around this limitation.

- 1-VS-1 Classification
  - Construct a support vector machine for each pair of categories.
  - For each classifier, record the prediction for each observation.
  - Have the classifiers vote on the prediction for each observation.

- 1-VS-All Classification
  - Construct a support vector machine for each individual category against all other categories combined.
  - Assign the observation to the classifier with the largest function value.

PART 5

Review

Review

- Part 1: Maximal Margin Classifier
  - Hyperplanes
    - Mathematically
    - Visually
  - Separating Hyperplanes
    - Mathematically
    - Visually
  - The Maximal Margin Classifier
    - Mathematically
    - Visually
- Part 2: Support Vector Classifier
  - Limitations of the Maximal Margin Classifier
  - The Support Vector Classifier
  - The Soft Margin
    - Visually
  - The Support Vector Classifier
    - Mathematically
    - Visually
- Part 3: Support Vector Machines
  - Limitations of the Support Vector Classifier
  - Feature Expansion
  - The Support Vector Machine
  - Kernels
    - Linear
    - Polynomial
    - Radial
    - Visually
- Part 4: Multi-Class SVMs