UNIVERSITY of SOUTH FLORIDA

Nonlinear Models

Spring 2020

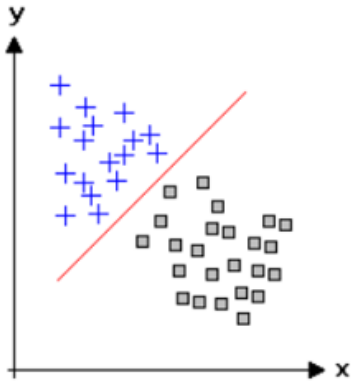
Instructor: Ankit Shah, Ph.D.

Types of Supervised Learning Methods

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} {x_1}^2 + \beta_{22} {x_2}^2$$

$$y = e^{\beta x_1}$$



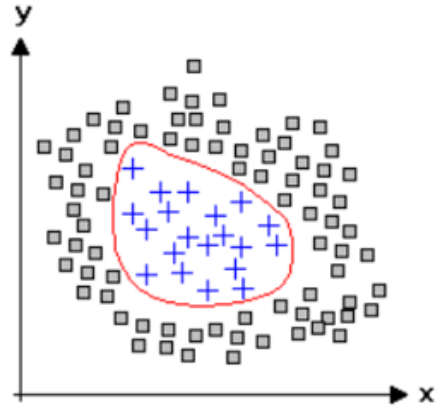
Supervised Learning

Linear

Linear Regression
Logistic Regression
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)

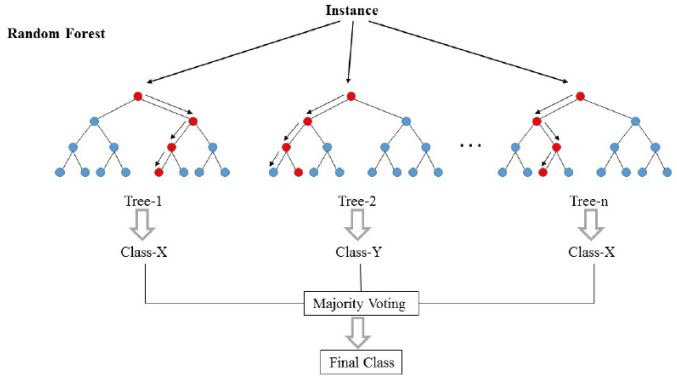
Nonlinear

Neural Networks
Support Vector Machine/Regression



Tree-based

Regression and Classification Trees
Random Forests

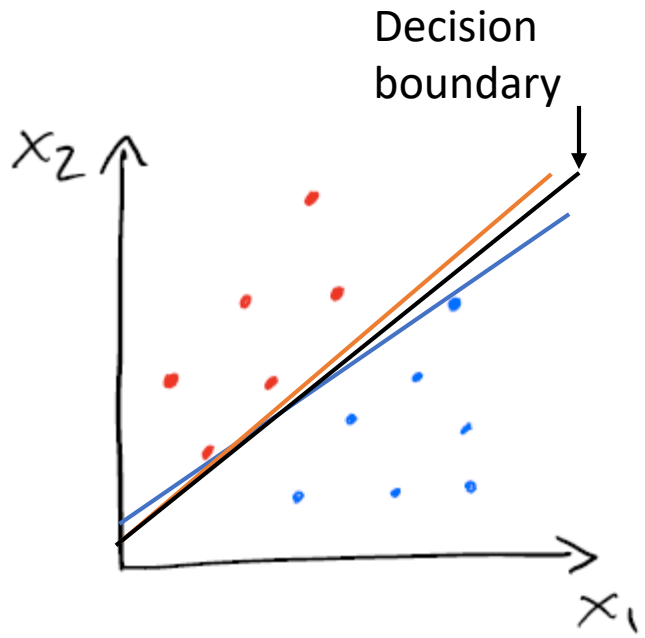


Linear models are linear in the parameters that need to be estimated, but not necessarily in the independent variables

Nonlinear Models

We are first going to motivate the need for a nonlinear model by discussing 2 classifiers that produce linear decision boundaries: 1) Maximal Margin Classifier and 2) Support Vector Classifier

Maximal Margin Classifier



- It is a simple method – more intuitive
- Used when classes can be separated with a linear boundary
- i.e., the classes can be separated with a (p-1)-dimensional hyperplane
- The decision boundary for the maximal margin classifier is the optimal separating hyperplane, which is the farthest away from any of the sample points in the training data

What is a hyperplane?

A hyperplane in a p-dimensional space can be defined by an equation of the form:

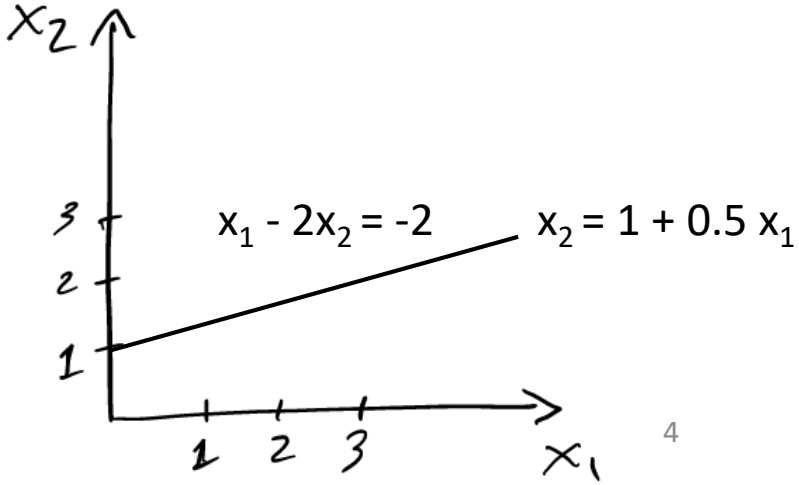
$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ \ldots + \beta_p x_p = 0$

Why is the above (p-1)-dimensional?

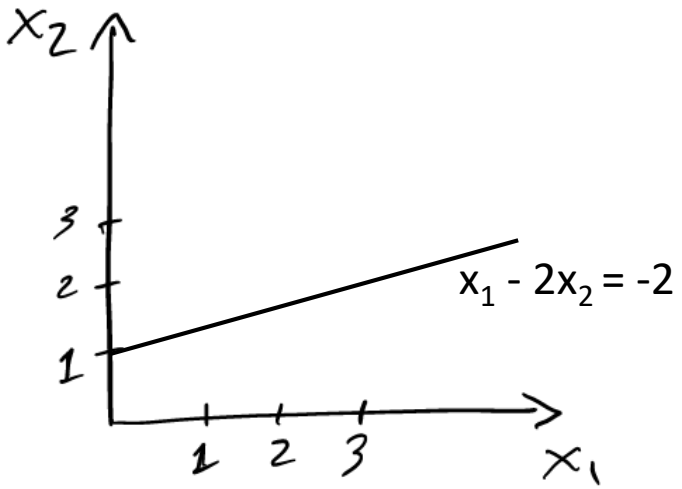
For 2 predictor variables:

$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$

$$x_2 = -\frac{\beta_0}{\beta_2} - \frac{\beta_1}{\beta_2} x_1$$



Maximal Margin Classifier



- Say, we have the following data points (x_1 , x_2) in the training data:
  (1,3), (3,1), (3,3), (1,1)
  Classify them in 2 different classes: -1 and +1
- *Hyperplane:* $\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$
  $2 + x_1 - 2x_2 = 0$
- (1,3)
- (3,1)
- (3,3)
- (1,1)

- If vector X satisfies $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p > 0$, then it belongs to one side of the p-dimensional space: y = 1
- If vector X satisfies $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p < 0$, then it belongs to another side of the p-dimensional space: y = -1
- Generalizing the above: a separating hyperplane has the following property
  $y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0$ for (i = 1, …, n)

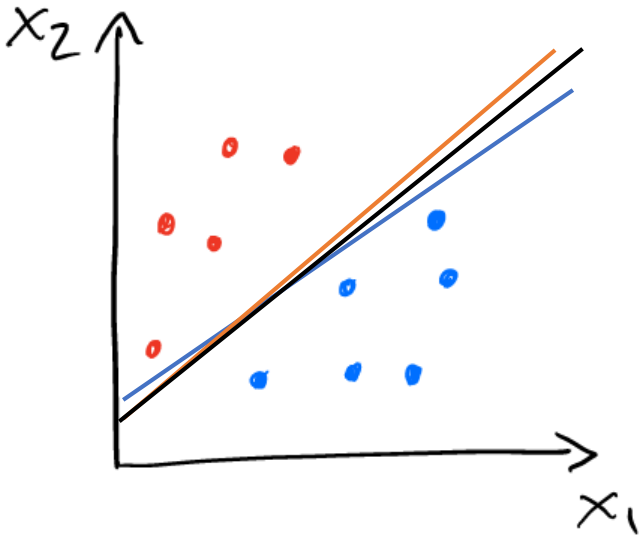
Classification using a Separating Hyperplane

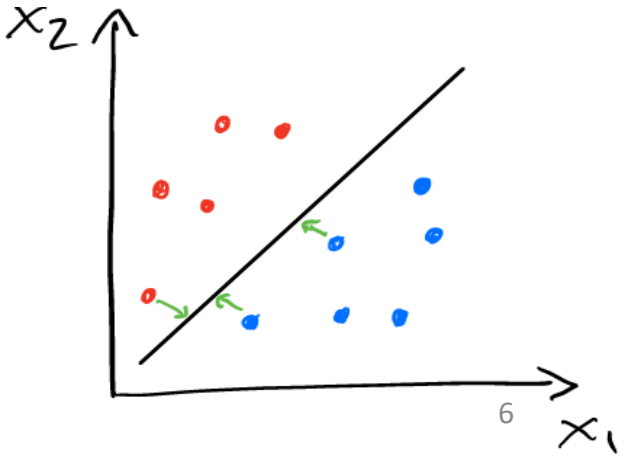
- If a separating hyperplane exists, then there will be an infinite collection of them
- We can adjust the hyperplane by a little and still separate the classes
- We multiply the β_j by a non-zero constant such that the equation still holds true: $y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0$
- Our objective is to construct a classifier that optimally separates the classes such that the separating hyperplane is the farthest distance from the training observations
  - We compute the perpendicular distance from each training data point to the given separating hyperplane
  - The smallest such distance is the minimal distance from the observations to the hyperplane – Margin
  - Our objective is to maximize the margin
- Mathematical Formulation:

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

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

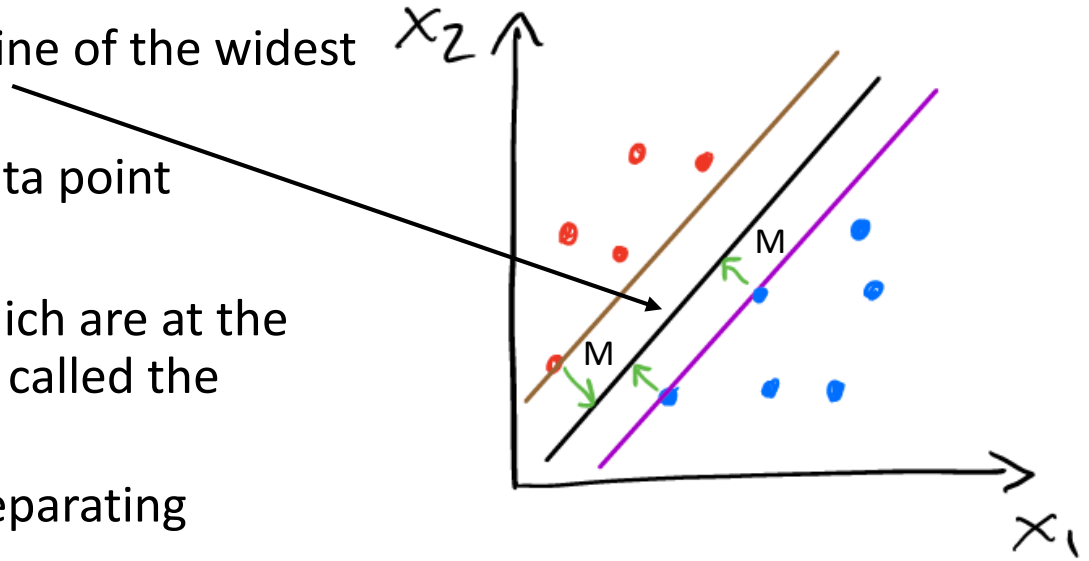
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M$$





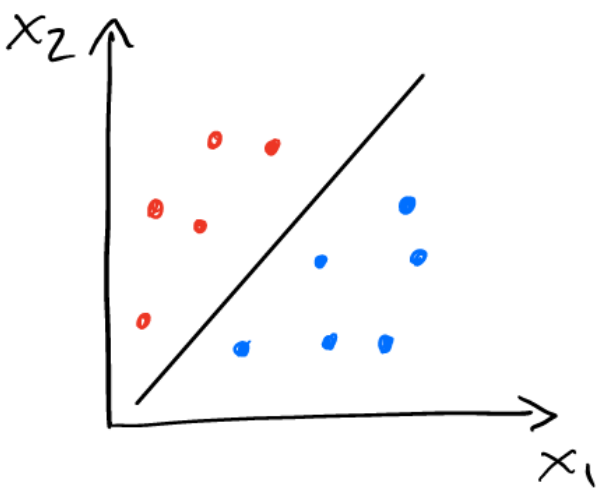
Classification using a Separating Hyperplane

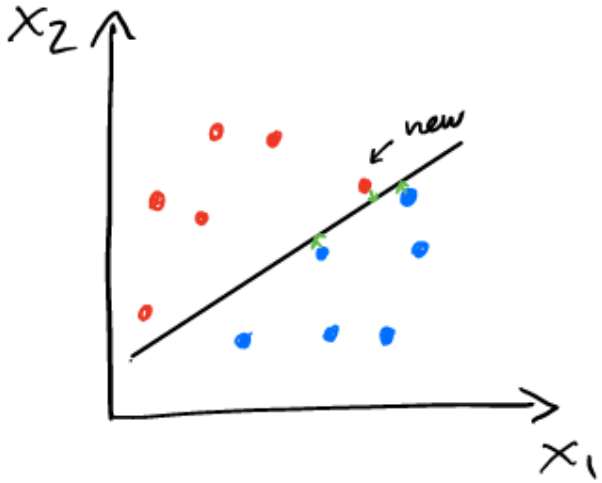
- Think of the maximal margin hyperplane as the mid-line of the widest "slab" that we can insert between the 2 classes
- The value M (smallest distance between a training data point and the hyperplane) is called the margin
- Observations (data points in the training data set) which are at the exact distance M from the separating hyperplane are called the support vectors
- Each data point is at least a distance of M from the separating hyperplane



- How many support vectors do we have in the example above?
- What is the dimensionality of the support vector?

Maximal Margin Classifier





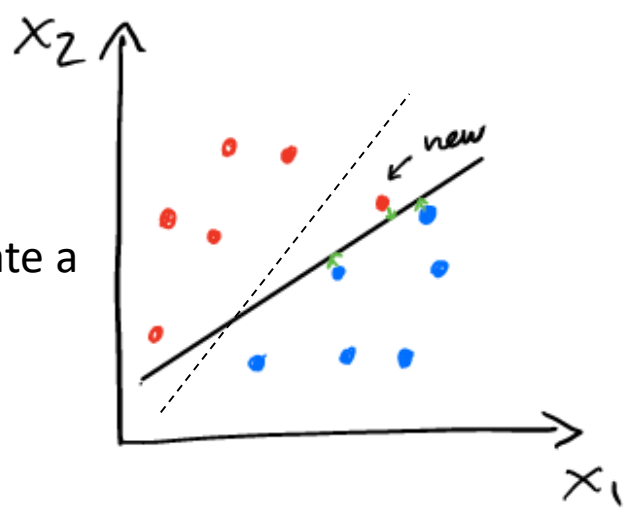
- A small change in the data set can result in a fairly large change in the decision boundary of the maximal margin classifier
- Such sensitivity suggests that the maximal margin classifier overfits the training data

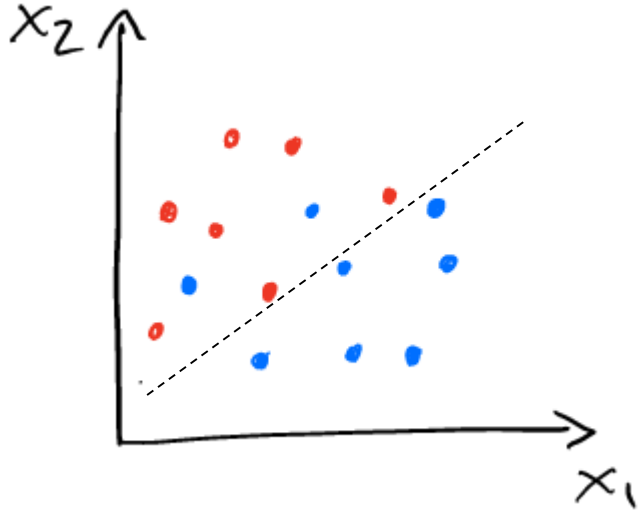
  When the model is tested on unseen data points – would the classification accuracy be lower/higher than that obtained on the training data points? Why?
- This motivates a need for a classifier that can be used to produce a more appealing decision boundary

Support Vector Classifier

- Support vector classifier is also called the “soft” margin classifier
- Used when the classes of the training data cannot be perfectly separated with a hyperplane and also when a more appealing decision boundary is needed in which separation is possible

We may have to misclassify a few data points in order to create a more robust model (unseen data points)

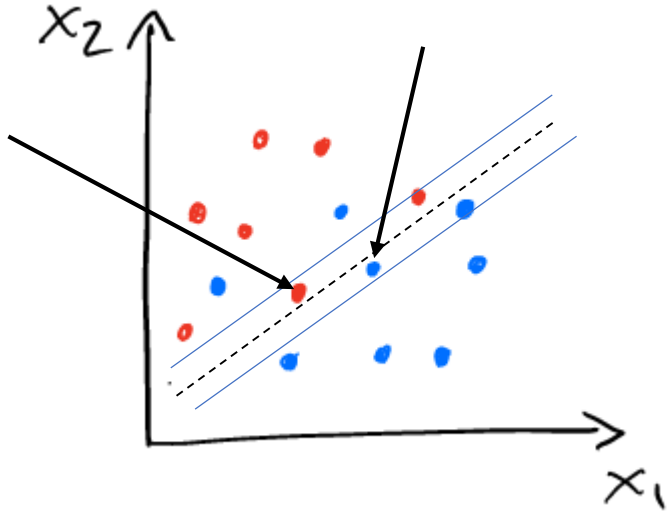




- In the above cases, we may consider a hyperplane that does not separate the 2 classes perfectly
- Our objective is to provide 1) robustness to the model and 2) better classification for majority of the data points

Support Vector Classifier

- As shown in the previous example, we allowed some data points to be on the incorrect side of the hyperplane and also be closer to the hyperplane on the correct side but with a value less than the margin
- Hence, it is a soft margin
  - i.e., it can be violated by some data points to obtain more robust models



- The support vector classifier has a (p-1)-dimensional hyperplane for the decision boundary and has a soft margin that could be violated by a few data points

What is the distance of the closest point to the decision boundary in support vector classifier?

Support Vector Classifier

- Mathematical Formulation:

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

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

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

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

Slack variable

Tuning parameter

Where will the data point lie, given the following conditions?

$\varepsilon_i = 0$

$0 < \varepsilon_i < 1$

$\varepsilon_i = 1$

$\varepsilon_i > 1$

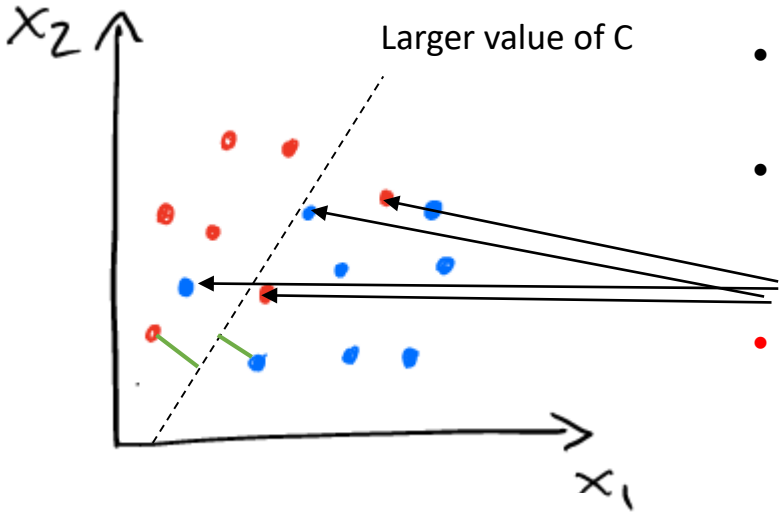
} Support vectors: All data points with $\varepsilon_i > 0$

- ε_i allows the data point *i* to be on the wrong side of the margin or the hyperplane
- C determines the number and severity of the violations to the margin and the hyperplane

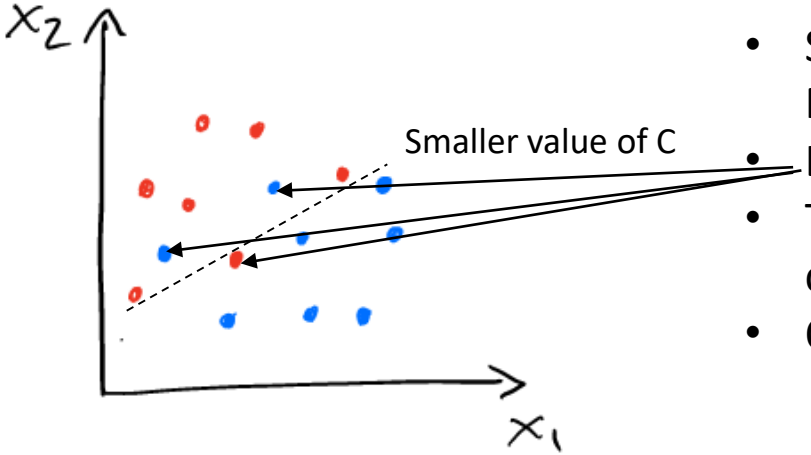
What happens when C = 0?

When C > 0, how many data points at most can be on the wrong side of the decision boundary?

Relationship of C with M



- Larger value of C -> larger value of M
- More slack variables come into play and hence more violators (support vectors)
- This is a condition with?
  Low variance – High bias
  High variance – Low bias



- Smaller value of C -> smaller value of M
- Less number of support vectors
- This is a condition where we are chasing the data points
- Overfit -> Low bias

How do we choose the value of C?
Cross-validation

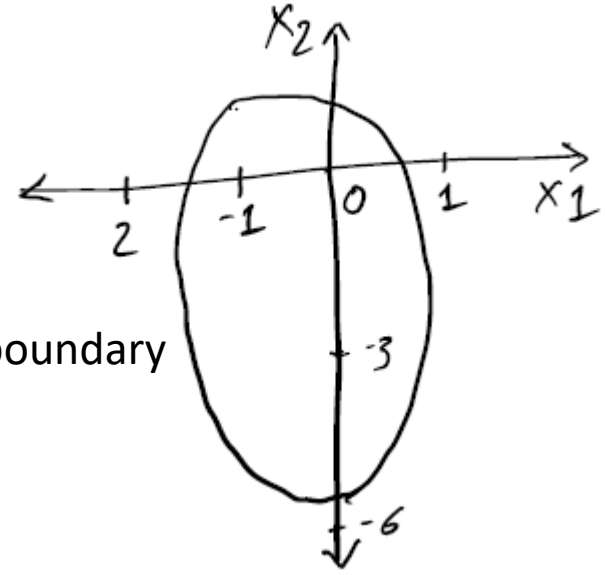
Support Vector Machines

Support Vector Machines

- Suppose we have 2 predictor variables x_1 and x_2 and we add 2 more predictor variables $x_3 = x_1^2$ and $x_4 = x_2^2$
- Next, we want to apply the support vector classifier and we obtain the decision boundary that satisfies the condition:
- $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p = 0$

⬇ Obtained after fitting on a training data set

- $-0.13 + 0.7\, x_1 + 0.5\, x_2 + 0.5\, x_1^2 + 0.1\, x_2^2 = 0$ Note that $\sum_j \beta_j^2 = 1$



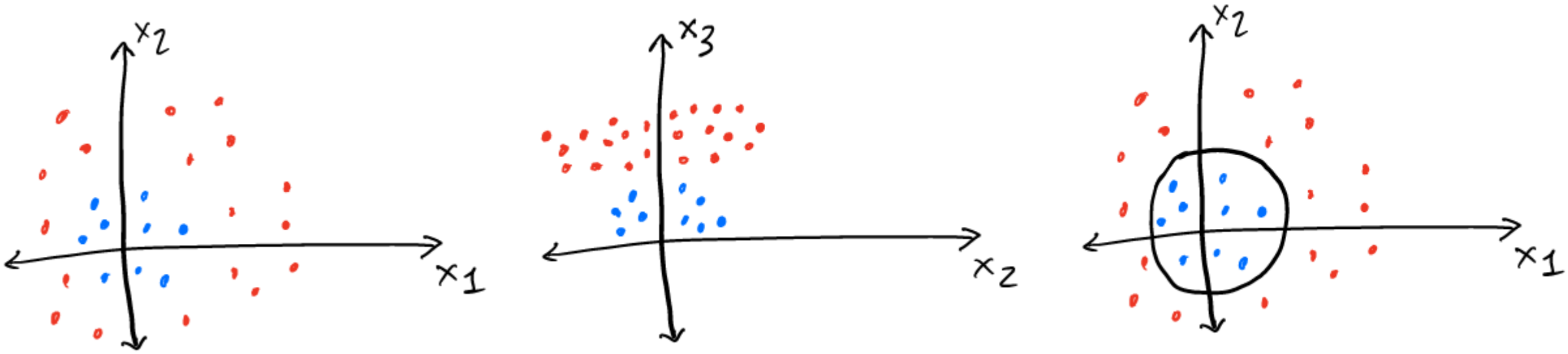
Elliptical decision boundary

- From the above we can arrive at:
- $0.5(x_1^2+1.4\, x_1) + 0.1(x_2^2+5\, x_2) = (1-0.87)$
- $0.5\,(x_1^2+1.4\, x_1+0.49) + 0.1\,(x_2^2+5\, x_2+6.25) = 1$
- $0.5(x_1+0.7)^2 + 0.1(x_2+2.5)^2 = 1$ ⬅ Equation of an ellipse

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*)

- Kernel method is an efficient computational approach for achieving the above

Support Vector Machines



- There is no separating hyperplane that can segregate the 2 classes
- Apply a transformation (kernel function) that maps the original data points into a higher-dimensional space in which they become separable
- Upon transforming it back to the original plane, we obtain the nonlinear circular boundary as shown above