STAT5003

Week 5 : Introduction to classification techniques

Dr. Justin Wishart

THE UNIVERSITY OF
SYDNEY

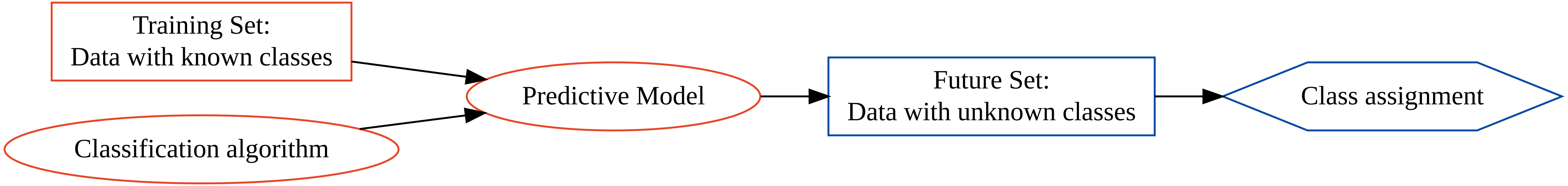
Readings 🏛

- Classification covered in Chapter 4 in James, Witten, Hastie, and Tibshirani (2013)
- Support Vector Machines covered in Chapter 9 in James, Witten, Hastie, et al. (2013)
- **Optional** for SVMs
    - Section 4.5.2 and Sections 14.1-14.3 in Hastie, Tibshirani, and Friedman (2017)

Classification

Basic principles of classification

- Each observation has two properties
  - A class label or response, y
  - A feature vector (vector of predictor variables), $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$
- Goal is to classify y using $\boldsymbol{x}$



Classification vs Clustering

Clustering: classes are unknown, want to discover them from the data (unsupervised)

Classification: classes are predefined, want to use a (training or learning) set of labeled objects to form a classifier for classification of future observations (supervised)

Classification vs Regression

Regression: no class definition, the response variable is a continuous value. Model the relationship between explanatory variables and the response variable.

Classification: samples are predefined to be from a given class. Classification models produce a continuous valued prediction, which is usually in the form of a probability (i.e. the predicted values of class membership for any individual sample are between 0 and 1 and sum to 1). A predicted class is required in order to make a decision.

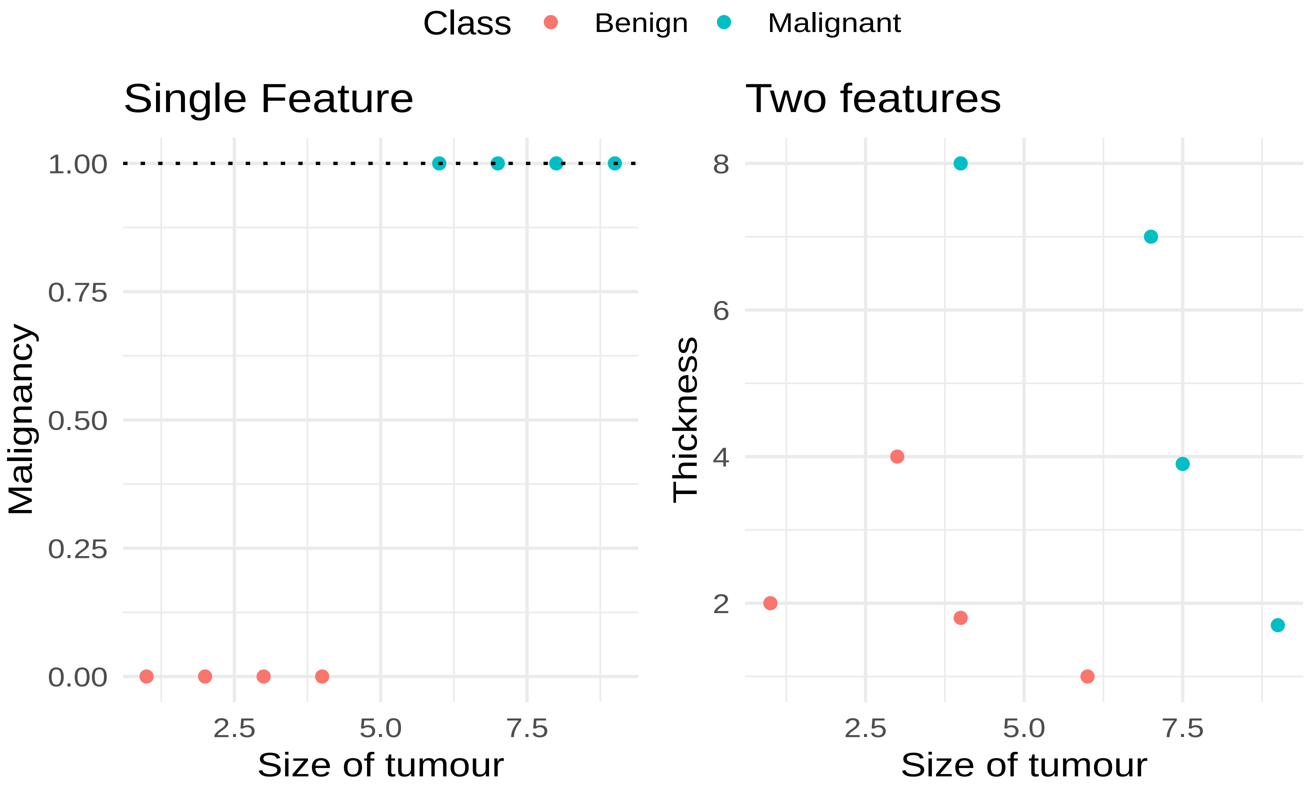
Classification algorithms to discuss

- Logistic Regression
- Linear discriminant analysis (LDA)
- k-nearest neighbours
- Supper vector machines (SVM)

Binary or Two class classification

- Binary in there are two possible values (0 or 1, `TRUE` or `FALSE`)
- Examples of binary classification:
  - Email: Spam / Not Spam
  - Tumour: Malignant /Benign
- Labels are similarly described, $y \in \{0, 1\}$
  - 0: "negative class"
  - 1: "positive class"

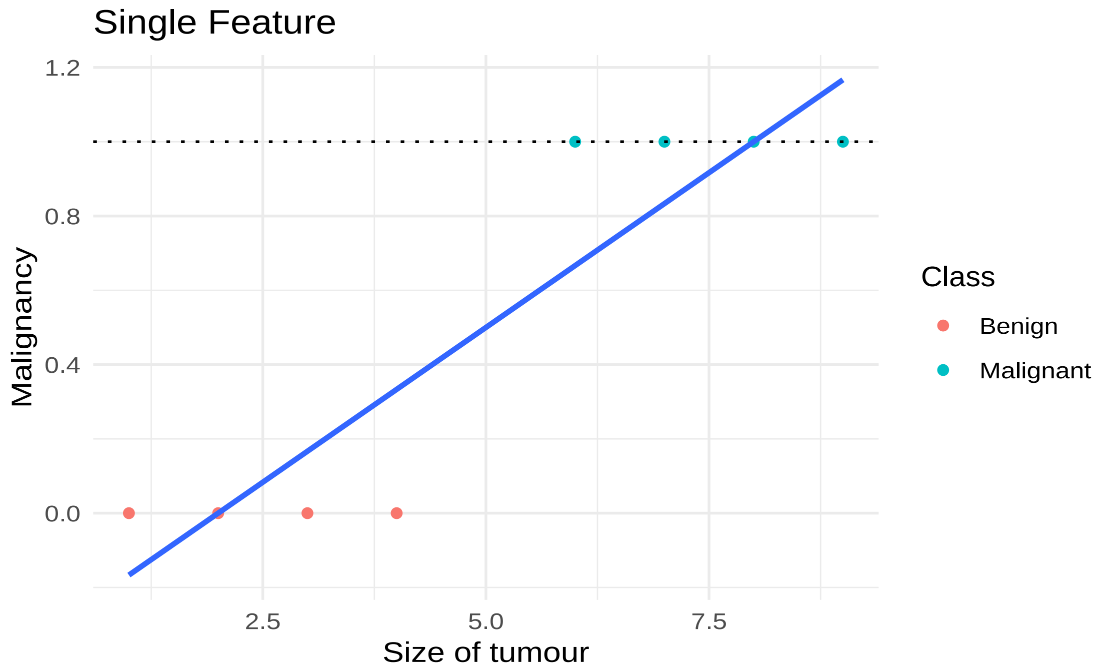
Problem setup



- Threshold classifier output $h_\theta(x)$ at 0.5:
  - if $h_\theta(x) > 0.5$, predict $y = 1$
  - if $h_\theta(x) < 0.5$, predict $y = 0$

Why not use simple linear regression?

- Y is the target value is a *binary* outcome.



- Linear regression is not constrained to $0 < y < 1$ for all x
  - What is the interpretation when $\hat{y} > 1$ (or < 0)

Linear regression misspecifications here

- The regression line $\beta_0 + \beta_1 x$ can span the entire real line
    - all values between $-\infty$ to ∞
- In the tumour diagnosis problem, the target variable y only takes two values: 0 or 1.
- The linear regression model is not well specified for this purpose.

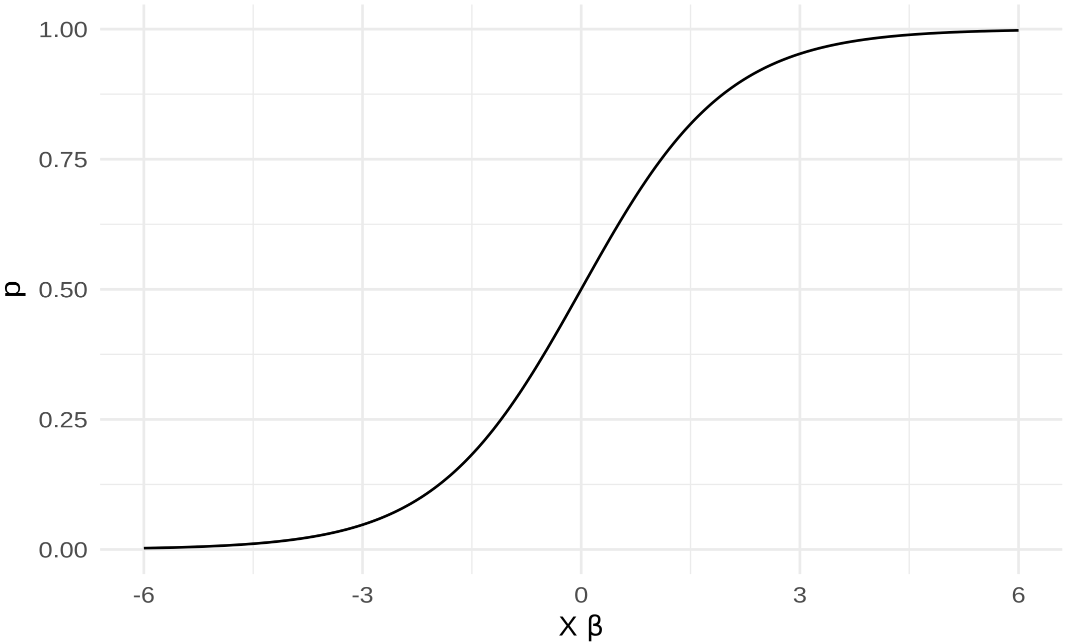
Logistic regression

Logistic regression

- Previously we had the multiple regression

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$
$$\text{Define } \boldsymbol{\theta}^T \boldsymbol{x} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

  - Could write this as $\mathbb{E}Y = \theta^T \boldsymbol{x} = \mu$
- Can **generalise** this to $g(\mathbb{E}Y) = \theta^T \boldsymbol{x} = g(\mu)$
- Logistic regression is a special case of one of these generalised linear models.
- $\log\left(\dfrac{p}{1-p}\right) = \boldsymbol{\theta}^T \boldsymbol{x}$
- Solve for p gives
  - $$p = P(Y = 1|\boldsymbol{x}) = g^{-1}(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})}$$



Logistic regression terminology

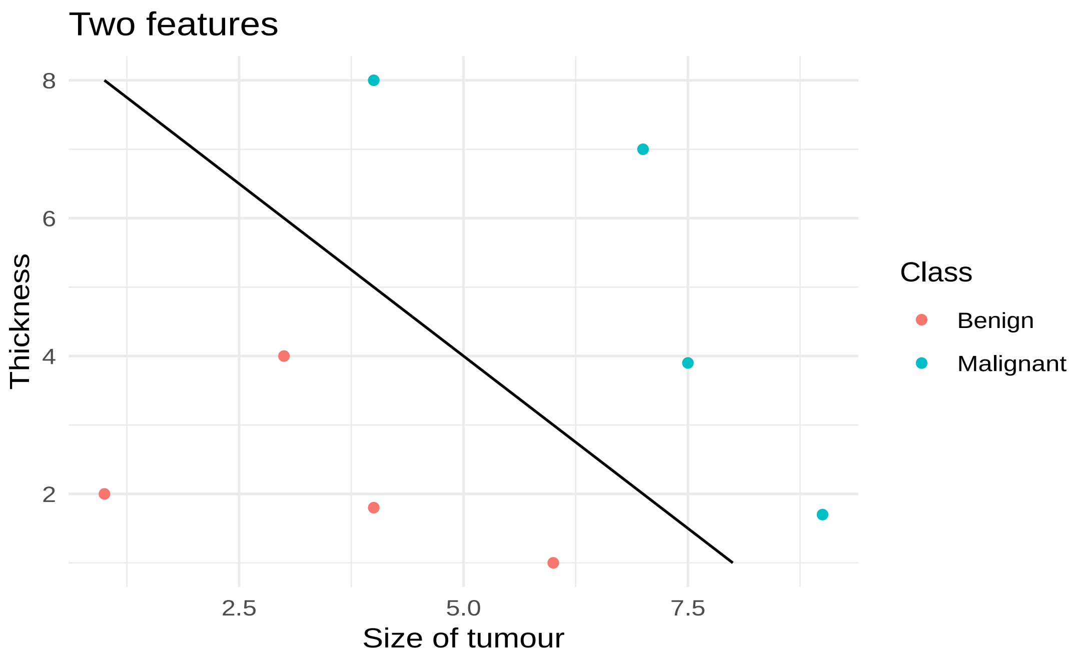
- Logistic function $\frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$
  - Responsible from mapping the features from $(-\infty, \infty) = \mathbb{R}$ to (0, 1)
- Odds ratio: $\frac{p}{1-p}$
  - Maps the probability from (0, 1) to $(0, \infty)$
- Log-odds or logit: $\log\left(\dfrac{p}{1-p}\right)$
- In logistic regression we want the values in the logit space to be linear in X

Logistic regression: decision boundary

- Decision boundary

$$P(Y = 1|\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})}$$

- Predict $Y = 1$ if $\boldsymbol{\theta}^T \boldsymbol{x} \geq 0$



Linear Discriminant Analysis (LDA)

THE UNIVERSITY OF SYDNEY

Linear Discriminant Analysis (LDA)

LDA undertakes the same task as Logistic Regression. It classifies data based on categorical variables

- Malignant or benign
- Making profit or not
- Buy a product or not
- Satisfied customer or not

Bayes' Theorem in the classification context

$$p_k(x) = P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}$$

Posterior: The probability of classifying observation to group k given it has features x

Prior: The prior probability of an observation in general belonging to group k

- $f_k(x) = P(X = x|Y = k)$ is the density function for feature x given it's in group k

Logistic Regression vs LDA formulations

- With Logistic Regression we modeled the probability of Y being from the k^{th} class as

$$p_k(x) = P(Y = k|X = x) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

- Bayes' Theorem states

$$p_k(x) = P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}$$

$$\begin{aligned} \pi_k &: \text{Probability of coming from class } k \text{ (prior probability)} \\ f_k(x) &: \text{Density function for } X \text{ given that } X \text{ is an observation from class } k. \end{aligned}$$

LDA estimates of π_k and $f_k(x)$

- We can estimate π_k and $f_k(x)$ to compute $p_k(x)$
- The most common model for $f_k(x)$ is the Normal Density (LDA)

$$f_k(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right)$$

- Using the above density, we only need to estimate three quantities to compute $p_k(x)$
  - That is, μ_k, σ_k^2 and π_k
- For simplicity, assume common variance.

Use training data set for estimation

- The mean $\widehat{\mu_k}$ could be estimated by the average of all training observations from the k^{th} class.
- The variance σ^2 could be estimated as the weighted average of variances of all k classes.
- The proportion π_k is estimated as the proportion of the training observations that belong to the k^{th} class.

$$\widehat{\mu_k} = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\sigma^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \widehat{\mu}_k)^2$$

Simple example with one predictor

- Suppose we have only one predictor
- Two normal density function $f_1(x)$ and $f_2(x)$, represent two distinct classes
- The two density functions overlap, so there is some uncertainty about the class to which an observation with an unknown class belongs
- The dashed vertical line represents Bayes' decision boundary

Deriving LDA for one predictor

- Assuming one predictor (and common variance)

$$f_k(x) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

$$p_k(x) = P(Y=k|x) = \frac{\frac{\pi_k}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)}{\sum_{l=0}^{K} \frac{\pi_\ell}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)}$$

- Find the class k which we maximize:

$$x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

LDA Decision boundary

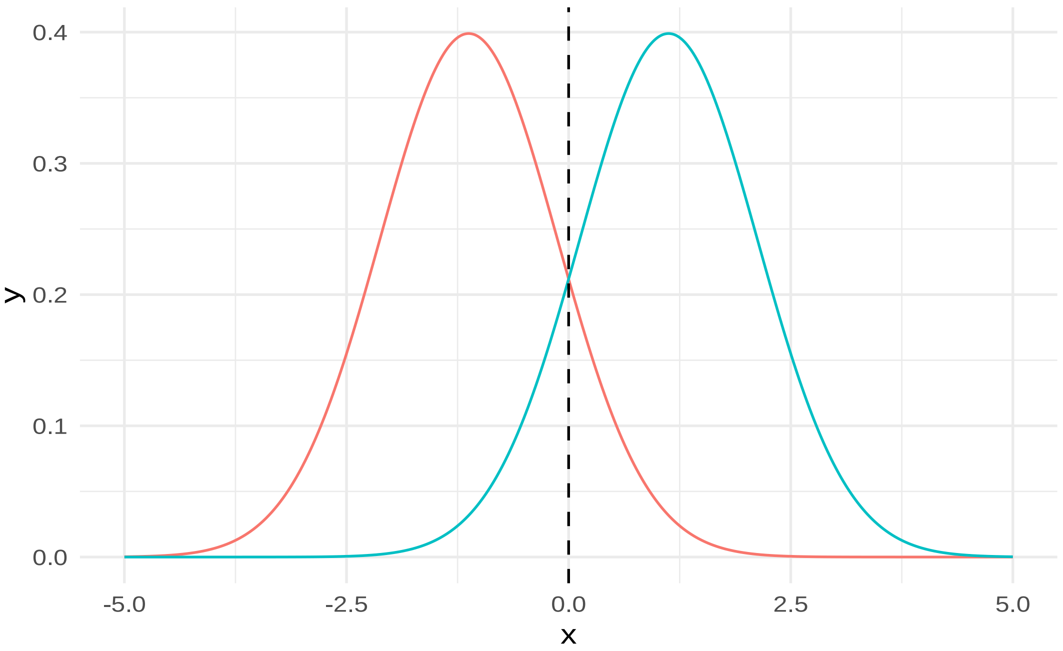
If $K = 2$ and $\pi_1 = \pi_2$, then assigns an observation to class 1 if $\log p_1(x) > \log p_2(x) \rightsquigarrow \log\left(\frac{p_1(x)}{p_2(x)}\right) > 0$

Substituting in the previous equation (assuming $\sigma_i = \sigma$) we have,

$$\begin{aligned} \log\left(\frac{p_1(x)}{p_2(x)}\right) &> 0 \\ \log(\pi_1) - \log(\pi_2) + \frac{x\mu_1}{2\sigma^2} - \frac{x\mu_2}{2\sigma^2} - \frac{\mu_1^2}{2\sigma^2} + -\frac{\mu_1^2}{2\sigma^2} &> 0 \\ 2x(\mu_1 - \mu_2) &> \mu_1^2 - \mu_2^2 \end{aligned}$$

- Decision boundary at

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$



Why not logistic regression?

- In the case where n is small, and the distribution of predictors X is approximately normal, then LDA is more stable than Logistic Regression
- LDA is more popular when we have more than two response classes. More intuitive to predict class assignment.
- When the classes are well separated, the parameter estimates for logistic regression are unstable. However, LDA doesn't suffer any stability issues in this case.

Logistic Regression vs LDA

Similarity:

- Both Logistic Regression and LDA produce linear boundaries

Differences:

- LDA assumes that the observations are drawn from the normal distribution with common variance in each class, while logistic regression does not have this assumption.
- LDA would do better than Logistic Regression if the assumption of normality hold, otherwise logistic regression may outperform LDA

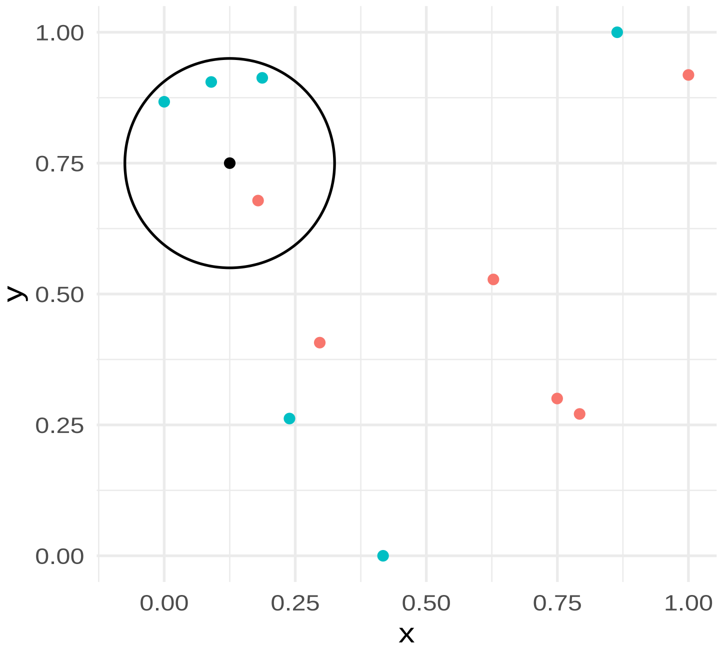
k-Nearest Neighbours (kNN)

k-Nearest Neighbours

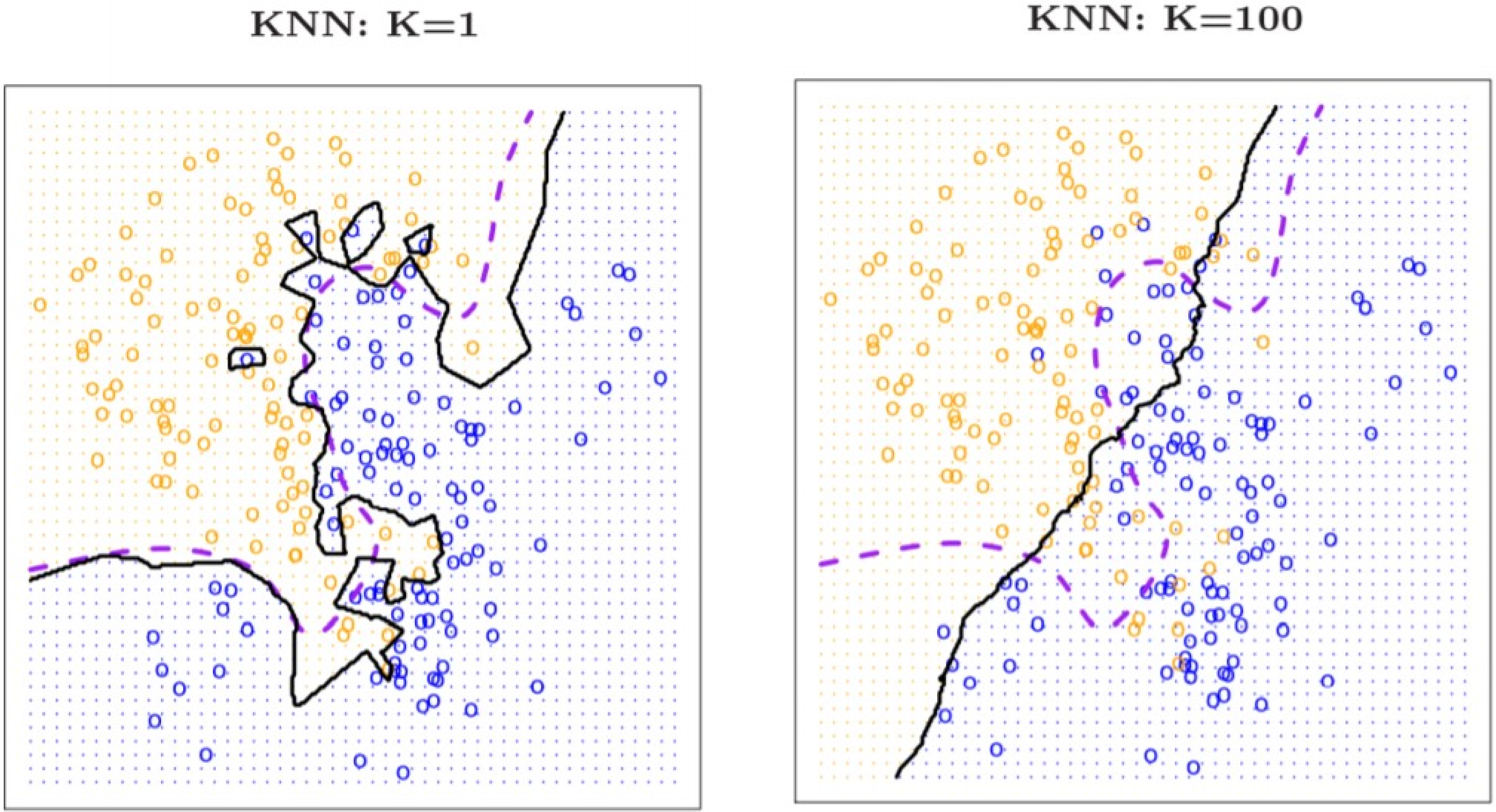
- kNN model is probability of an observation with features $\boldsymbol{x}$ belonging to group ℓ depends on the membership of the nearest points to $\boldsymbol{x}$

$$P(Y = \ell | \boldsymbol{x}) = \frac{1}{k} \sum_{N_x^k} 1_{\{y=\ell\}} = \frac{1}{k} \times \text{Count of the closest } k \text{ points that belong to group } \ell$$

- Suppose $k = 4$ is chosen. At the candidate black point. The four nearest neighbours are inspected. There is probability 3/4 of being in the green group and 1/4 for being in the orange group.



k-Nearest Neighbours



KNN: K=1

KNN: K=100

kNN vs (LDA and Logistic Regression)

- kNN takes a completely different approach
- kNN is completely non-parametric: No assumptions are made about the shape of the decision boundary
- Advantage of kNN: We can expect kNN to dominate both LDA and Logistic Regression when the decision boundary is highly non-linear
- Disadvantage of kNN: kNN does not tell us which predictors are important (no table of coefficients)

Support Vector Machines (SVM)

Support Vector Machines (SVM)

- Basic idea behind SVM

> Find a plane that separates the classes in the feature space.

- If a basic mathematical plane is not possible due to overlap
  - Relax the idea of complete separation (allow points to violate the boundary)
  - Enrich and enlarge the feature space so that separation is possible
    - Think dimension expansion

What is a hyperplane?

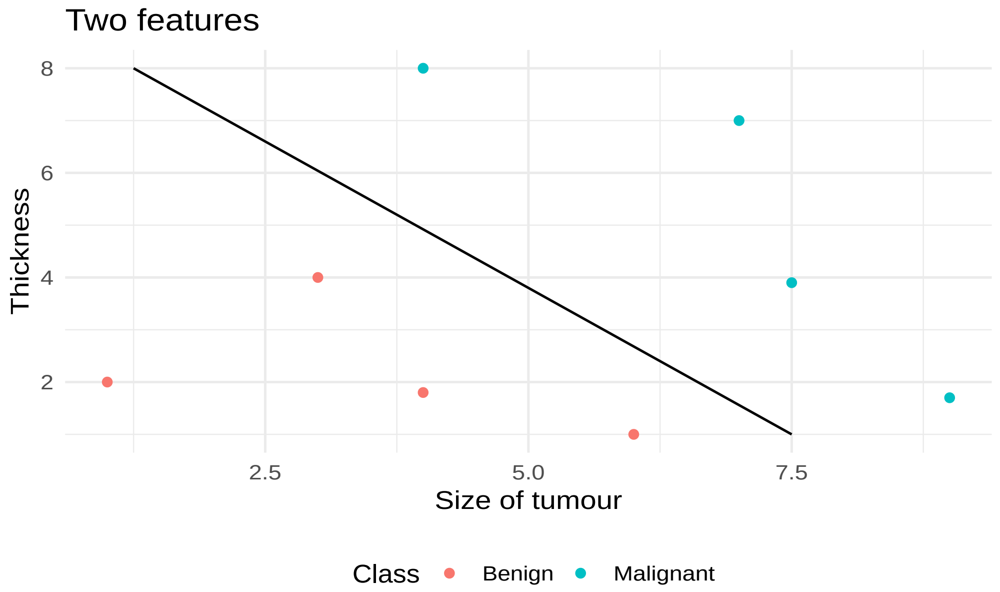
- In p dimensions it is a flat affine subspace of dimension $p-1$
- General equation has the form

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

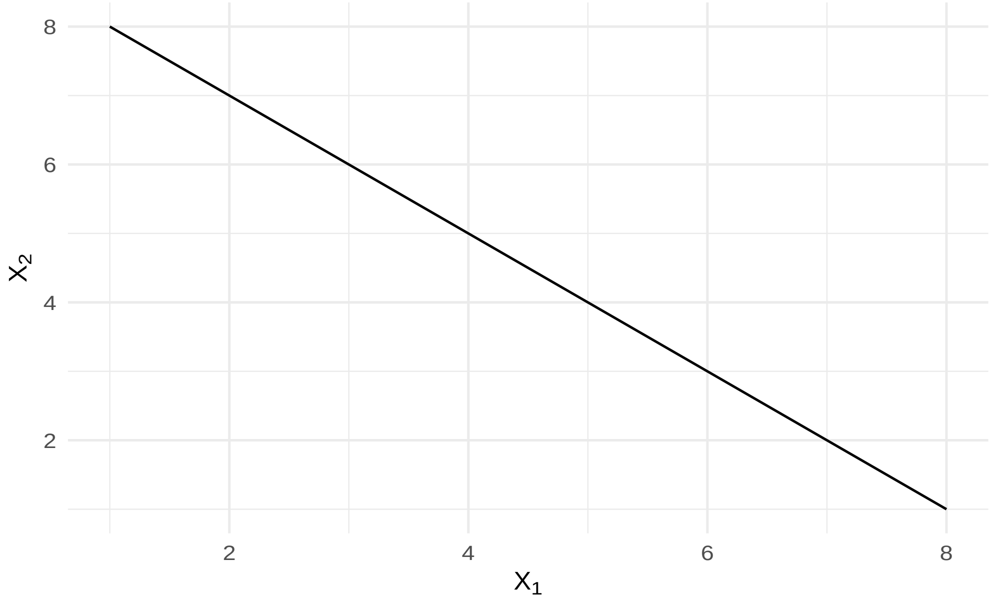
- In $p = 2$ dimensions, the hyperplane is a line.
- If $\beta_0 = 0$, the hyperplane passes through the origin, otherwise it does not.
- The vector $(\beta_1, \beta_2, \ldots, \beta_p)$ is called the normal vector
    - It points in a direction orthogonal to the surface of the hyperplane

Hyperplane example

- Earlier hypothetical example

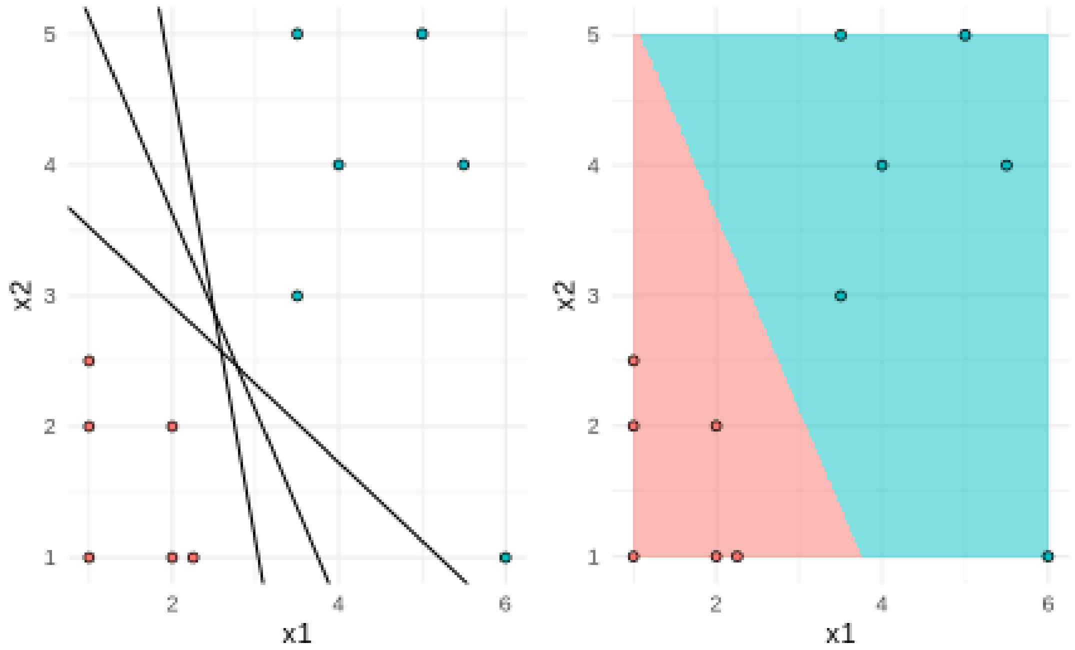


- Consider just the line (hyperplane)



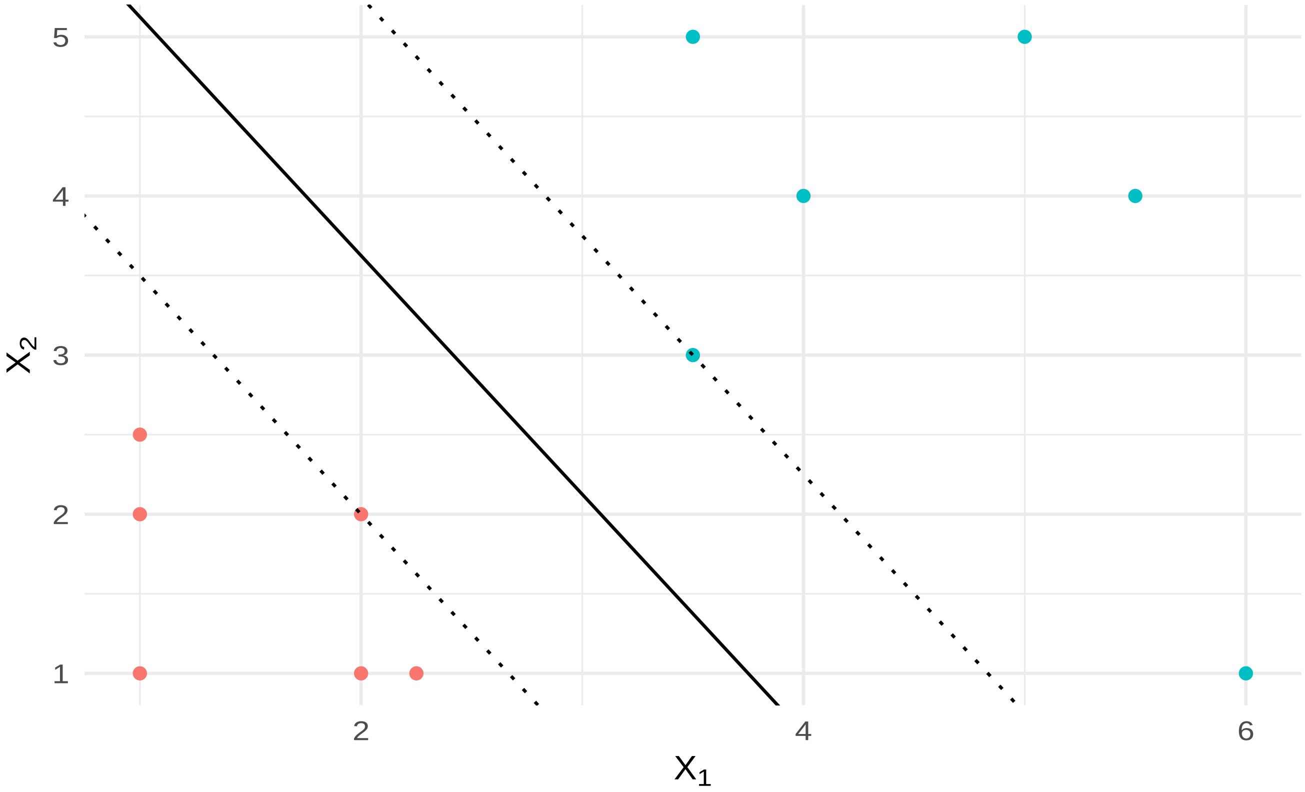
- Equation of the hyperplane here is $-9 + X_1 + X_2 = 0$

Separating hyperplanes



- Consider coding Benign (red?) as $y_i = -1$ and malignant (blue?) as $y_i = 1$
- Then $y_i f(x_i) > 0$ for all i, $f(x_i)$ defines a separating hyperplane.
- If $f(x_i) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ defines a hyperplane
  - $f(x) > 0$ defines a region on one side of the hyperplane
  - $f(x) < 0$ defines a region on one other side of the hyperplane

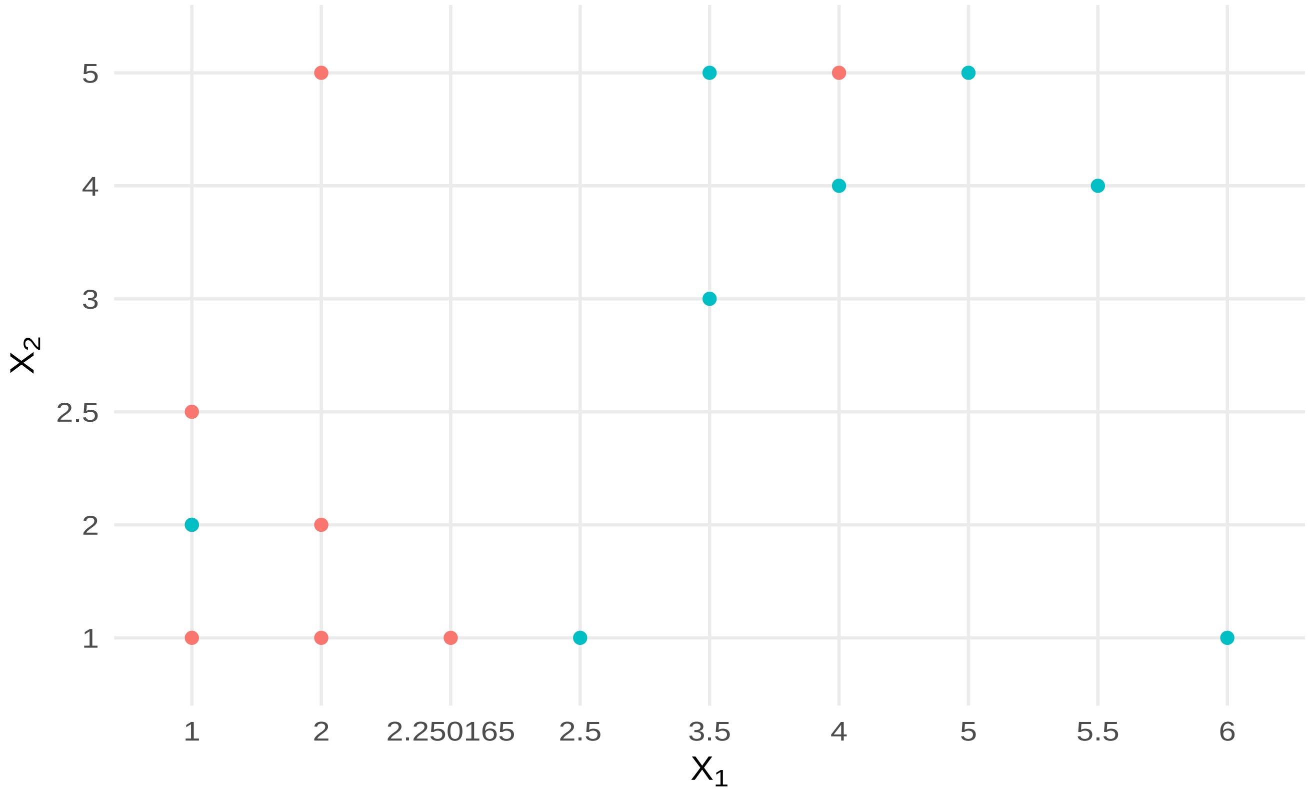
Maximal Margin Classifier



$$\max_{\beta_0,\beta_1,\beta_2,\ldots,\beta_p} M \quad \text{such that} \quad \sum_{j=1}^{p} \beta_j^2 = 1$$

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

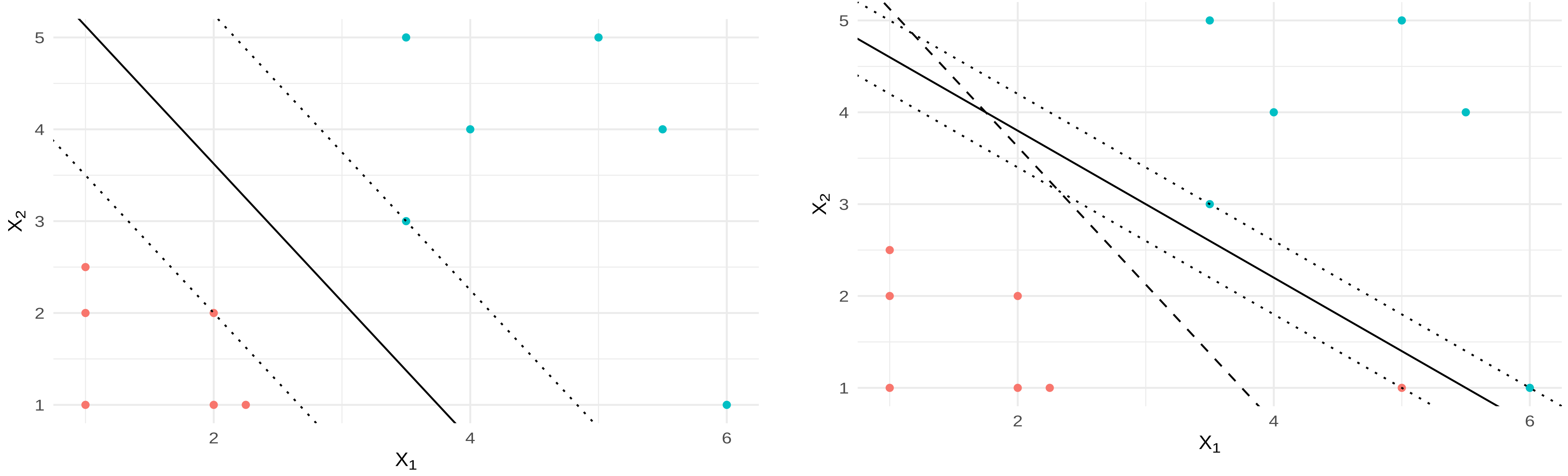
Non-perfect separation



- There is no linear boundary (hyperplane) that perfectly separates the classes.
- This is typically the case that observations don't have a perfect boundary of separation.
  - Except in the case when $n < p$ (more features than observations)

Effect of noisy data

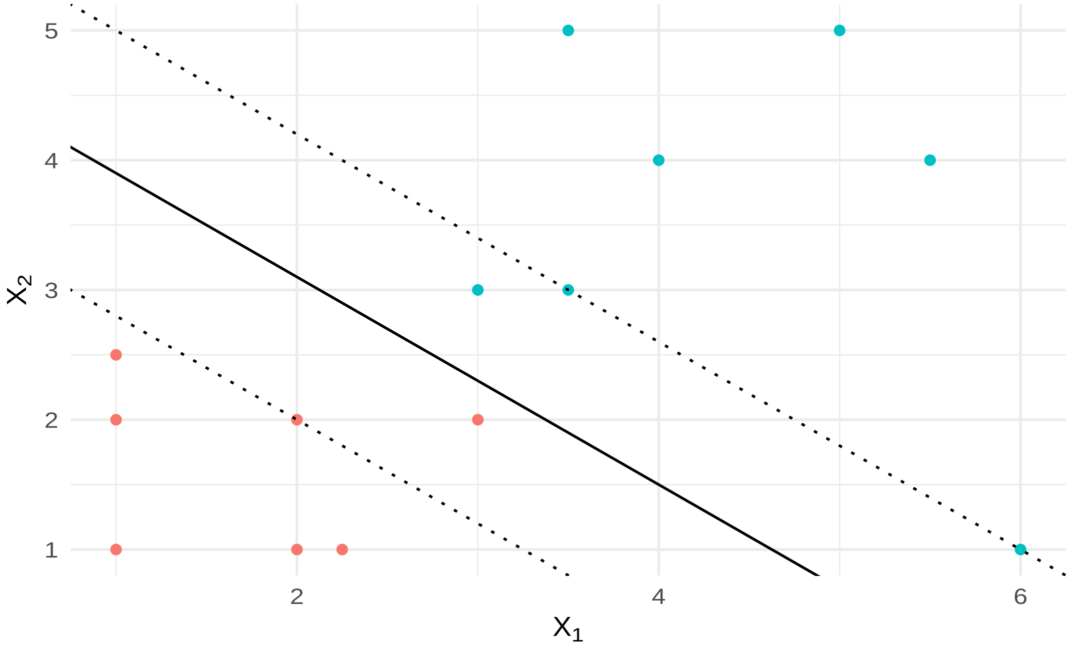
Consider the impact of one extra observation

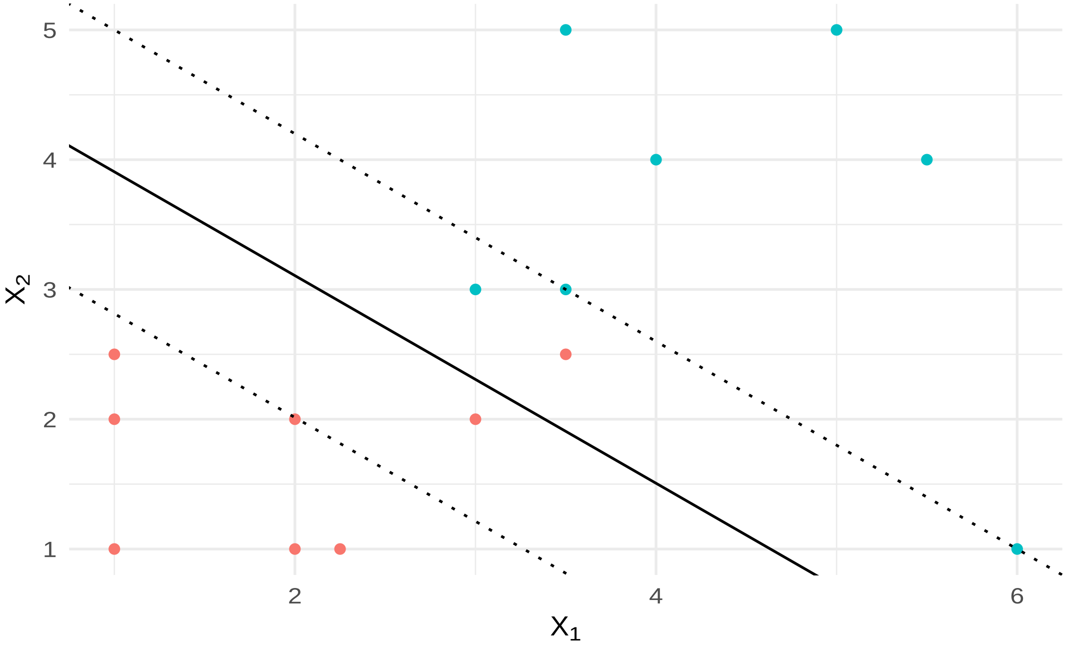


- Data could be separable, but noisy $\rightsquigarrow$ unstable solution for the maximal margin classifier.
- The support vector classifier maximizes a soft margin.
    - relaxes requirement for all observations to be on the correct side of the margin

Soft margin examples

- Observations allowed in the margin





- Observations on the correct side of hyperplane

- Allow observations on incorrect side of hyperplane

Support Vector Classifier

Support Vector Classifier solves the following optimization problem:

$$\max_{\beta_0,\beta_1,\beta_2,...,\beta_p,\epsilon_1,...,\epsilon_n} M \quad \text{such that} \quad \sum_{j=1}^{p} \beta_j^2 = 1$$

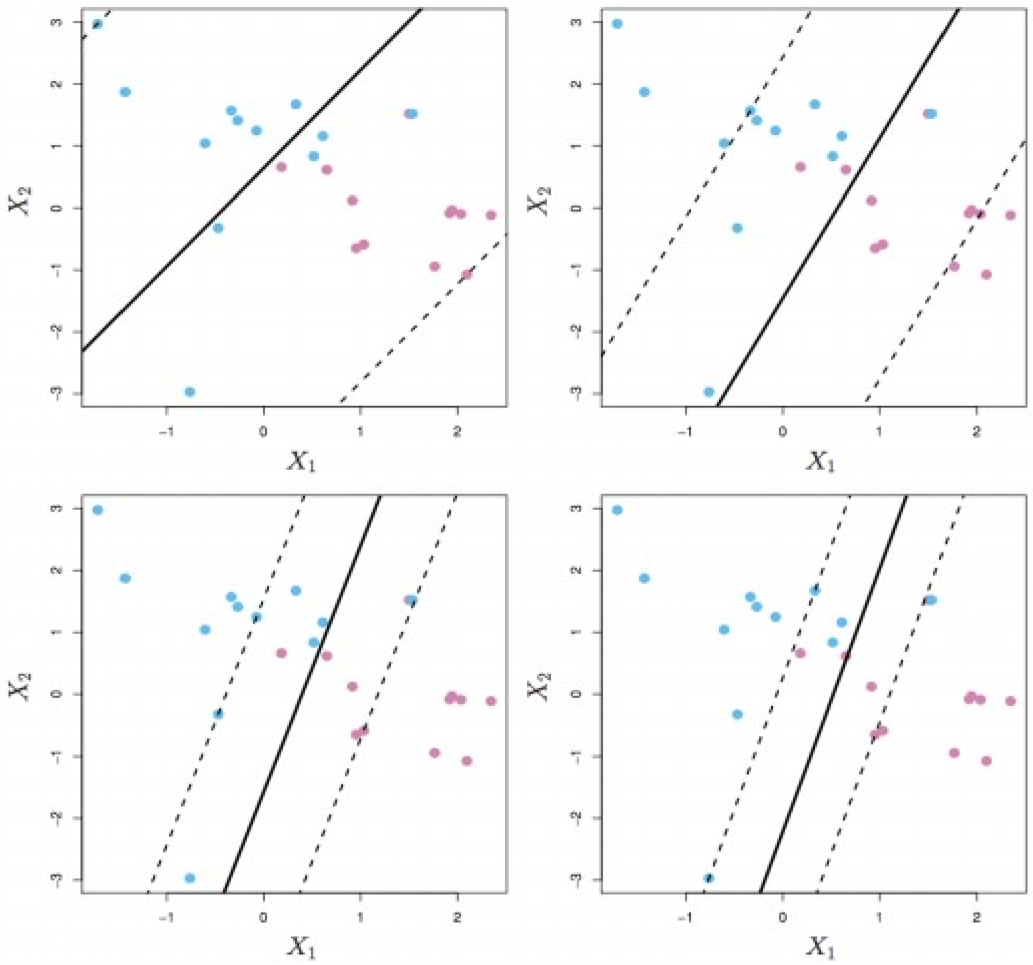
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots \beta_p x_{ip}) \geq M(1 - \epsilon_i)$$

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

- C is a non-negative tuning parameter,
- M is the width of the margin,
- ϵ_i are slack variables that allow observations to be on the wrong side of the margin,
  - if $\epsilon_i > 1$, then observation i is on the wrong side of the hyperplane
  - if $0 < \epsilon_i \leq 1$, then observation i is on the correct side but inside margin
  - if $\epsilon_i = 0$, then observation i is on the correct side and past the margin.

Impact of cost parameter C

Largest C



Smallest C

Limitations of support vector classifier

- Single linear boundary can be insufficient

Feature space expansion

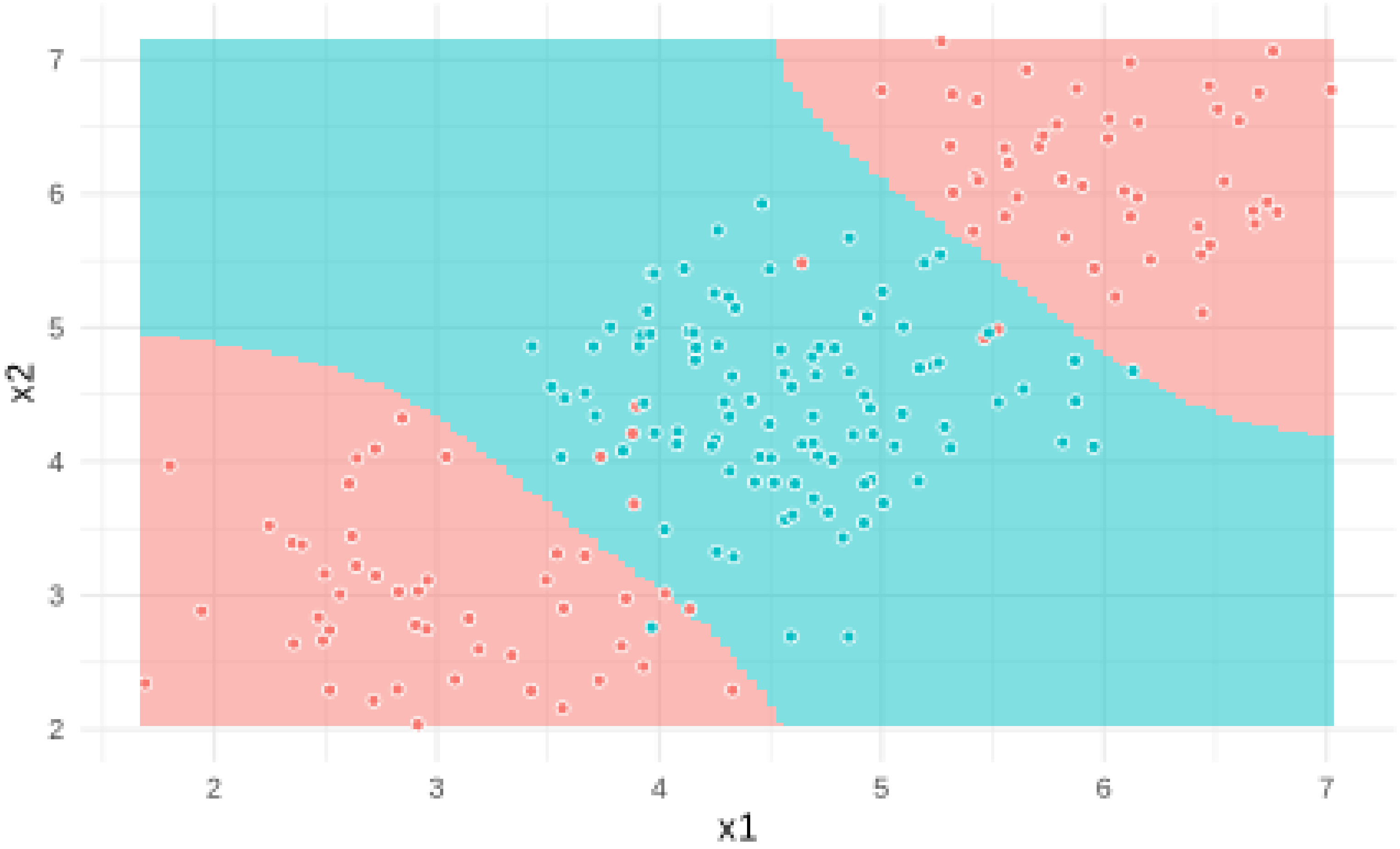
- Enlarge the space of features by including transformations:
  - e.g. new features that are powers and products X_1^2, X_1^3, X_1X_2
  - Hence go from p-dimensional space to $P > p$ dimensional
- Fit (linear) support vector classifier in the expanded feature space.
  - Impact is a **non-linear** decision boundary in original feature space.
- Example: Suppose we start off in 2-dimensional feature space (X_1, X_2).
  - Make new feature space $(X_1, X_2, X_1^2, X_2^2, X_1X_2)$.
  - Then the decision boundary would be of the form:

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

- This leads to non-linear decision boundary in the original space (quadratic conic sections)

Sixth order polynomial

- Using degree six polynomial expansion



Non-linearity and kernels

- Polynomials get complicated and a burden very quickly as dimension increases.
- More elegant solution is to induce non-linear structure in Support vector classifier with **kernels**
- The elegance comes from the role of the **inner product** in the support vector classifier definition

Inner products and support vectors

- Recall $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p$
- Inner product between vectors given by,

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \sum_{k=1}^{p} x_{ik} x_{jk}$$

- Recall the hyperplane equation

$$f(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

- The linear support vector classifier can be represented as

$$f(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{n} \alpha_j \langle \boldsymbol{x}, \boldsymbol{x}_j \rangle$$

Support set

- Estimation of the parameters $\alpha_1, \alpha_2, \ldots, \alpha_n$ and β_0 required
  - All that is required are the inner products between pairs of training observations.
- Usually $\widehat{\alpha}_i = 0$ with the non-zero values occuring on the support vectors
  - I.e. ones that lie on the margin.

$$f(\boldsymbol{x}) = \beta_0 + \sum_{j \in S} \alpha_j \langle \boldsymbol{x}, \boldsymbol{x}_j \rangle$$

- Here, S is the support set of indices such that $\alpha_j > 0$

Kernel functions

Suppose now we replace the inner product with a generalized function of the form

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

This function is called a kernel.

In this context is quantifies the similarity of two observations.

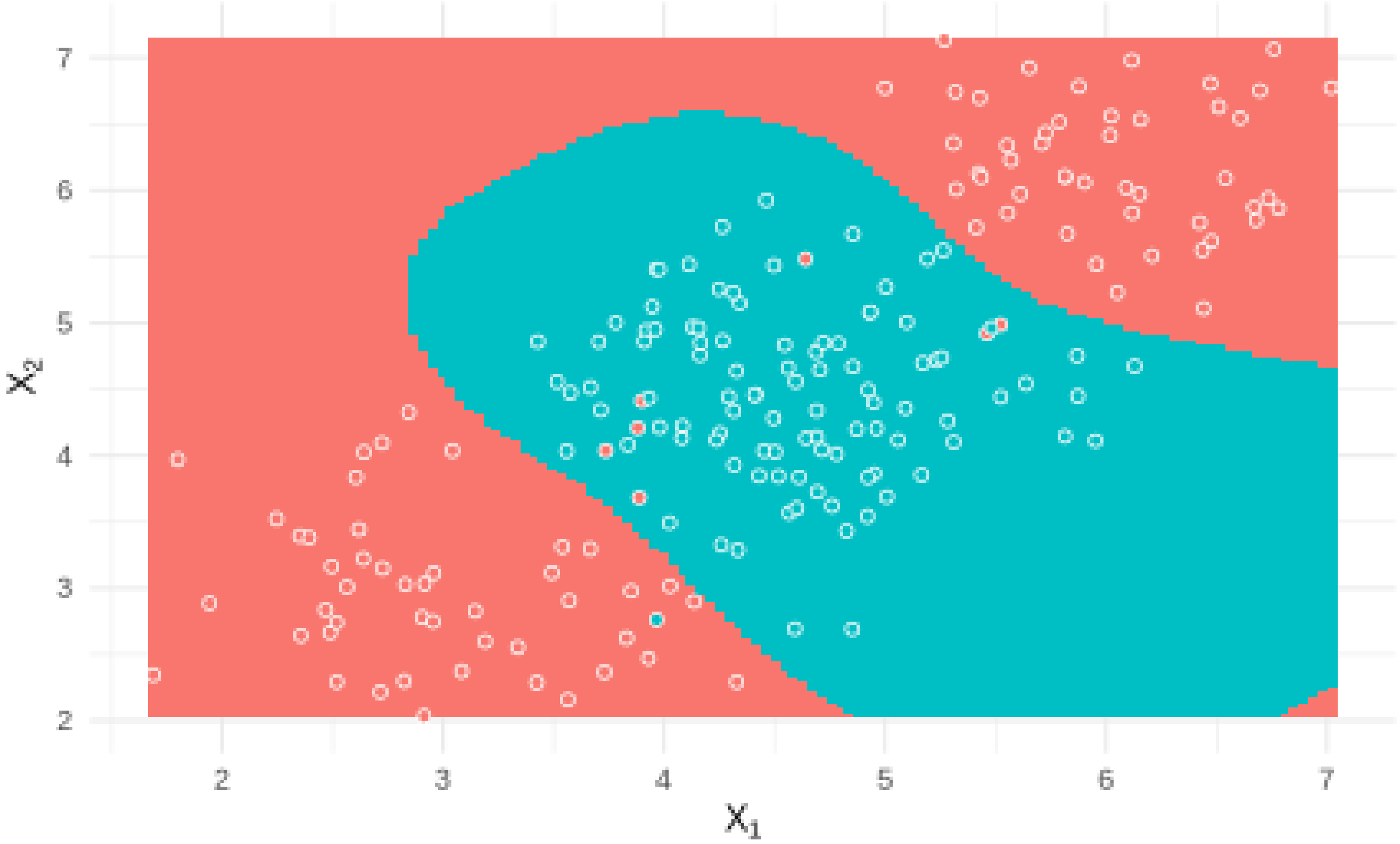
- Examples
  - Polynomial kernel

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle\right)^d$$

  - Gaussian radial kernel

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \sum_{k=1}^{p} (x_{ik} - x_{jk})^2\right)$$

Support vector machines with the radial kernel



SVM with more than two classes

- SVM covered previously are design for binary classification. To expand this to K classes there are two options

1. One vs all:
   - Fit K different binary classifiers, $f_k(\boldsymbol{x})$ for $k = 1, 2, \ldots, K$ where each boundary attempts to separate class k vs the rest.
   - Then $\boldsymbol{x}_i$ is classified to k^* where $f_{k^*}(\boldsymbol{x}_i) > f_j(\boldsymbol{x}_i)$ for all $j \neq k^*$. (i.e. the largest distance from the boundary).
2. One vs one:
   - Fit all $\binom{K}{2}$ pairwise classifiers
   - Fit $\boldsymbol{x}_i$ to the class that wins the most pairwise comparisons.

- Which to use?
  - If K is small, do one vs one. Otherwise recommended One vs all.

References

Hastie, T., R. Tibshirani, and J. Friedman (2017). *The elements of statistical learning: data mining, inference, and prediction*. Second Edition, 12th printing. Springer Science & Business Media.

James, G., D. Witten, T. Hastie, et al. (2013). *An introduction to statistical learning*. Vol. 112. Springer.