MONASH University

ETC3250/5250: Introduction to Machine Learning

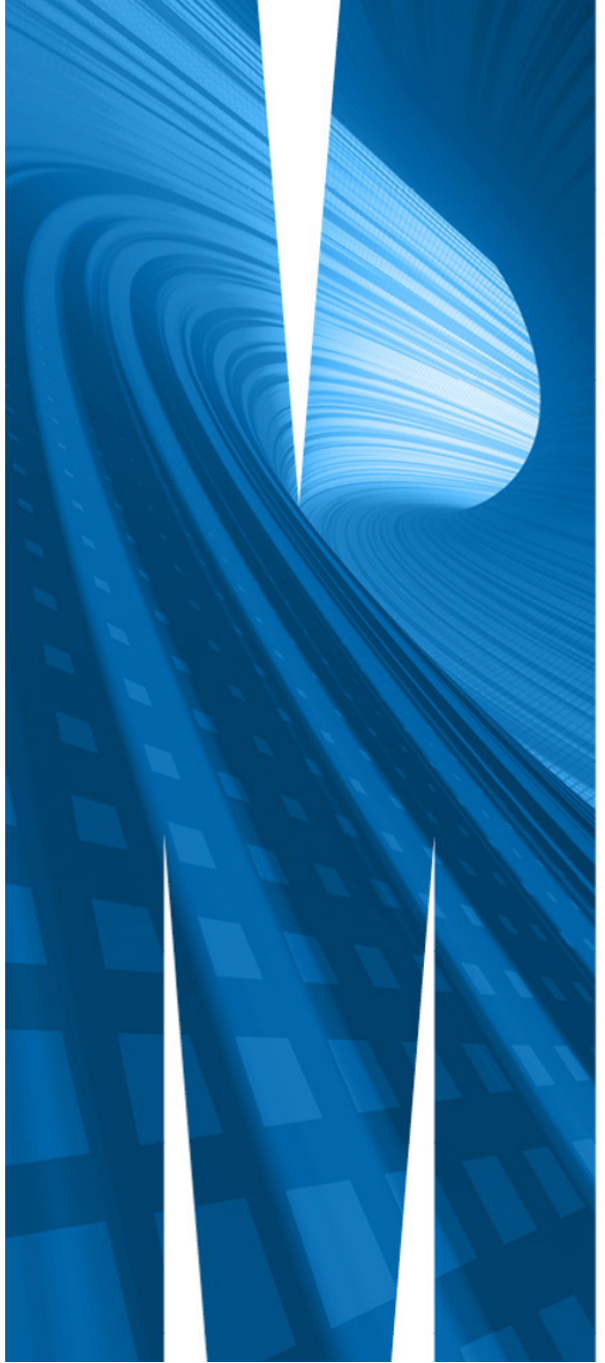
Support vector machines

Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

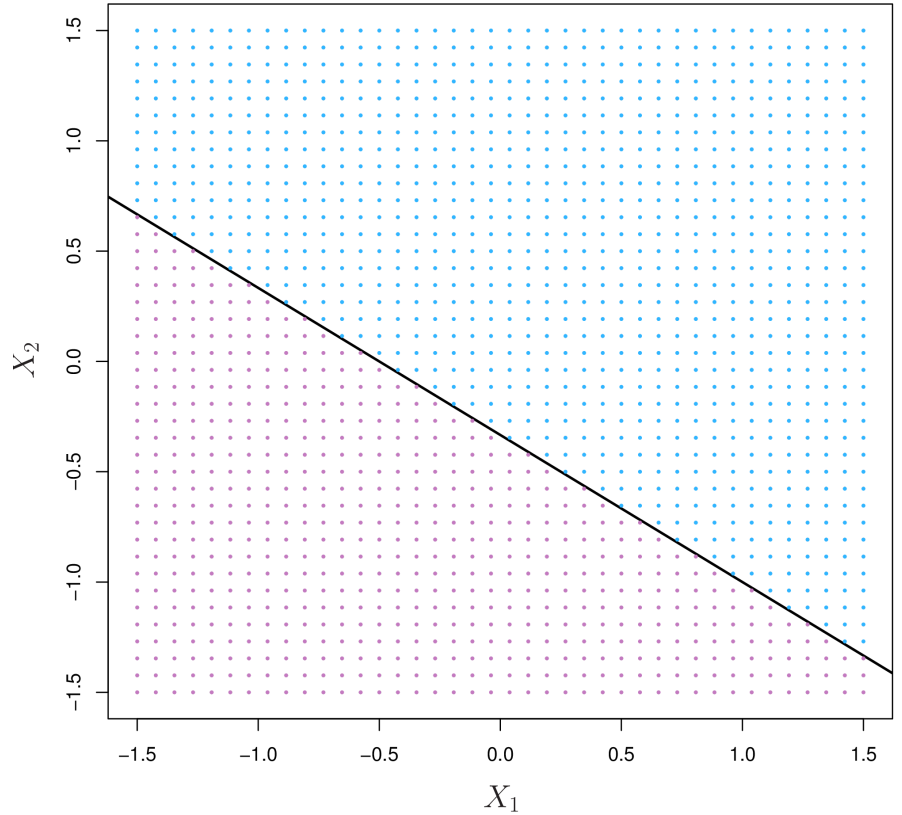
ETC3250.Clayton-x@monash.edu

Week 7b



Separating hyperplanes

In a p-dimensional space, a hyperplane is a linear subspace of dimension $p - 1$.



(ISLR: Fig 9.1)

Separating hyperplanes

The equation of p-dimensional hyperplane is given by

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

For the i^{th} observation,

$$x_i = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{ip} \end{pmatrix}$$

and y_i coded as $\{-1, 1\}$, $i = 1, \ldots, n$, then

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} > 0 \text{ if } y_i = 1,$$

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} < 0 \text{ if } y_i = -1$$

Equivalently,

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

Separating hyperplanes

- A new observation is assigned a class depending on which side of the hyperplane it is located
- Classify the test observation x_0 based on the sign of

$$s(x_0) = \beta_0 + \beta_1 x_{01} + \cdots + \beta_p x_{0p}$$

- If $s(x_0) > 0$, class 1, and if $s(x_0) < 0$, class -1, i.e. $h(x_0) = \text{sign}(s(x_0))$.
- $s(x_0)$ far from zero $\rightarrow x_0$ lies far from the hyperplane + **more confident** about our classification
- $s(x_0)$ close to zero $\rightarrow x_0$ near the hyperplane + **less confident** about our classification

Separating hyperplane classifiers

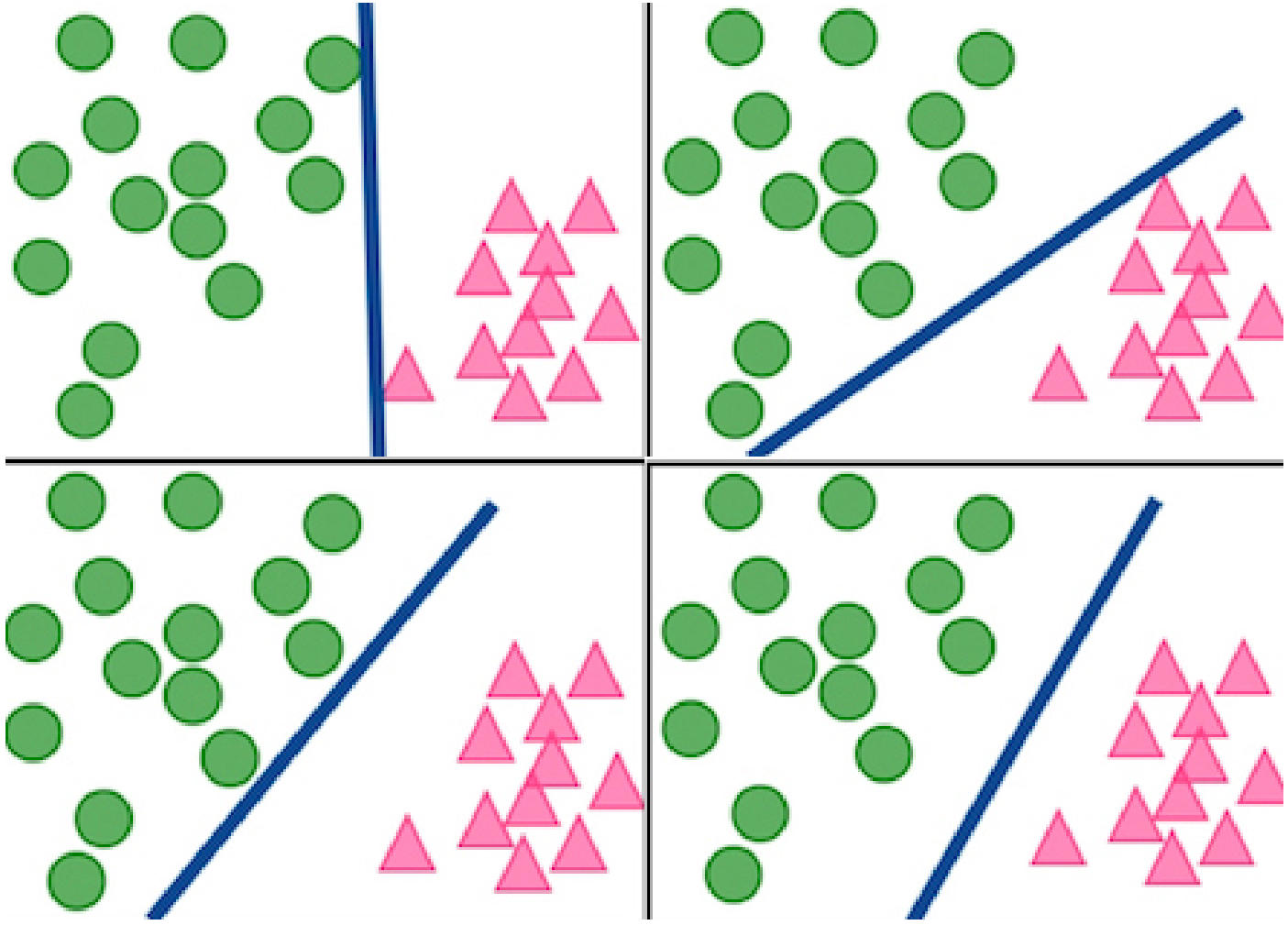
Three different types of hyperplane classifiers.

- Maximal marginal classifier for when the data is perfectly separated by a hyperplane
- Support vector classifier/soft margin classifier for when data is NOT perfectly separated by a hyperplane but still has a linear decision boundary, and
- Support vector machines used for when the data has nonlinear decision boundaries.

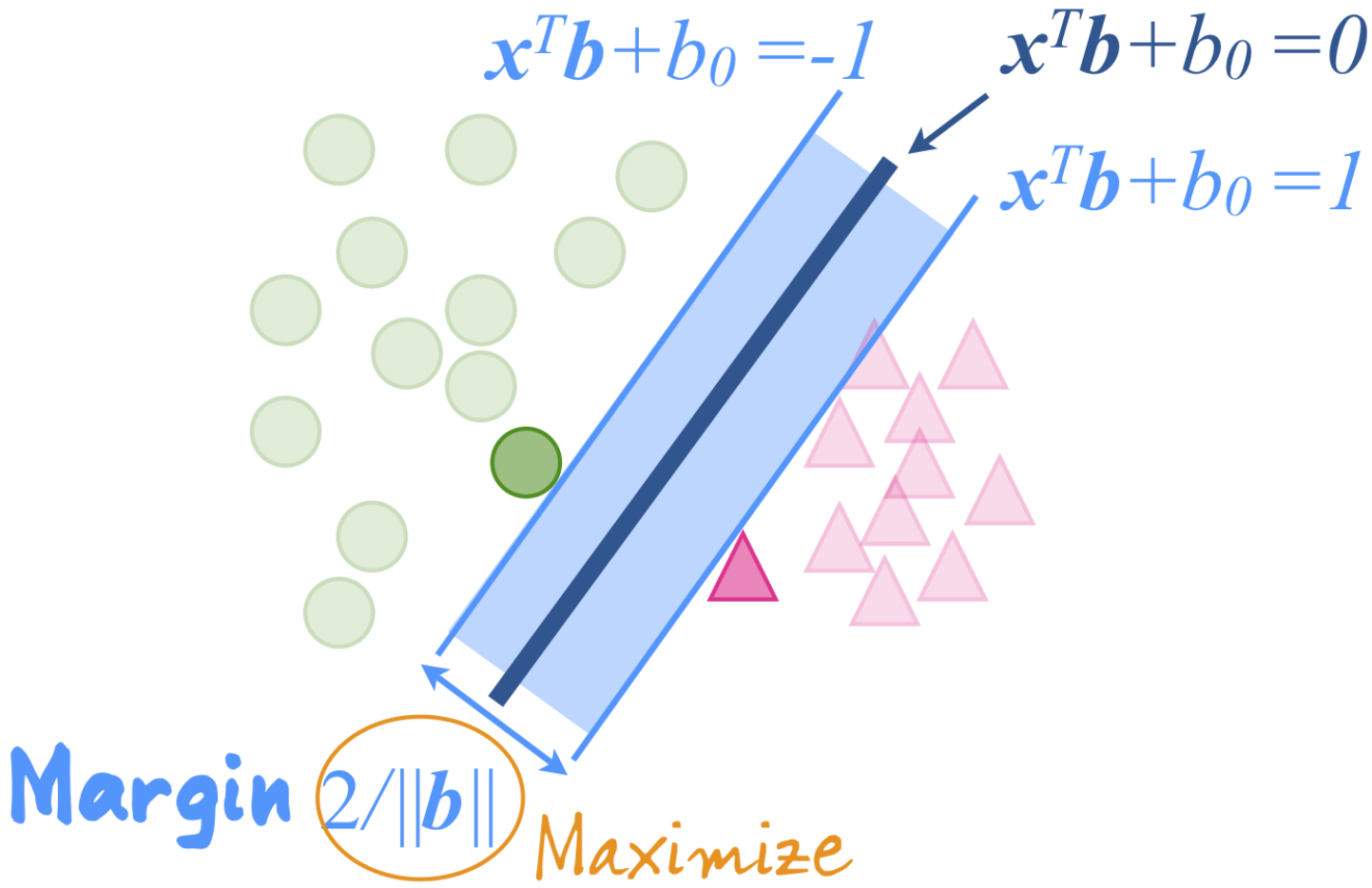
All are support vector machines.

Maximal margin classifier

All are separating hyperplanes. Which is best?

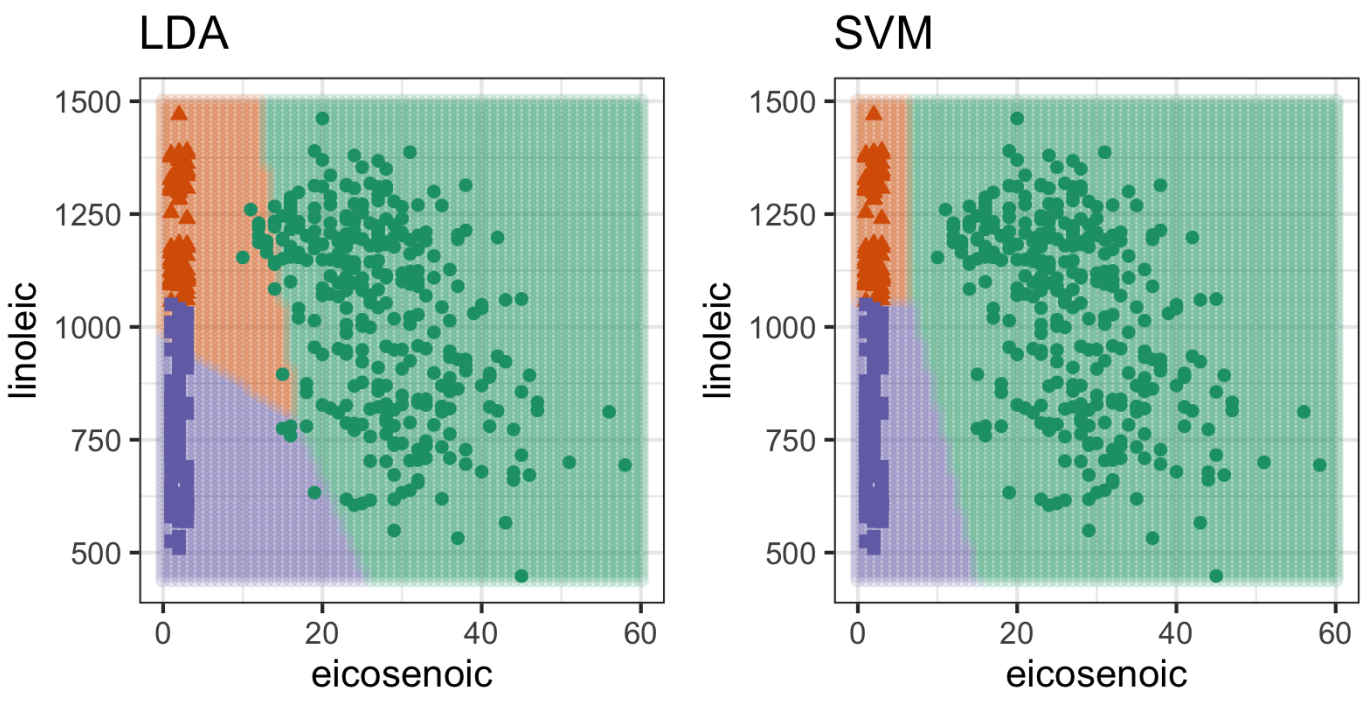


Maximal margin classifier



From LDA to SVM

- Linear discriminant analysis uses the difference between means to set the separating hyperplane.
- Support vector machines uses gaps between points on the outer edge of clusters to set the separating hyperplane.



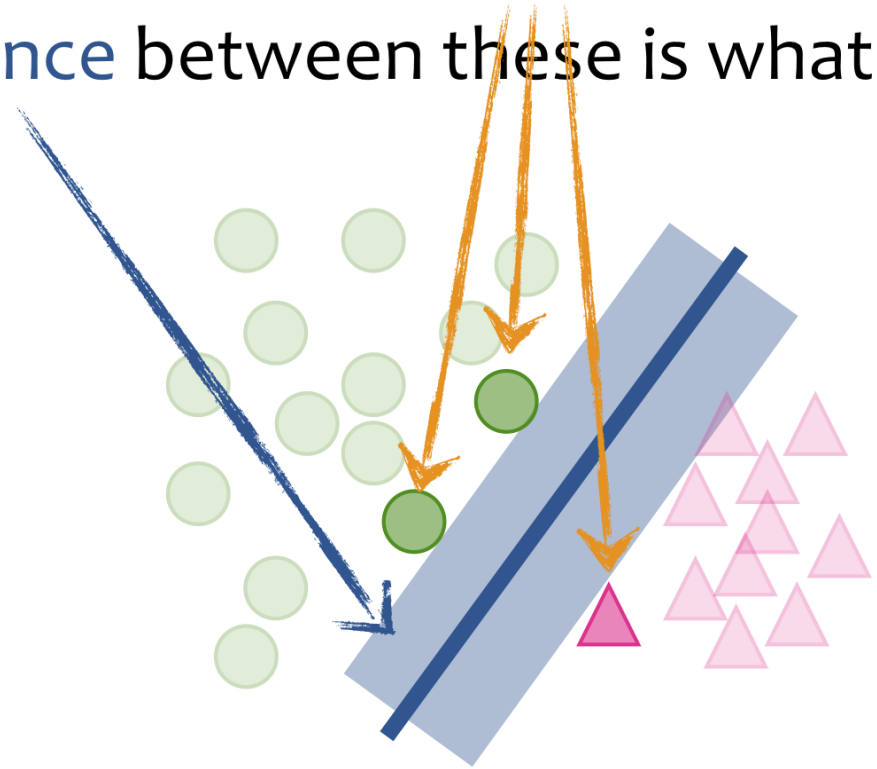

```
svm_mod <-
  svm_rbf(cost = 10) %>%
  set_mode("classification")
  set_engine("kernlab",
  kernel="vanilladot")

olive_svm <- svm_mod %>%
  fit(region~eicosenoic+linc
  data=olive)
```

SVM

- If our data can be perfectly separated using a hyperplane, then there will in fact exist an **infinite number of such hyperplanes**.
- We compute the (perpendicular) distance from each training observation to a given separating hyperplane. The smallest such distance is known as the margin.
- The optimal separating hyperplane (or maximal margin hyperplane) is the separating hyperplane for which the margin is largest.
- We can then classify a test observation based on which side of the maximal margin hyperplane it lies. This is known as the maximal margin classifier.

Support vectors

- Hyperplane is defined by a subset of the point, the support vectors
- Distance between these is what is maximized



See more detailed explanations here.

Support vectors

- The support vectors are equidistant from the maximal margin hyperplane and lie along the dashed lines indicating the width of the margin.
- They support the maximal margin hyperplane in the sense that if these points were moved slightly then the maximal margin hyperplane would move as well

The maximal margin hyperplane depends directly on the support vectors, but not on the other observations

Support vectors define the maximal margin classifier

The separating hyperplane is defined as

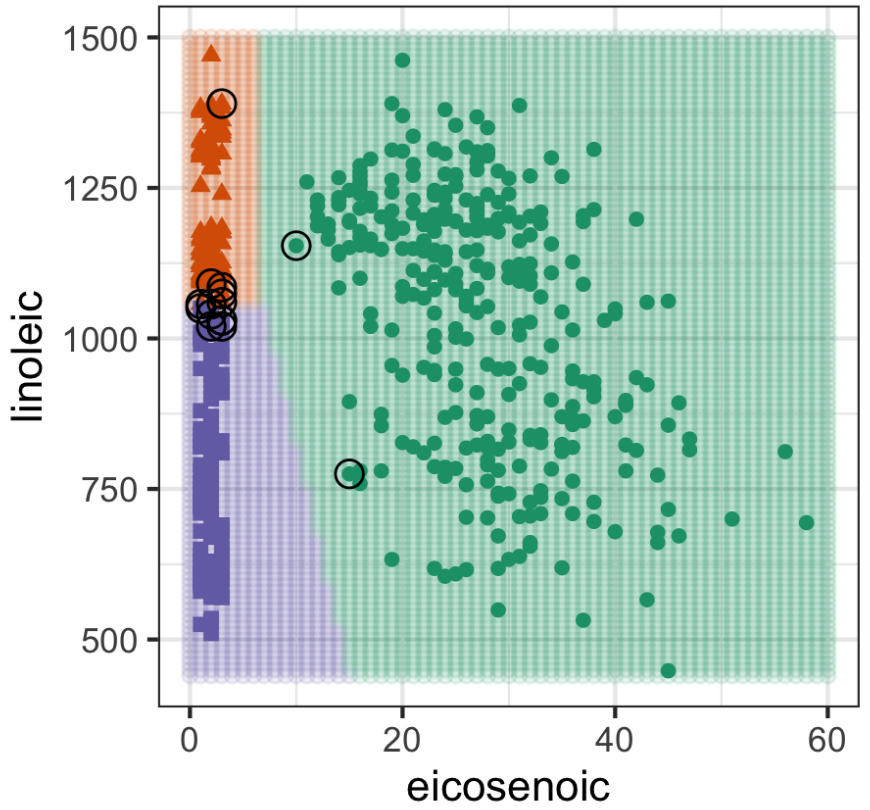
$$\{x : \beta_0 + x^T\beta = 0\}$$

where $\beta = \sum_{k=1}^{s}(\alpha_k y_k)x_k$ and s is the number of support vectors. Then the maximal margin hyperplane is fitted by finding β (ie α) to

maximise the margin $M = \frac{2}{||\beta||}$, subject to $\sum_{j=1}^{p}\beta_j^2 = 1$, and $y_i(x_i^T\beta + \beta_0) \geq M, i = 1, \ldots, n$.

Example: Support vectors

```
indx <- olive_svm$fit@SVindex
svs <- olive[indx,]
```



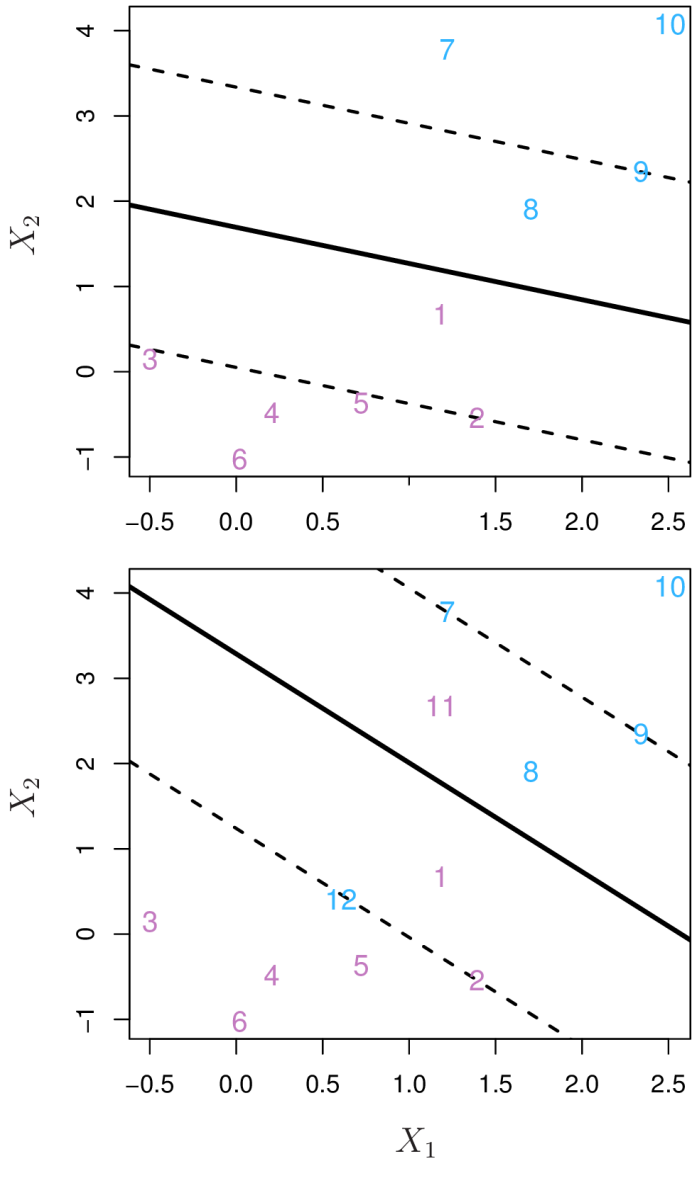
Non-separable case

The maximal margin classifier only works when we have perfect separability in our data.

What do we do if data is not perfectly separable by a hyperplane?

The support vector classifier allows points to either lie on the wrong side of the margin, or on the wrong side of the hyperplane altogether.

Right: ISLR Fig 9.6



Support vector classifier - optimisation

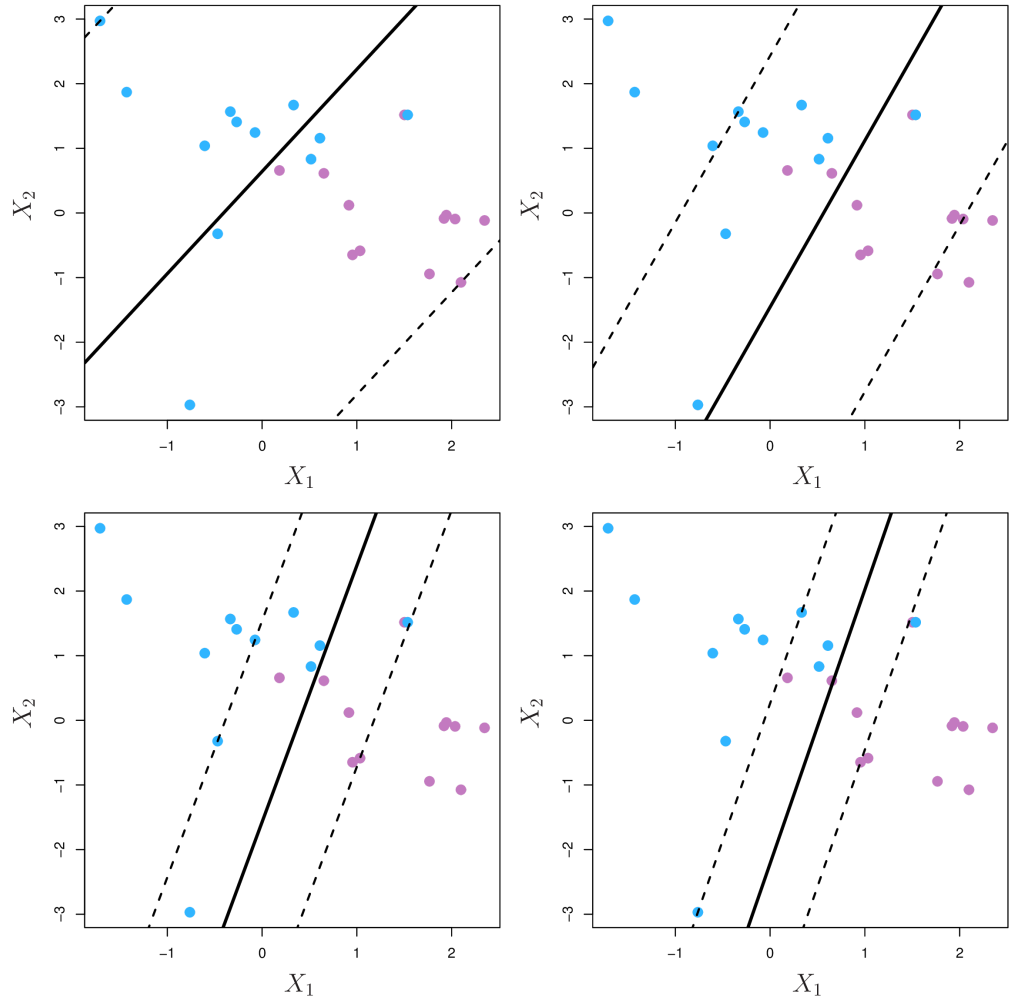
Maximise M, subject to $\sum_{j=1}^{p} \beta_j^2 = 1$, and $y_i(x_i^T\beta + \beta_0) \geq M(1 - \varepsilon_i), i = 1, \ldots, n$, AND $\varepsilon_i \geq 0, \sum_{i=1}^{n} \varepsilon_i \leq C$.

ε_i indicates where the ith observation is located and C is a nonnegative tuning parameter.

- $\varepsilon_i = 0$: correct side of the margin,
- $\varepsilon_i > 0$: wrong side of the margin (violation of the margin),
- $\varepsilon_i > 1$: wrong side of the hyperplane.

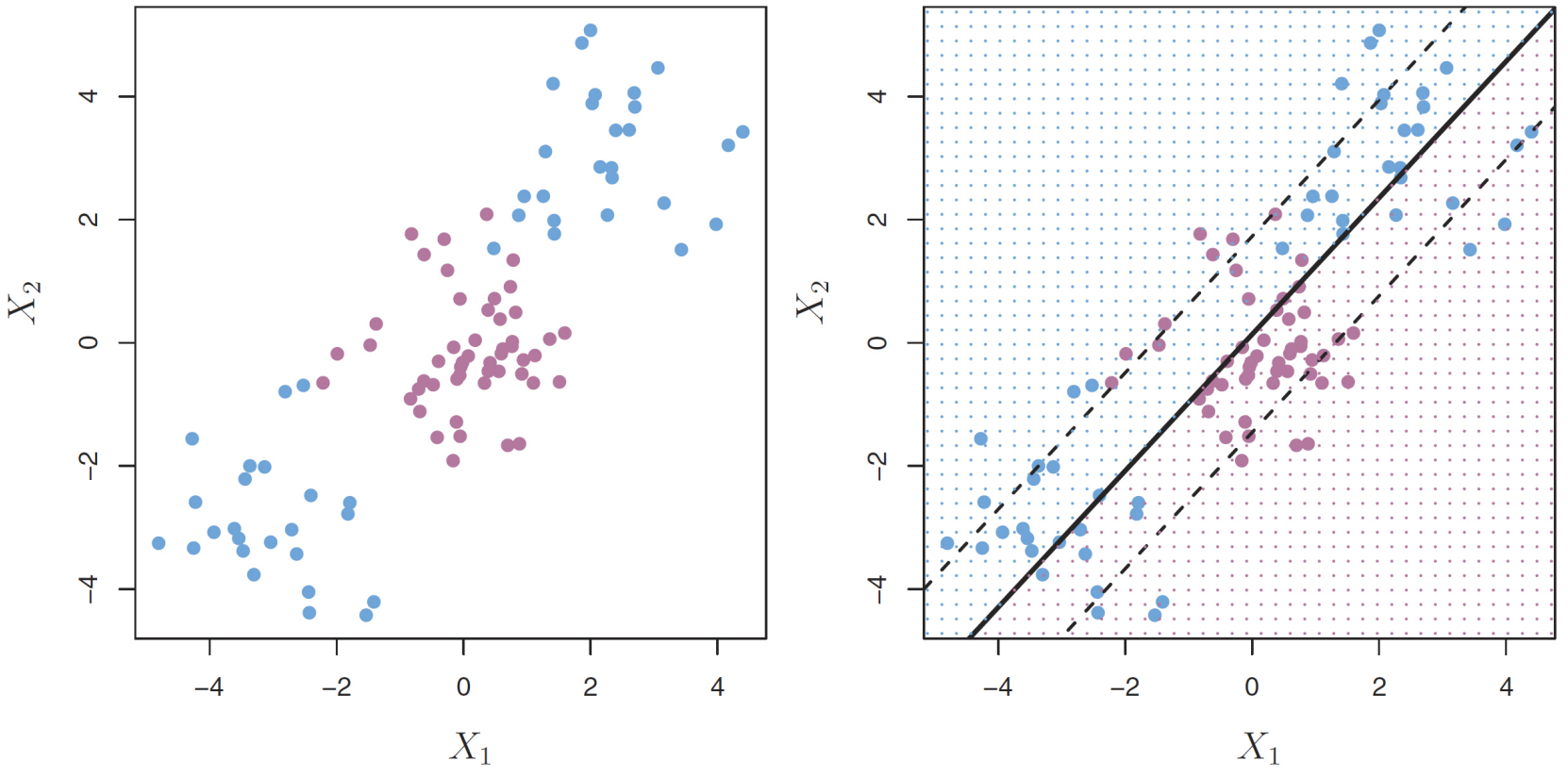
Non-separable case

Tuning parameter: decreasing the value of C



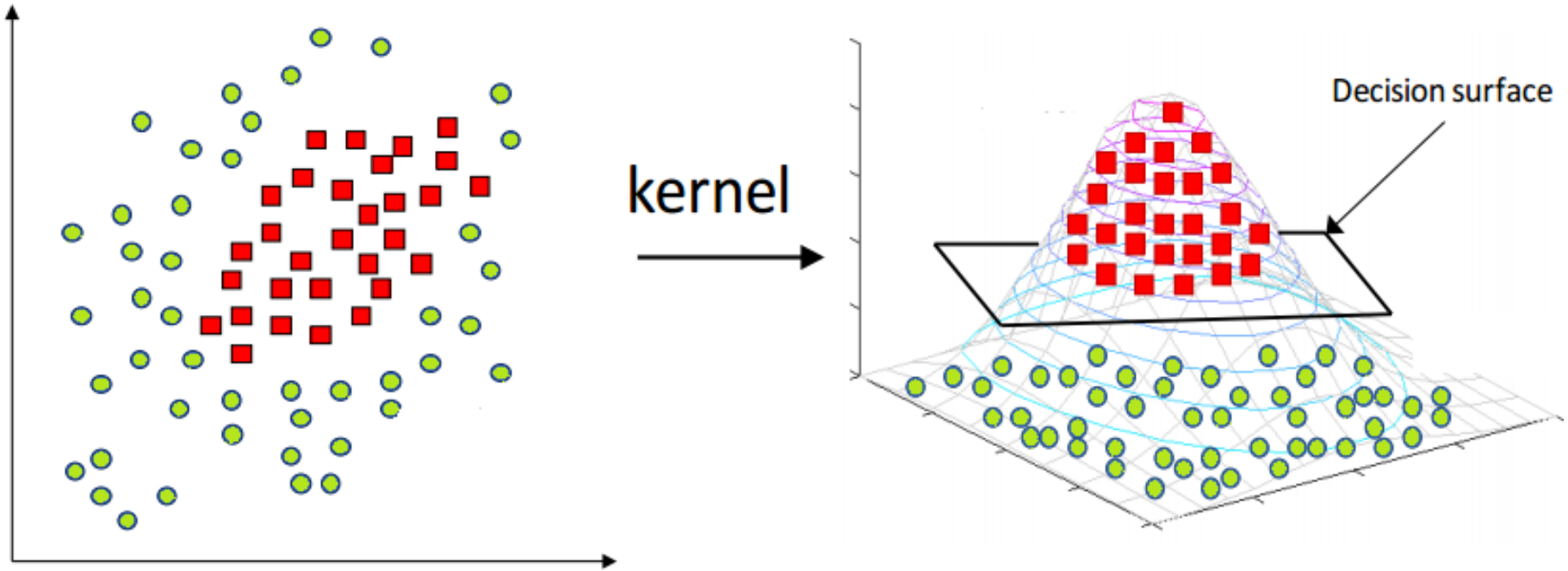
Nonlinear boundaries

The support vector classifier doesn't work well for nonlinear boundaries. What solution do we have?



Enlarging the feature space

Consider the following 2D non-linear classification problem. We can transform this to a linear problem separated by a maximal margin hyperplane by introducing an additional third dimension.



Source: Grace Zhang @zxr.nju

The inner product

Consider two p-vectors

$$\begin{aligned} & \mathbf{x} = (x_1, x_2, \ldots, x_p) \in \mathbb{R}^p \\ \text{and} \quad & \mathbf{y} = (y_1, y_2, \ldots, y_p) \in \mathbb{R}^p. \end{aligned}$$

The inner product is defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + x_2 y_2 + \cdots + x_p y_p = \sum_{j=1}^{p} x_j y_j$$

A linear measure of similarity, and allows geometric constrctions such as the maximal marginal hyperplane.

Kernel functions

A kernel function is an inner product of vectors mapped to a (higher dimensional) feature space $\mathcal{H} = \mathbb{R}^d, d > p$.

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle$$

$$\psi : \mathbb{R}^p \to \mathcal{H}$$

Non-linear measure of similarity, and allows geometric constructions in high dimensional space.

Examples of kernels

Standard kernels include:

$$\begin{aligned} \text{Linear} \quad & \mathcal{K}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle \\ \text{Polynomial} \quad & \mathcal{K}(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + 1)^d \\ \text{Radial} \quad & \mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp(-\gamma ||\mathbf{x} - \mathbf{y}||^2) \end{aligned}$$

Support Vector Machines

The kernel trick

The linear support vector classifier can be represented as follows:

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

We can generalise this by replacing the inner product with the kernel function as follows:

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

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

Name	Function
Polynomial	$(\|\boldsymbol{x}_i{}^T \boldsymbol{x}_j\|+d)^p$
Gaussian radial basis	$exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma^2)$
Sigmoid	$tanh(a\|\boldsymbol{x}_i{}^T \boldsymbol{x}_j\|+d)$

Your turn

Let $\mathbf{x}$ and $\mathbf{y}$ be vectors in $\mathbb{R}^2$. By expanding $\mathcal{K}(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^2$ show that this is equivalent to an inner product in $\mathcal{H} = \mathbb{R}^6$.

Remember: $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^{p} x_j y_j$.

Solution

$$
\begin{aligned}
\mathcal{K}(\mathbf{x}, \mathbf{y}) &= (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^2 \\
&= \left(1 + \sum_{j=1}^{2} x_j y_j \right)^2 \\
&= (1 + x_1 y_1 + x_2 y_2)^2 \\
&= (1 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2) \\
&= \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle
\end{aligned}
$$

where $\psi(\mathbf{x}) = (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2)$.

The kernel trick - why is it a trick?

We do not need to know what the high dimensional enlarged feature space $\mathcal{H}$ really looks like.

We just need to know the which kernel function is most appropriate as a measure of similarity.

The Support Vector Machine (SVM) is a maximal margin hyperplane in $\mathcal{H}$ built by using a kernel function in the low dimensional feature space $\mathbb{R}^p$.

Non-linear boundaries

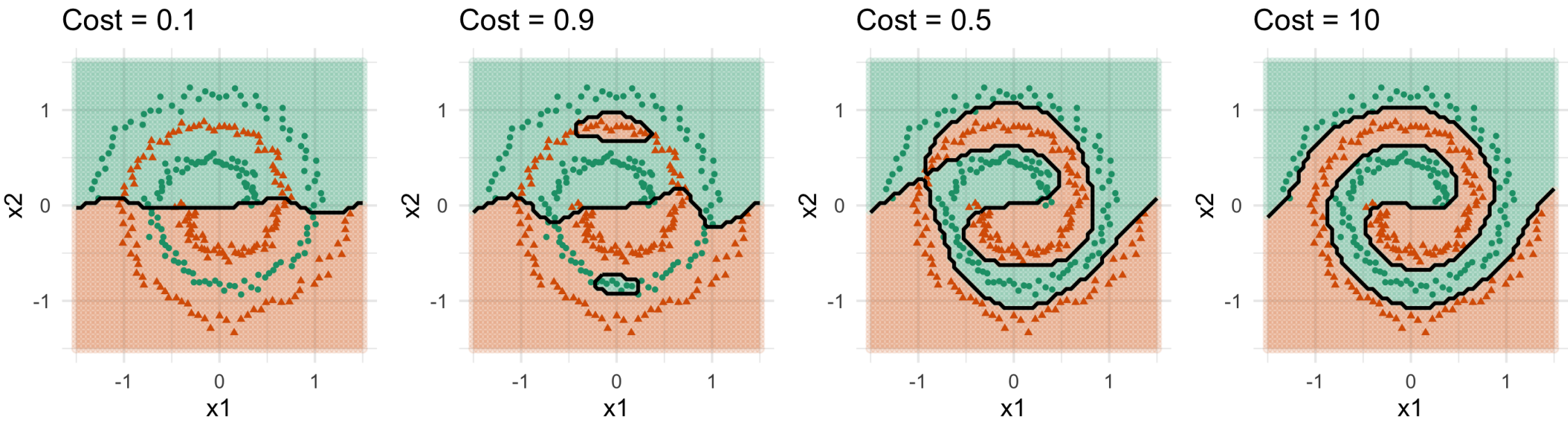
Polynomial and radial kernel SVMs

Non-linear boundaries

Italian olive oils: Regions 2, 3 (North and Sardinia)

Comparing decision boundaries

Increasing the value of `cost` in `svm`



SVM in high dimensions

Examining misclassifications and which points are selected to be support vectors

SVM in high dimensions

Examining boundaries

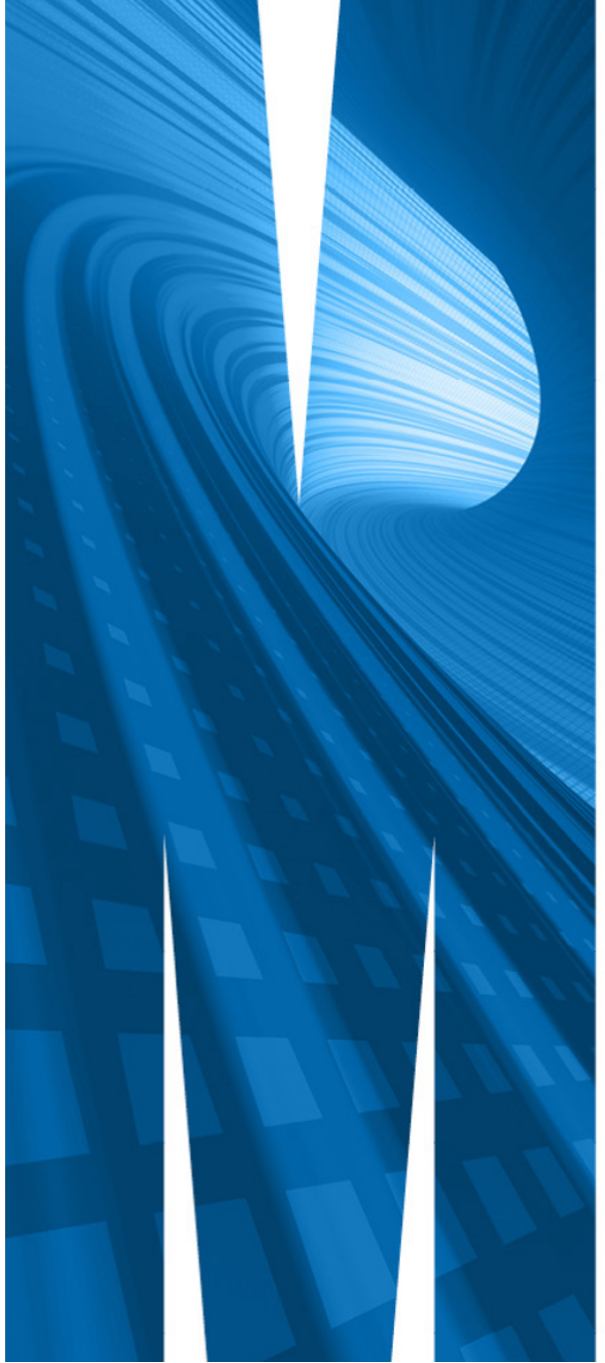
SVM in high dimensions

Boundaries of a radial kernel in 3D

SVM in high dimensions

Boundaries of a polynomial kernel in 5D

MONASH University



Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

ETC3250.Clayton-x@monash.edu

Week 7b