FIT5149: Applied Data Analysis
Support Vector Machines

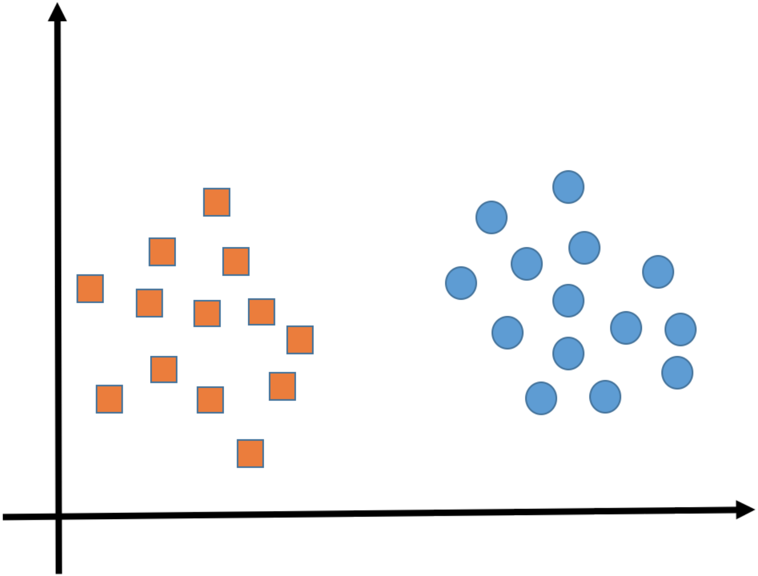
Dr. Du, Lan

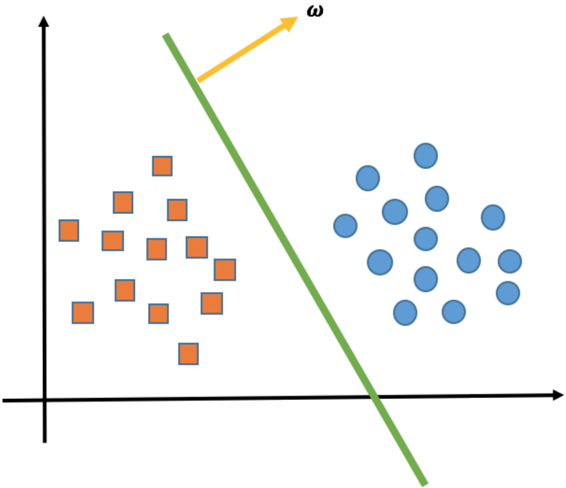
Faculty of Information Technology, Monash University, Australia

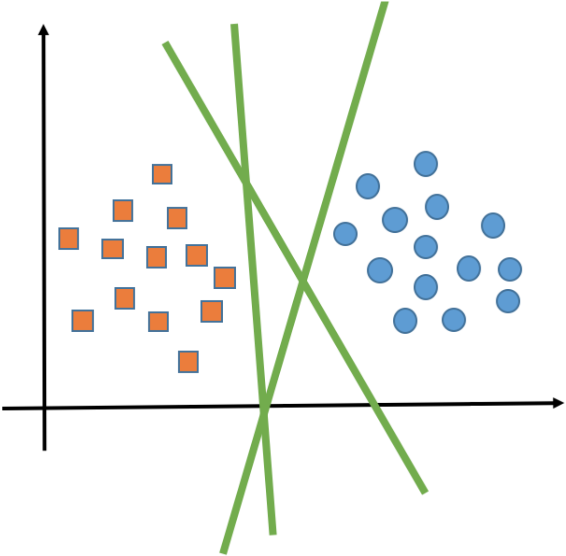
Week 9

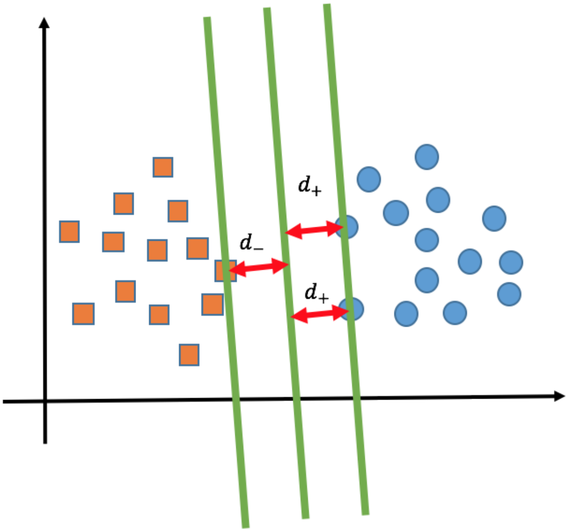
Outline

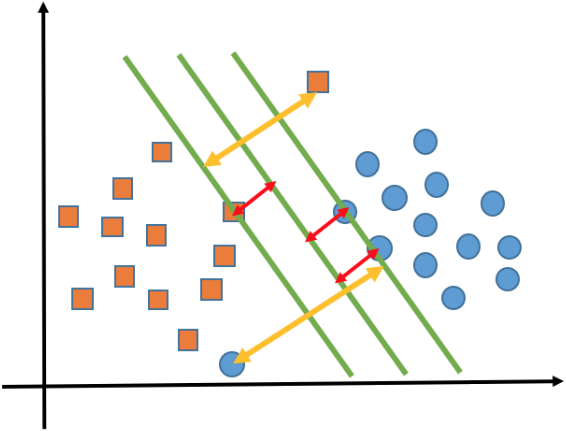
Introduction




ω




d_+
d_-
d_+



Maximal Margin Classifier

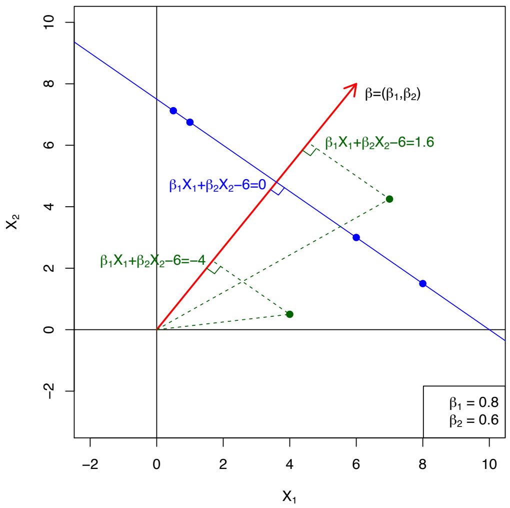
What is a Hyperplane?

- In a p-dimensional space, a hyperplane is a flat affine subspace of dimension $p-1$
  - In 2 dimensional space: a flat one-dimensional subspace is a line.
  - In 3 dimensional space: a flat two-dimensional subspace is a plane.
  - More than 3 dimensional space: No visualisation
- $\mathbf{X} = (X_1, X_2, \ldots, X_p) \in \mathbb{R}^p$ the p-dimensional hyperplane is

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

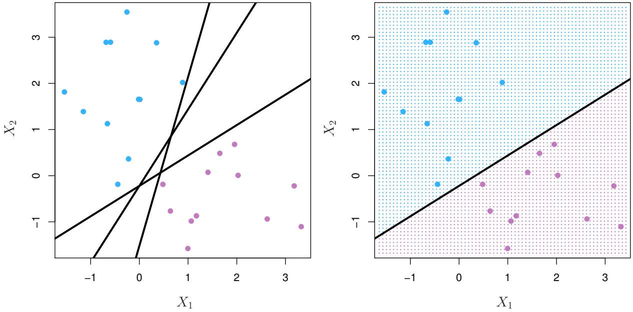
  - The vector $\boldsymbol{\beta} = \beta_1, \beta_2, \ldots, \beta_p$ is called the normal vector.
    - It is orthogonal to the surface of a hyperplane.

What is a Hyperplane?



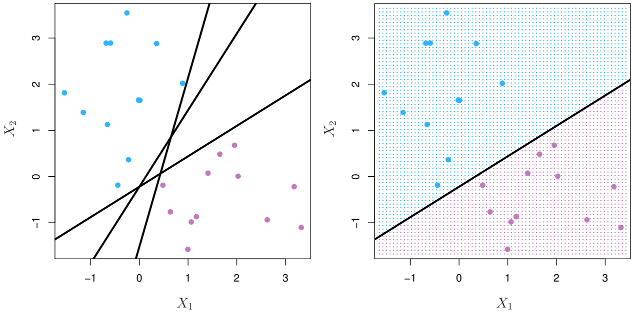
- The magnitude of $f(X) = \beta_1 X_1 + \beta_2 X_2 - 6$: the certainty about the class assignment of X.

Separating Hyperplane



- If $f(\mathbf{X}) = \beta_0 + \boldsymbol{\beta}^T\mathbf{X}$, then $f(\mathbf{X}) > 0$ for points on one side of the hyperplane, and $f(\mathbf{X}) < 0$ for points on the other.
- If we code the coloured points as $Y_i = +1$ for blue and $Y_i = -1$ for purple, then
  - $Y_i \times f(\mathbf{X}_i) > 0$ for all i,
  - $f(X) = 0$ defines a separating hyperplane.
- The magnitude of $f(\mathbf{X})$ can measure the confidence of the class assignment of $\mathbf{X}$.

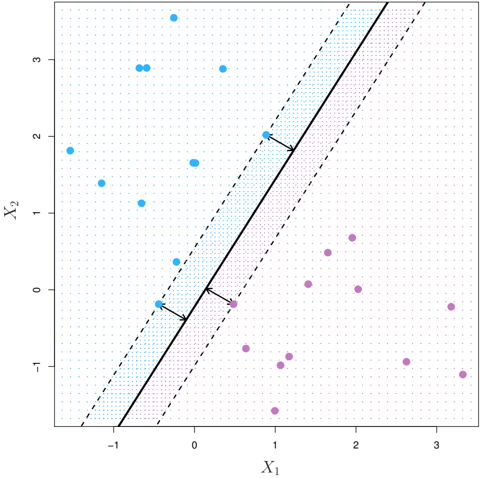
Separating Hyperplane



- Decide which of the infinite possible separating hyperplanes to use.

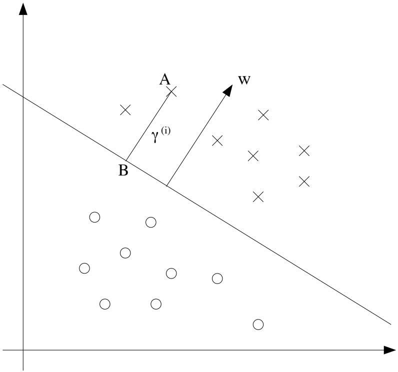
Maximal Margin Classifier

The idea: find the hyperplane that makes the biggest gap or margin between the two classes.



1. Compute the (perpendicular) distance from each training observation to a given separating hyperplane;
2. find the minimal distance from the observations to the hyperplane, known as margin.
3. **Target**: Maximise the margin.

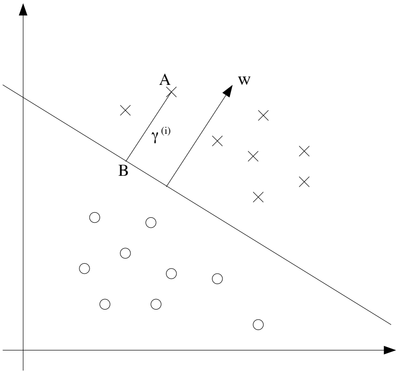
Functional margin vs geometric margin



Functional margin: $\hat{M}_i = y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0)$

- if $y_i = +1$, for the functional margin to be large (i.e., for our prediction to be confident and correct), then we need $\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0$ to be a large positive number.
- if $y_i = -1$, for the functional margin to be large, then we need $\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0$ to be a large negative number.

Functional margin vs geometric margin



Functional margin: $\hat{M}_i = y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0)$

- For a linear classifier with a sign function g, if we replace $\boldsymbol{\beta}$ with $2\boldsymbol{\beta}$ and β_0 with $2\beta_0$, then

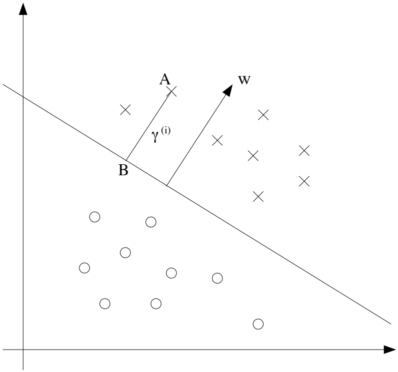
$$g(\boldsymbol{\beta}^T \mathrm{x}_i + \beta_0) = g(2\boldsymbol{\beta}^T \mathrm{x}_i + 2\beta_0)$$

  Conclusion: by exploiting our freedom to scale on the parameters, we can make the functional margin arbitrarily large without really changing anything meaningful.

- Given a set of training set $S = (\mathrm{x}_i, y_i); i = 1, \ldots, N$, the functional margin

$$\hat{M} = \min_{i=1,\ldots,N} \hat{M}_i$$

Functional margin vs geometric margin



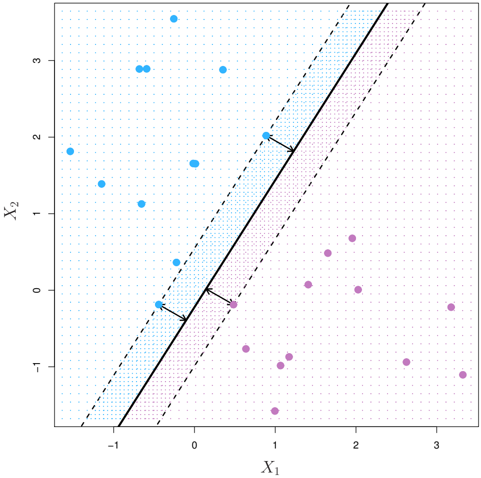
Geometric margin: $M_i = y_i \frac{\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0}{\|\boldsymbol{\beta}\|}$

- if $\| \boldsymbol{\beta} \| = 1$, the functional margin equals the geometric margin.
- Given a set of training set $S = (\mathbf{x}_i, y_i); i = 1, \ldots, N$, the geometric margin

$$M = \min_{i=1,\ldots,N} M_i$$

Construct Maximal Margin Classifier

MONASH University



Constrained optimisation problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\; M \quad (1)$$

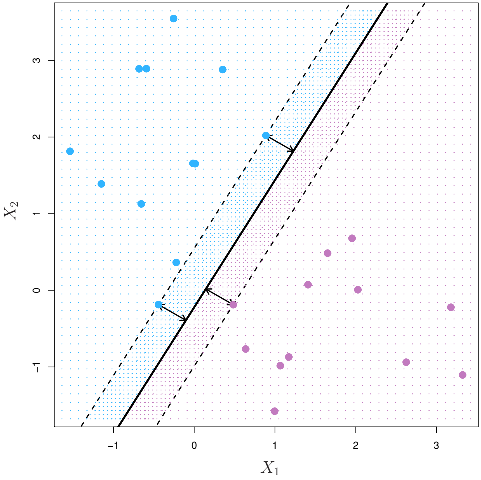
$$\text{subject to } \|\boldsymbol{\beta}\|^2 = 1 \quad (2)$$

$$y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq M$$

$$\text{for all } i = 1, \ldots, N \quad (3)$$

Now, we can use the Lagrange duality to solve the constrained optimisation problem.

Construct Maximal Margin Classifier



Constrained optimisation problem:

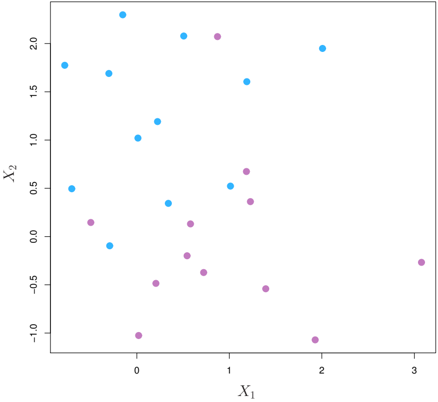
$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \frac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 \tag{1}$$

$$\text{s.t. } y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq 1,$$

$$\text{for all } i = 1, \ldots, N \tag{2}$$

Now, we can use the Lagrange duality to solve the constrained optimisation problem.

The Non-separable Case



- No separating hyperplane exists
- This often the case, unless $N < p$

Sensitivity to Noisy Data

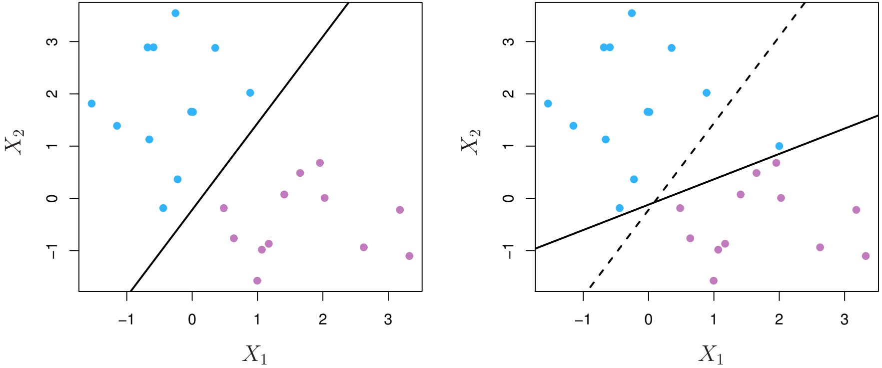


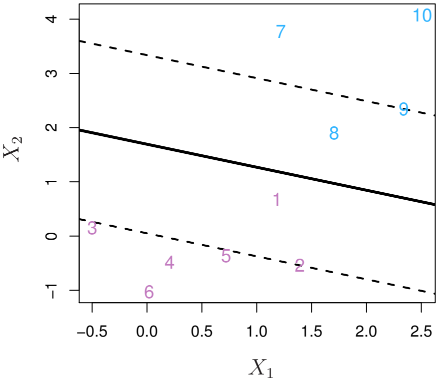
Figure: Right: An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.

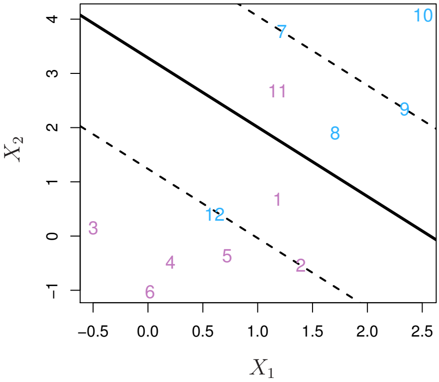
Support Vector Classifiers

Support vector classifier

MONASH University

The idea: use a soft margin





- Soft margin classifier: allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane

Construct the soft margin classifier

MONASH University

$$\max_{\beta_0, \boldsymbol{\beta}, \boldsymbol{\epsilon}} M \tag{3}$$

Subject to

$$\| \boldsymbol{\beta} \| = 1 \tag{4}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \quad \text{for all } i \tag{5}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{6}$$

where

- ϵ_i: tells us where the ith observation is located, relative to the hyperplane and relative to the margin.
  - $\epsilon_i = 0$: $\mathbf{x}_i$ is on the correct side of the margin
  - $\epsilon_i > 0$: $\mathbf{x}_i$ is on the wrong side of the margin
  - $\epsilon_i > 1$: $\mathbf{x}_i$ is on the wrong side of the hyperplane

Construct the soft margin classifier

MONASH University

$$\max_{\beta_0,\boldsymbol{\beta},\boldsymbol{\epsilon}} M \tag{3}$$

Subject to

$$\| \boldsymbol{\beta} \| = 1 \tag{4}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \quad \text{for all } i \tag{5}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{6}$$

where

- C: the tuning parameter
  - $C = 0$: no budget for violations to the margin.
  - $C > 0$: no more than C observations can be on the wrong side of the hyperplane.

Construct the soft margin classifier

MONASH University

$$\max_{\beta_0, \boldsymbol{\beta}, \boldsymbol{\epsilon}} M \tag{3}$$

Subject to

$$\| \boldsymbol{\beta} \| = 1 \tag{4}$$

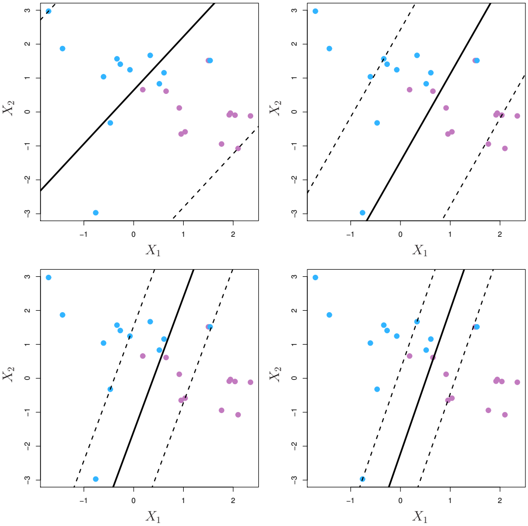
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \quad \text{for all } i \tag{5}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{6}$$

where

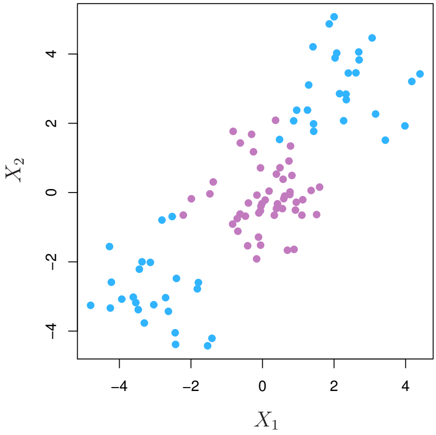
- C: the tuning parameter
  - the bias-variance trade-off
    - Large C: wide margin, many support vectors, highly fit to the data, high bias, low variance
    - Small C: narrow margin, few support vectors, fit the data less hard, low bias, high variance

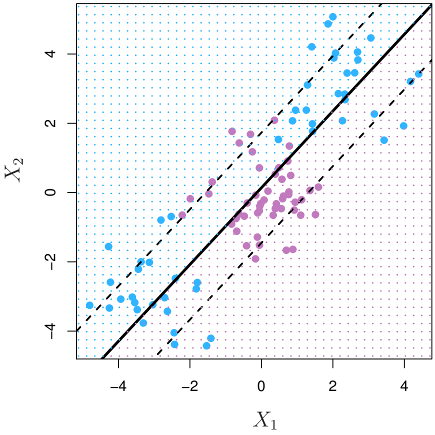
C is a regularisation parameter



- 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.
- The support vector classifier's decision rule is based only on the support vectors

Linear boundary can fail





What to do?

Support Vector Machines

Feature expansion

- Enlarge the space of features by including transformations;

$$X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2$$

- Fit a support-vector classifier in the enlarged space results in non-linear decision boundaries in the original space.
- Example: Suppose we use $(X_1,\ X_2,\ X_1^2,\ X_2^2,\ X_1X_2)$ instead of just $(X_1,\ X_2)$. Then the decision boundary would be of the following form

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

Feature expansion

MONASH University

- Then the problem is

$$\max_{\beta_0,\boldsymbol{\beta},\boldsymbol{\epsilon}} M \tag{7}$$

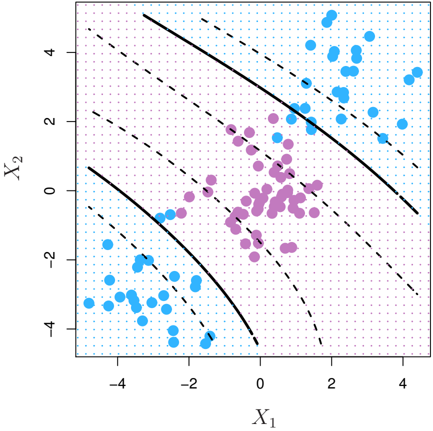
Subject to

$$y_i\left(\beta_0+\sum_{j=1}^{p}\beta_{j1}x_{ij}+\sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right)\geq M(1-\epsilon_i) \quad \text{for all } i \tag{8}$$

$$\epsilon_i\geq 0,\quad \sum_{i=1}^{n}\epsilon_i\leq C,\quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2=1 \tag{9}$$

$$\tag{10}$$

Cubic polynomial



- A basis expansion of cubic polynomials
- From 2 variables to 9 variables
- The support-vector classifier in the enlarged space solves the problem in the lower-dimensional space

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 + \beta_6 X_1^3 + \beta_7 X_2^3 + \beta_8 X_1 X_2^2 + \beta_9 X_1^2 X_2$$

Support Vector Machine (SVM)

MONASH University

- The use of kernels: a more elegant and controlled way to introduce nonlinearities in support-vector classifiers.
  - An efficient computational approach.
- So,
  - What are the Kernels?
  - How do they work?

Learning SVM: Minimisation

The optimisation problem for finding the optimal margin classifier

$$\min_{\boldsymbol{\beta},\beta_0} \frac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 \tag{11}$$

$$\text{Subject to } y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) \geq 1, \; i = 1, \ldots, N \tag{12}$$

Reformulate the constraints as

$$g_i(\boldsymbol{\beta}) = -y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) + 1 \leq 0$$

Construct the Lagrangian

$$\mathcal{L}(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) = \frac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) - 1 \right]$$

Learning SVM: Minimisation

$$\mathcal{L}(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) = \frac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0) - 1 \right]$$

- Set the derivative of $\mathcal{L}$ with respect to $\boldsymbol{\beta}$ to zero, we have

$$\boldsymbol{\beta} = \sum_{i}^{N} \alpha_i y_i \mathbf{x}_i \tag{13}$$

- Set the derivative of $\mathcal{L}$ with respect to β_0 to zero, we have

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{14}$$

Learning SVM: Maximization

MONASH University

$$\mathcal{L}(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j$$

- The dual optimisation problem:

$$\max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) \tag{15}$$

subject to

$$\alpha_i \geq 0, \quad i = 1, \ldots, N \tag{16}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{17}$$

SVM: Inner product

MONASH University

- The inner product from

$$\boldsymbol{\beta}^T x + \beta_0 = \left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i\right)^T \mathbf{x} + \beta_0 \tag{18}$$

$$= \sum_{i=1}^{N} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + \beta_0 \tag{19}$$

- To estimate the parameter $\boldsymbol{\alpha}$ and β_0, all we need are the $\binom{n}{2}$ inner products.
- It turns out that most of the α_i can be zero:

$$\sum_{i \in \mathcal{S}} \hat{\alpha}_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + \beta_0$$

where $\mathcal{S}$ is the support set of indices i such that $\hat{\alpha}_i > 0$.

SVM: Kernels

$$\langle \mathbf{x}_i, \mathbf{x}_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

can be generalised to

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_{i'})$$

where, for instance,

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

Kernels: implicit feature mapping

MONASH University

Suppose $\mathbf{x} \in \mathbb{R}^p$

$$
\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_{i'}) \\
&= \left(\mathbf{x}_i^T \mathbf{x}_{i'}\right)^2 && (20) \\
&= \left(\sum_{j=1}^{p} x_{ij} x_{i'j}\right)\left(\sum_{j'=1}^{p} x_{ij'} x_{i'j'}\right) && (21) \\
&= \sum_{j=1}^{p}\sum_{j'=1}^{p} x_{ij} x_{ij'} x_{i'j} x_{i'j'} && (22) \\
&= \sum_{j,j'=1}^{p} (x_{ij} x_{ij'})(x_{i'j} x_{i'j'}) && (23)
\end{aligned}
$$

If $p = 3$

$$
\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}
$$

- Calculating the high-dimensional $\phi(\mathbf{x})$ requires $\mathcal{O}(p^2)$ time,
- Finding $K(\mathbf{x}_i, \mathbf{x}_{i'})$ takes only $\mathcal{O}(p)$ time

Kernels: Polynomial kernel

MONASH University

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \left(\mathbf{x}_i^T \mathbf{x}_{i'} + c\right)^d \tag{24}$$

computes the inner-product needed for d dimensional polynomials.

- In general, the kernel above corresponds to a feature mapping to

$$\binom{p+d}{d}$$

  feature space, corresponding to all monomials of the form $x_{i1}x_{i2}\ldots x_{ip}$ that are up to order d.

Kernels: Polynomial kernel

If $d = 2$,

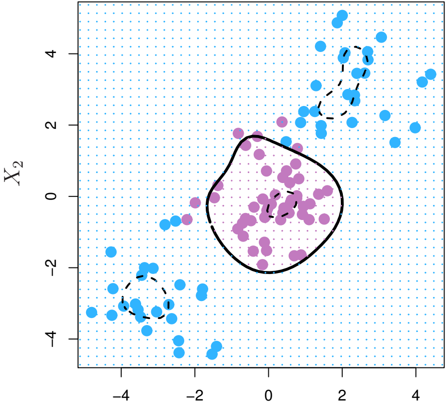
$$
\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_{i'}) \\
&= \left(\mathbf{x}_i^T \mathbf{x}_{i'} + c\right)^2 && (25) \\
&= \sum_{j,j'=1}^{p} (x_{ij} x_{ij'})(x_{i'j} x_{i'j'}) \\
&\quad + \sum_{j=1}^{p} (\sqrt{2c} x_{ij})(\sqrt{2c} x_{i'j}) \\
&\quad + c^2 && (26)
\end{aligned}
$$

If $p = 3$,

$$
\phi(x) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_3 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \\ \sqrt{2c} x_1 \\ \sqrt{2c} x_2 \\ \sqrt{2c} x_3 \\ c \end{bmatrix}
$$

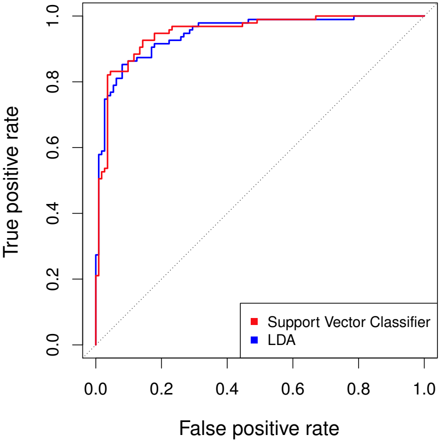
SVM: Radial Kernel

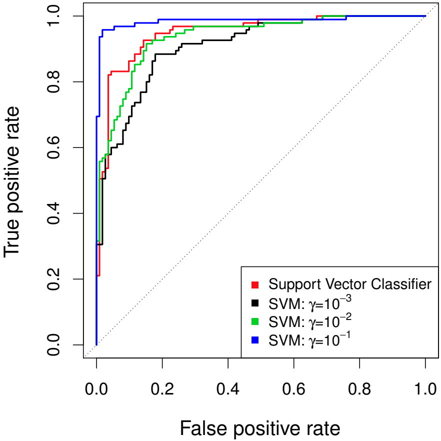
$$K(\mathbf{x}, \mathbf{x}_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$



- Implicit feature space: very high dimensional.
- Controls variance by squashing down most dimensions severely.

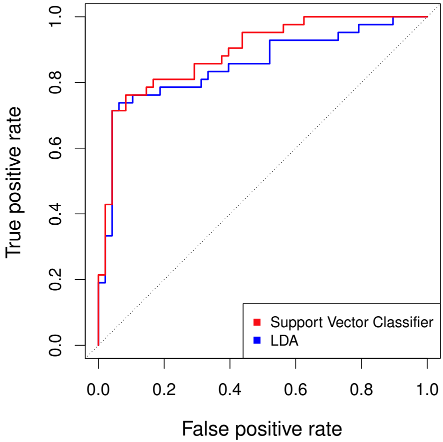
SVM: Heart Data

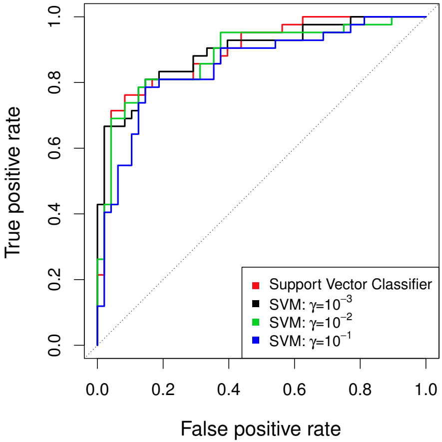




ROC curve is obtained by changing the threshold 0 to threshold t in $f(X) > t$, and recording false positive and true positive rates as t varies. Here we see ROC curves on training data.

SVM: Heart Data





More than 2 classes:

- One-Versus-All Classification
  - Fit K different 2-class SVM classifiers $\hat{f}_k(x)$, $k = 1, ..., K$; each class versus the rest. Classify $x*$ to the class for which $\hat{f}_K(x*)$ is largest.
- One-Versus-One Classification
  - Fit all $\binom{K}{2}$ pairwise classifiers $\hat{f}_{k,l}(x)$. Classify $x*$ to the class that wins the most pairwise competitions.
- Which to choose? If K is not too large, use OVO.

Summary

- SVM
  - "Support Vector Machines", Chapter 9 of "Introduction to Statistical Learning", 6th edition
- Acknowledgement:
  - Figures in this presentation were taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
  - Some of the slides are reproduced based on the slides from T. Hastie and R. Tibshirani
  - Some of the deduction formulas are adapted from Andrew Ng's note on SVM.