MA5832: Data Mining & Machine Learning

Collaborate Week 4: Support Vector Machines (SVM)

Martha Cooper, PhD

JCU Masters of Data Science

2021-02-04

Housekeeping

- Collaborates = Thursdays 6-7:30pm

For my Collaborate Sessions, you can get the slides & R code for each week on Github:

https://github.com/MarthaCooper/MA8532

Today's Goals

- Support Vector Machines (SVM)
- Assessment 1 Common Mistakes/Q&A

Support Vector Machines (SVM)

SVM

- Support Vector Machines
  - Classification using Hyperplanes
  - Maximal Margin Classifier (linear decision boundary)
  - Support Vector Classifiers (*linear* decision boundary, *soft margin*)
  - Support Vector Machine (supports *non-linear decision boundary*)

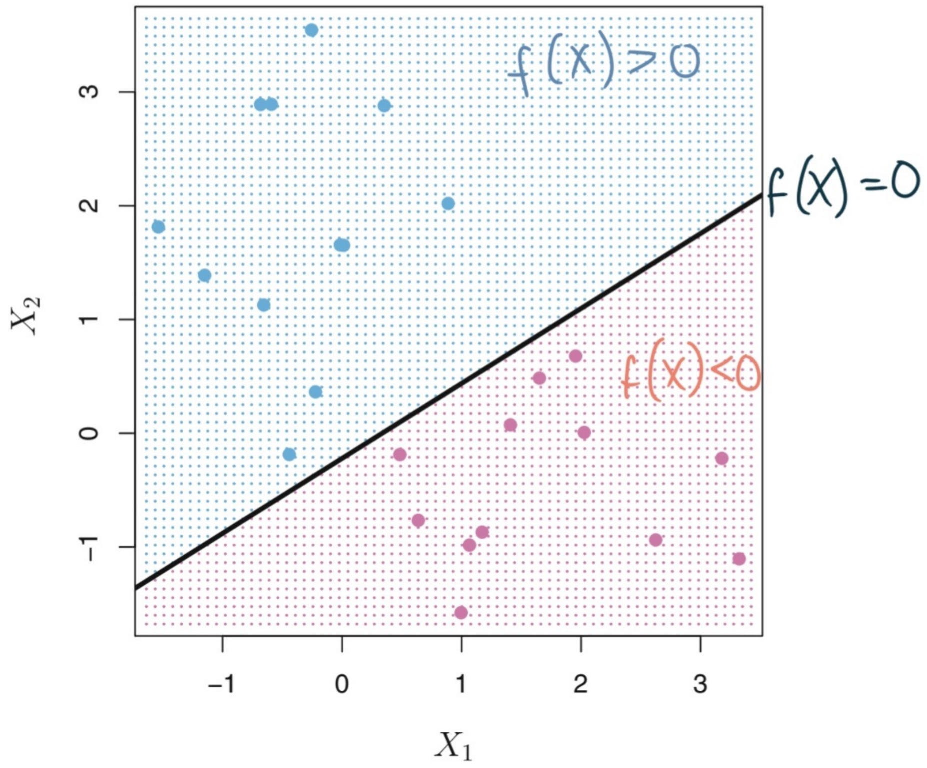
Classification using hyperplanes

- Classification setting - 2 groups
- Try to find a hyperplane that separates the classes in feature space

Classification using hyperplanes

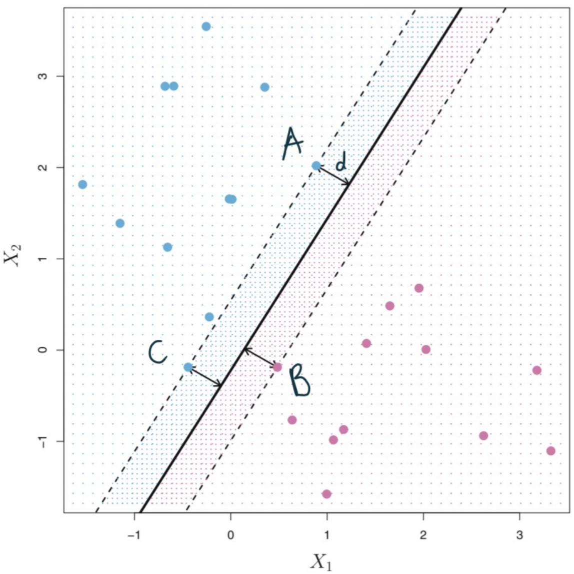
- A hyperplane in p dimensions is a set of points $(x_1, \ldots X_p)$ that satisfy a linear equation $\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p = 0$
- $p = 2$?
- Classification using hyperplanes
  - $f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$
  - $f(X) > 0$ - all points of one side of the hyperplane
  - $f(X) < 0$ - all points on the other side of the hyperplane
  - $f(X) = 0$ defines the separating hyperplane

Classification using hyperplanes



Maximal margin classifier

- Amongst all separating hyperplanes, it is the one for which the margin is largest
- It has the farthest minimum distance to the training observations.
- Points A, B, C that lie on the margins are called support vectors
- The distance between these points and the hyperplane is called d. The Margin, M , is twice the absolute distance of d.



Maximal margin classifier

- How to find the optimal hyperplane for classification where $y \in \{1, -1\}$?

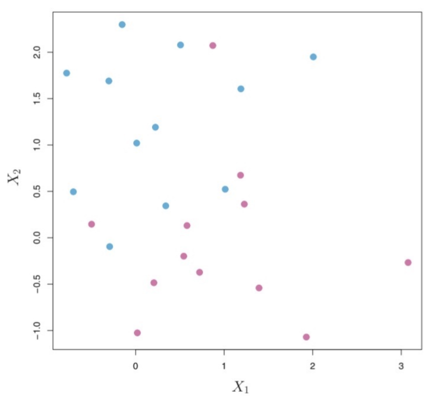
Optimisation problem:

- $maximise_{\beta_0,\beta_1,\ldots,\beta_p} \quad M$
  - Maximise the width of the margin
- $subject\,to \quad \sum_j^p \beta_j^2 = 1$
  - Find a unique hyperplane, and define the perpendicular distance between any point, i, and the hyperplane
- $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M \quad \forall\, i = 1, \ldots, n$
  - Make sure that each observation will be on the correct side of the margin

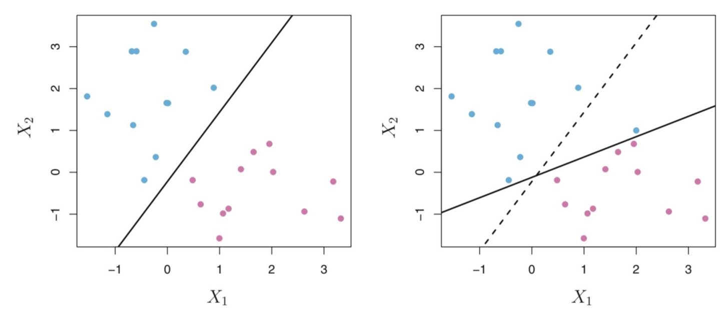
Support Vector Classifier & Soft Margins

- Sometimes the data are separable, but noisy.
- Sometimes the data aren't perfectly separable
- We might want to use a hyperplane that *almost* separates the classes, called a soft margin
- This extension of the *Maximal Margin Classifier* is called a Support Vector Classifier

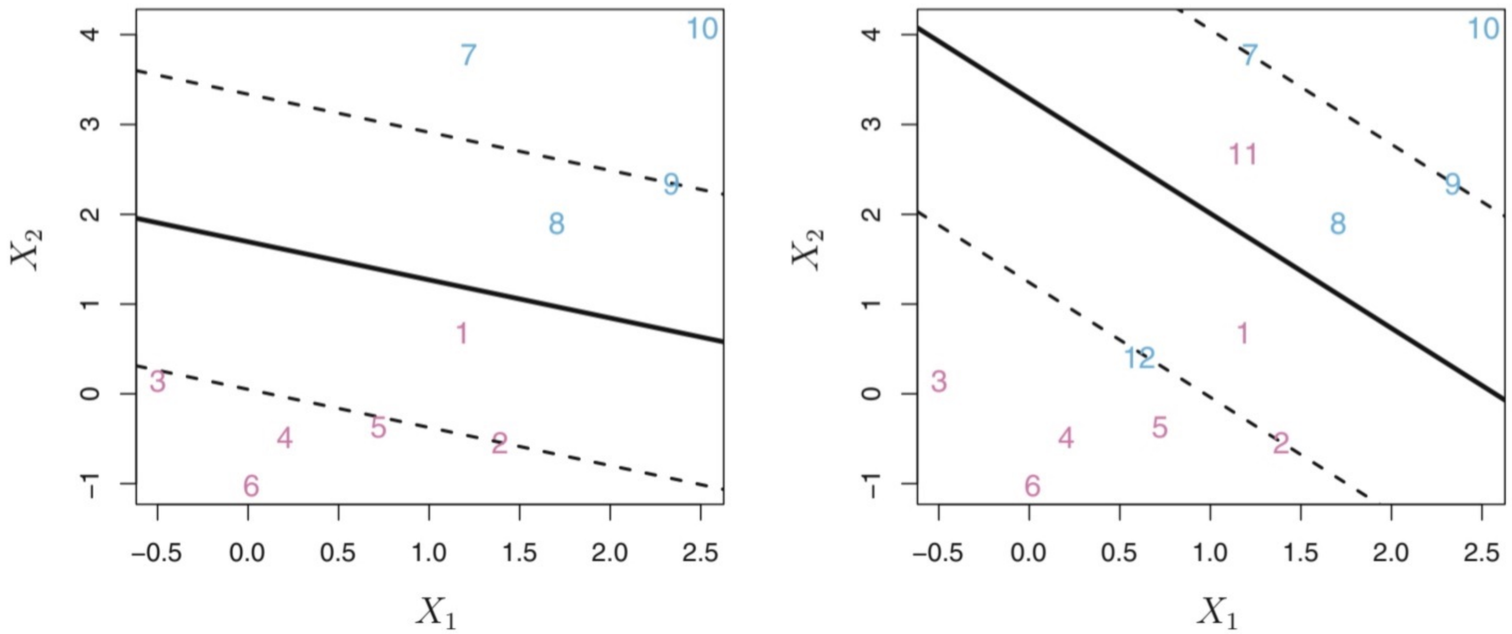
Non-Separable



Noisy



Support Vector Classifier & Soft Margins



Support Vector Classifier & Soft Margins

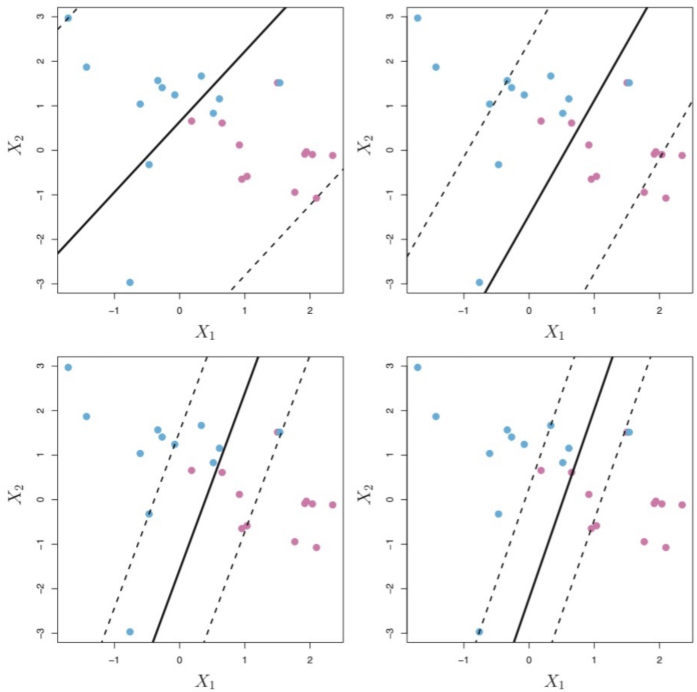
- How to find the optimal soft margin hyperplane for classification where $y \in \{1, -1\}$?

Optimisation problem

- $maximise_{\beta_0,\beta_1,\ldots,\beta_p} \quad M$
- $subject\,to \quad \sum_j^p \beta_j^2 = 1$
- $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^n \epsilon_i \leq C$
- where C is the permissible misclassification (a tuning parameter)
- $\epsilon_i, \ldots, \epsilon_n$ are slack variables - they allow observations to be on the wrong side of the margin or hyperplane

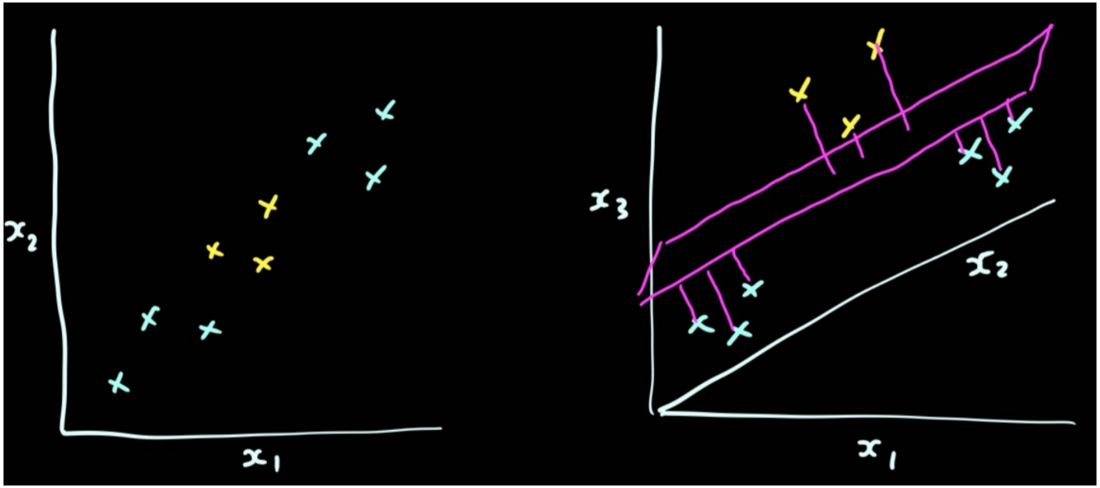
Choosing C

- If $C = 0$, there is no budget for observations to be misclassified
- C controls the Bias-Variance trade-off, and we choose it by CV
  - C is small = low bias, high variance
  - C is large = high bias, low variance



Non-Linear Classification with the Support Vector Machine

- Sometimes a linear boundary won't work regardless of the value of C
- What if we enlarged the feature space *i.e.* added extra dimensions?



- The *Support Vector Machine* uses kernals to enlarge the feature space, without actually performing any transformations - Kernal Trick

Inner Products

- The solution to the support vector classifier only involves the *inner products* of the observations.
- Inner products are defined by:
- $\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$
- The linear support vector classifier can be represented as:
- $f(x) = \beta_0 + \sum_{n=1}^{n} \alpha_i \langle x, x_i \rangle$

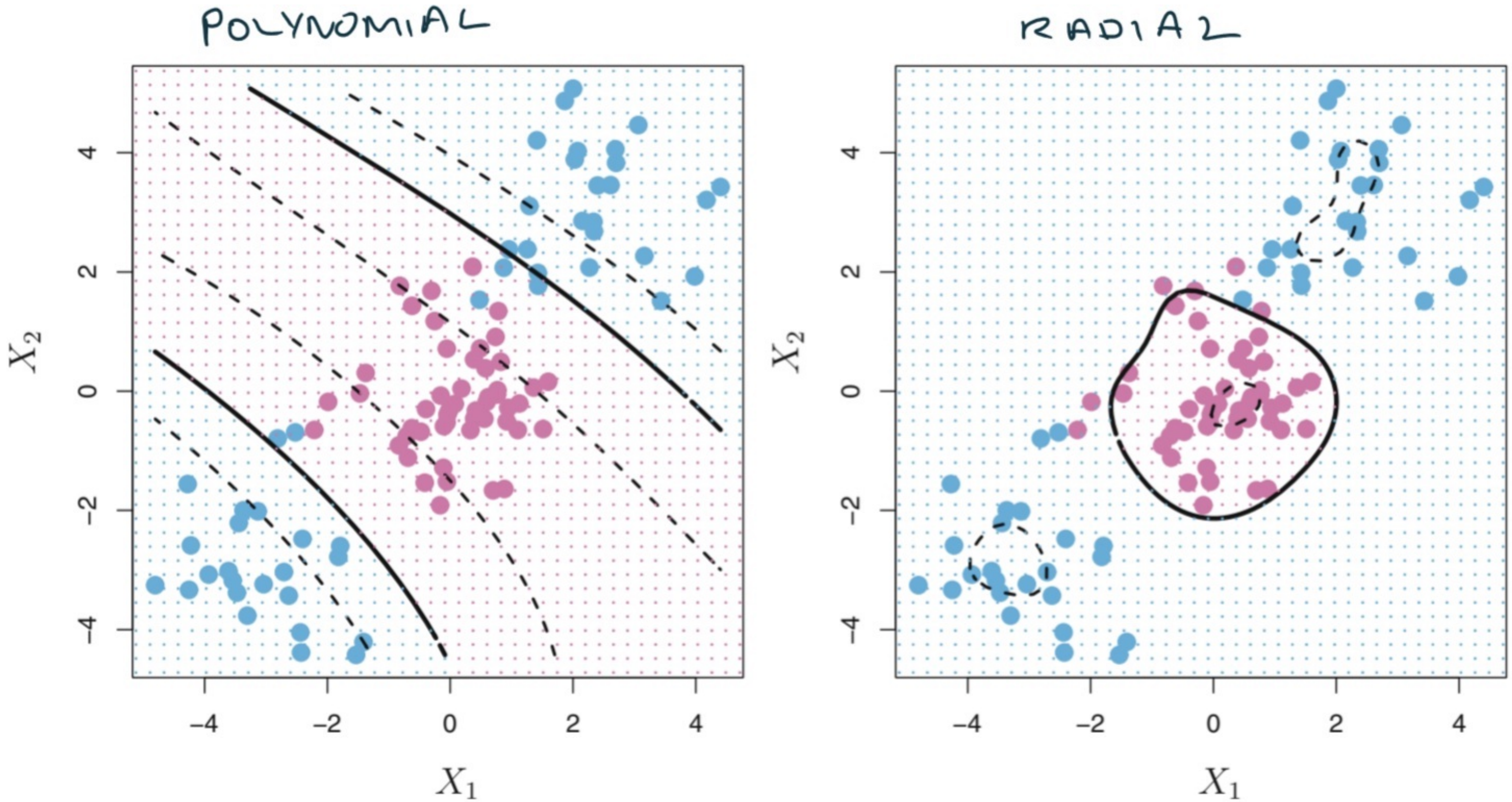
Inner products & support vectors

- The linear support vector classifier can be represented as:
- $f(x) = \beta_0 + \sum_{n=1}^{n} \alpha_i \langle x, x_i \rangle$
- The hyperplane only depends on the support vectors. If S is a collection of the support vectors then:
- $f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle$
- Summary: in representing the linear classifier $f(x)$ and in computing its coefficients, all we need are the inner products.

The Kernal Trick

- K is a function we will refer to as a Kernal. The non linear support vector classifier can be presented as
- $f(x) = \beta_0 + \sum_{i \in S} \alpha_i K\langle x, x_i \rangle$
- Kernals for non-linear Support Vector Machines
  - Polynomial kernal $K\langle x, x_{i'} \rangle = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$
  - Radial kernal $K\langle x, x_{i'} \rangle = exp(-\gamma \sum_{j=1}^{p} (x_{ij} x_{i'j})^2)$ where γ is a positive constant.

SVM with polynomial and radial kernals



SVM in R

Can you build a machine learning model to accurately predict whether or not the patients in the dataset have diabetes or not?

pregnant	Number of times pregnant
glucose	Plasma glucose concentration (glucose tolerance test)
pressure	Diastolic blood pressure (mm Hg)
triceps	Triceps skin fold thickness (mm)
insulin	2-Hour serum insulin (mu U/ml)
mass	Body mass index (weight in kg/(height in m)\^2)
pedigree	Diabetes pedigree function
age	Age (years)
diabetes	Class variable (test for diabetes)

Discuss Approaches...

To scale or not to scale?

- Tree based methods
- SVM
- Neural Networks

To scale or not to scale?

- Tree based methods
- SVM
- Neural Networks

Optimisation problems = error inflation if variables on larger scales

SVM in R

Check .Rmd, not all code on slide...

```r
library(caret) # to fit svm
library(mlbench) # to obtain data
data("PimaIndiansDiabetes2", package = "mlbench") #diabetes dataset

# Exploratory Analysis
###Checks
#####- missing values
#####- distribution/skewness
#####- outliers
#####- diagnostic plots
head(PimaIndiansDiabetes2)
summary(PimaIndiansDiabetes2)
df <- na.omit(PimaIndiansDiabetes2)
summary(df)

# Defining test and training data
##### Why?
set.seed(6)
test_index <- createDataPartition(df$diabetes, p = 0.3, list = F)
traindat  <- df[-test_index,]
testdat <- df[test_index,]
```

Extra reading

- Chapter 9 ISLR
- Chapters 12 ESL

Extra watching

- The linear algebra we missed: MIT Learning: Support Vector Machines
- General Intro: StatQuest: SVM

References

- Chapter 9 ISLR

Slides

- xaringhan, xaringanthemer, remark.js, knitr, R Markdown