Predictive Analytics – Session 4

Machine Learning Tools: Unsupervised Learning
Ensembles Methods

Associate Professor Ole Maneesoonthorn

Associate Professor in Econometrics and Statistics

Melbourne Business School

O.Maneesoonthorn@mbs.edu

What predictive tools are available?

The Predictive Tools

Ensembles Methods

The basic idea:

- There are a number of candidate base models
- Why choose one? Why not combine them all?
- Ensembles = a collection of base models used for predictive purposes

The Predictive Tools

Ensembles Methods

Also known as “Forecast Combination”

- Plain vanilla – linear combination with equal weights to all models
- Optimal linear combination – some optimization involved
- Aim: minimize the final prediction error

The Predictive Tools

Ensembles Methods

We will explore three machine learning ensembles:

- Bagging
- Boosting
- Random forest
- All three can handle classification and numerical variables

The Predictive Tools

Ensembles Methods – Bagging

- Train the base models on “subset” of the training data
- Subset obtained by bootstrapping process
- Bootstrapping – random sampling with replacement of the data
- If we do this B times, we B functions for prediction $\hat{f_i}(x)$ for $i = 1,2, \ldots, B$

The Predictive Tools

Ensembles Methods – Bagging

- Bagged prediction is the simple average of all predictions

$$\text{Bagged prediction} = \frac{1}{B}\sum_{i=1}^{B}\hat{f}_i(x)$$

- Why do we do this?

- Reduce variance of predictions by averaging over bootstrapped samples

- Smaller variance = stability of prediction

The Predictive Tools

Ensembles Methods – Random Forest

- Forest = a collection of trees
- Similar idea to bagging – bootstrap samples
- But the base learner is always of a tree structure
- There is also a random component in the tree construction

The Predictive Tools

Ensembles Methods – Random Forest

- For each bootstrap sample, construct a tree. At the terminal node:

  - Select a set of "m" predictors at random (m<p, with p=# of inputs)
  - Pick the best variable amongst the "m" variables to split
  - Split the node into two children nodes
  - Repeat until you get the desired minimum number of nodes
  - Return the trained tree T_b

- Random forest prediction = $\frac{1}{B}\sum_{i=1}^{B} T_i(x)$

The Predictive Tools

Ensembles Methods – Boosting

- For problems when you have a collection of “weak learners”
- That is, models only produce predictions that are slightly better than random guesses
- The algorithm is designed to “boost” the performance of these weak learners
- Typically applied to a regression tree structure

The Predictive Tools

Ensembles Methods – Boosting

The basic idea of Adaptive Boosting (AdaBoost)

- Train all training data points with the “weak learners” models
- Work out the errors associated with each data point
- Give more weight to the data point with larger errors
- Train the “weak learners” again
- Repeat this process many (M) times
- Aggregate the prediction across the M sets of predictions using a weight function

The Predictive Tools

Ensembles Methods – Boosting

- Extension to AdaBoost → Gradient Boosting

- Can handle a variety of loss functions

- Applicable to regression, ranked, classification variables

The Predictive Tools

Ensembles Methods – Boosting

- Gradient Boosting (GB) for Regression – key option

- Loss functions
  - Squared loss – conventional, but outliers are given too much weight
  - Absolute loss (Laplace) – downplay outliers
  - Huber loss – identical to squared loss in some region, but still downplay outliers

The Predictive Tools

Ensembles Methods – Boosting

- Gradient Boosting – importance of loss function
- The “gradient” in GB is the change of loss if the prediction changes
- The boosting algorithm focuses on adapting the “weak learners” to observations with large gradients
    - i.e. observations that need improvement
- So the choice of loss function is critical

The Predictive Tools

Ensembles Methods – Boosting

- Gradient Boosting for Regression – the learning rate ("nu" or ν)
  - "nu" is a number between 0 and 1
- Defines how fast the algorithm adapts to the gradient
  - Large "nu" = faster learner, smaller number of steps/models in the ensembles
  - Large "nu" → risk of over-correction
  - Smaller "nu" = slower learner, larger number of steps/models in the ensembles
  - Mboost package default "nu" = 0.1
  - General rule of thumb: between 0.1 and 0.3

The Predictive Tools

Support Vector Machine (SVM)

The basic idea:

- A partitioning algorithm
- Searching for **a function** that partitions the data
- The function needs a buffer zone around it, defined by “support vectors”
    - Vectors are basically collection of numbers
- Buffer zone should be as large as possible – well defined partitions

The Predictive Tools

Support Vector Machine (SVM)

The basic idea:

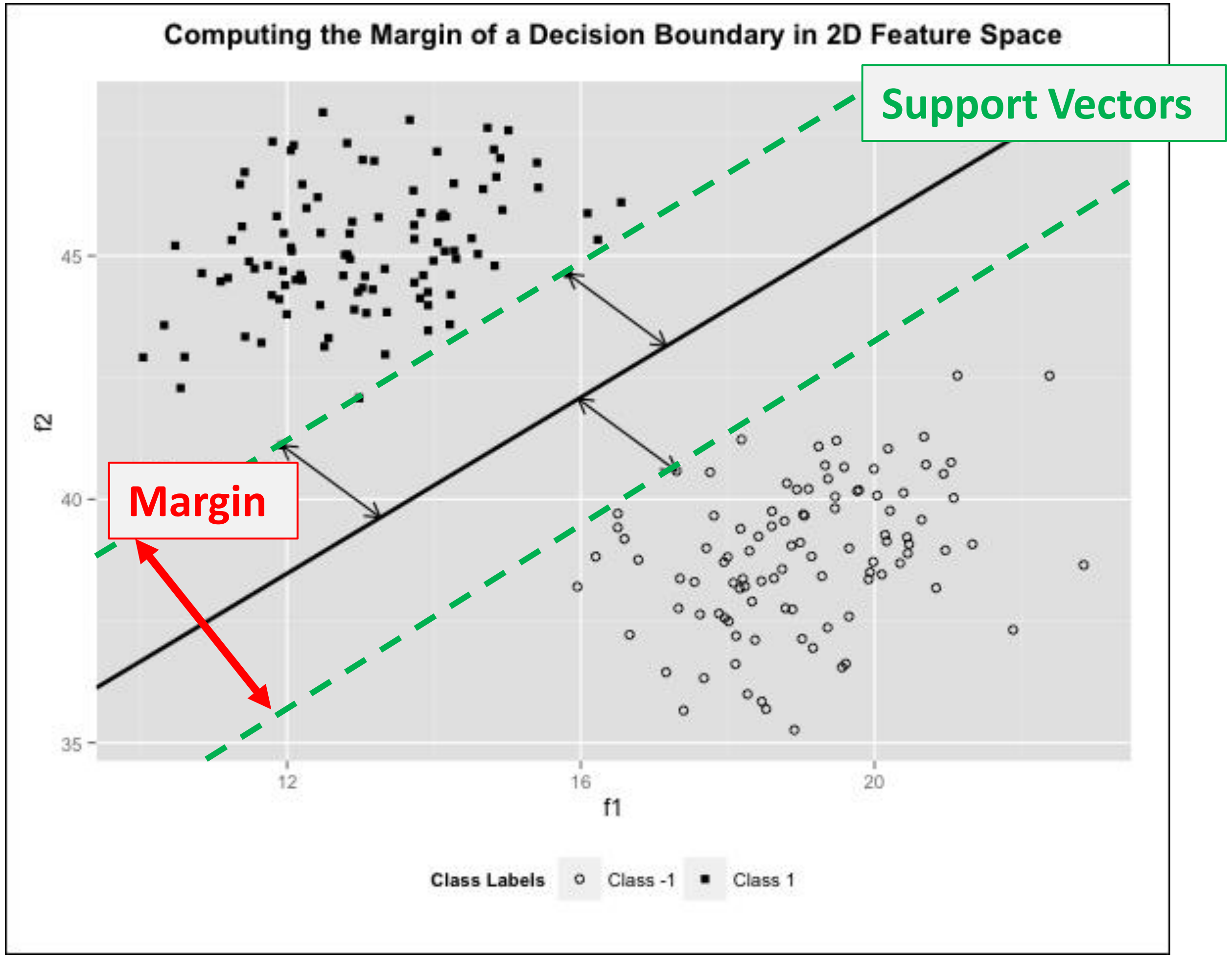
- Initially developed for classification problems
- Classification?
- Predicting/modelling outcomes that are "labels" or non-numeric, e.g.
    - Yes/No
    - Agree/Neutral/Disagree
    - Brands A/B/C

The Predictive Tools

Support Vector Machine (SVM)

The basic idea:

- The clear separation case →
- Objective: maximize the margin
- The partition classifies data to different labels
- Support vectors are also known as "maximal margin hyperplane"



The Predictive Tools

Support Vector Machine (SVM)

The basic idea:

- In practice: clear separation not realistic
- Algorithm allows for “slack” parameters – allowing for some observations to be within the margin
- But these “slack” parameters should be as small as possible collectively
- → A very complex constraint optimization problem!

The Predictive Tools

Support Vector Machine (SVM)

Regression with SVM

- The SVM algorithm can also be applied to numeric variables
- Objective: fit a flexible function to predict the numeric outcome
- How is it different to conventional regression?
  - Optimization: find the “most flat” function
  - i.e. the slope coefficients are smallest in magnitude
  - Subject to the regression residuals (actual – predicted) being smaller than the margin size

The Predictive Tools

Support Vector Machine (SVM)

Regression with SVM

- Flexibility of predictive function comes with a choice of “kernels”
    - Linear
    - Radial (discontinuous in some parts)
    - Polynomials

- We will look into the applications of SVM for classification in Session 5