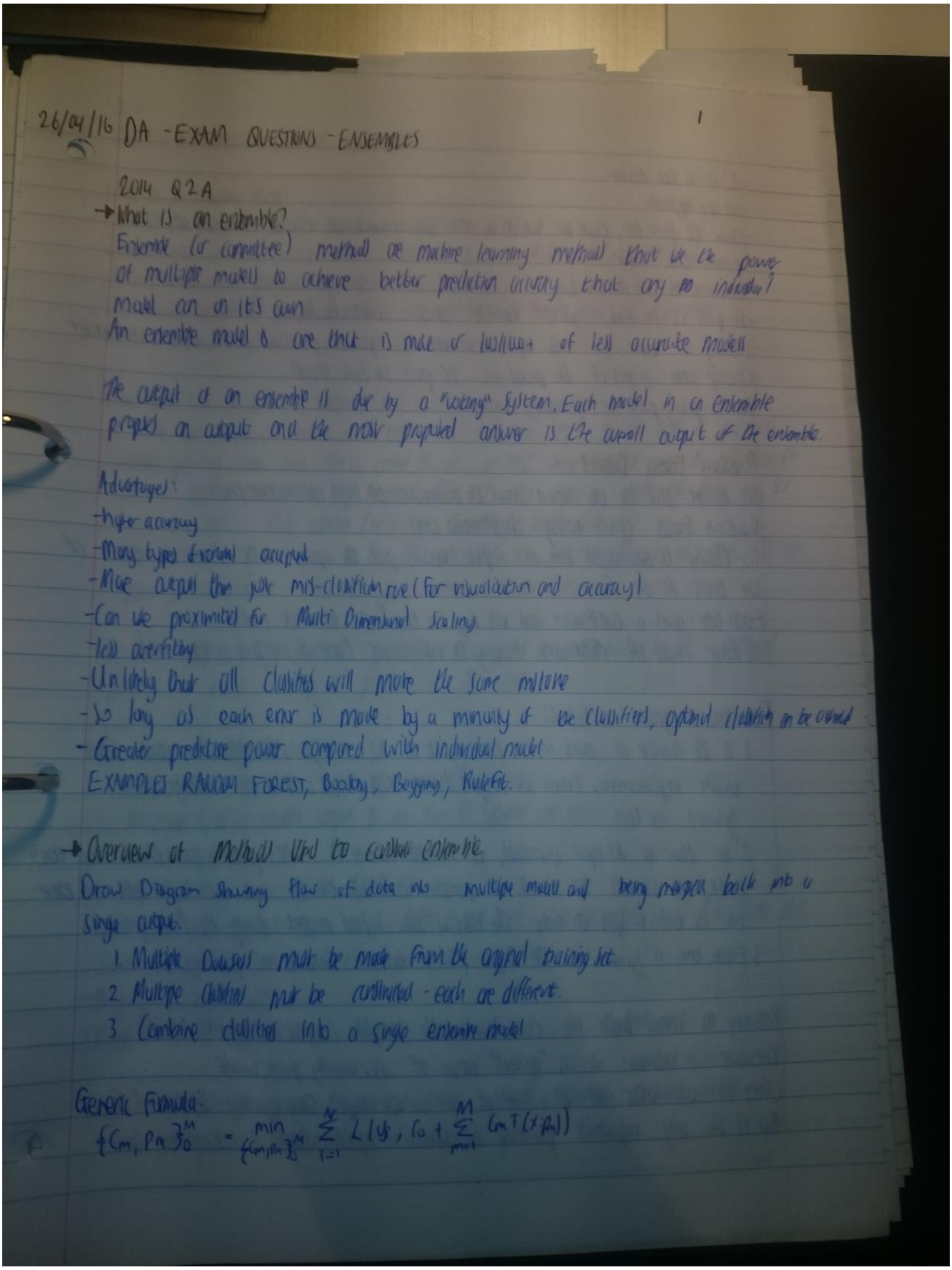


26/04/16 DA - EXAM QUESTIONS - ENSEMBLES

2014 Q2A

→ What is an ensemble?

Ensemble (or committee) methods are machine learning methods that use the power of multiple models to achieve better prediction accuracy than any ~~no~~ individual model can on its own

An ensemble model is one that is made of collection of less accurate models

The output of an ensemble is done by a "voting" system. Each model in an ensemble proposes an output and the most proposed answer is the overall output of the ensemble.

Advantages:

- higher accuracy
- Many types of variables accepted
- More outputs than just mis-classification rate (For visualisation and accuracy)
- Can use proximities for Multi Dimensional Scaling
- less overfitting
- Unlikely that all classifiers will make the same mistake
- So long as each error is made by a minority of the classifiers, optimal classification can be achieved
- Greater predictive power compared with individual model

EXAMPLES RANDOM FOREST, Boosting, Bagging, RuleFit.

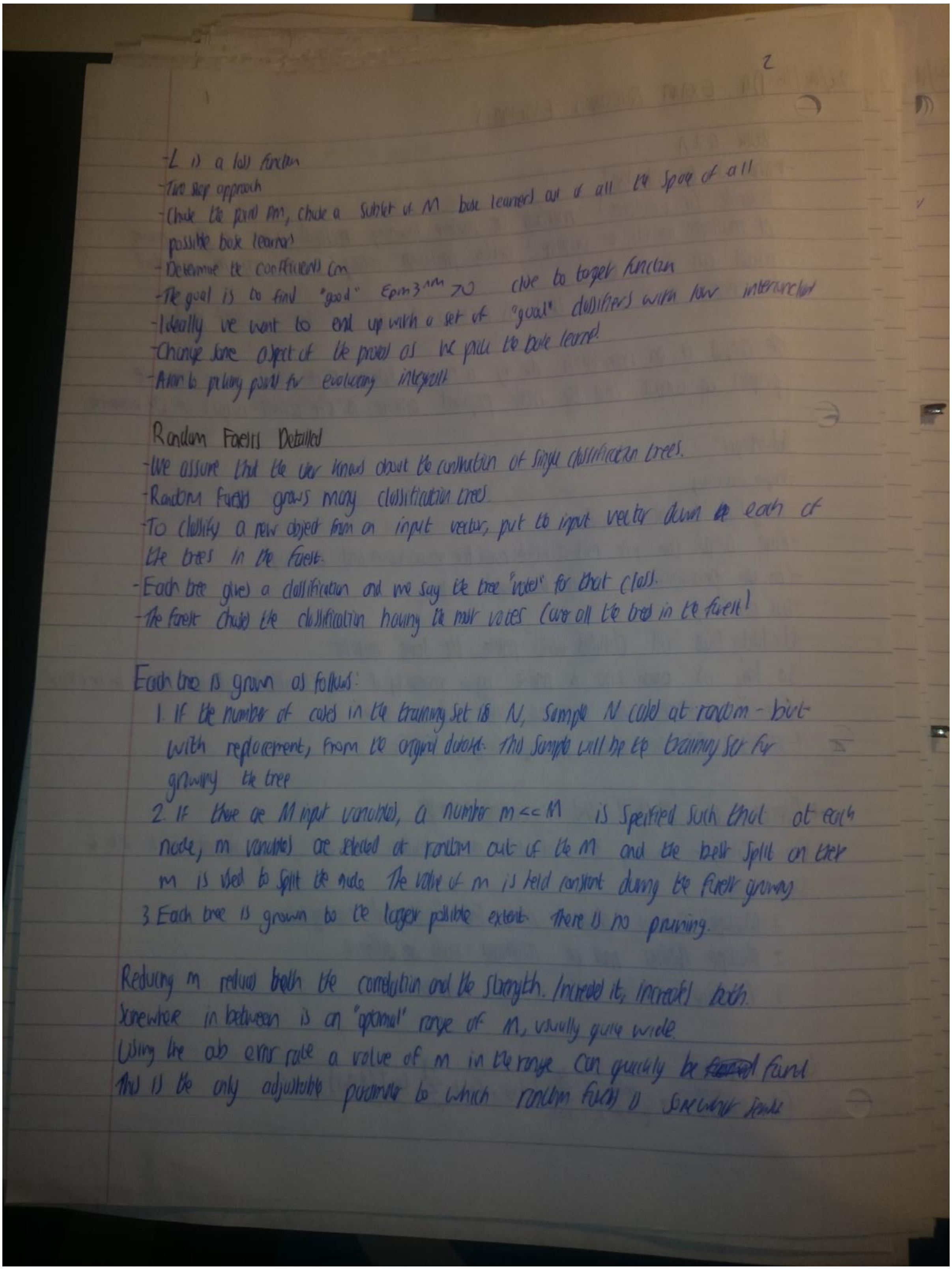
→ Overview of Methods Used to construct ensemble

Draw Diagram showing flow of data into multiple models and being merged back into a single output.

1. Multiple Datasets must be made from the original training set.
2. Multiple classifiers must be constructed - each are different.
3. Combine classifiers into a single ensemble model

Generic Formula

$$\{c_m, p_m\}_0^M = \min_{\{c_m, p_m\}_0^M} \sum_{i=1}^{N} L(y_i, c_0 + \sum_{m=1}^{M} c_m T(x, p_m))$$



- L is a loss function
- Two step approach
- Choose the points pm, choose a subset of M base learners out of all the space of all possible base learners
- Determine the coefficients cm
- The goal is to find "good" $\{p_m\}_1^M$ to close to target function
- Ideally we want to end up with a set of "good" classifiers with low intercorrelation
- Change some aspect of the process as we pick the base learner
- Akin to picking points for evaluating integrals

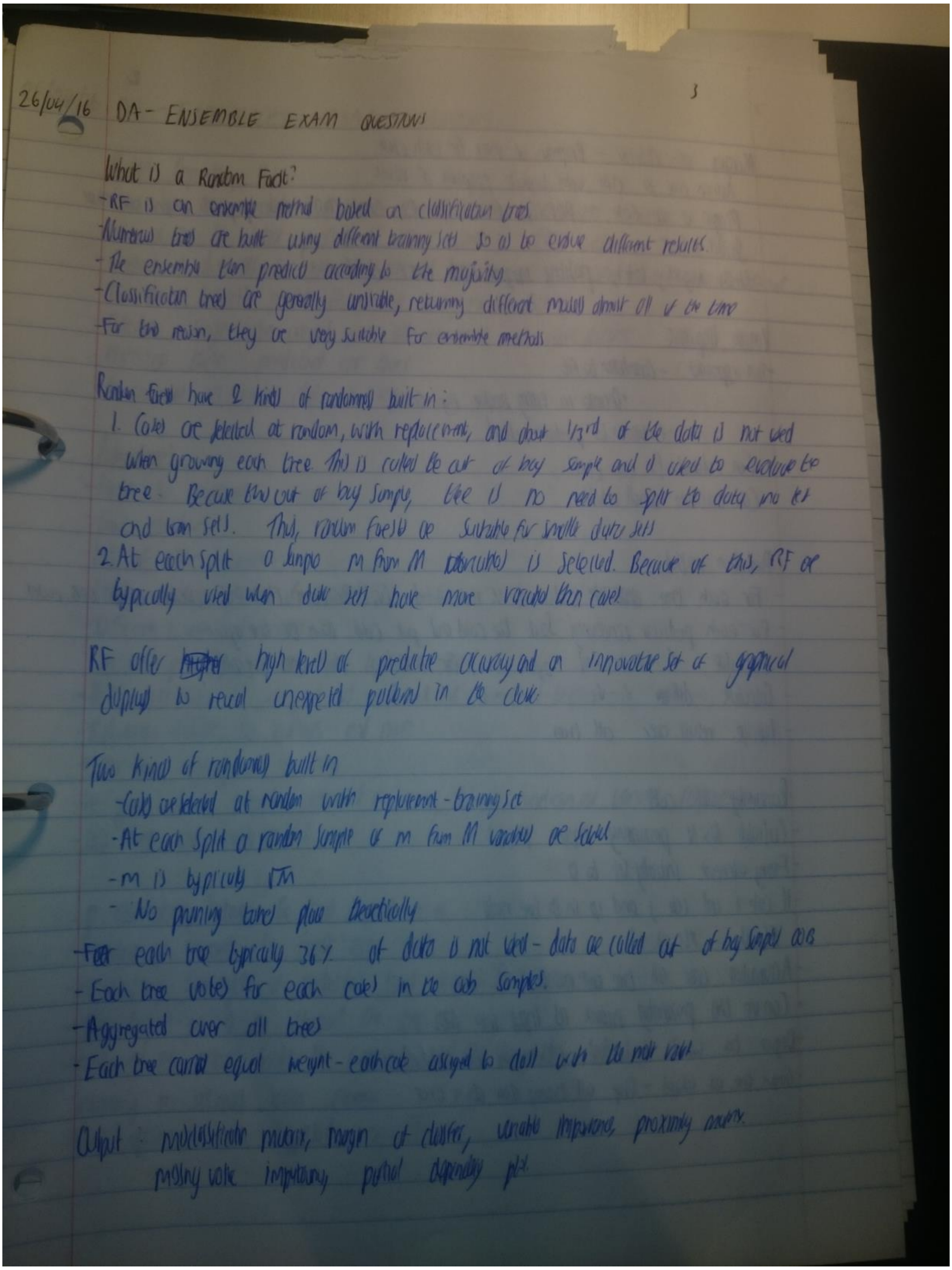
Random Forests Detailed

- We assume that the user knows about the construction of single classification trees.
- Random Forests grows many classification trees.
- To classify a new object from an input vector, put the input vector down each of the trees in the forest.
- Each tree gives a classification and we say the tree "votes" for that class.
- The forest chooses the classification having the most votes (over all the trees in the forest)

Each tree is grown as follows:

1. If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original dataset. This sample will be the training set for growing the tree
2. If there are M input variables, a number $m << M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing
3. Each tree is grown to the largest possible extent. There is no pruning.

Reducing m reduces both the correlation and the strength. Increasing it, increases both.
Somewhere in between is an "optimal" range of m, usually quite wide.
Using the oob error rate a value of m in the range can quickly be found
This is the only adjustable parameter to which random forests is somewhat sensitive



26/04/16 DA- ENSEMBLE EXAM QUESTIONS

What is a Random Forest?

- RF is an ensemble method based on classification trees
- Numerous trees are built using different training sets so as to evolve different results
- The ensemble then predicts according to the majority
- Classification trees are generally unstable, returning different models almost all of the time
- For this reason, they are very suitable for ensemble methods

Random forests have 2 kinds of randomness built in:

1. Cases are selected at random, with replacement, and about 1/3rd of the data is not used when growing each tree. This is called the out of bag sample and is used to evaluate the tree. Because the out of bag sample, there is no need to split the data into test and train sets. Thus, random forests are suitable for smaller data sets
2. At each split a sample m from M variables is selected. Because of this, RF are typically used when data sets have more variables than cases

RF offer ~~higher~~ high levels of predictive accuracy and an innovative set of graphical displays to reveal unexpected patterns in the data

Two kinds of randomness built in

- Cases are selected at random with replacement - training set
- At each split a random sample of m from M variables are selected
- m is typically $\sqrt{M}$
- No pruning takes place theoretically

- For each tree typically 36% of data is not used - data are called out of bag samples OOB
- Each tree votes for each cases in the oob samples.
- Aggregated over all trees
- Each tree carries equal weight - each case assigned to class with the most votes

Output : misclassification matrix, margin of classifier, variable importance, proximity matrix, missing value imputation, partial dependency plot.

Margins of classifier - Proportion of votes for each class

Assign case to class with highest proportion of votes.

Margin of a case = proportion of votes for correct class - max proportion assigned any class

Should be large

Under majority vote, positive margins mean correct classification

Variable Importance

- Two approaches: - Contribution to Fit
  - Decrease in fitting measure e.g. GINI
  - Contribution to prediction method

Can calculate for each class

Can calculate overall result

Prediction method:

- For each tree calculate % misclassified (v_1) for each class and overall using OOB cases
- For each predictor randomly sort the cases and put cases down the tree again
- Calculate % misclassified again (v_2) for each class and overall
- Calculate difference $v_1 - v_2$
- Average results over all trees

Proximity of cases

- Calculate NxN proximity matrix $P(i,j)$
- Every element initially set to 0
- If case i and case j end up in the same node

  $P(i,j) = P(i,j) + 1$
- Accumulate over all trees and normalise
- Can use this proximity matrix as input into MDS
- Degree to which individual observations are classified alike
- Grow tree as usual - Drop all training data down tree

6/04/16 DA - ENSEMBLES EXAM QUESTIONS

Partial Dependency Plot

- Shows how each predictor is related to response holding other variables constant.
- Let x be the initial predictor of interest with v distinct values
- Construct v data sets for each of the v values of x leaving all other variables unchanged
- For the v datasets, predict the response using random forests.
- Calculate a single value averaged over all observations for each dataset
- Average these predictions over trees.
- K categories in output class then K-1

Advantages

- Simple to implement
- Good classifier
- lots of other information
- Works with large number of variables
- Different types of variables
- Can use proximity matrix as input into MDS
- It's accuracy is as good as adaboost and sometimes better.
- Relatively robust to outliers and noise
- Faster than bagging or boosting
- It gives useful internal estimates of error, strength, correlation and variable importance
- It's simple and easily parallelised

- ρ correlation between the trees depends on m
- Increasing ρ increases the forest error rate.
- Increasing strength of individual trees decreases forest error rate
- The larger m is the "better" the tree
- Reducing m reduces both the correlation and the strength
- Increasing m increases both correlation " "
- Find optimum m Suggested $m = \sqrt{M}$

26/04/16

1 DA - ENSEMBLES EXAM QUESTIONS

Partial Dependency Plot

- Shows how each predictor is related to response holding other variables constant.
- Let x be the initial predictor of interest with v distinct values
- Construct v data sets for each of the v values of x leaving all other variables untouched
- For the v datasets, predict the response using random forests
- Calculate a single value averaged over all observations for each dataset
- Average these predictions over trees
- K categories in output class where k=2

A Advantages

- Simple to implement
- Good classifier
- lots of other information
- Works with large number of variables
- Different types of variables
- Can use proximity matrix as input into MDS
- It's accuracy is as good as adaboost and sometimes better
- Relatively robust to outliers and noise
- Faster than bagging or boosting

(4)
- It gives useful internal estimates of error, strength, correlation and variable importance
- It's simple and easily parallelized

- ρ correlation between the trees depends on m
- Increasing ρ increases the forest error rate.
- Increasing strength of individual trees decreases forest error rate
- The larger m is the "better" the tree
- Reducing m reduces both the correlation and the strength
- Increasing m increases both correlation " "
- Find optimum m suggested $m = \sqrt{M}$

x4

Tuning Parameters - Node Size for growing tree?
- Number of trees
- Number of predictors sampled

- Cannot include costs like a single tree
- Can alter priors

From the random Forest

Margin is the difference between the maximum class and the next largest class
i.e. predicted hair colour is brown, black, blonde
For people, say 80% say black, 10% blond 8% brown the margin is 80 - 10% = 70%
Margin is big therefore good

Missing data can be dealt with in 2 ways:
- Simplest method is to replace any missing data with the median
- More comprehensive method involves beginning with a rough guess of the values of missing data
  → Full size random forest is grown
  → For variable M with a missing value, an average of over all ~~missing~~ non-missing cases is taken and weighted by proximity
- Above process is repeated until the values converge.

ADA Boost:
- Variation of an ensemble methods
- Comprises of a # of weak learners - individual trees which are only slightly more accurate than random guessing - and weights for each of these learners, depending on which local area of the graph it is operating on
- Adaptive in the sense that subsequent weak learners are tweaked in favour of those instances misclassified by previous classifiers
- Sensitive to noisy data and outliers
- Individual learners can be weak but as long as learners are better than random guessing the final model can be proven to converge to a strong learner.

DA - ENSEMBLES EXAM QUESTIONS

- Heart of algorithm is how it examines each weak learner and assigns an alpha weight

- You go through the pool of classifiers and find the one which does the best job (minimising classification error). You then increase the weight of the samples which are wrongly classified - so the next classifier has to work better on these samples. Then you go through the pool again

- emphasise incorrect data to focus on valuable info

Adv DisAdv

- Powerful classification algorithm
- Can achieve similar classification results with much less tweaking of parameters or settings
- Can be sensitive to noisy data and outliers
- Can be less susceptible to overfitting problem
- Simple to implement
- Does feature selection resulting in relatively simple classifiers
- Instead of re-sampling, uses training set re-weighting
- Possibly suboptimal solution
- Lose the simple interpretability of classification trees.
- Computation is more difficult.
- No prior knowledge needed about weak learner.
- No parameters to tune except (T).
- From empirical evidence, Adaboost is particularly vulnerable to uniform noise

- Boosting performs an exhaustive search for best predictor to split on, whereas random forests' model only searches a small subset of data
- Boosting grows trees in series, with later trees dependent on the results of previous trees, RF grows in parallel, independent of one another
- On v large training sets, boosting can be slow with many predictors, while RF models select only a subset of predictors for each split which can handle significantly larger problems before slowing