



Random forests

RUG-Philippines

Isaac Toh Wei Zhong

20/7/2016

A little bit about me

- Graduated from National University of Singapore, Computational biology
 - Statistics and computing onto biology and healthcare
- Data scientist at Singtel, Globe

Agenda for this evening

- Some materials on trees
 - Terminologies
 - Measuring performance
 - Pruning
- Ensemble modelling
 - Intuition and math
- Bagging, or bootstrap aggregating
- Random forest (RF)
 - Bagging vs. RF
 - Out-of-bag (OOB) assessment of model performance
 - Variable importance measures
 - Multidimensional scaling (MDS) plot on proximity matrix
 - Hyperparameters tuning
- Hands-on / code walkthrough

Decision trees

Some basic materials

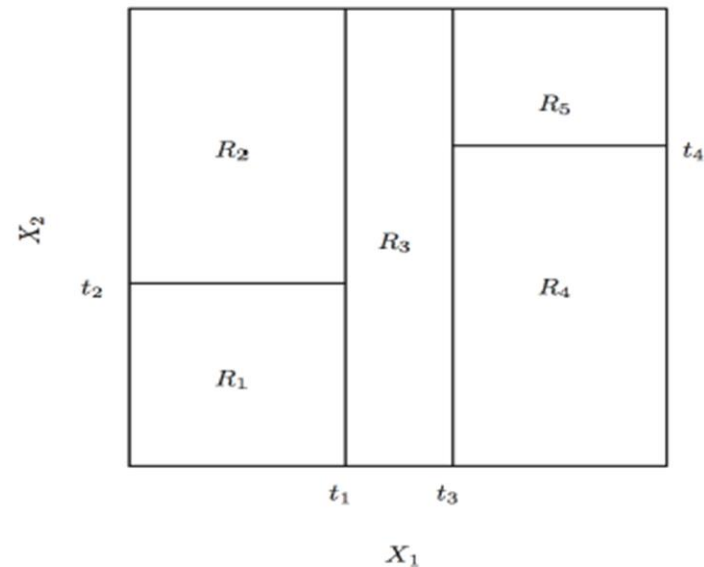
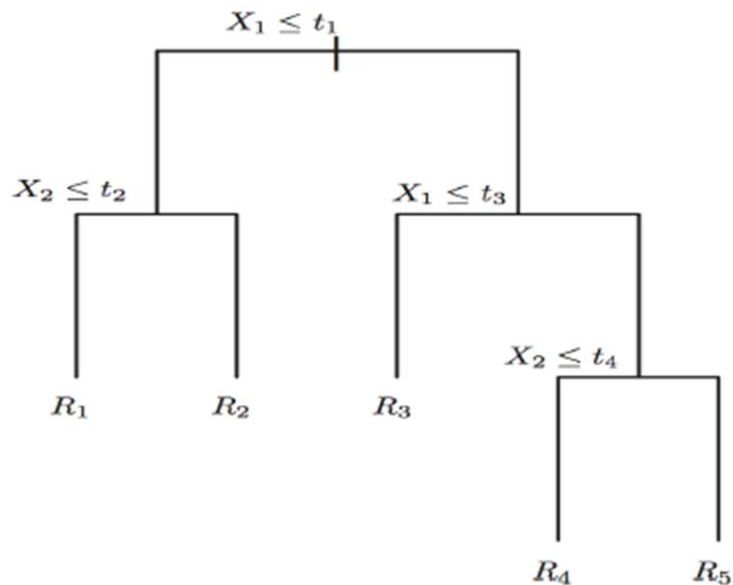


Decision trees

- A simple model used in supervised learning
- CART, C4.5 – amongst top 10 most popular data mining algorithms
- Classification (response variable is categorical) and regression (response variable is continuous or numerical)
- [R] The **tree** package that we are using uses the recursive partitioning algorithm

Equivalents

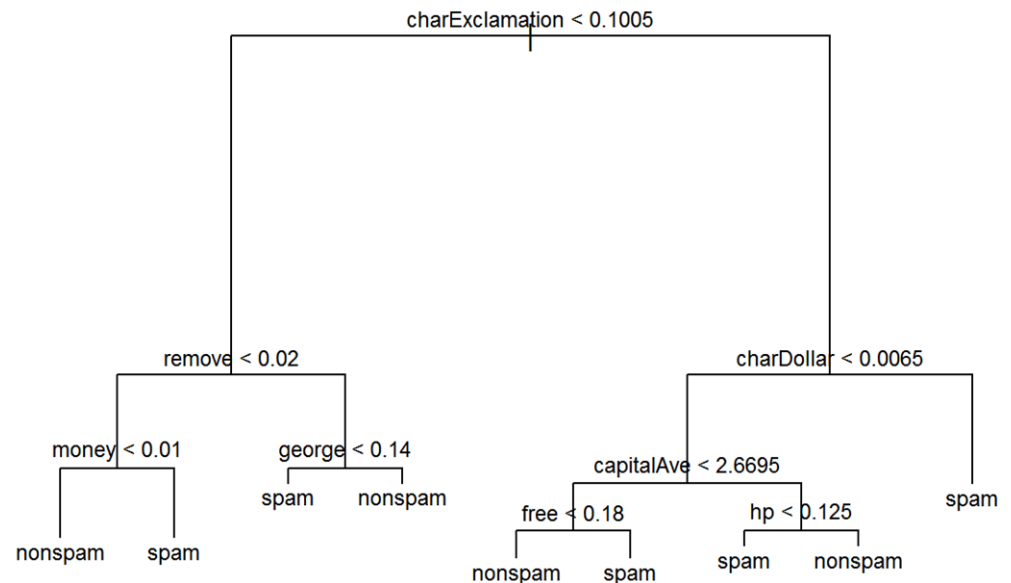
- Tree == Binary partitioning of dataset
- Each partition is represented by the mode (classification) or mean (regression)



Terminologies

- Depth
- Node
 - Leaf nodes
 - Non-leaf nodes
- The size of a tree sometimes refers to the number of leaf nodes
- Parents and children
- Branching factor

Pruned decision tree (9 leaf nodes)

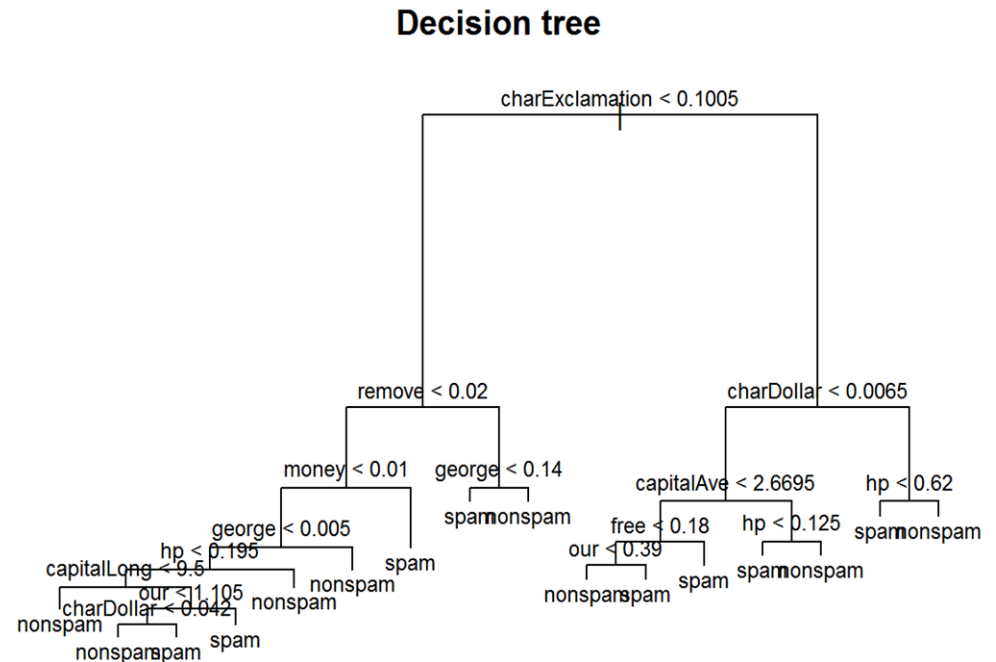


Assessing the performance of a supervised learning model

- Classification: accuracy / error rate
 - Sensitivity, specificity etc.
- Regression: mean squared error
 - $MSE = \frac{1}{n} \sum (prediction - actual)^2$
- Also, there are two types of classification models:
 - (1) Those that output classes / categories as predictions
 - (2) Those that output probabilities as predictions
- (2): can use ROC-AUC as a measure of performance

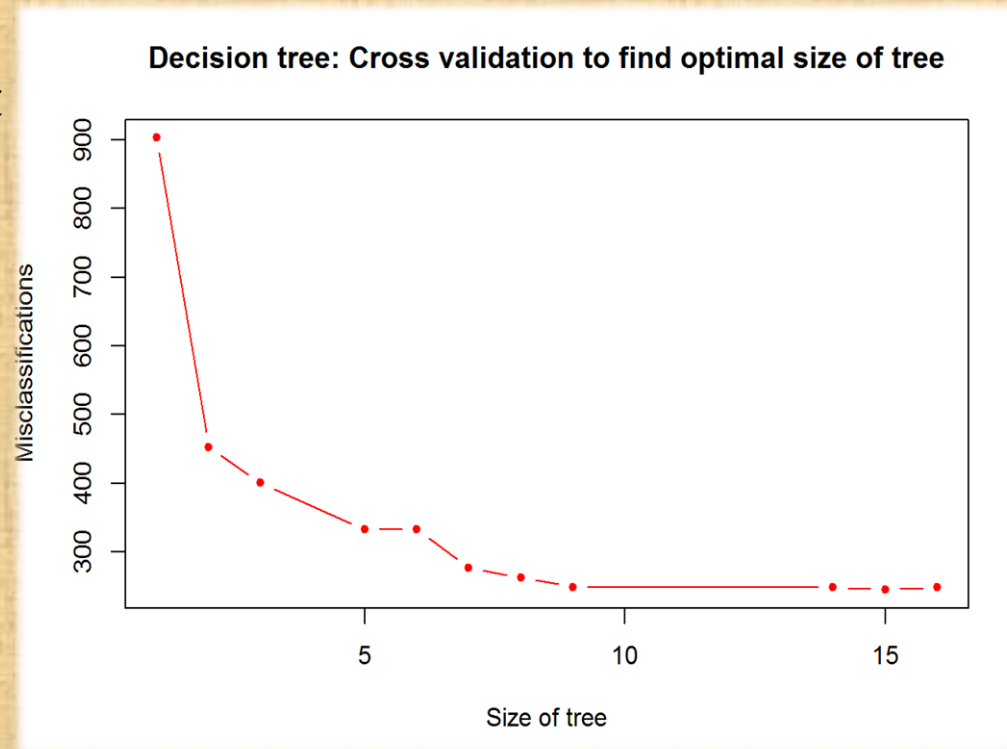
Pruning

- Typically after the construction of a decision tree, we would want to prune the tree, because the tree may be overly complicated



Pruning (2)

- Pruning refers to the process of trimming the tree to a more compact and concise one, without sacrificing much performance
- [R] The **tree** package uses cost-complexity pruning
 - Comparing the relationship between number of leaf nodes and performance of model



Pros and cons of decision trees

- Pros:
 - Very easy to interpret and communicate to others, because it is similar to how humans think and make decisions
 - Easy to construct
- Cons:
 - Generally unstable
 - Low predictive accuracy

Ensemble learning

Putting models together



Ensemble learning

- Putting multiple models / learners together in an ensemble
- Voting: can be shown mathematically that, to minimise prediction errors, for
 - Classification: use majority vote (mode)
 - Regression: use mean of all predictions

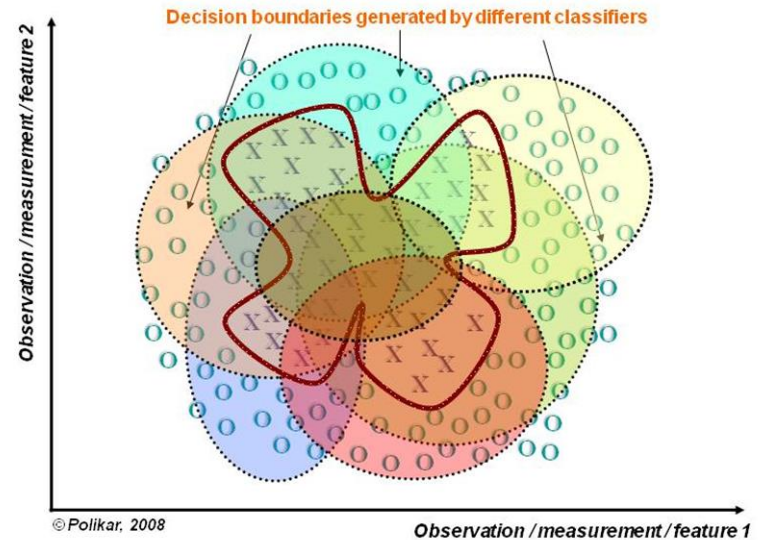
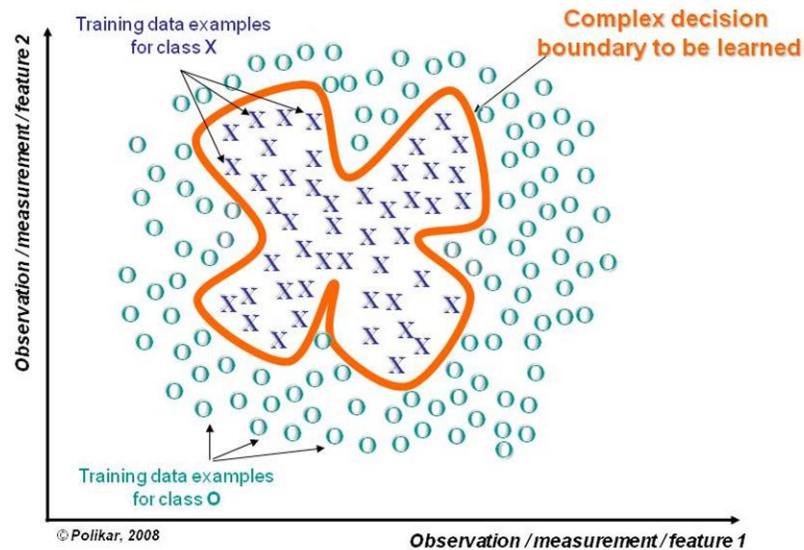
Intuition of rationale behind ensemble learning

- Binary classification: a single classifier has probability $p > 0.5$ of giving the correct answer
- Let's assume $p = 0.6$
- Putting three classifiers together:
 - Predicted answer is correct if 2 out of 3 classifiers give correct answer
 - Overall probability of giving correct answer, $p^* = 0.648$
- Generally, p^* increases as number of classifiers increases
- **This result is valid only if the individual classifiers are independent, or at least uncorrelated, with each other**

Mathematical rationale

- Each classifier c signifies a Bernoulli random variable, with mean of p , variance of $p(1 - p)$
- Putting 3 classifiers together and assuming independence,
 - $(ens.) = \frac{1}{3}(c_1 + c_2 + c_3)$
 - $E(ens.) = p$ (unbiased)
 - $Var(ens.) = \frac{1}{3}p(1 - p) < p(1 - p) = Var(c)$
- Without independence, we need to consider pairwise covariance terms: $Var(ens.)$ increases
- Analogous for a regression problem: $Var(ens.)$ increases

Another way to look at it



Bagging

Bootstrap aggregating



Bagging

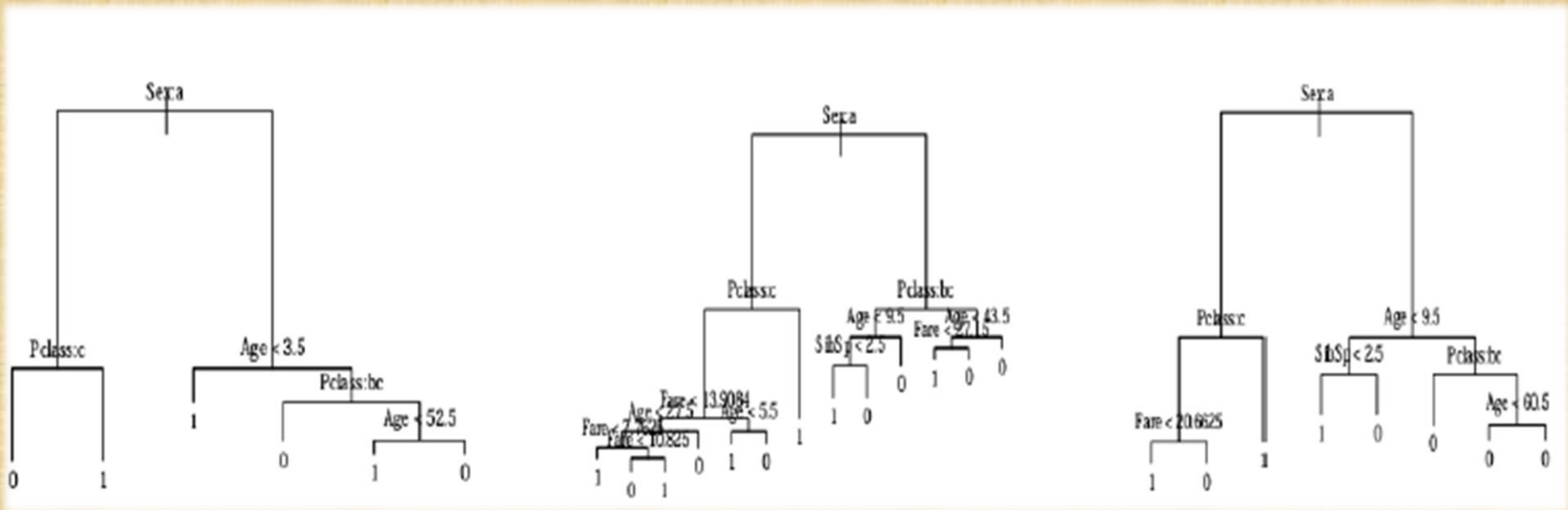
- Voting only works well if the individual models are uncorrelated, or at least less correlated with one another
- Bagging, a.k.a. **bootstrap aggregating**, aims to alleviate this problem
- Idea: build decision trees on different subsets of the training data. Each subset is known as a “bag”
- Each bag is a sample from the training data, **with replacements**
- Each decision tree gives a vote, overall classification / regression is based on the votes
- **Size of each bag is the same as the sample size of the dataset**

Bagging (2)

		ID	V_1	V_2	V_3	...	V_M	RESPONSE
Bagging: sample with replacement, one tree each	1							
	2							
	3							
	4							
	...							
	N							

Not good enough

- Bagging is an attempt to reduce the amount of correlation / similarity in the individual trees





Random forest

Random subspace sampling

Random forest

- To further reduce correlation / similarity between trees, RF uses a technique called “**random subspace sampling**”
- For each tree, for each node, instead of choosing one variable from all variables to split on, choose one from only a random subset of variables
- “Space” refers to feature space, i.e. all variables in training data

Bagged forest vs. random forest

Random subspace sampling: sample **without** replacement, choose one to split on for each tree, each node

ID						RESPONSE
	V_1	V_2	V_3	...	V_M	
1						
2						
3	Bagged forest: bagging					
4	Random forest: bagging + random subspace sampling					
...						
N						

Bagging:
sample **with**
replacement,
one tree each

Bagged forest vs. random forest (2)

Tree

-> (ensemble learning + bagging) bagged forest

-> (random subspace sampling) random forest

- The only difference between bagged forest and random forest is the use of a subset of variables to do splitting on
- [R] Only the **mtry** argument differs

Pros and cons of random forest

- Pros:
 - One of the top-performing models in supervised learning
 - With some basic understanding of sampling and bootstrapping, RF can be easy to communicate. The intuition of voting as a mechanism to make decisions is simple
 - Able to derive variable importance measures
- Cons:
 - Computationally intensive



More about random forest

- Out-of-bag (OOB) assessment of model performance
- Variable importance measures
- Multidimensional scaling (MDS) plot on proximity matrix
- Hyperparameters tuning

Out-of-bag assessment of model performance

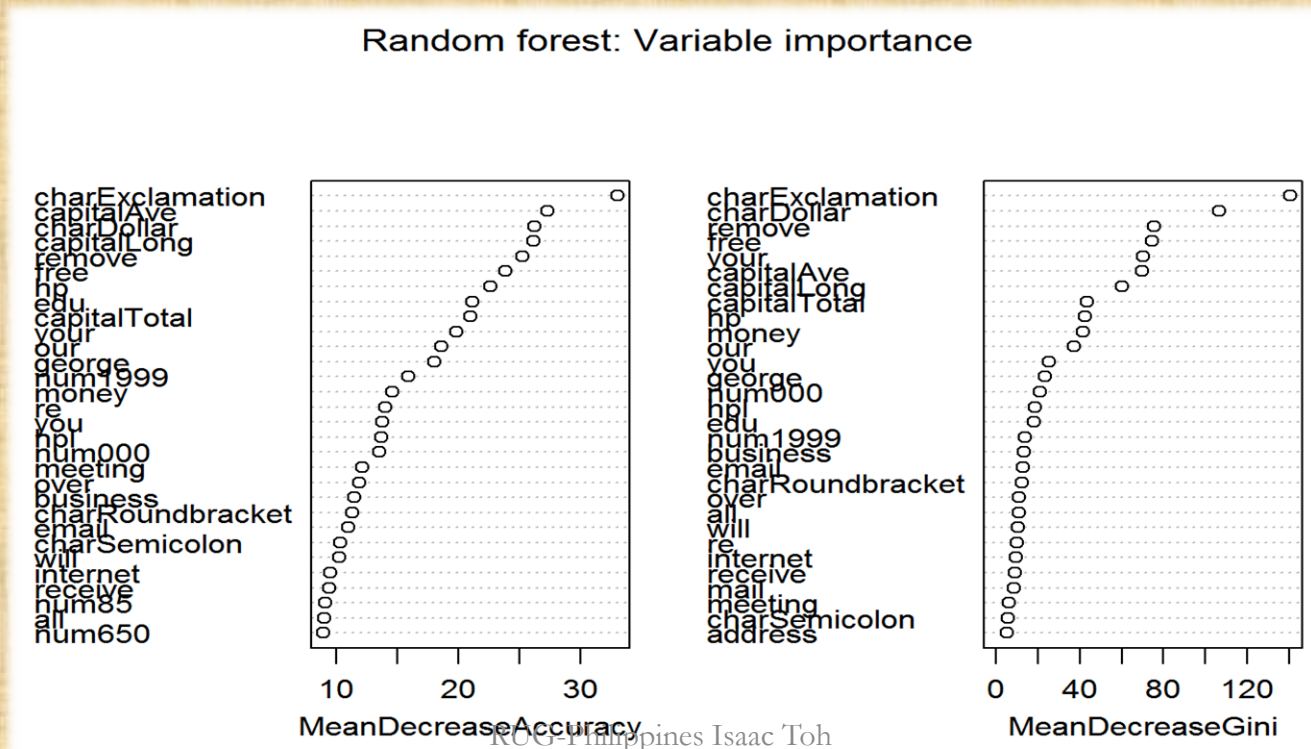
- Similar to cross validation
- Recall that in bagging, we have multiple bags – each bag is a subset of samples in the dataset
- Individual models are then built on each bag
- For a given bag, there are samples in the dataset that is either in the bag or out of bag (OOB)

Out-of-bag assessment of model performance (2)

- For each sample s_i , take the set of models in the ensemble that did not use s_i in its construction. Call this sub-ensemble E_{-i}
- We then get a prediction of s_i , using E_{-i} , by voting
- The prediction of s_i using E_{-i} may incur
 - A classification error $(err.)_i$
 - A regression error ε_i
- The OOB error estimate of the entire ensemble is then
 - $(err.rate)_{OOB} = \frac{1}{n} \sum (err.)_i$
 - $MSE_{OOB} = \frac{1}{n} \sum \varepsilon_i^2$

Variable importance

- To assess relative variance importance in RF model,
 - Mean decrease in accuracy (MDA)
 - Mean decrease in Gini (MDG)



Mean decrease in accuracy

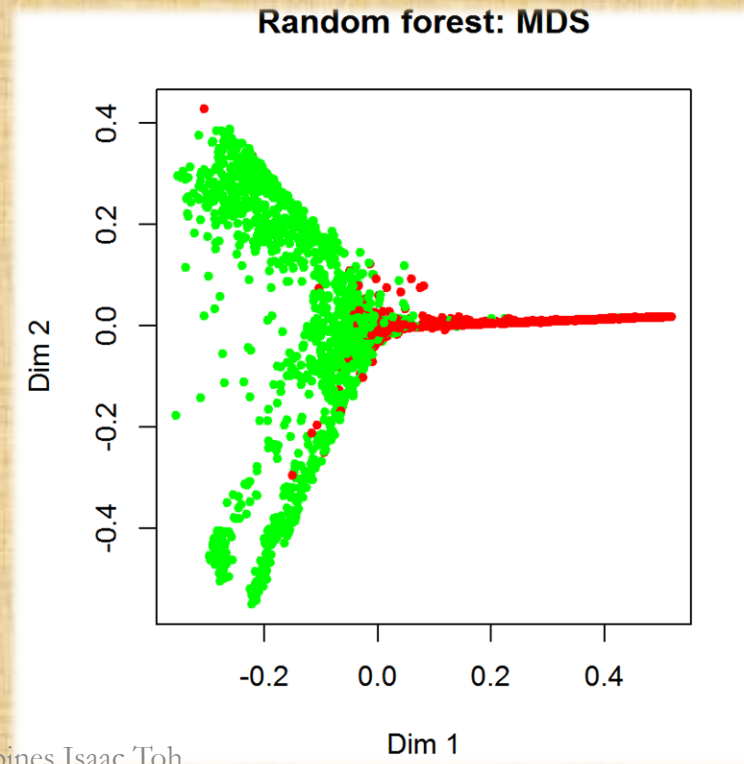
- For each tree T_k in the ensemble, take its OOB samples (samples that were not used in the construction of T_k). Call them $(oob)_k$
- Run all $(oob)_k$ down T_k , and get a classification accuracy
- Now, for each variable v_j in $(oob)_k$, randomly shuffle its values. Run the v_j -shuffled- $(oob)_k$ down T_k
- Measure the decrease in accuracy for v_j on T_k , call it $(da)_{jk}$. Repeat for all j, k
- To get MDA for v_j ,
 - $(mda)_j = \frac{1}{ntree} \sum (da)_{jk}$
- Analogous to regression: use MSE

Mean decrease in Gini

- In the construction of each tree T_k in the ensemble, for each split, the variable used reduces the Gini impurity criterion
- Simply add up the Gini decreases accumulated by each variable, and divide by **ntree**

Multidimensional scaling plot on proximity matrix

- Very good tool to visualize samples in the dataset in relations to each other, in the context of the RF model
- Two things here:
 - Multidimensional scaling
 - Proximity matrix

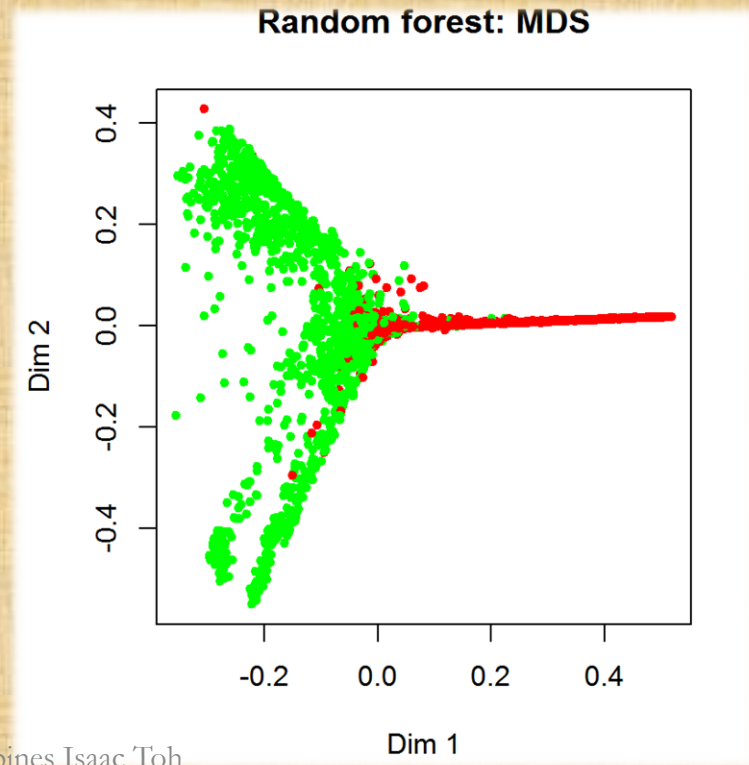


Proximity matrix in RF

- Create a n -by- n identity matrix (n is the number of samples). Call it P
- For each sample s_i , run it down all trees in E_{-i}
 - s_i will end up in particular leaf nodes in each tree in E_{-i}
- Take another sample s_j , run it down in E_{-j}
- Each time s_i and s_j end up in the same leaf node, increment P_{ij} and P_{ji} by 1
- Finally, standardise by dividing the off-diagonal elements of P by **ntree**
- This gives the proximity matrix P

Multidimensional scaling plot on proximity matrix

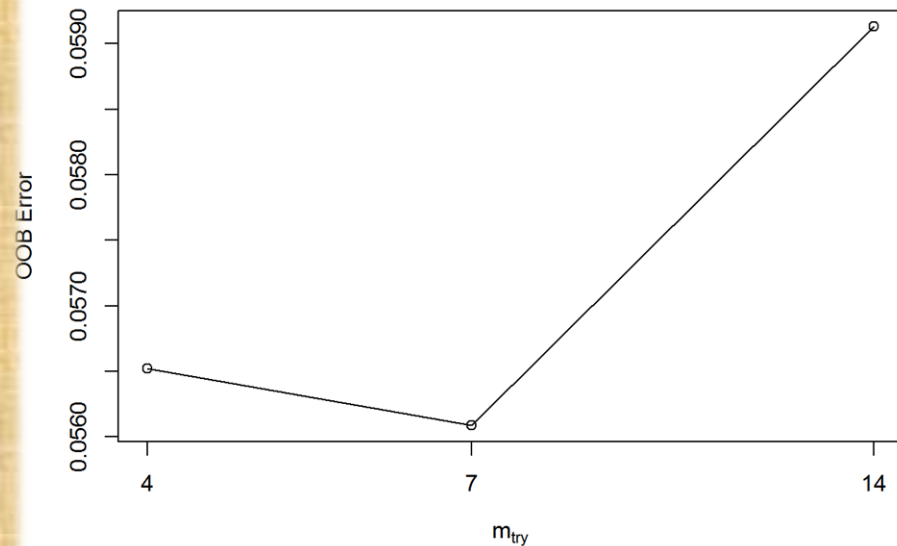
- With the proximity matrix P , do principal components analysis (PCA)
- Plot PC1 and PC2



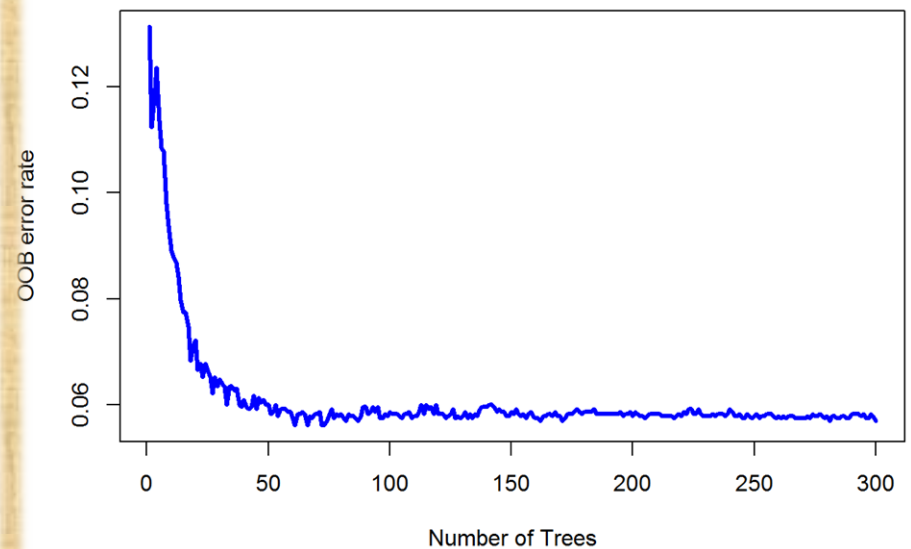
Tuning of RF parameters in R

- **mtry**: number of variables to try from for each split
- **ntrees**: number of trees in ensemble

Random forest: Tuning the mtry hyperparameter




Random forest: OOB estimate of error rate



[illegible][illegible]

What's next?

- Gradient boosting
 - Ensemble of ensembles
 - Model stacking
- 
- Well-liked by Kagglers

Thanks! Questions?

github.com/tohweizhong/RUGS-RF.git

sg.linkedin.com/in/tohweizhong

tohweizhong@u.nus.edu

