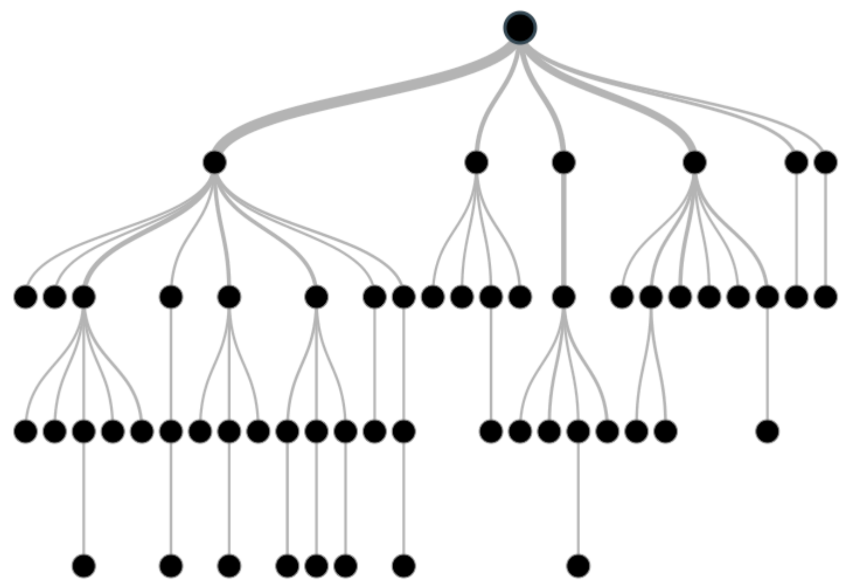
Random Forest



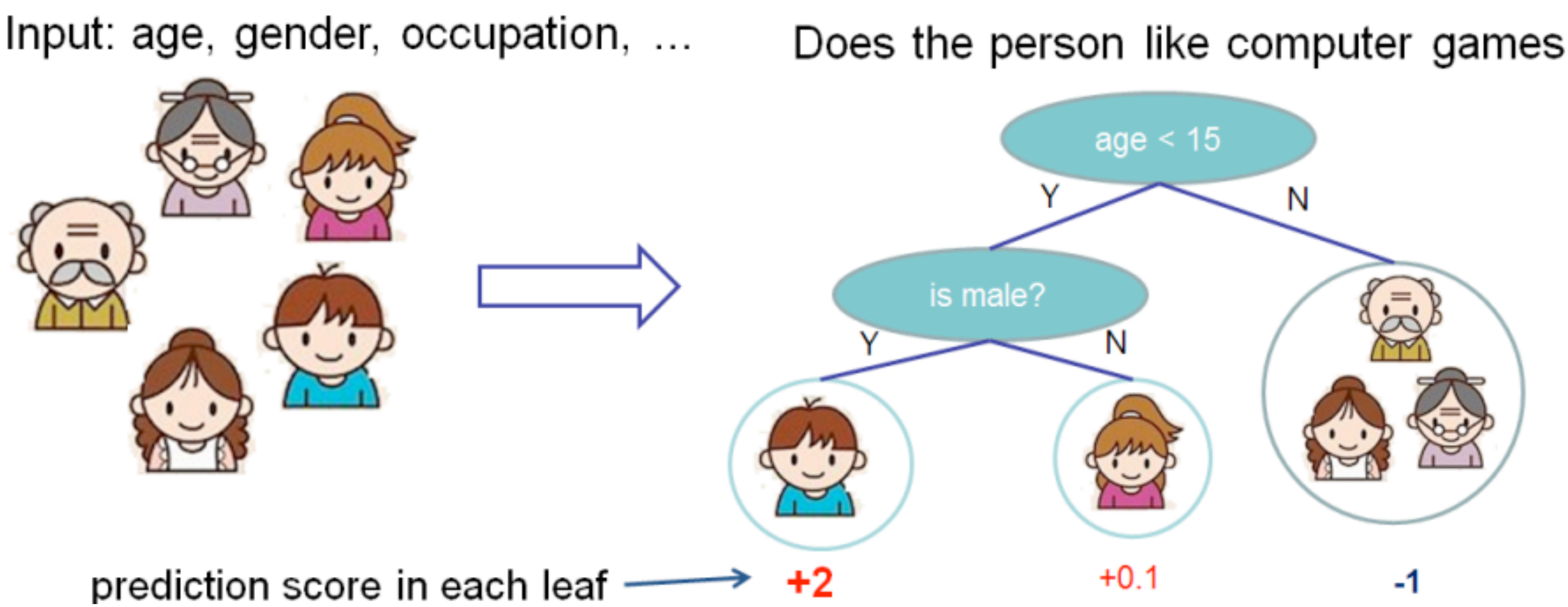
Yingkai Zhang
Department of Chemistry, New York University
NYU-ECNU Center for Computational Chemistry at NYUSH

Outline

- Decision Trees - Classification and Regression Trees (CART)
- Bagging: Averaging Trees
- Random Forest: Clever Averaging of Trees

Decision Trees: classification and regression trees (CART)

- Separate the data according to a series of decision rules (age < 15)



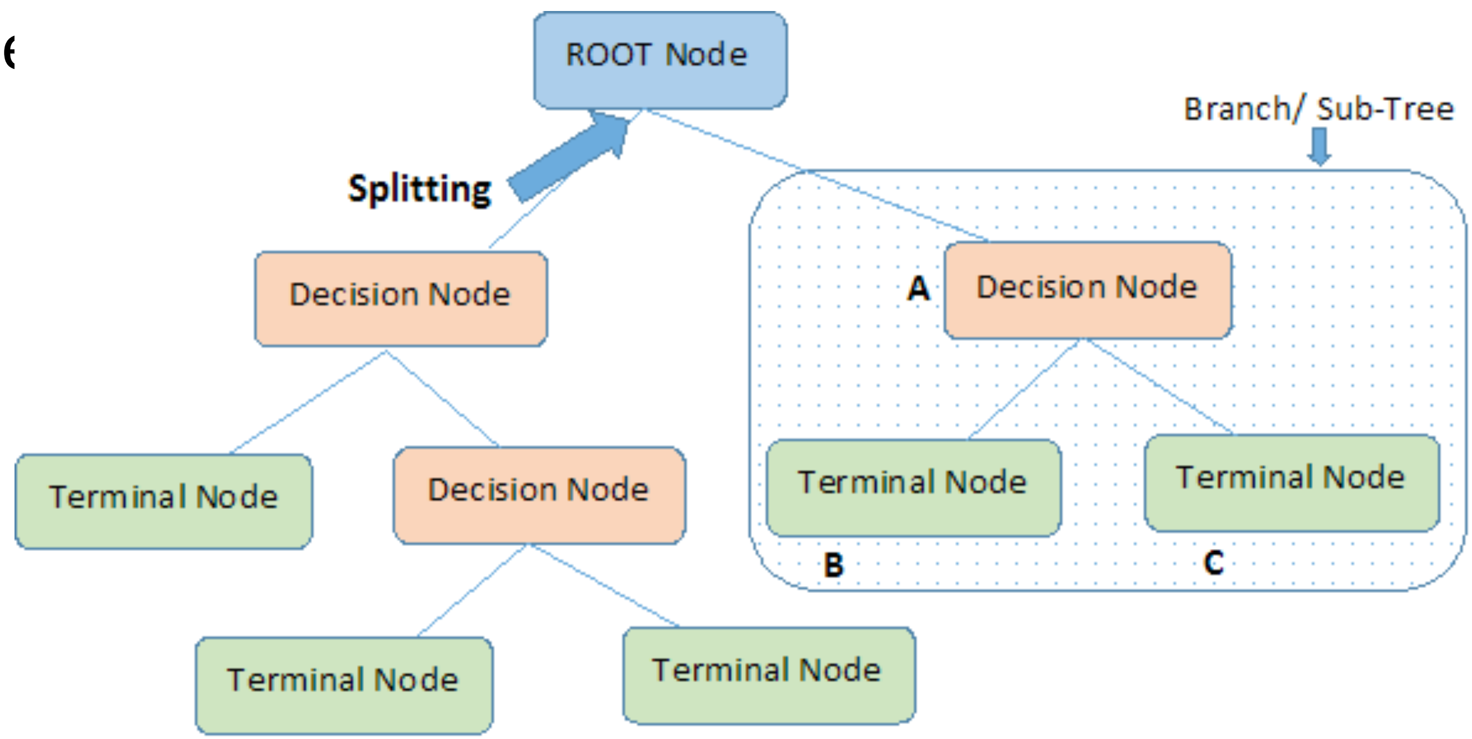
http://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf

Decision Tree Terminology

- Root node
- Splitting: a process of dividing a node into two or more subnodes
- Decision node
- Leaf

(terminal node

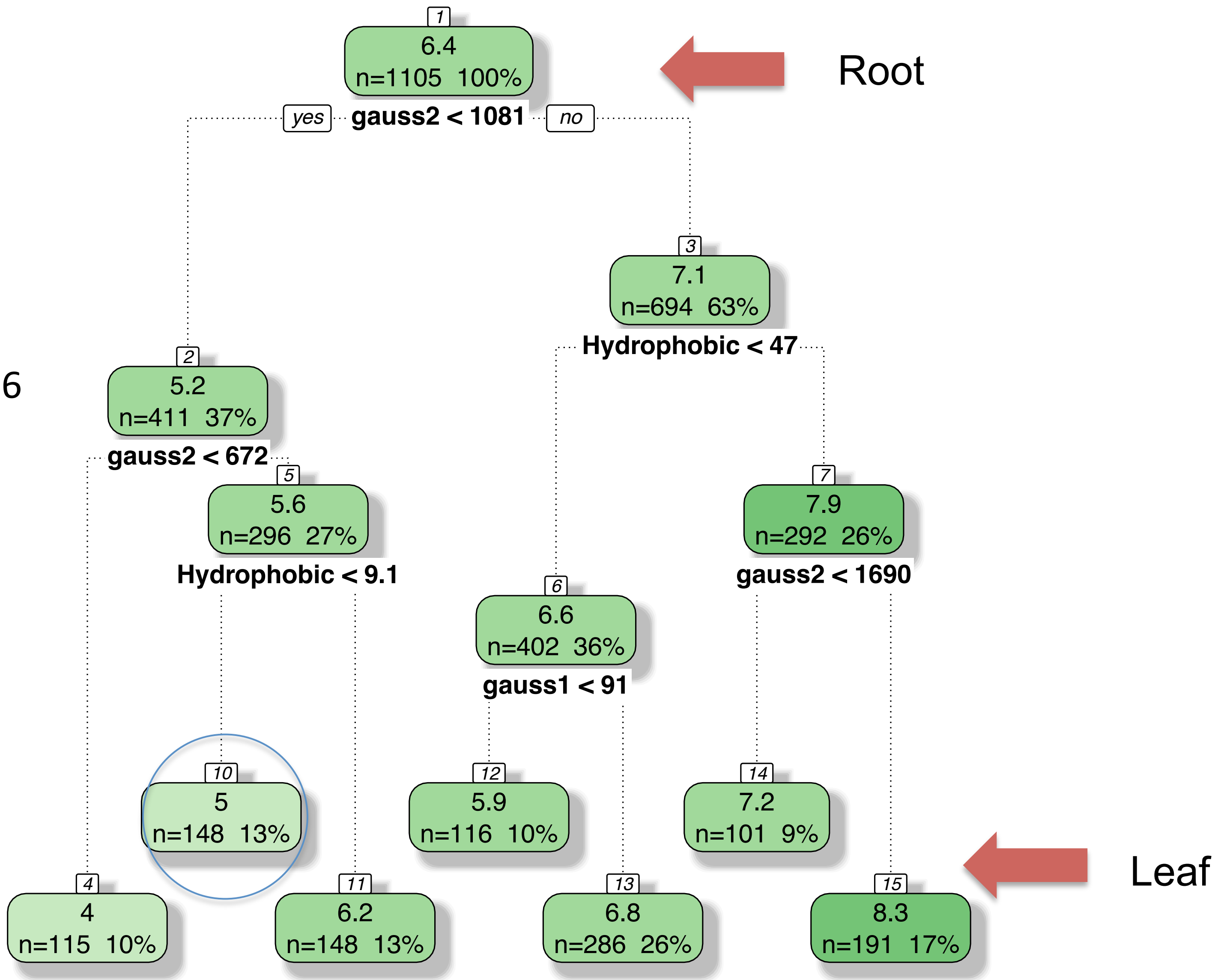
- Pruning
- Branch/ Sub-Tree
- Parent and child node



Note:- A is parent node of B and C.

Regression using tree-based method

New data:
Gauss2= 900
Hydrophobic=7.6
Gauss1 = 80



Recursive Binary Splitting

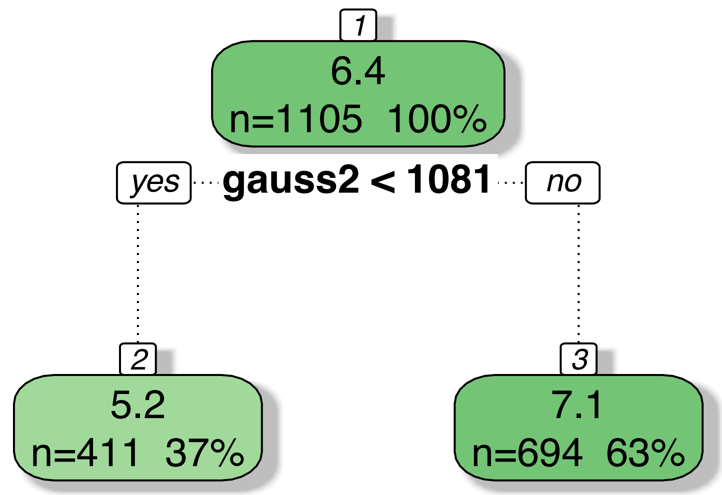
- A top-down, greedy approach
- Each node
  - Find feature X_j and cut-point s
  - split the data points into two regions

$$R_1(j,s) = \{X \mid X_j < s\}$$
$$R_2(j,s) = \{X \mid X_j \geq s\}$$

  - with lowest residual sum of square (RSS)

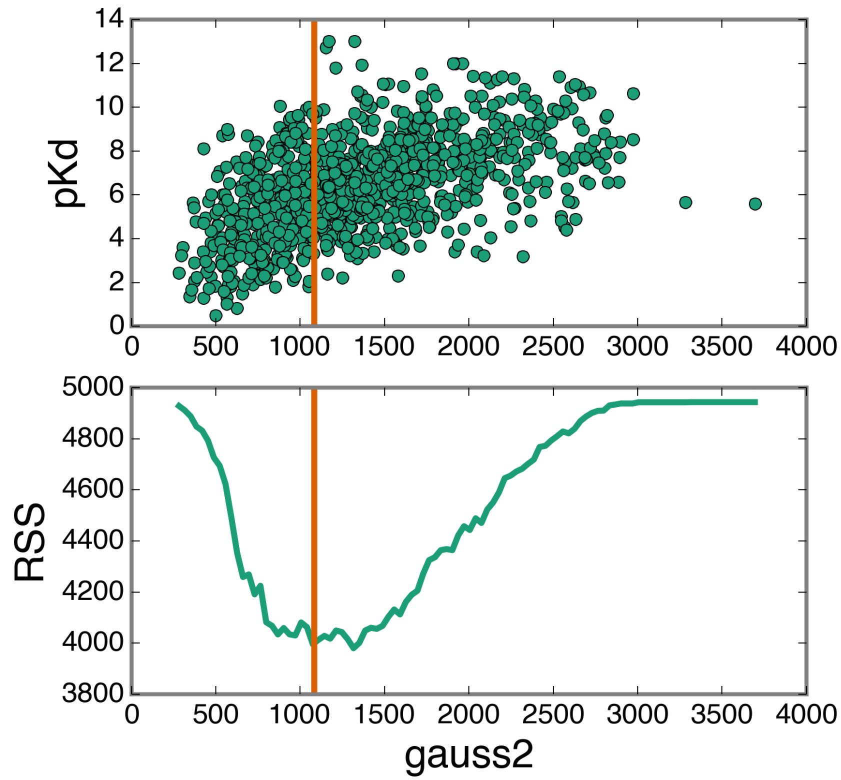
$$RSS = \sum_{i: x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2$$

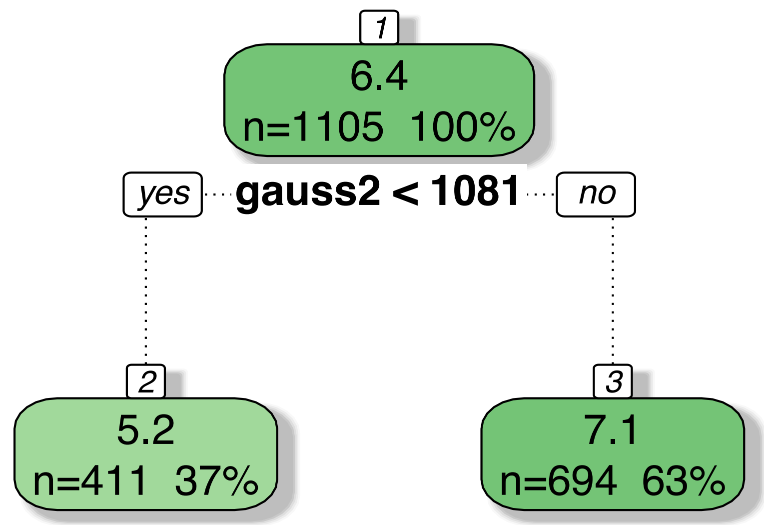
  - Each node is represented by the mean



Selects the split which results in most homogeneous sub-nodes

Reduction in Variance of Sub-Nodes

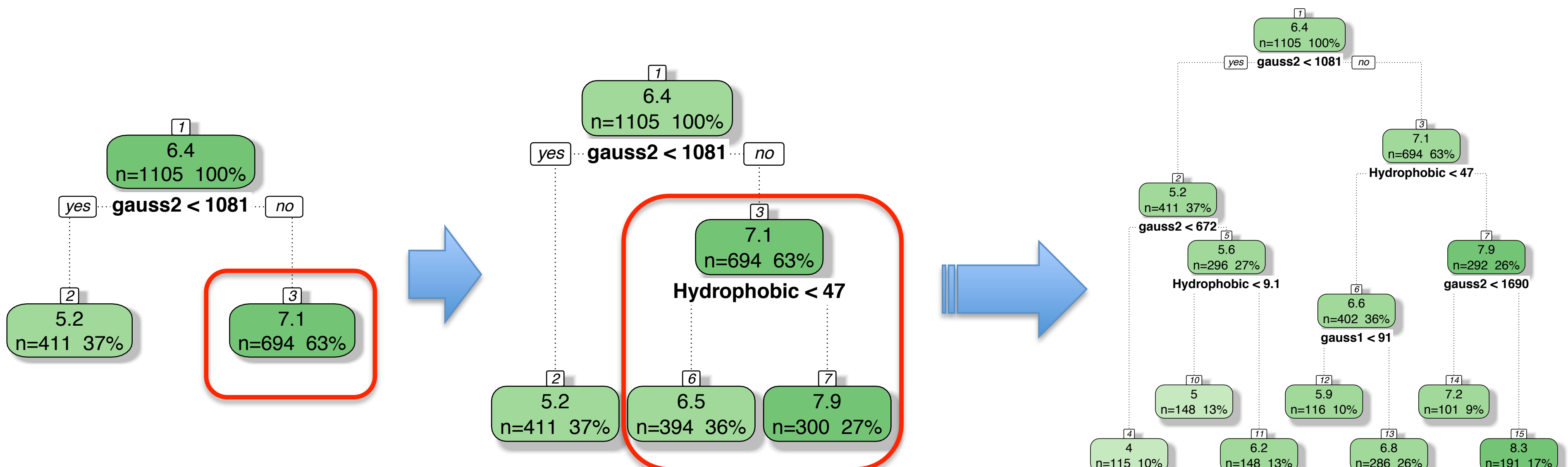




$$RSS = \sum_{i:x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2$$

Build Regression Tree

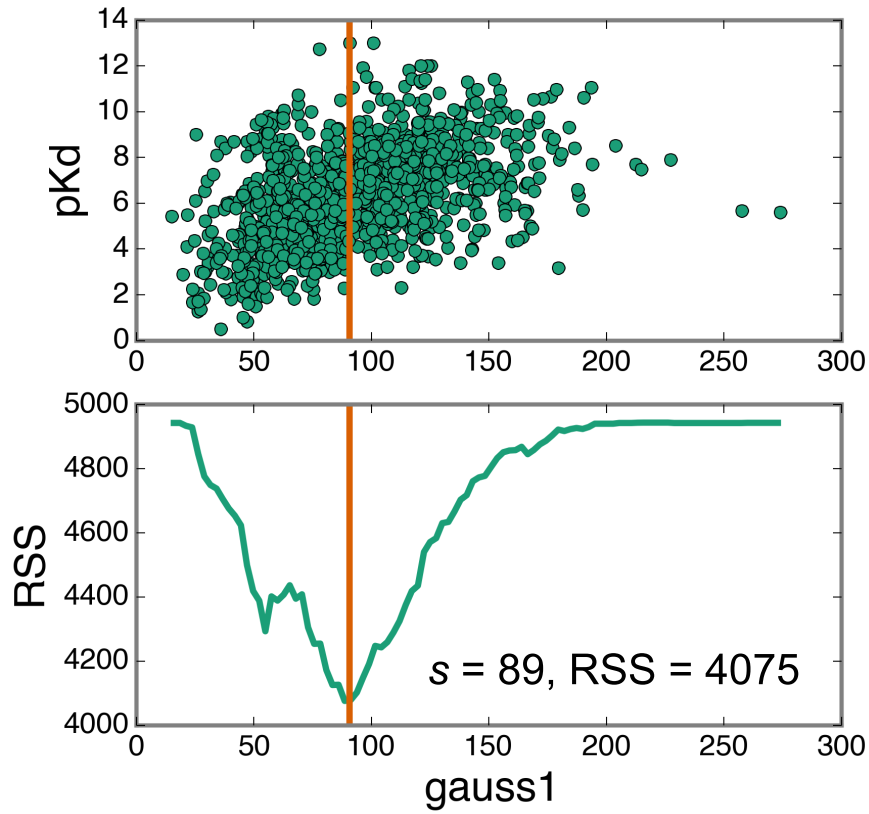
- ❑ Split each node using the same procedure until a stopping criteria is reached
  - ❑ i.e. number of data points in each region lower than cutoff



Reduction in Variance of Sub-Nodes

- Each feature X_j
  - Find the cut-point s with lowest RSS
- Select the feature have lowest RSS

Feature	RSS	s
gauss1	4075	89
gauss2	3980	1081
Replusion	4838	3.6
Hydrophobic	4131	9.7
HBonding	4880	2.0
Nrot	4668	6.5



$$RSS = \sum_{i:x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2$$

Pros and Cons of Decision Trees

- Non linear
- Robust to correlated feature
- Robust to feature distributions
- Robust to missing values
- Easy to understand
- Fast to train and predict
- Non parametric method

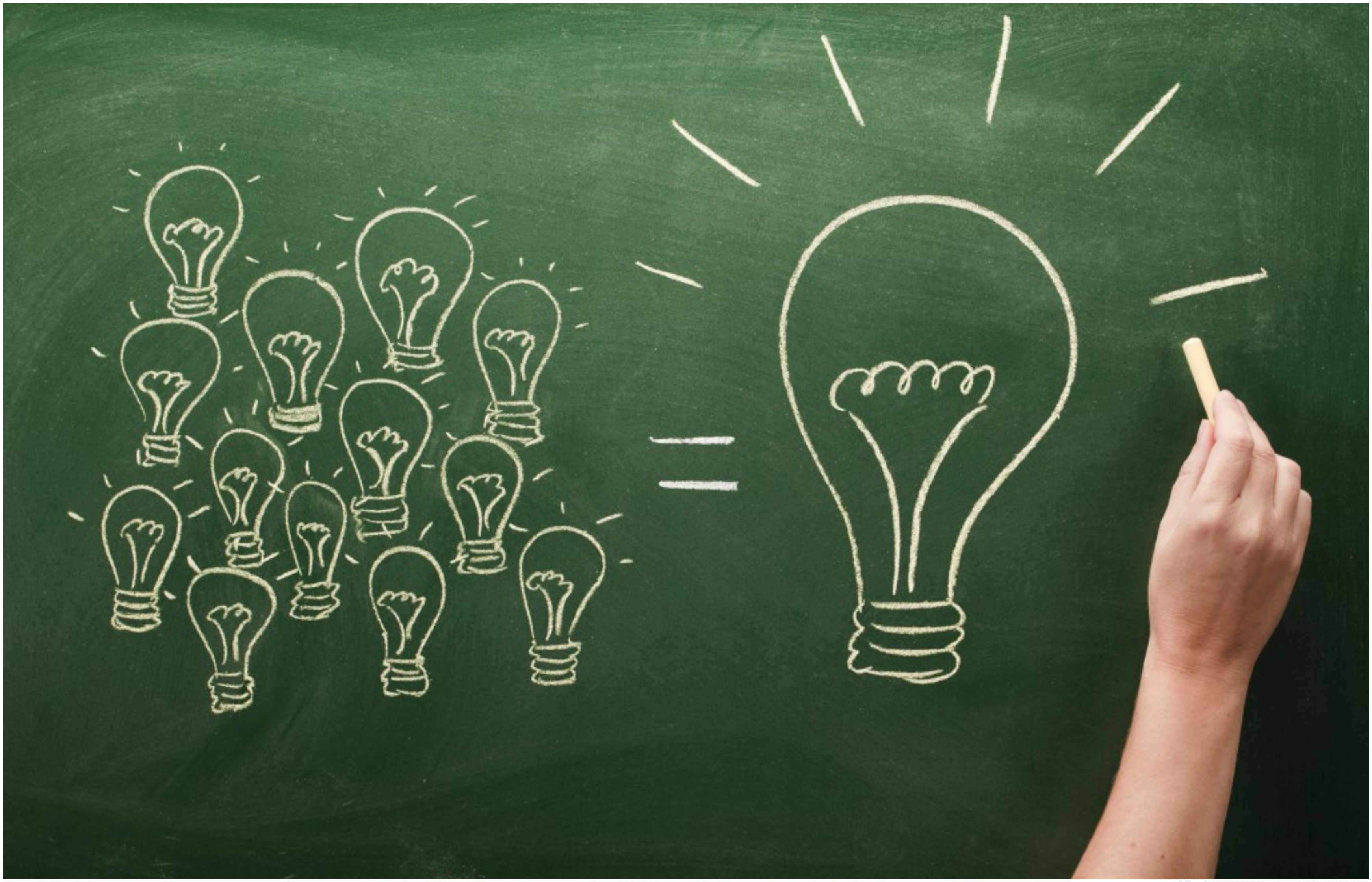
- Poor accuracy
- Over-fitting
- Cannot extrapolate
- Inefficiently fits linear relationships

Ensemble Models

- Ensemble methods combine multiple models
- Parallel ensembles
  - Each model is built **independently**
  - Combine many models to reduce variance
  - e.g. **random forest**
- Sequential ensembles
  - Models are generated **sequentially**
  - Try to add new models that do well where previous models lack
  - e.g. gradient boosting machine

https://davidrosenberg.github.io/mlcourse/Lectures/9b.adaboost.pdf

Power of the crowds



http://www.scaasymposium.org/portfolio/part-v-the-power-of-innovation-and-the-market/

Why does it work?

- Suppose there are 25 decision trees
- Each tree has error rate, $\varepsilon = 0.35$
- Assume independence among trees
- Probability that the combined tree makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{\mathrm{i}} (1-\varepsilon)^{25-i} = 0.07 = \varepsilon / \sqrt{25}$$

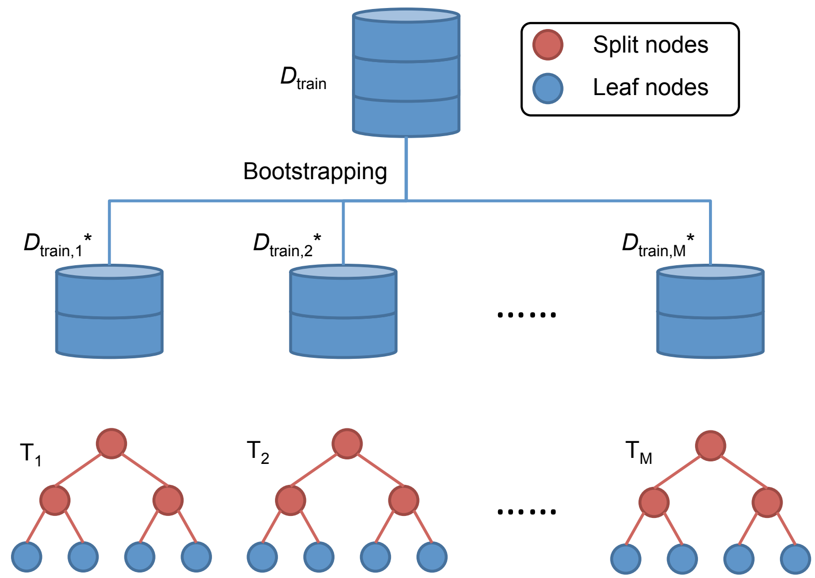
How about for correlated trees ?

- For each Tree T_i with $\mathrm{Var}(T) = \sigma^2$
- If $T_1, \ldots, T_B$ are i.i.d.

$$\mathrm{Var}\left[\frac{1}{B}\sum_{i=1}^{B} T_i\right] = \frac{\sigma^2}{B}$$

- Trees are correlated with $\mathrm{Corr}(T_i, T_j) = \rho$

$$\mathrm{Var}\left[\frac{1}{B}\sum_{i=1}^{B} T_i\right] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$



Reduce the correlation between trees (ρ)

$$f(X) = \frac{1}{B}\sum_{i=1}^{B} T_i(X;\Theta)$$

Breiman, L. *Machine Learning* **2001**, 45, 5-32
Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*, 2nd ed.; Springer New York Inc.: New York, 2009

Bagging (Bootstrap aggregation)

- Reducing the variance
- Regression tree
    - Training based on all data point to get one decision tree for prediction

- Bagging
    - Generated B different training (small) set
    - Each training set is random selected 2/3 data from full training set
    - Build regression tree based on bootstrapped training set
    - Prediction at point x is $f^{*b}(x)$ for *b*-th tree

- Average all the prediction to get

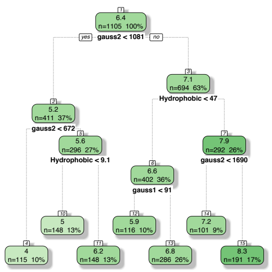
$$f_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} f^{*b}(x)$$

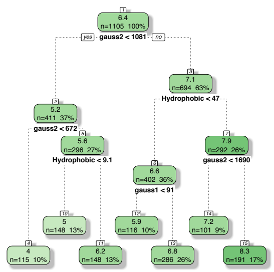
- Lower variance of the prediction

Bagging

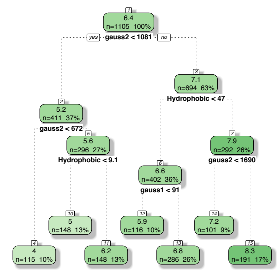
2/3 train data

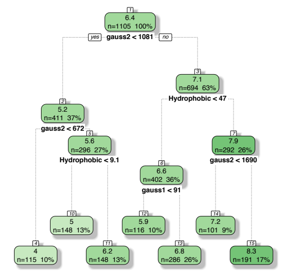
B Trees





...





Predict point x

$f^{*1}(x)$ $f^{*2}(x)$ $f^{*B-1}(x)$ $f^{*B}(x)$

$$f_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} f^{*b}(x)$$

Random Forest

- Bagging
  - Several strong features will be in the top split
  - all the bagged trees will be similar to each other and correlated

- Random forest
  - Improvement over bagged trees by **decorrelating** the trees

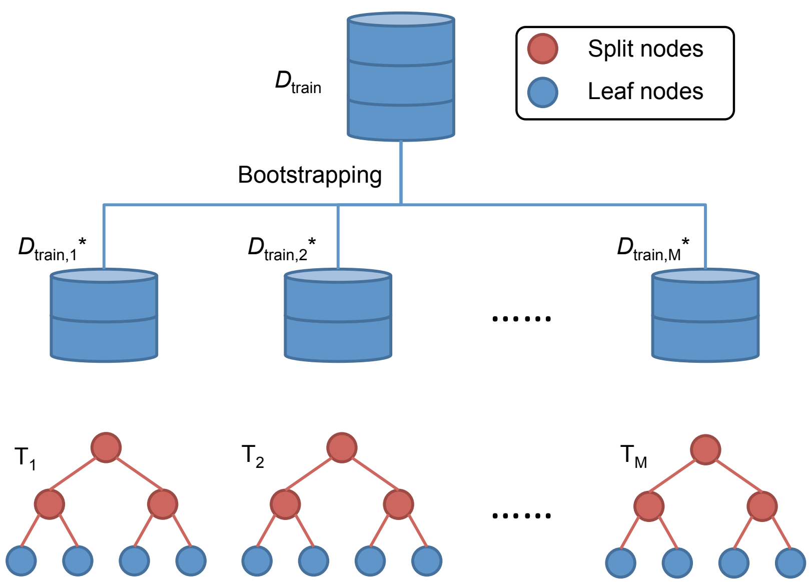
- Suppose we have p features
- Random pick m (<p) features as candidates for splitting each node

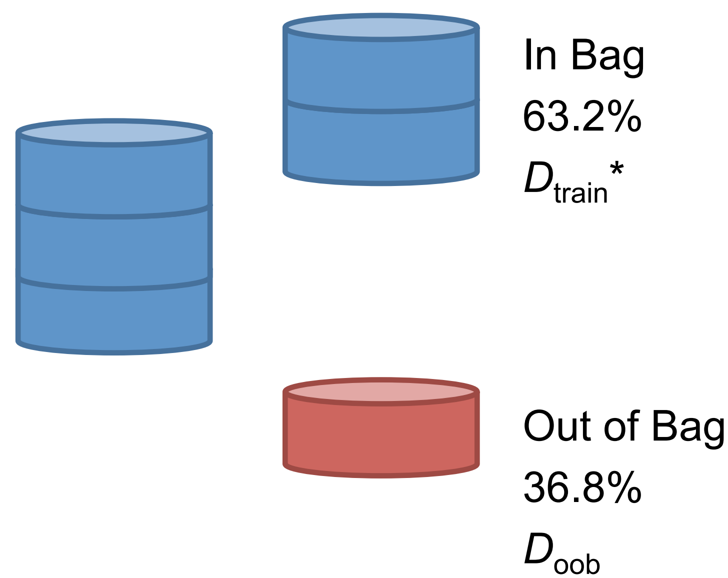
Randomization in Random Forest

Reduce the correlation between trees (ρ)

Randomization

1. **Data: bootstrap samples(bagging)**
2. Tree build: random selection of m variable to split each node





OOB can be used to evaluate the model, and it is similar to CV

Randomization in Random Forest

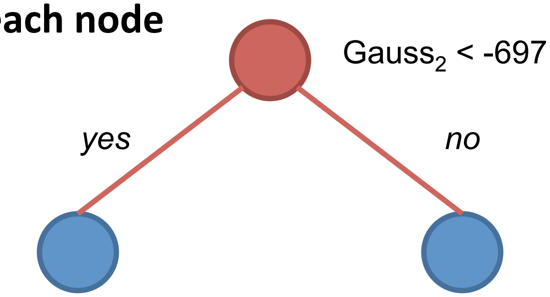
Reduce the correlation between trees (ρ)

Randomization

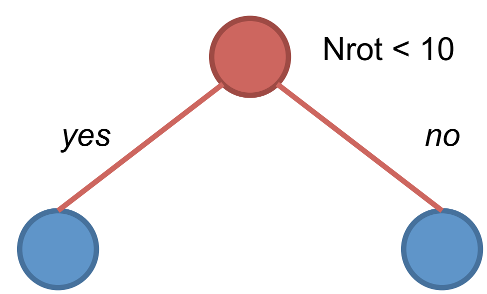
1. Data: bootstrap samples
2. **Tree build: random selection of *m* variable to split each node**

Feature	RSS	*s*
gauss1	12744	-69
gauss2	**12378**	**-697**
Repulsion	14859	-1.24
Hydrophobic	12524	-16.10
HBond	15034	-0.09
Nrot	14358	10

m = 6

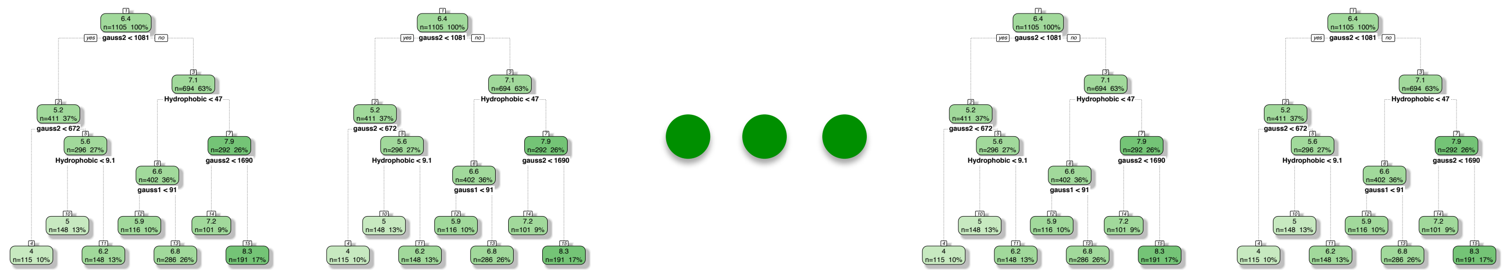


m = 2



Random Forest

B Trees



Predict point x

$f^{*1}(x)$ $f^{*2}(x)$ $f^{*B-1}(x)$ $f^{*B}(x)$

$$f_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} f^{*b}(x)$$

- Each tree is build on 2/3 (random) of train data points and each node is split by (random) p features

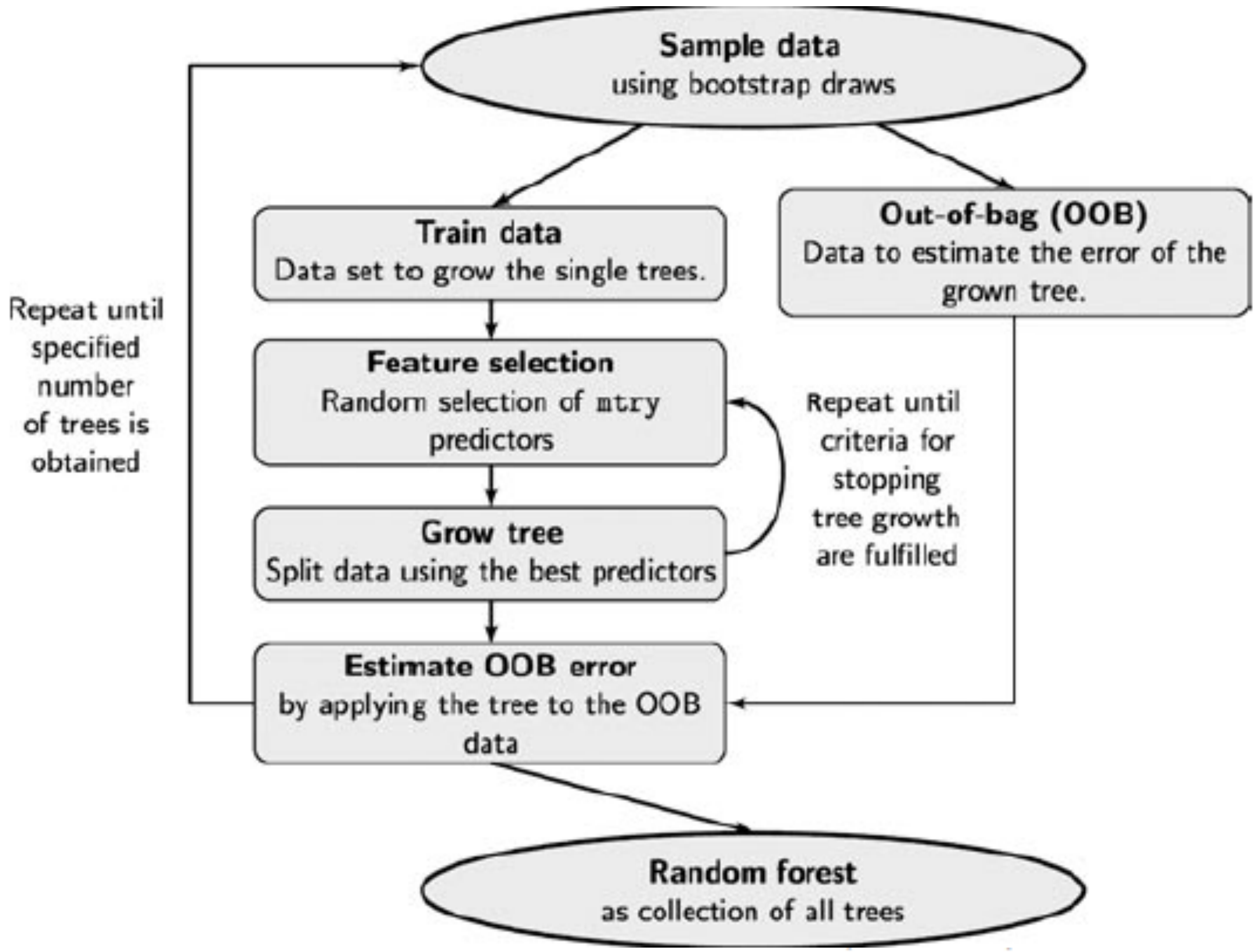
- Out of Bag (OOB): predict y on 1/3 of train data points not used in building tree. This is similar to cross validation.

Breiman, L. *Machine Learning* **2001,** 45, 5-32

Out-of-Bag Error Estimation

- Remember, in bootstrapping we sample with replacement, and therefore **not all observations are used for each bootstrap sample**. On average 1/3 of them are not used!
- Out-of-bag samples (OOB)
- Can predict the response for the i-th observation using each of the trees in which that observation was OOB and do this for *n* observations
- Calculate overall OOB MSE (Similar to leave-one-out cross validation)

Random Forest Algorithm



WIREs Data Mining Knowl Discov 2012, 2: 493–507 doi: 10.1002/widm.1072

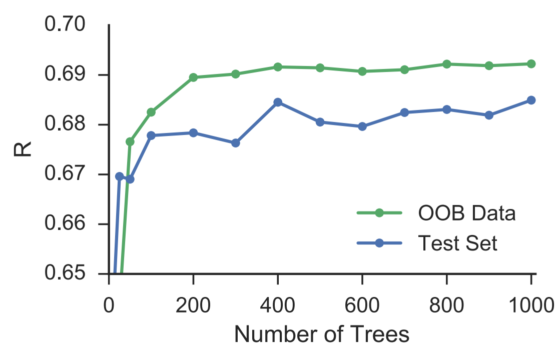
Choice of Parameters

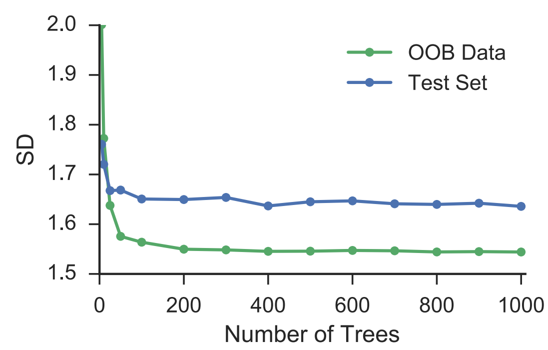
- Number of Trees (The default value is ~ 500)
- Number of Candidate features (m_{try}, a default value is p/3 for regression)
- Size of Trees

Much less parameters than other ML algorithms.

Number of trees

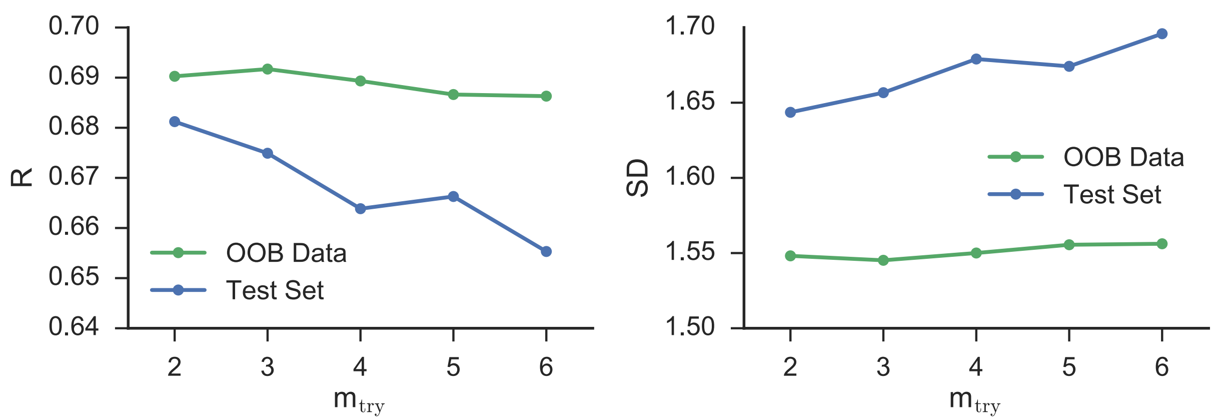
- Should increase with the number of candidate features. Stable after enough trees.
- A larger value always yield more reliable results than a smaller one.





Number of Candidate features (m_{try})

- A real parameter in RF: its optimal value depends on the data at hand
- A default value is p/3 for regression.



Size of Trees

- Tuning parameters but their influence on the results is expected to be lower than m_{try}

1. The minimal size that a node should have to split.
2. The maximal number of layers
3. A threshold value for the splitting criterion
4. Minimal size of leaves

Feature Importance

- Permutation importance indices: The increasing in mean square error when the observed values of this feature are randomly permuted in the OOB samples.

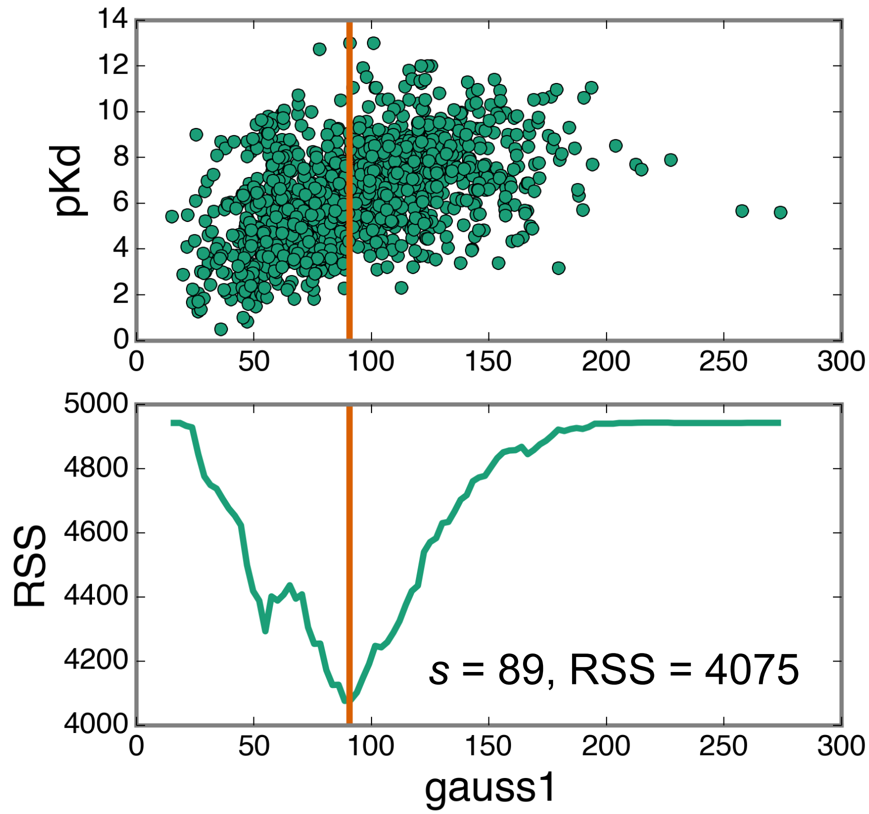
$$\%\mathrm{IncMSE}_i = \frac{\mathrm{MSE}_i^{\mathrm{OOB}} - \mathrm{MSE}^{\mathrm{OOB}}}{\mathrm{MSE}^{\mathrm{OOB}}} \times 100\%$$

- Gini indices: decrease of RSS during the tree splitting. (can be normalized)

Reduction in Variance of Sub-Nodes

- Each feature X_j
  - Find the cut-point s with lowest RSS
- Select the feature have lowest RSS

Feature	RSS	s
gauss1	4075	89
gauss2	3980	1081
Replusion	4838	3.6
Hydrophobic	4131	9.7
HBonding	4880	2.0
Nrot	4668	6.5



$$RSS = \sum_{i:x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2$$

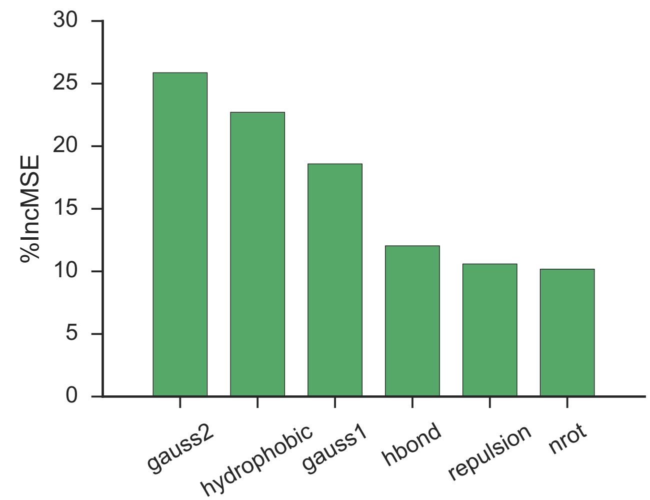
Random Forest: a popular machine learning algorithm

Random Forest advantages as a ML algorithm

- ❑ performs remarkably well with very little tuning required
- ❑ handles large feature set and correlated features
- ❑ is used not only for prediction, but also to access feature importance

Feature Importance

$$\%\mathrm{IncMSE}_i = \frac{\mathrm{MSE}_i^{\mathrm{OOB}} - \mathrm{MSE}^{\mathrm{OOB}}}{\mathrm{MSE}^{\mathrm{OOB}}} \times 100\%$$



Breiman, L. *Machine Learning* **2001**, 45, 5-32
Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*, 2nd ed.; Springer New York Inc.: New York, 2009

AutoDock Vina (Performance)

Vina 6	Train (3336)		Test (195)	
Model	R_p	SD	R_p	SD
Original	0.520	1.83	**0.567**	1.85
Linear Reg	0.573	1.75	**0.627**	1.75
Reg Tree (2)	0.543	1.80	**0.560**	1.86
Reg Tree (20)	0.920	0.84	**0.462**	1.99
Random Forest	0.690*	1.55*	**0.686**	1.63

*The result is from out of bag prediction

Further reading

- The Elements of Statistical Learning

  Trevor Hastie, Robert Tibshirani, Jerome Friedman

  http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf

- Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics

  Boulesteix et al

  WIREs Data Mining Knowl Discov 2012, 2: 493–507 doi: 10.1002/widm.1072

Acknowledgement

Dr. Cheng Wang