DS 4400

Machine Learning and Data Mining I

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

October 29 2020

Outline

- Ensemble learning
- Bagging
  - Bootstrap samples
  - Random Forest algorithm
- Boosting
  - General method
  - AdaBoost algorithm

Ensemble Learning

REGRESSION — AVERAGE — WEIGHTED AVERAGE

Consider a set of classifiers $h_1, ..., h_L$

Idea: construct a classifier $H(\mathbf{x})$ that combines the individual decisions of $h_1, ..., h_L$

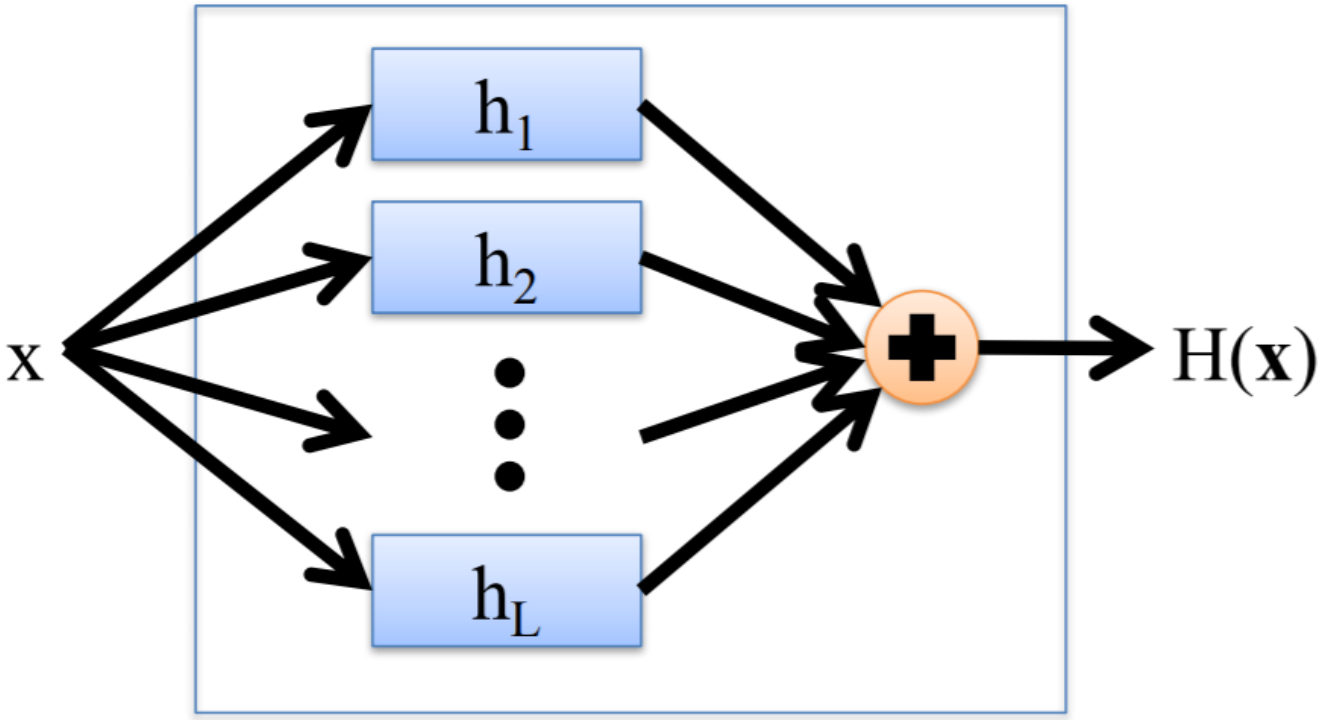
- e.g., could have the member classifiers vote, or
- e.g., could use different members for different regions of the instance space

CLASSIFICATION — VOTING (WEIGHTED) — AVG. OF PROB

Successful ensembles require **diversity**

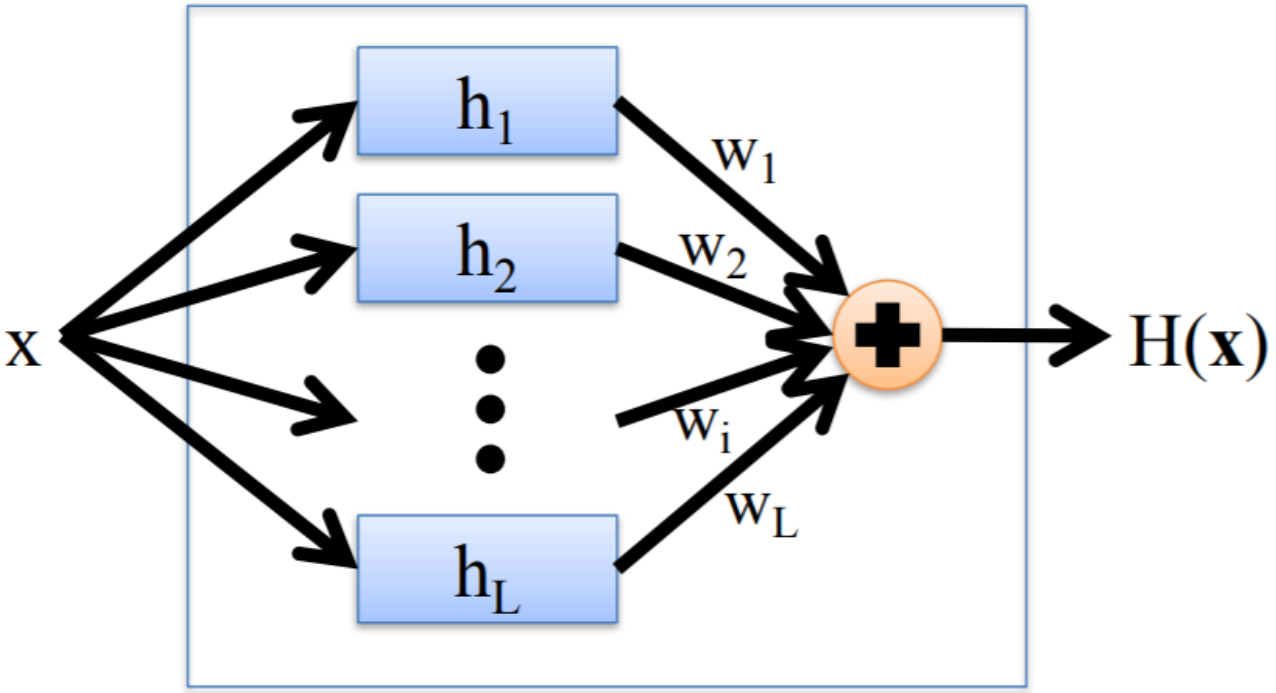
- Classifiers should make different mistakes
- Can have different types of base learners

Combining Classifiers: Averaging



- Final hypothesis is a simple vote of the members

Combining Classifiers: Weighted Averaging



- Coefficients of individual members are trained using a validation set

Ensembles Reduce Error

- Suppose there are 25 base classifiers
- Each classifier has error rate, $\varepsilon = 0.35$
- Assume independence among classifiers
- Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{\mathrm{i}} (1-\varepsilon)^{25-i} = 0.06$$

Ensembles Reduce Variance

$X_1, \ldots, X_n$ INDEPENDENT OF CLASSIFIER

$E[X_i] = \mu$

$Var[X_i] = \sigma^2$

$\bar{X} = \frac{X_1 + \ldots + X_n}{n}$

$Var(X+Y) = Var(X) + Var(Y)$

$Var(c \cdot X) = c^2 Var(X)$

$Var(X) = E(X^2) - E^2(X)$

$$E[\bar{X}] = E\left[\frac{X_1 + \ldots + X_n}{n}\right] = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = \mu$$

$$Var[\bar{X}] = Var\left[\frac{X_1 + \ldots + X_n}{n}\right] = \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i) = \frac{\sigma^2}{n}$$

↓

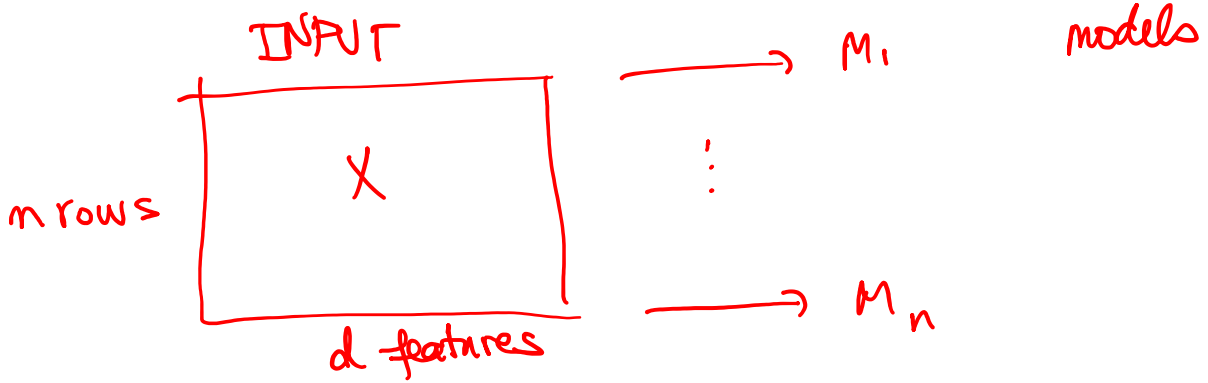
REDUCTION IN VARIANCE BY A FACTOR BY n

How to Achieve Diversity

INPUT: TRAINING DATA

1) VARY THE TRAINING DATA

2) VARY THE FEATURE SET



How to Achieve Diversity

- Avoid overfitting
  - Vary the training data
- Features are noisy
  - Vary the set of features

Two main ensemble learning methods

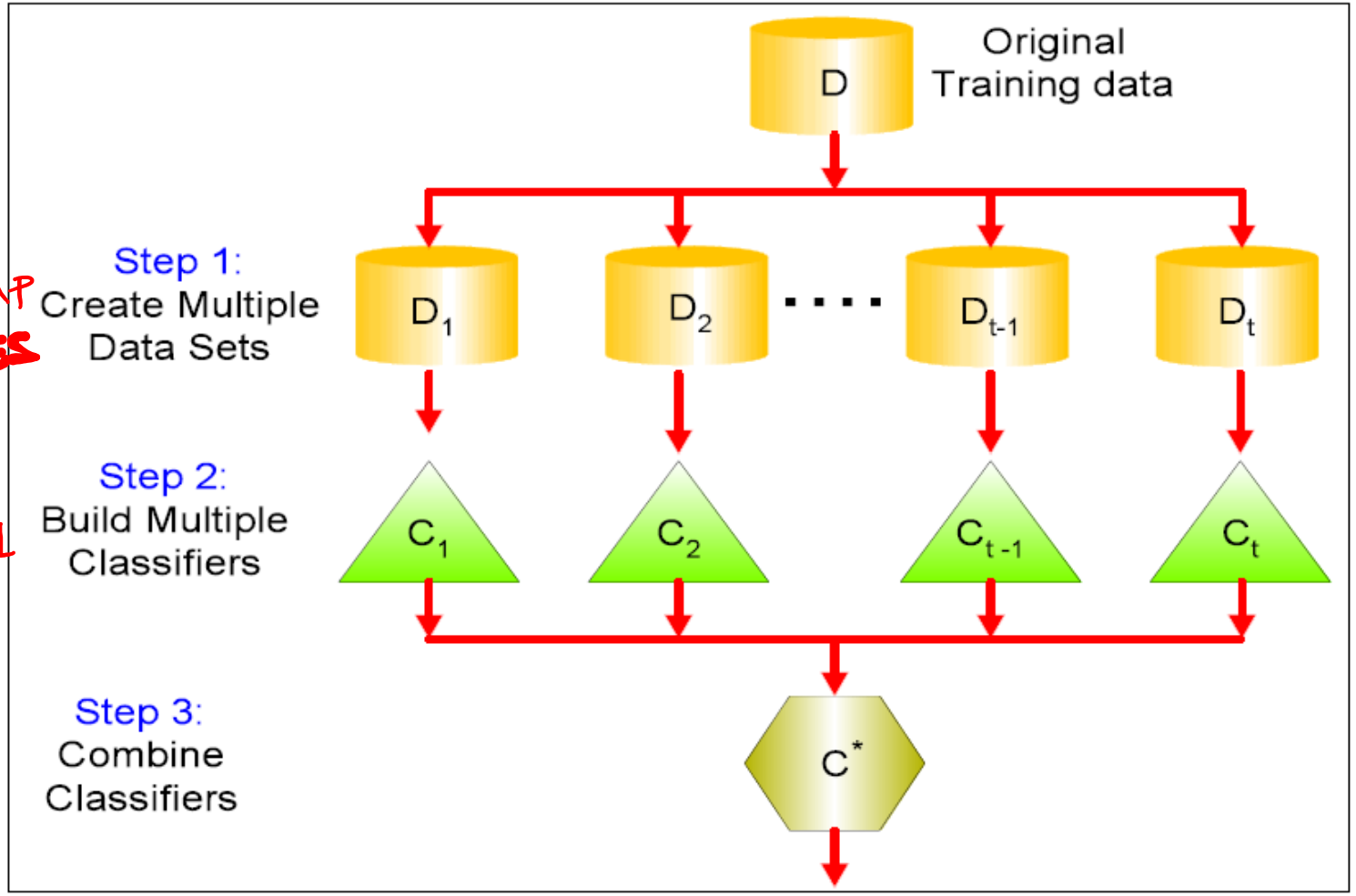
- Bagging (e.g., Random Forests)
- Boosting (e.g., AdaBoost)

Bagging

- Leo Breiman (1994)
- Take repeated bootstrap samples from training set *D*
- *Bootstrap sampling*: Given set *D* containing *N* training examples, create *D'* by drawing *N* examples at random with replacement from *D*.

- Bagging:
  - Create *k* bootstrap samples D_1 ... D_k.
  - Train distinct classifier on each D_i.
  - Classify new instance by majority vote / average.

General Idea



BOOTSTRAP SAMPLES

PARALLEL

Majority Votes

Example of Bagging

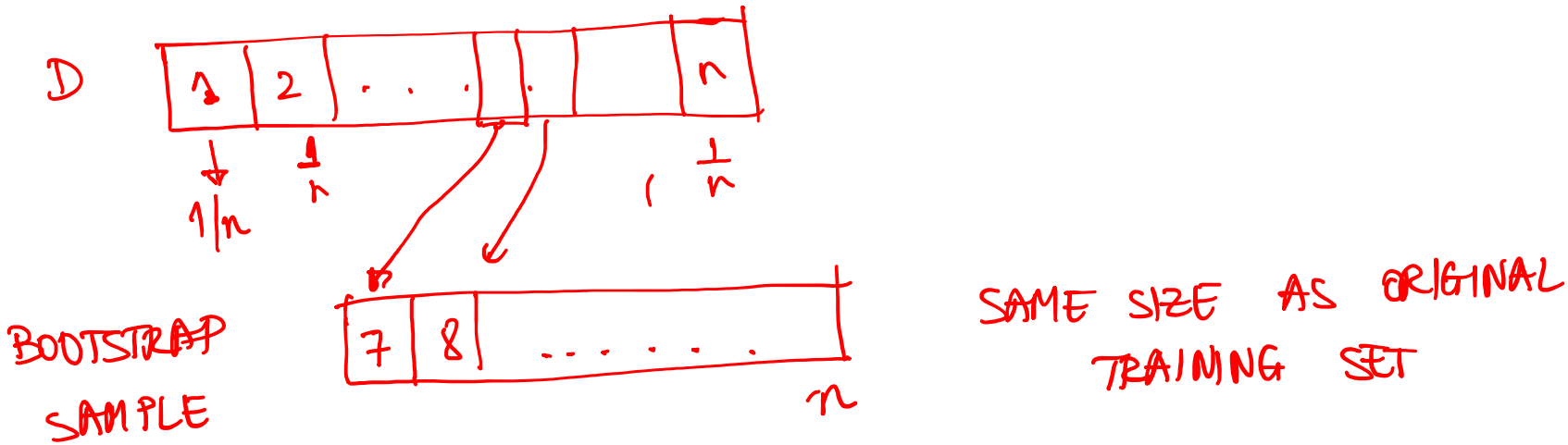
- Sampling with replacement

Training Data

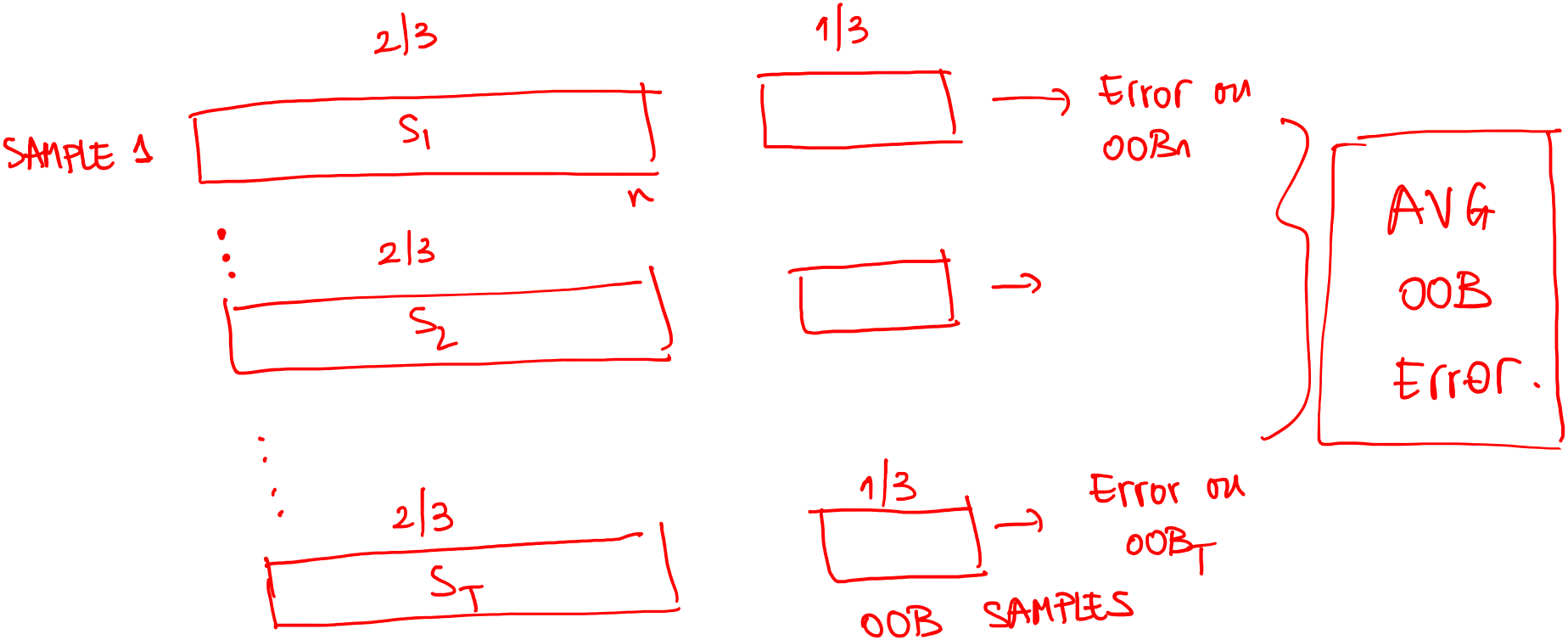
Data ID

	1	2	3	4	5	6	7	8	9	10
Original Data										
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Sample each training point with probability 1/n
- Out-Of-Bag (OOB) observation: point not in sample



Bootstrap Samples

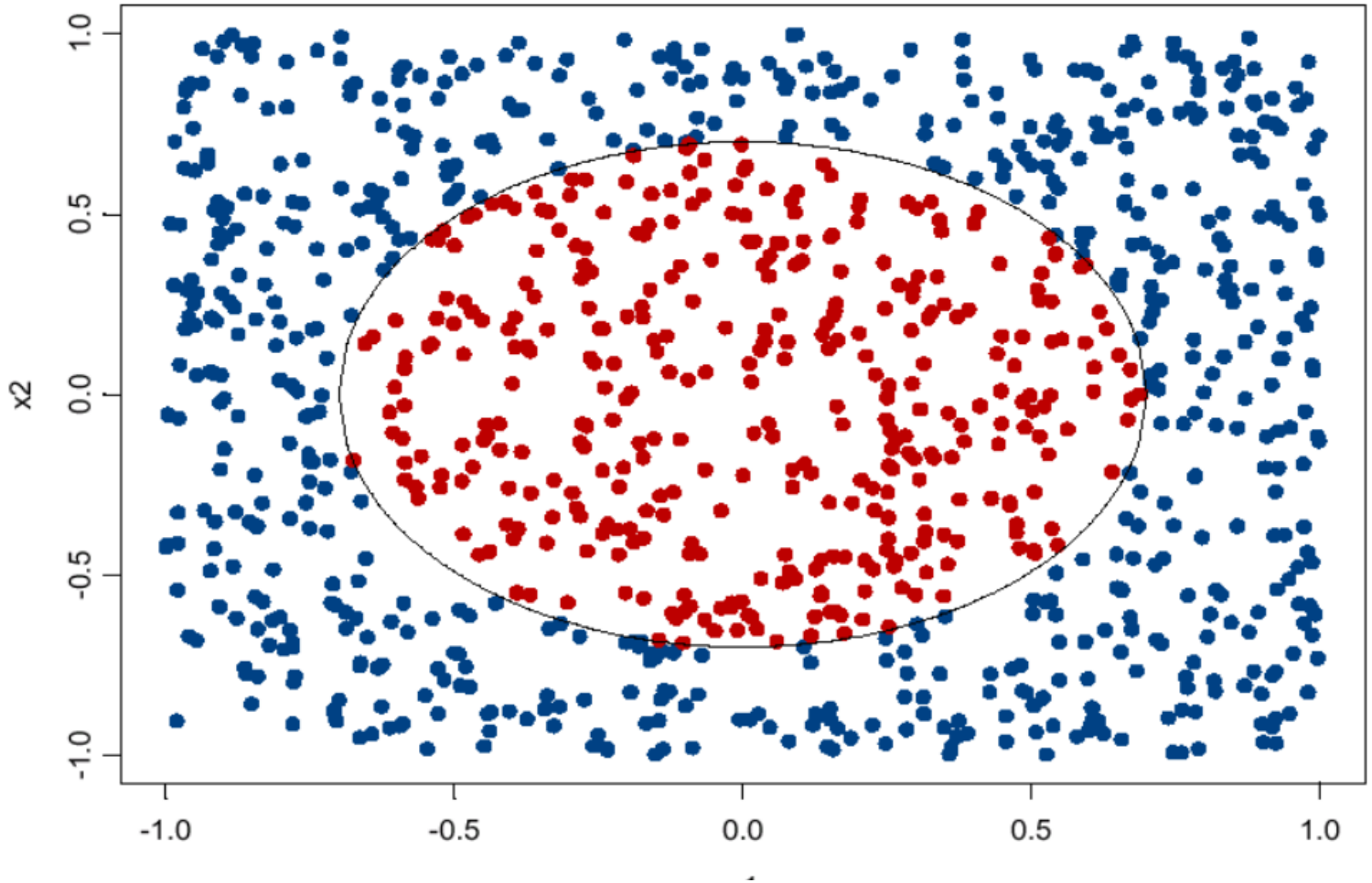


$$P[\text{TRAINING EXAMPLE } i \text{ IS NOT IN } S_1] = \left(1 - \frac{1}{n}\right)^n \simeq \frac{1}{e} = \frac{1}{2.71}$$

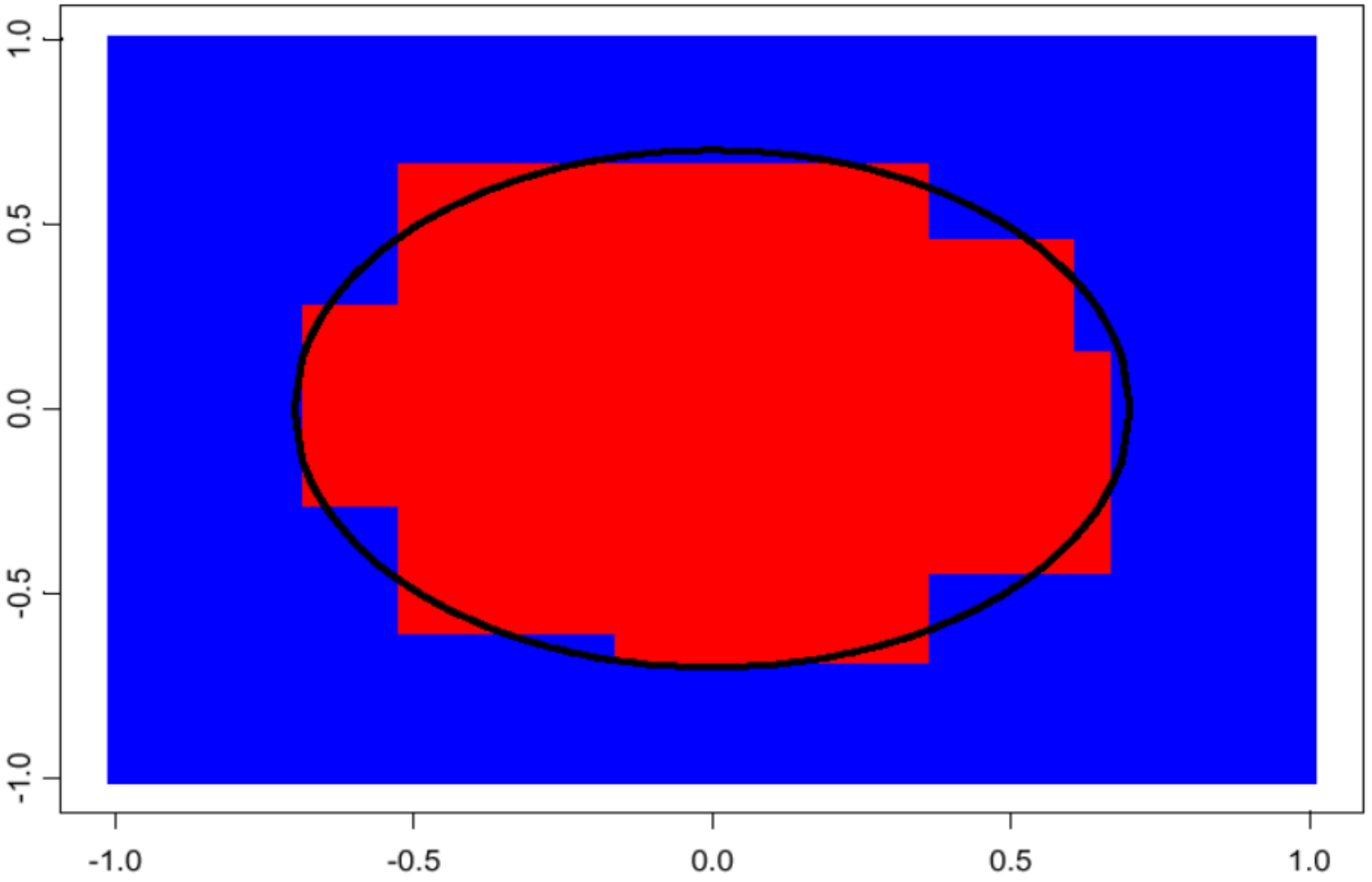
Bagging

- Can be applied to multiple classification models
- Very successful for decision trees
  - Decision trees have high variance
  - Don't prune the individual trees, but grow trees to full extent
  - Precision accuracy of decision trees improved substantially
- OOB average error used instead of Cross Validation

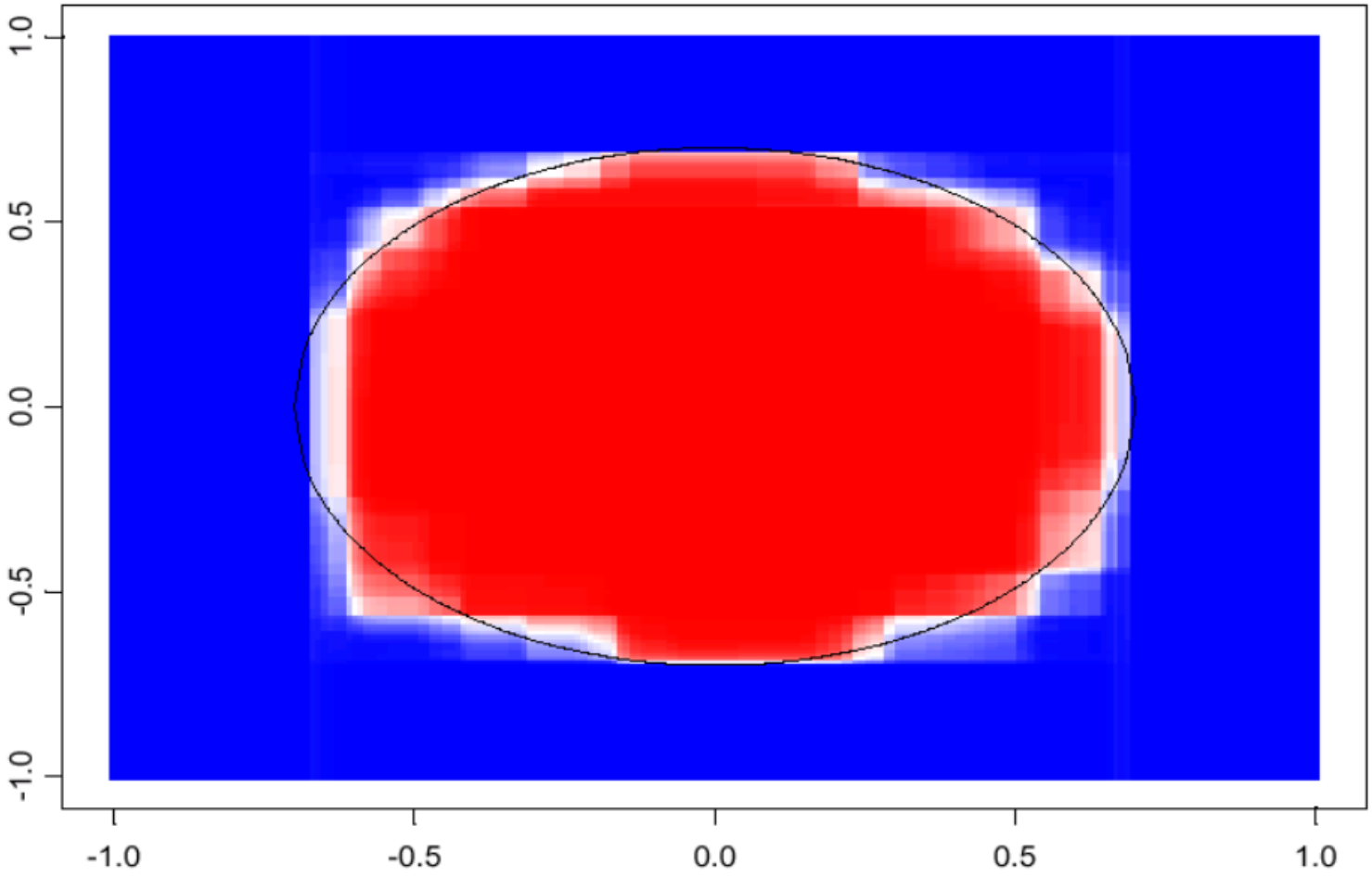
Example Distribution



Decision Tree Decision Boundary



100 Bagged Trees



shades of blue/red indicate strength of vote for particular classification

Random Forests

- Ensemble method specifically designed for decision tree classifiers

- Introduce two sources of randomness: "Bagging" and "Random input vectors"
  - Bagging method: each tree is grown using a bootstrap sample of training data
  - Random vector method: At each node, best split is chosen from a random sample of *m* attributes instead of all attributes

VARY TRAINING

VARY FEATURES

m — HYPER-PARAMETER
AT EACH SPLIT: - SUBSET OF m FEATURES
FOR EVERY TREE

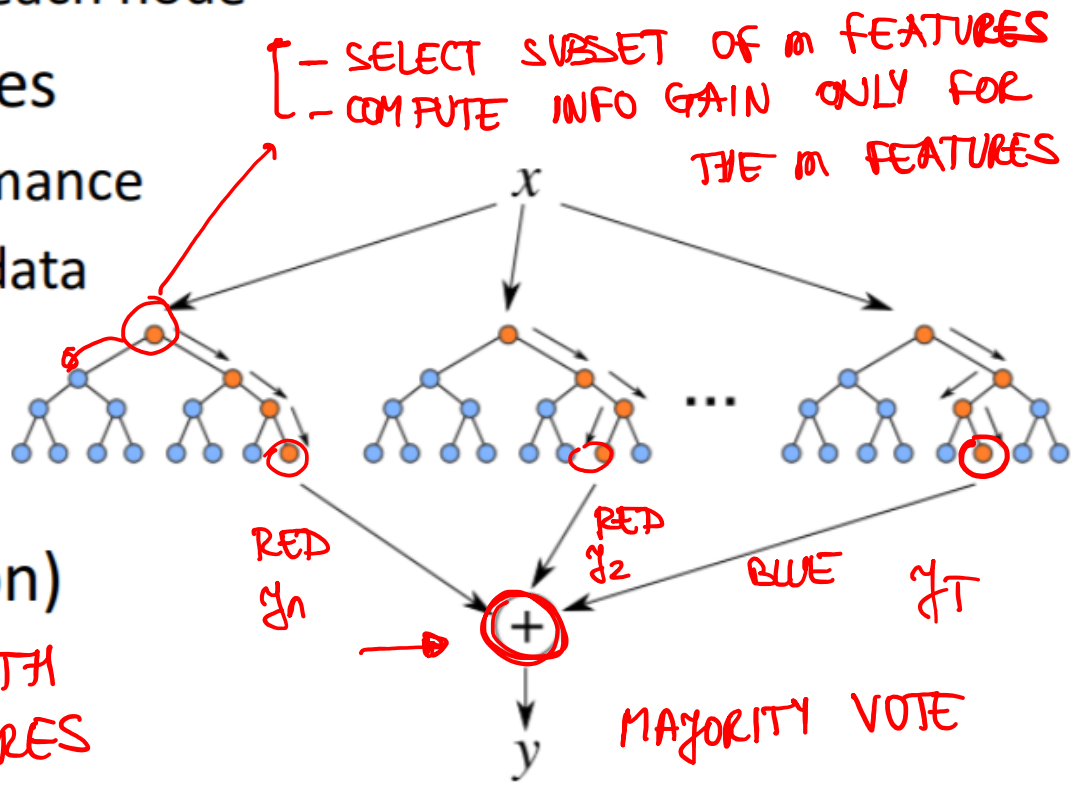
Random Forests

TRAIN

- Construct decision trees on bootstrap replicas
  - Restrict the node decisions to a small subset of features picked randomly for each node
- Do not prune the trees
  - Estimate tree performance on out-of-bootstrap data

TESTING

- Average the output of all trees (or choose mode decision)



NOT THE SAME WITH SELECTING m FEATURES AND TRAINING ON THOSE

Random Forest Algorithm

TRAIN:

1. For $b = 1$ to B: ONE ITERATION PER MODEL (DECISION TREE

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a ~~random forest~~ tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

   DECISION TREE TRAINING

   i. Select m variables at random from the d variables.

   ii. Pick the best variable/split-point among the m.

   iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$. → SET OF DECISION TREES

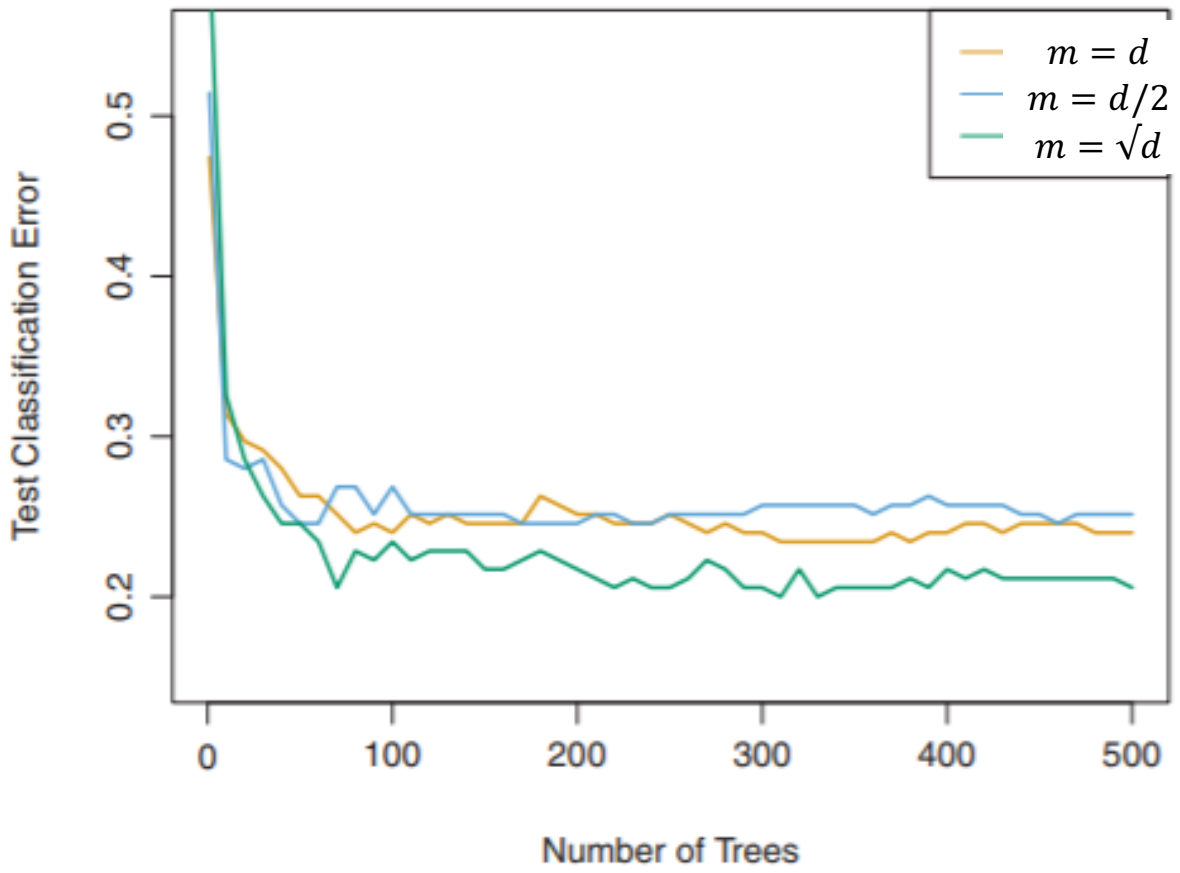
TESTING: To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$. AVERAGE

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

- $m = d$ => BAGGED DECISION TREES

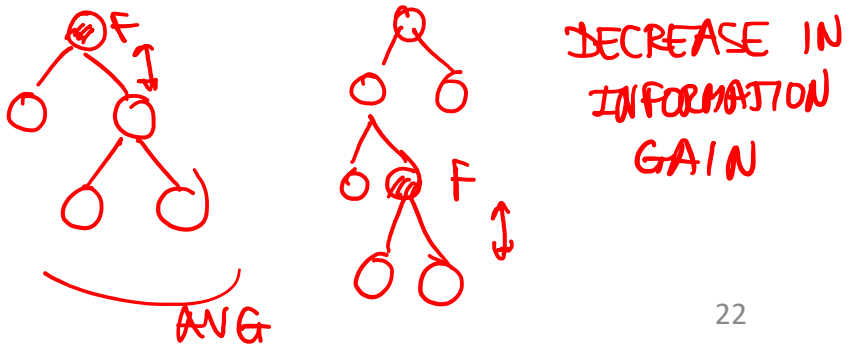
Effect of Number of Predictors



- d = total number of predictors; m = predictors chosen in each split
- Random Forests uses $m = \sqrt{d}$

Variable Importance

- Ensemble of trees looses somewhat interpretability of decision trees
- Which variables contribute mostly to prediction?
- Random Forests computes a Variable Importance metric per feature
  - For each tree in the ensemble, consider the split by the particular feature
  - How much information gain / Gini index decreases after the split
  - Average over all trees



Variable Importance Plots

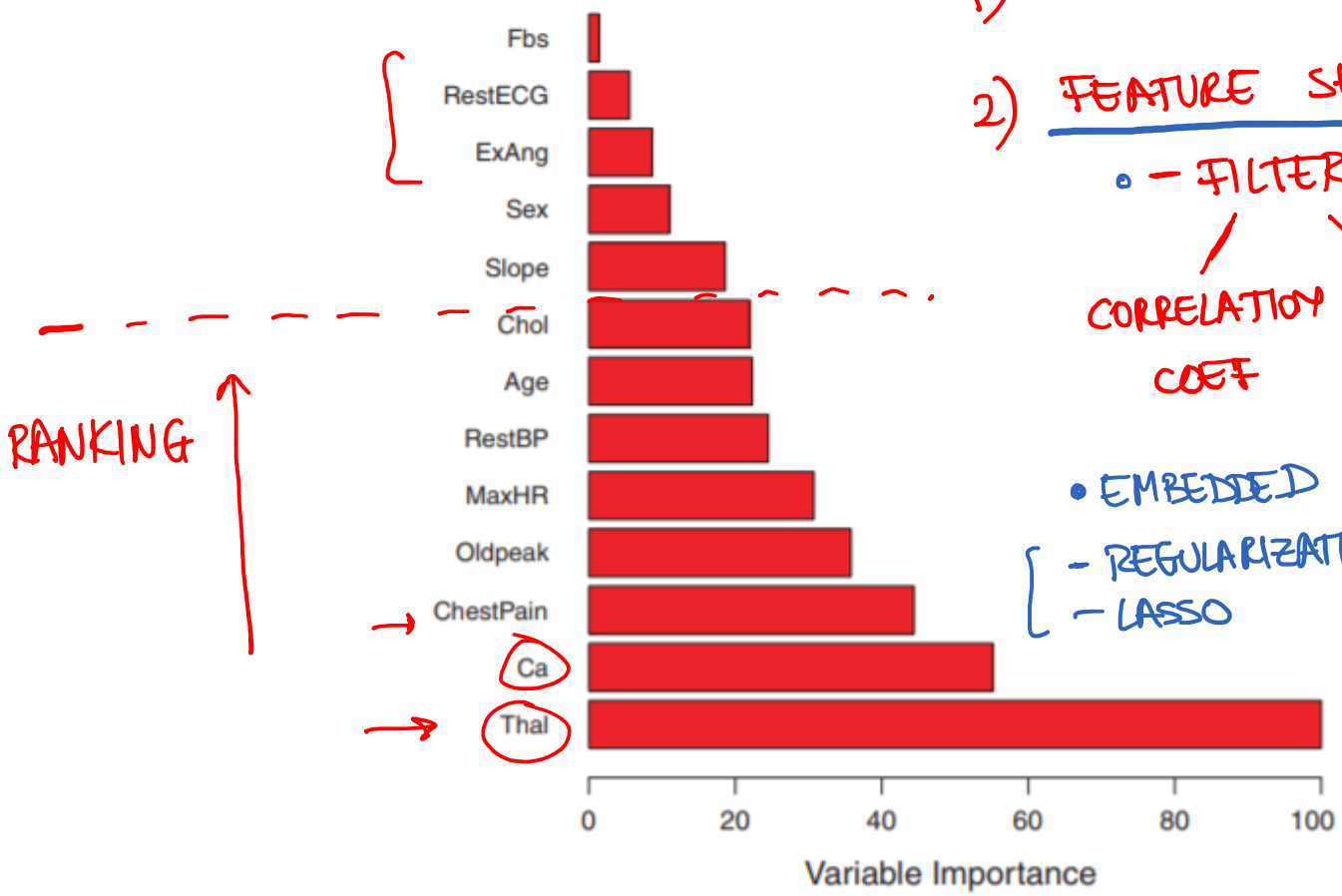


FIGURE 8.9. *A variable importance plot for the* `Heart` *data. Variable importance is computed using the mean decrease in Gini index, and expressed relative to the maximum.*

PROJECT PROPOSAL (1 PAGE)

MONDAY NOV. 2

- TITLE
- TEAM
- PB DESCRIPTION — REGRESSION / CLASSIFICATION
- DATASET — LINK ; DESCRIPTION ; FEATURES ; CHARACTERIZATION — NUMERICAL / CATEGORICAL
- METHODOLOGY — FEATURE SELECTION
  - MODELS — LINEAR / GENERATIVE / ENSEMBLE (BAGGING | BOOSTING)
- LANG. | PACKAGES
- METRICS
- REFERENCES (RELATED WORK).