Random Forest

Instructor: Junghye Lee

Department of Industrial Engineering
junghyelee@unist.ac.kr

Contents

1. Background
2. Bagging Predictors
3. Random Forest
4. Variable Importance

Decision Tree - Advantages

- Applicable to both regression and classification problems.
- Handle categorical predictors naturally.
- Computationally simple and quick to fit, even for large problems.
- No formal distributional assumptions (non-parametric).
- Can handle highly non-linear interactions and classification boundaries.
- Automatic variable selection.
- Handle missing values through surrogate variables.
- Very easy to interpret if the tree is small.

Decision Tree - Disadvantages

- Accuracy - current methods, such as support vector machines and ensemble classifiers often have 30% lower error rates than CART.
- Instability – if we change the data a little, the tree picture can change a lot. So the interpretation is not as straightforward as it appears.

We can do better!

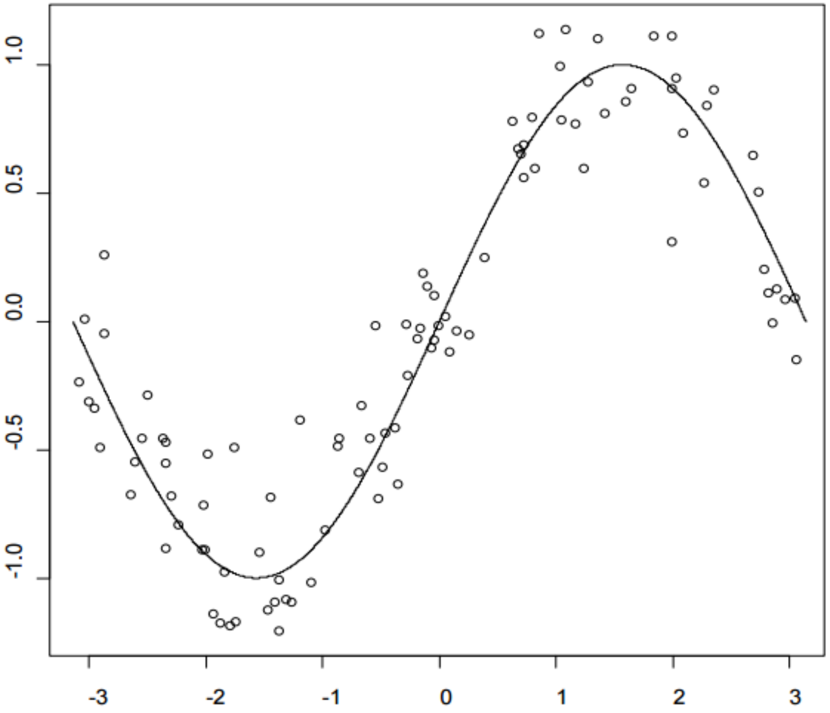
Random Forest

Outline

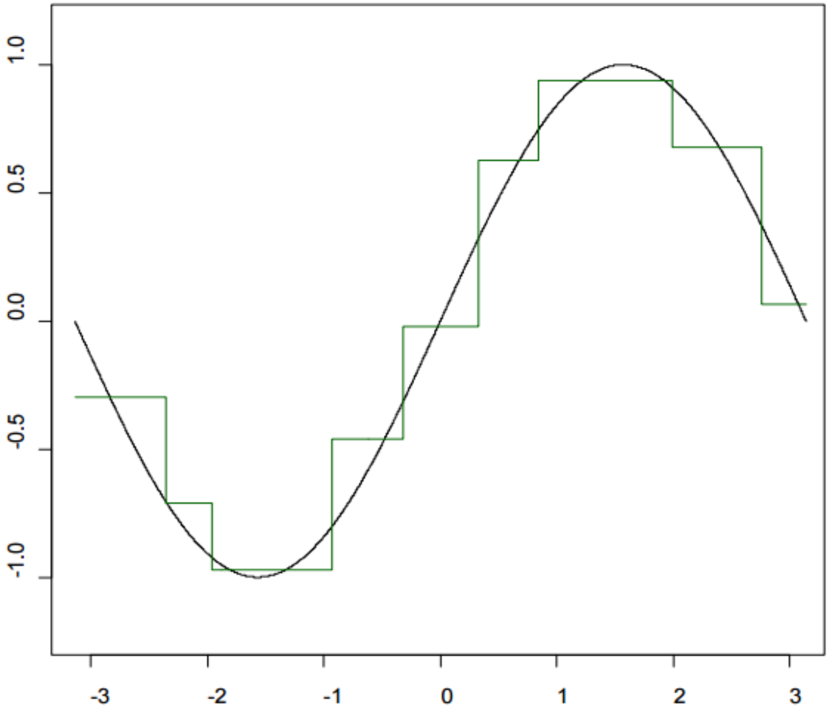
- **Bagging predictors**
- Random forest
- Variable importance

Aggregation of Regression Trees

Data and Underlying Function

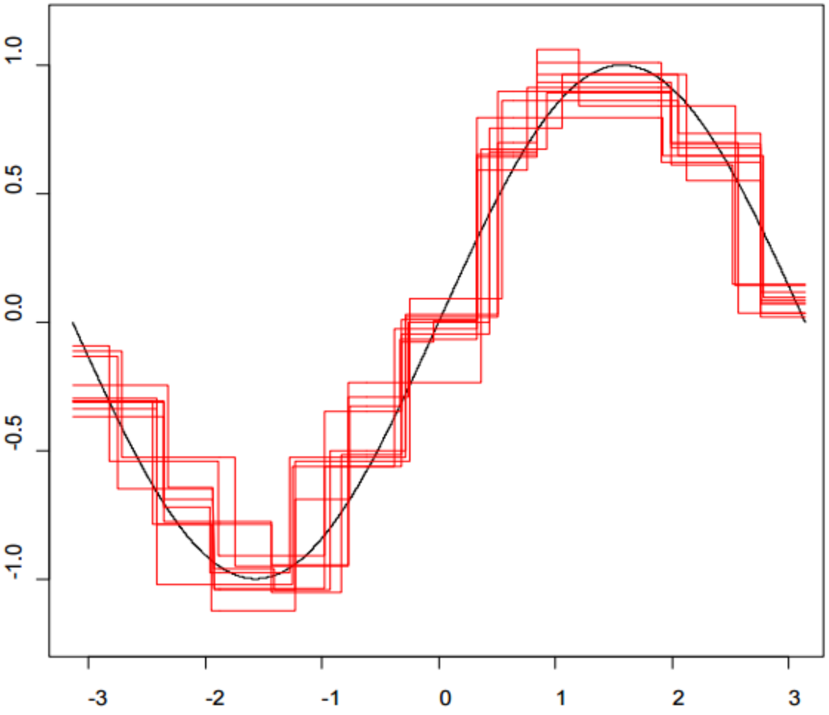


Single Regression Tree

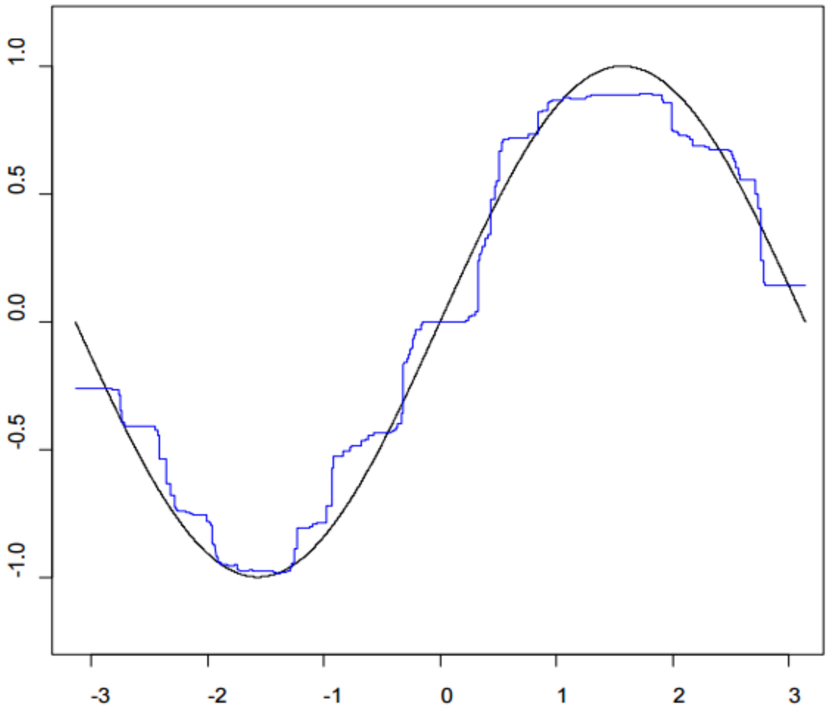


Aggregation of Regression Trees

10 Regression Trees

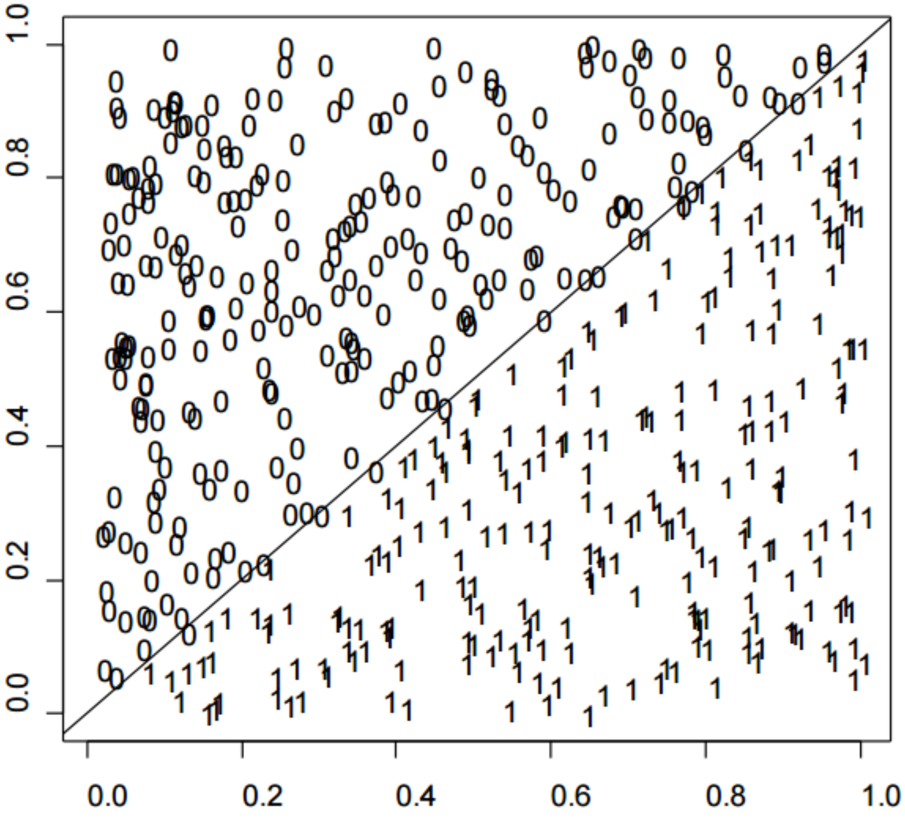


Average of 100 Regression Trees

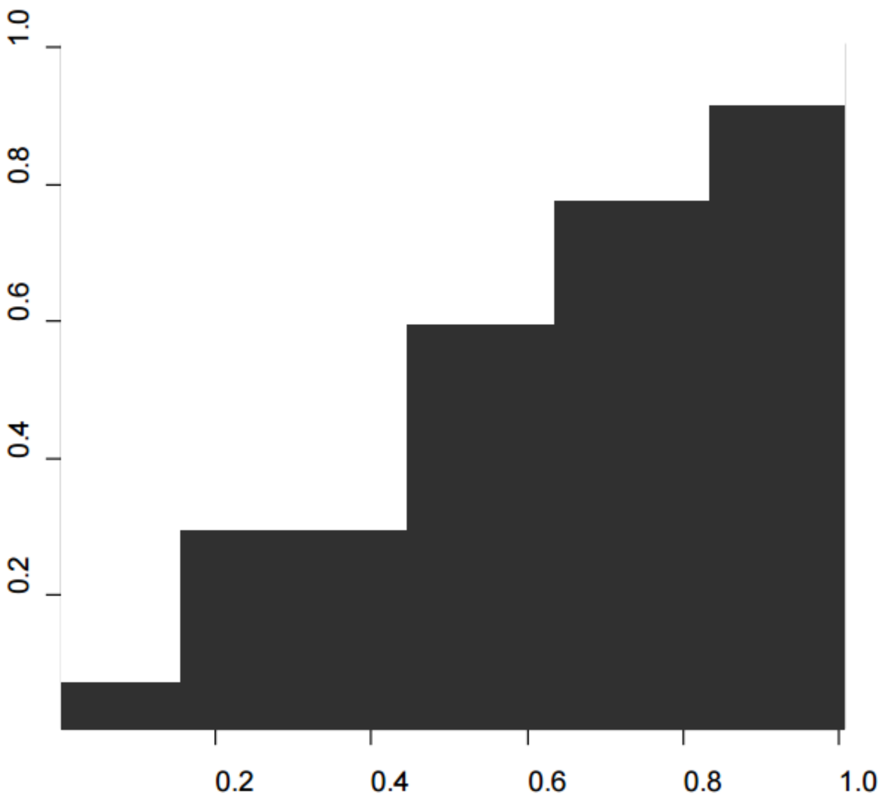


Aggregation of Classification Trees

Data and Underlying Function

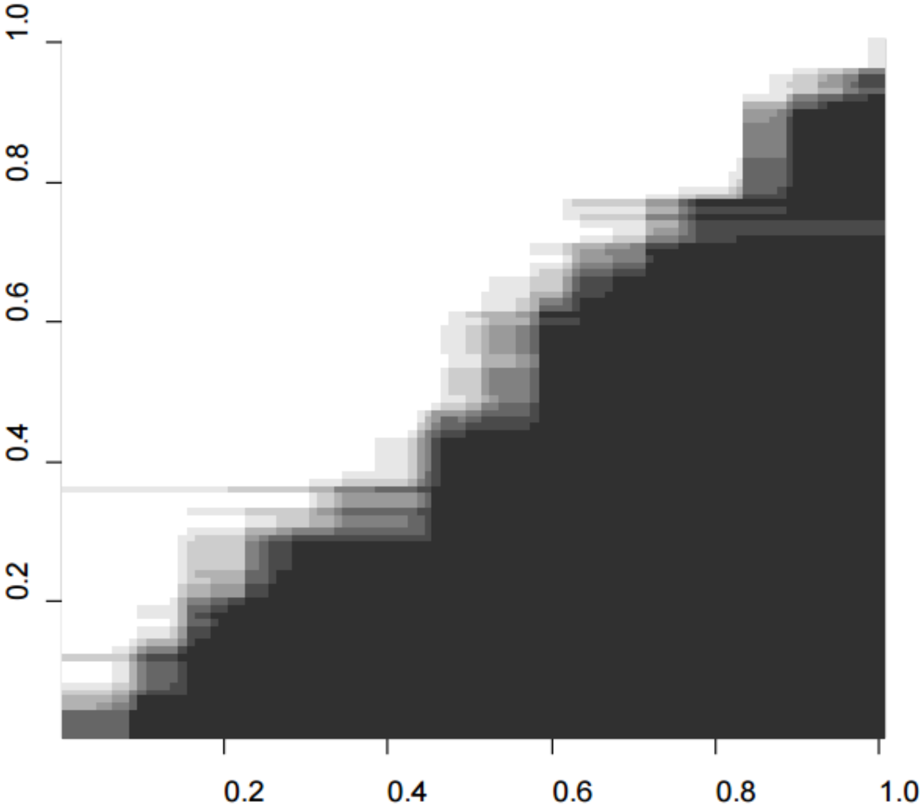


Single Classification Tree

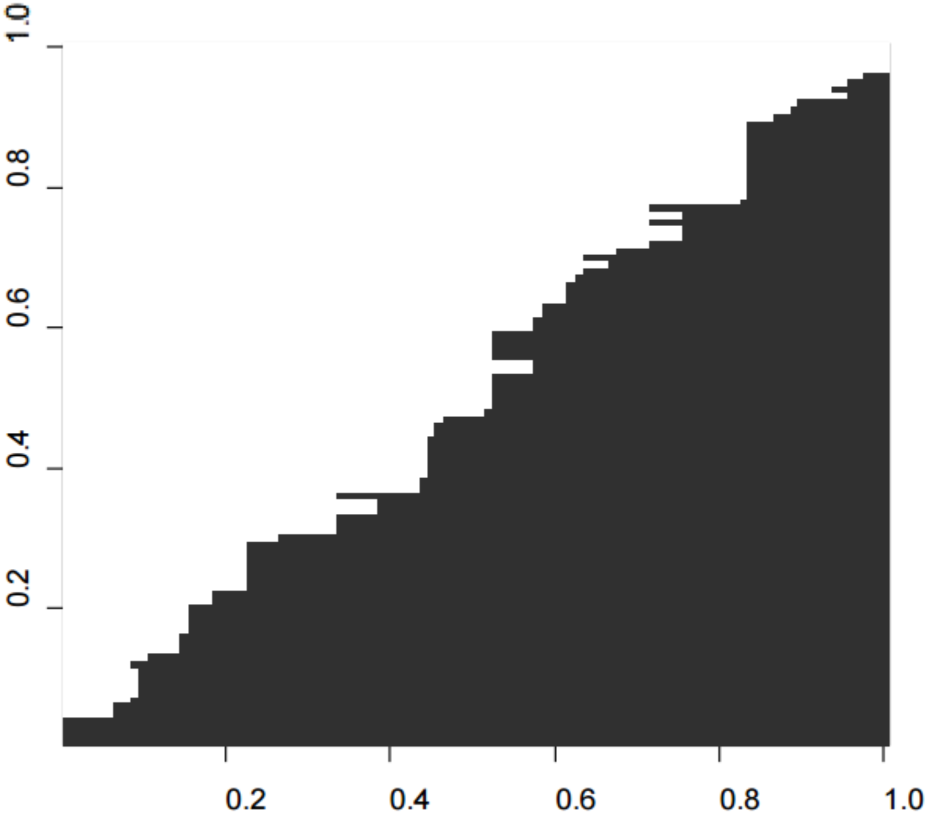


Aggregation of Classification Trees

25 Averaged Classification Trees



25 Voted Classification Trees



Bootstrap Aggregating (Bagging)

- Bootstrap Aggregating
- Breiman, “Bagging Predictors”, Machine Learning, 1996.

- Fit classification or regression models to bootstrap samples from the data and combine by voting (classification) or averaging (regression).

Bootstrap sample → $f_1(x)$
Bootstrap sample → $f_2(x)$
Bootstrap sample → $f_3(x)$
…
Bootstrap sample → $f_m(x)$

MODEL AVERAGING

Combine $f_1(x)$, …, $f_m(x)$

$f_i(x)s$ are “base learners”

Bagging

- A bootstrap sample is chosen at random with replacement from the data. Some observations end up in the bootstrap sample more than once, while others are not included (“out of bag”).
- Bagging reduces the variance of the base learner but has limited effect on the bias.
- It’s most effective if we use weak base learners that have very little bias but high variance (unstable). E.g. trees.
- Both bagging and boosting are examples of “ensemble learners” that were popular in machine learning in the ‘90s.

Bagging CART

- Leo Breiman (1996) “Bagging Predictors”, Machine Learning, 24, 123-140.

Dataset	# cases	# vars	# classes	CART	Bagged CART	Decrease %
Waveform	300	21	3	29.1	19.3	34
Heart	1395	16	2	4.9	2.8	43
Breast Cancer	699	9	2	5.9	3.7	37
Ionosphere	351	34	2	11.2	7.9	29
Diabetes	768	8	2	25.3	23.9	6
Glass	214	9	6	30.4	23.6	22
Soybean	683	35	19	8.6	6.8	21

Outline

- Bagging predictors
- **Random forest**
- Variable importance

Random Forest

- Grow a **forest** of many trees.
  - R default is 500
- Grow each tree on an independent **bootstrap sample** from the training data.
  - Sample **n cases out of all N cases** at random with replacement.
- At each node:
  1. Select **d variables at random out of all D possible variables** (independently for each node).
  2. Find the best split on the **selected d variables.**
- Grow the trees to maximum depth.
- Vote/average the trees to get predictions for new data.

Random Forest

- Inherit many of the advantages of decision tree:
  - Applicable to both regression and classification problems
  - Handle categorical predictors naturally
  - Computationally simple and quick to fit, even for large problems
  - No formal distributional assumptions (non-parametric)
  - Can handle highly non-linear interactions and classification boundaries
  - Automatic variable selection
  - Handle missing values
- But do not inherit:
  - The picture of the tree cannot give valuable insights into which variables are important and where.
  - The terminal nodes cannot suggest a natural clustering of data into homogeneous groups.

Random Forest

- Improve on decision tree with respect to:

1. Accuracy – Random Forest is competitive with the best known machine learning methods (but note the “no free lunch” theorem that suggests that there is no universally best learning algorithm).
2. Instability – if we change the data a little, the individual trees may change but the forest is relatively stable because it is a combination of many trees.

Why bootstrap? (Why subsample?)

- Diversity of training set
- out-of-bag data (like test set; semi-test set)
  - Estimated error rate and confusion matrix
  - Variable importance

The RF Predictor

- A case in the training data is ***not in the bootstrap sample for about one third of the trees*** (we say the case is “out of bag” or “oob”). Vote (or average) the predictions of these trees to give the RF predictor.

- For example, suppose we fit 1000 trees, and a case is out-of-bag in 339 of them, of which:

  283 say “class 1”

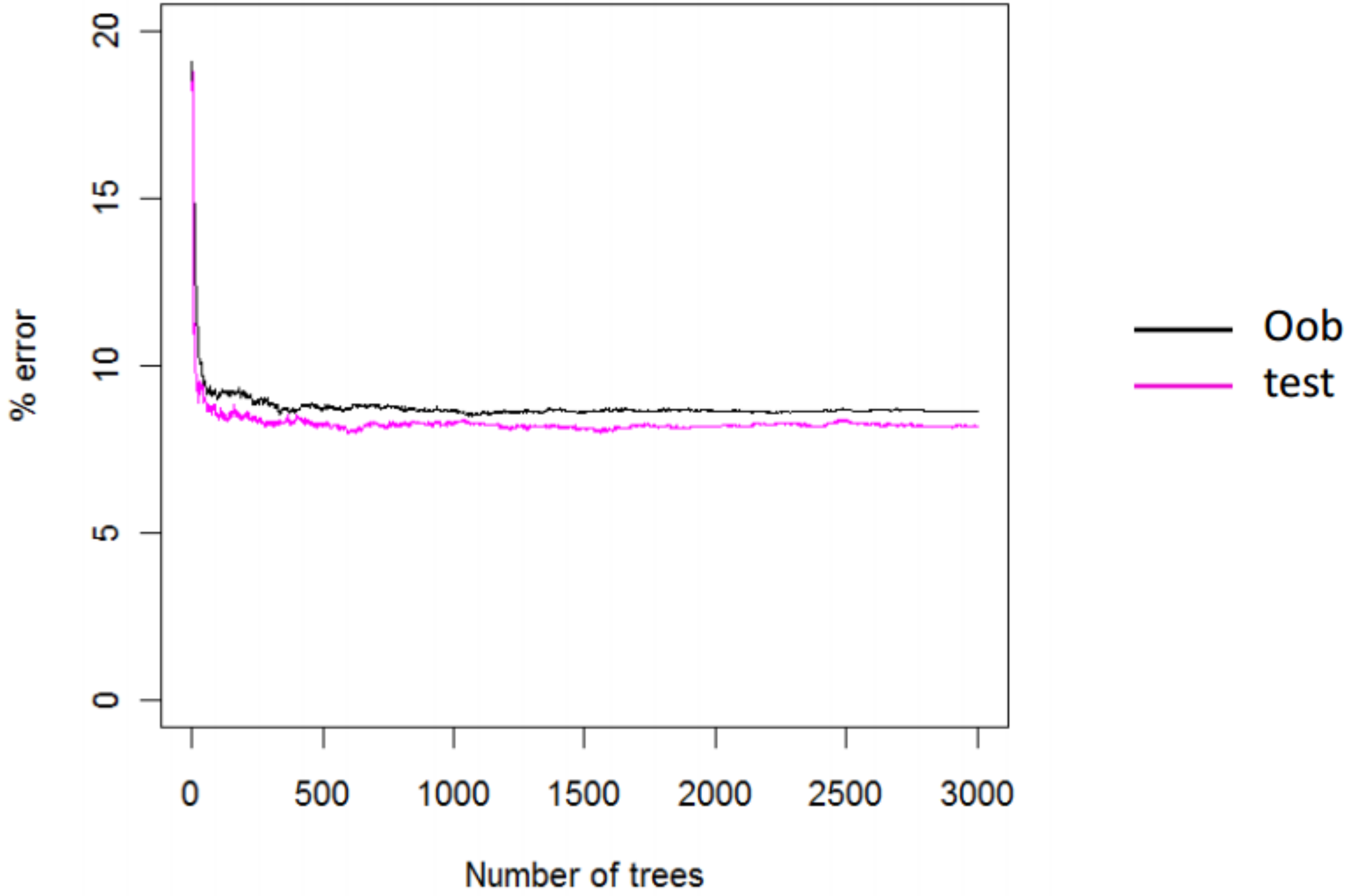
  56 say “class 2”

- The RF predictor for this case is class 1.

The RF Predictor

- The “oob” error gives an estimate of test set error (generalization error).
  - The *oob confusion matrix* is obtained from the *RF predictor*.
  - The *oob error rate* is the error rate of the *RF predictor*.
- For new cases, vote (or average) all the trees to get the *RF predictor*.

RF does not overfit as we fit more trees

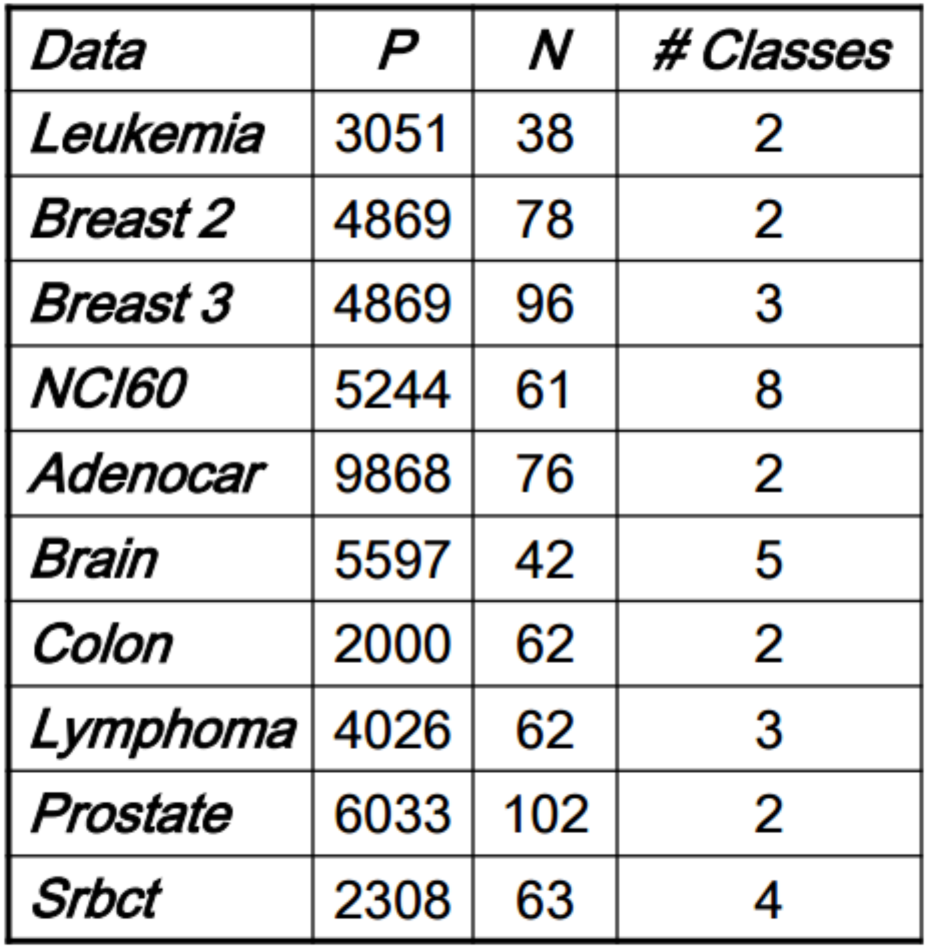


RF handles thousands of predictors

- Ramón Díaz-Uriarte, Sara Alvarez de Andrés Bioinformatics Unit, Spanish National Cancer Center March, 2005 http://ligarto.org/rdiaz
- Compared with
  - SVM, linear kernel
  - KNN
  - DLDA
  - Shrunken Centroids

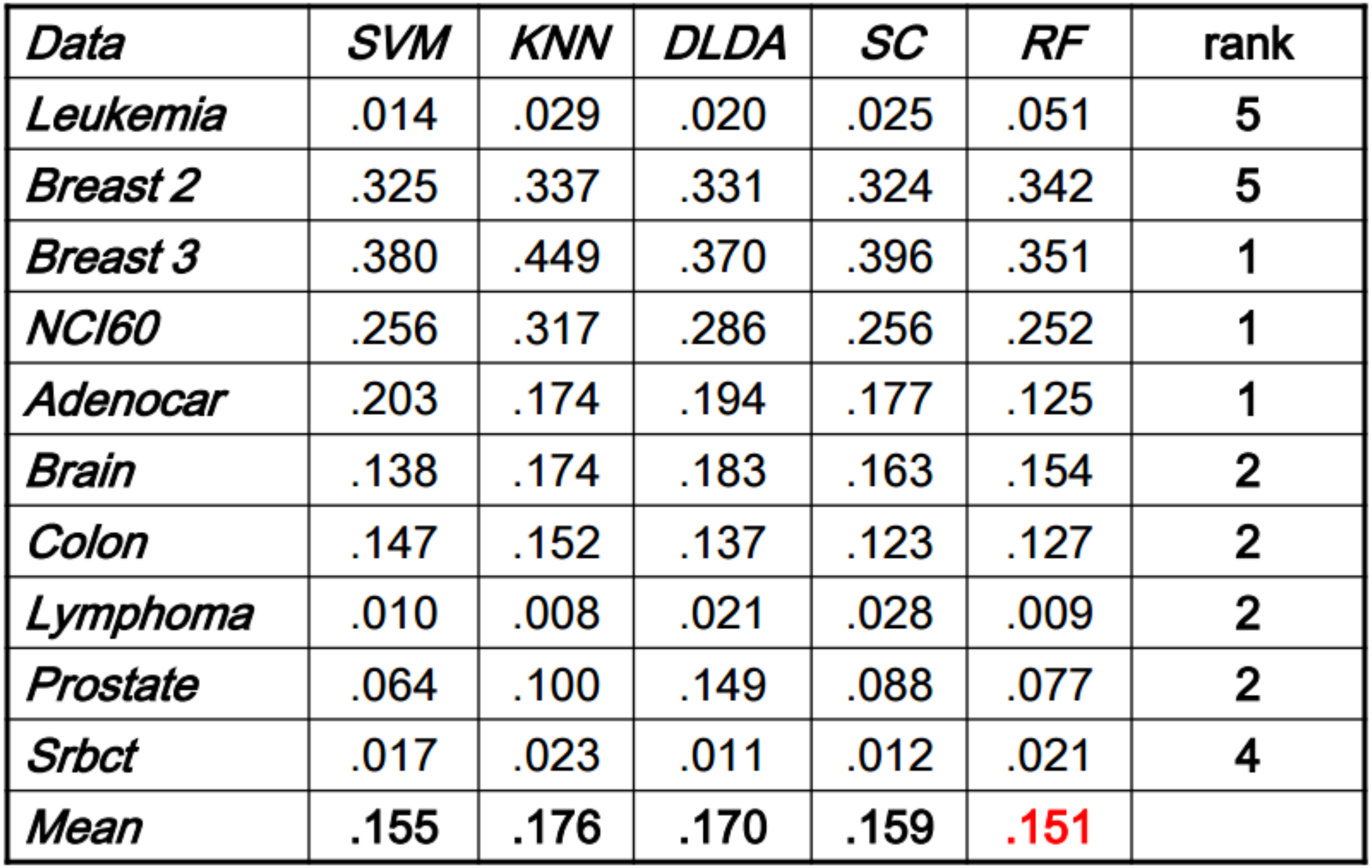
Microarray Datasets

- P: number of variables (genes), N: number of samples (people)



Data	*P*	*N*	*# Classes*
Leukemia	3051	38	2
Breast 2	4869	78	2
Breast 3	4869	96	3
NCI60	5244	61	8
Adenocar	9868	76	2
Brain	5597	42	5
Colon	2000	62	2
Lymphoma	4026	62	3
Prostate	6033	102	2
Srbct	2308	63	4

Microarray Error Rates



Data	*SVM*	*KNN*	*DLDA*	*SC*	*RF*	rank
Leukemia	.014	.029	.020	.025	.051	5
Breast 2	.325	.337	.331	.324	.342	5
Breast 3	.380	.449	.370	.396	.351	1
NCI60	.256	.317	.286	.256	.252	1
Adenocar	.203	.174	.194	.177	.125	1
Brain	.138	.174	.183	.163	.154	2
Colon	.147	.152	.137	.123	.127	2
Lymphoma	.010	.008	.021	.028	.009	2
Prostate	.064	.100	.149	.088	.077	2
Srbct	.017	.023	.011	.012	.021	4
Mean	**.155**	**.176**	**.170**	**.159**	**.151**	

Outline

- Bagging predictors
- Random forest
- **Variable importance**

Local Variable Importance

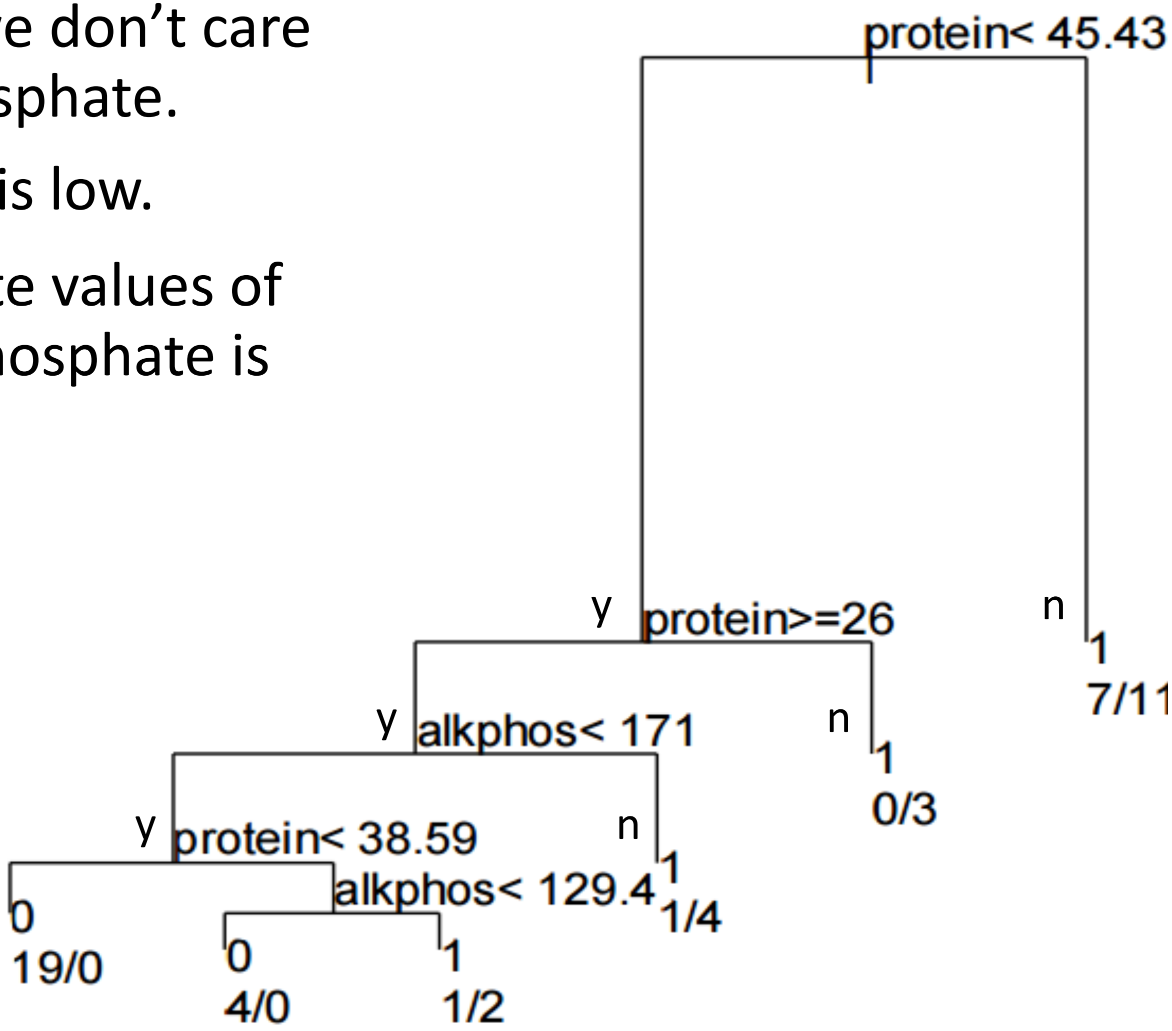
- We usually think about variable importance as an overall measure. In part, this is probably because we fit models with global structure (linear regression, logistic regression).

- In CART, however, variable importance is local.

- Different variables are important in different regions of the data.

Local Variable Importance

- If protein is high, we don’t care about alkaline phosphate.
- Similarly if protein is low.
- But for intermediate values of protein, alkaline phosphate is important.



Variable Importance Measures

- RF computes two measures of variable importance

1. Based on a **rough-and-ready measure** (i.e., impurity)

   Mean Decrease Impurity (MDI): summing total impurity reductions at all trees nodes where the variable appears

2. Based on **permutations** (using oob samples)

   Mean Decrease Accuracy (MDA): measuring accuracy reduction on oob samples when the values of the variable are randomly permuted

MDI

- Importance of variable X_i for an ensemble of m trees where $\{\phi_l\}_{l=1}^{m}$ with a tree ϕ_l is

$$Imp\left(X_j\right) = \frac{1}{m}\sum_{l=1}^{m}\sum_{t\in\phi_l} 1(V_t = X_i)[p(t)\Delta i(t)]$$

Where V_t denotes the variable used at node t, $p(t) = N_t/N$ and $\Delta i(t)$ is the impurity reduction at node t:

$$\Delta i(t) = i(t) - \frac{N_{t_L}}{N_t} i(t_L) - \frac{N_{t_r}}{N_t} i(t_R)$$

- Impurity $i(t)$ can be entropy, Gini index, variance, error

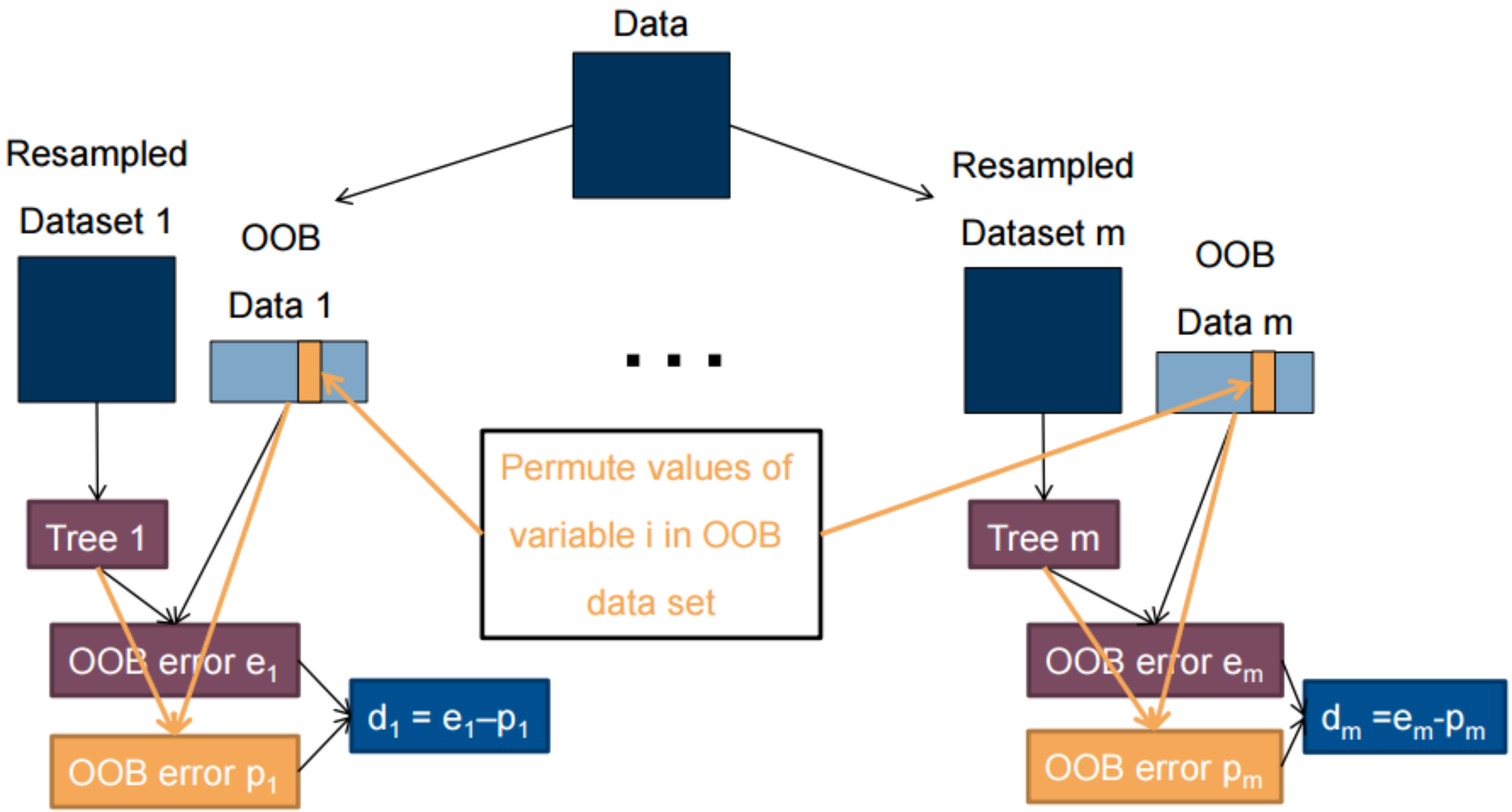
MDA

- For each tree, look at the out-of-bag (OOB) data:
  - Randomly permute the values of X_i
  - Pass these perturbed data down the tree, save the classes.
- For X_i, find

Error rate with X_i permuted − **Error rate with no permutation**

where the error rates are taken over all trees for which case is oob.

MDA



$$\bar{d}_i = \frac{1}{m}\sum_{l=1}^{m} d_{li}$$

$$s_{d_i}^2 = \frac{1}{m-1}\sum_{l=1}^{m}\left(d_{li} - \bar{d}_i\right)^2$$

➡

$$v_i = \frac{\bar{d}_i}{s_{d_i}}$$