Machine Learning

Lecture 18: Decision Tree / Bagging

Dr. Beilun Wang

Southeast University
School of Computer Science
and Engineering

Course Content Plan

- Regression (supervised) — Y is a continuous
- Classification (supervised) — Y is a discrete
- Unsupervised models — NO Y
- Learning theory — About f()
- Graphical models — About interactions among X1,... Xp
- Reinforcement Learning — Learn program to Interact with its environment

Three major sections for classification

- We can divide the large variety of classification approaches into roughly three major types
  - Discriminative
    - directly estimate a decision rule/boundary
    - ~~e.g., support vector machine,~~ decision tree, ~~logistic regression,~~ e.g. ~~neural networks (NN), deep NN~~
  - Generative:
    - build a generative statistical model
    - ~~e.g., Bayesian networks, Naïve Bayes classifier~~
  - ~~Instance based classifiers~~
    - Use observation directly (no models)
    - ~~e.g. K nearest neighbors~~

Decision Tree / Random Forest

Task

↓

Representation Score

↓

Function

↓

Search/Optimization

↓

Models,
Parameters

Classification

↓

Partition feature space
into set of rectangles,
local smoothness

↓

Greedy to find partitions

↓

Split with Purity measure /
e.g. IG / cross-entropy / Gini /

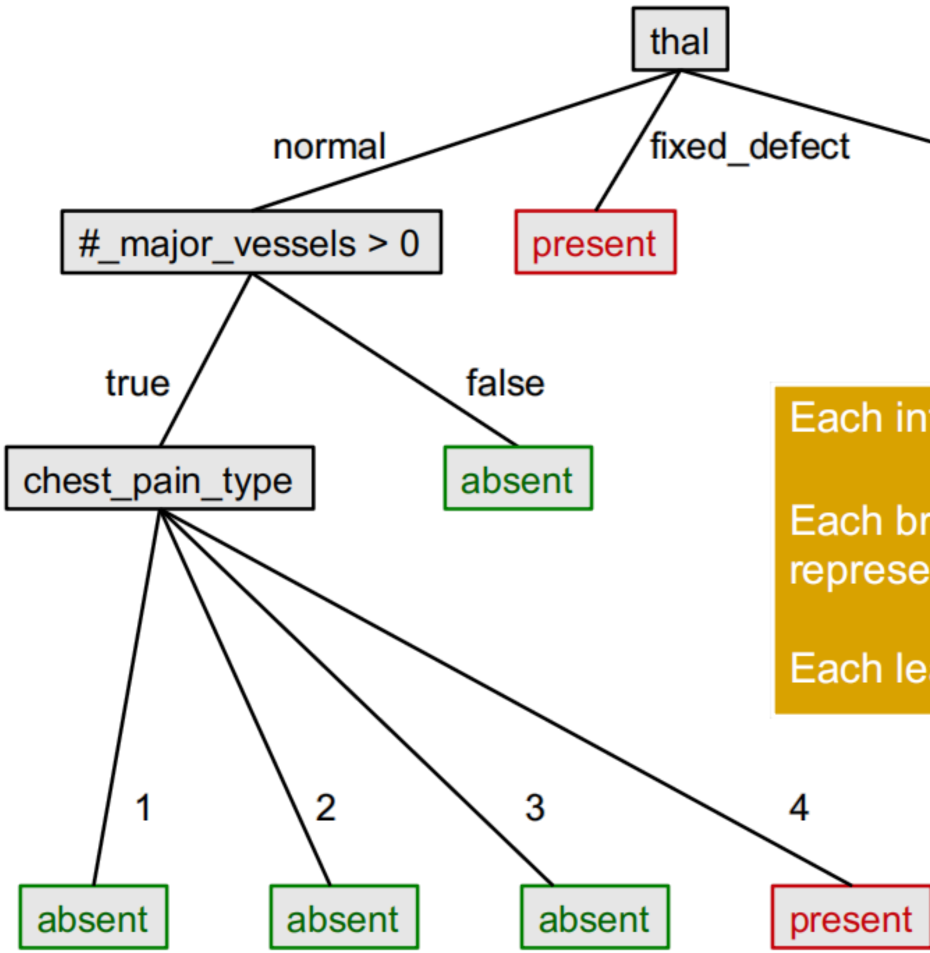
↓

Tree Model (s), i.e.
space partition

Today

- Decision Tree (DT):
    - Tree representation
- Brief information theory
- Learning decision trees
- Bagging
- Random forests: Ensemble of DT
- More about ensemble

A decision tree to predict heart disease

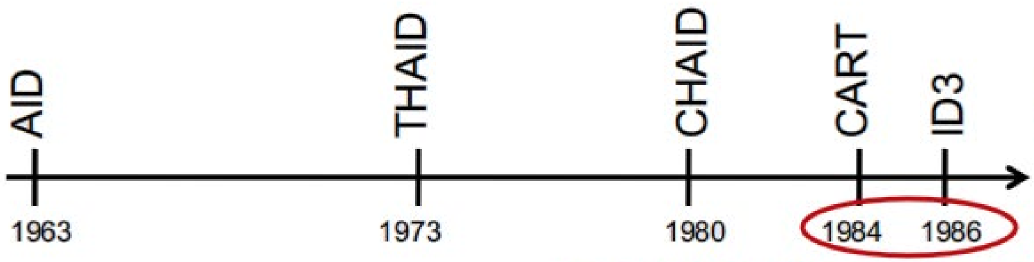


Each internal node tests one feature x_i

Each branch from an internal node represents one outcome of the test

Each leaf predicts y or $P(y \mid \boldsymbol{x})$

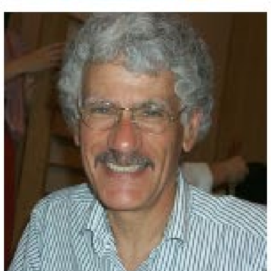
History of decision tree learning

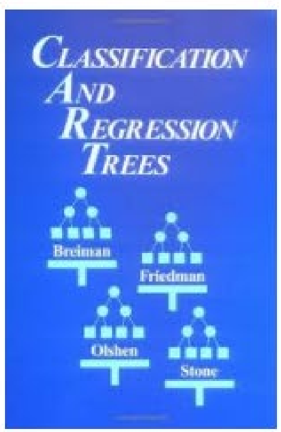


many DT variants have been developed since CART and ID3

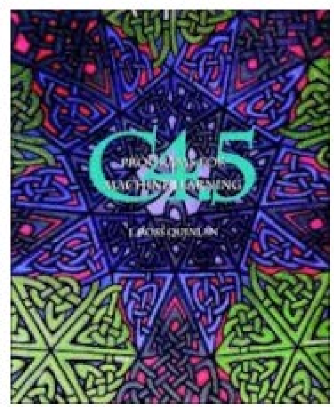
dates of seminal publications: work on these 2 was contemporaneous

CART developed by Leo Breiman, Jerome Friedman, Charles Olshen, R.A. Stone





ID3, C4.5, C5.0 developed by Ross Quinlan



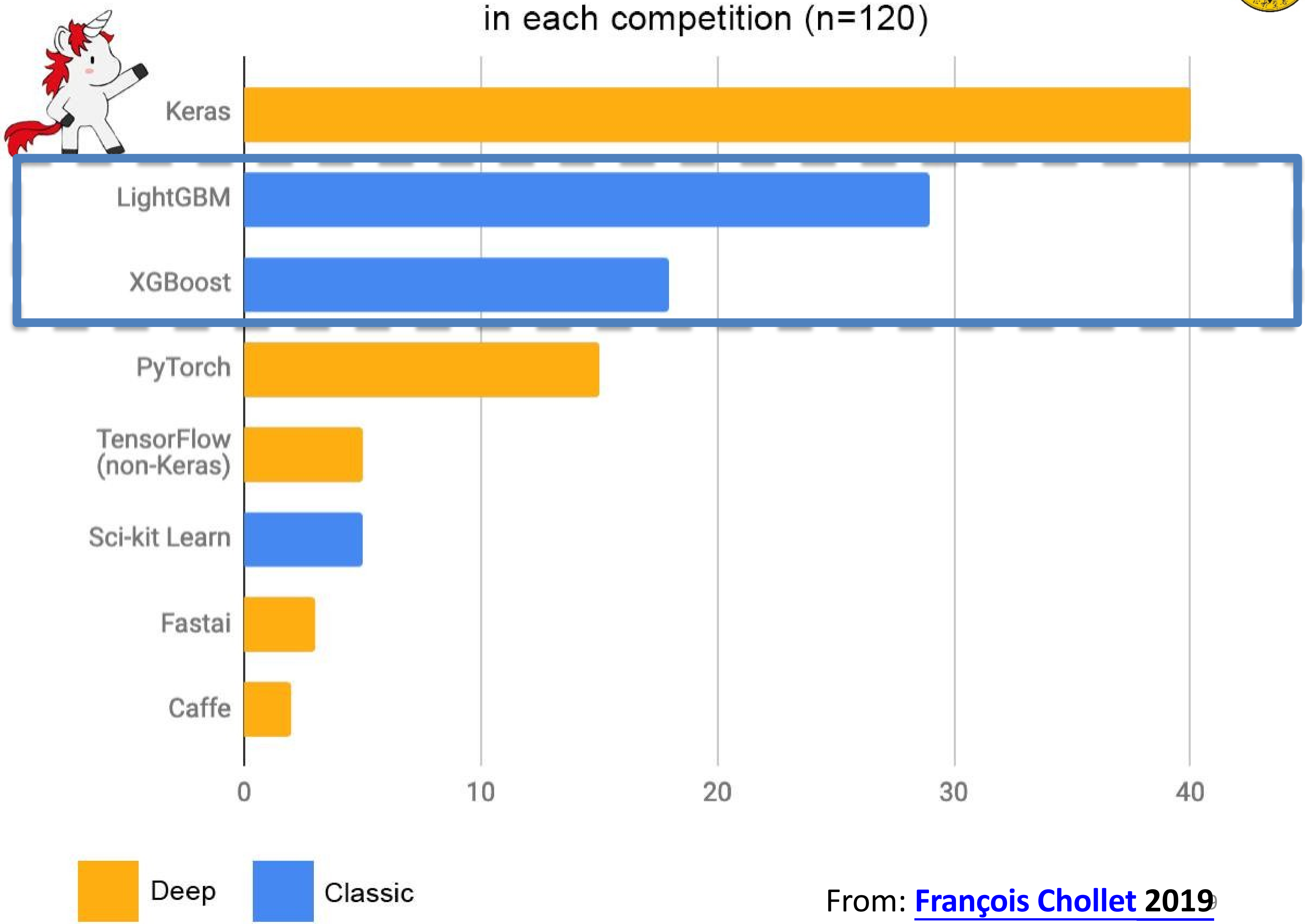
A study comparing Classifiers
→ 11 binary classification problems / 8 metrics

Top 8 Models

Table 2. Normalized scores for each learning algorithm by metric (average over eleven problems)

	CAL	ACC	FSC	LFT	ROC	APR	BEP	RMS	MXE	MEAN	OPT-SEL
BST-DT	PLT	.843*	.779	**.939**	**.963**	**.938**	.929*	**.880**	**.896**	**.896**	**.917**
RF	PLT	.872*	.805	.934*	.957	.931	**.930**	.851	.858	.892	.898
BAG-DT	–	.846	.781	.938*	.962*	.937*	.918	.845	.872	.887*	.899
BST-DT	ISO	.826*	.860*	.929*	.952	.921	.925*	.854	.815	.885	.917*
RF	–	**.872**	.790	.934*	.957	.931	**.930**	.829	.830	.884	.890
BAG-DT	PLT	.841	.774	.938*	.962*	.937*	.918	.836	.852	.882	.895
RF	ISO	.861*	**.861**	.923	.946	.910	.925	.836	.776	.880	.895
BAG-DT	ISO	.826	.843*	.933*	.954	.921	.915	.832	.791	.877	.894
SVM	PLT	.824	.760	.895	.938	.898	.913	.831	.836	.862	.880
ANN	–	.803	.762	.910	.936	.892	.899	.811	.821	.854	.885
SVM	ISO	.813	.836*	.892	.925	.882	.911	.814	.744	.852	.882
ANN	PLT	.815	.748	.910	.936	.892	.899	.783	.785	.846	.875
ANN	ISO	.803	.836	.908	.924	.876	.891	.777	.718	.842	.884
BST-DT	–	.834*	.816	**.939**	**.963**	**.938**	.929*	.598	.605	.828	.851
KNN	PLT	.757	.707	.889	.918	.872	.872	.742	.764	.815	.837
KNN	–	.756	.728	.889	.918	.872	.872	.729	.718	.810	.830
KNN	ISO	.755	.758	.882	.907	.854	.869	.738	.706	.809	.844
BST-STMP	PLT	.724	.651	.876	.908	.853	.845	.716	.754	.791	.808
SVM	–	.817	.804	.895	.938	.899	.913	.514	.467	.781	.810
BST-STMP	ISO	.709	.744	.873	.899	.835	.840	.695	.646	.780	.810
BST-STMP	–	.741	.684	.876	.908	.853	.845	.394	.382	.710	.726
DT	ISO	.648	.654	.818	.838	.756	.778	.590	.589	.709	.774

Primary ML software tool used by **top-5 teams** on Kaggle in each competition (n=120)



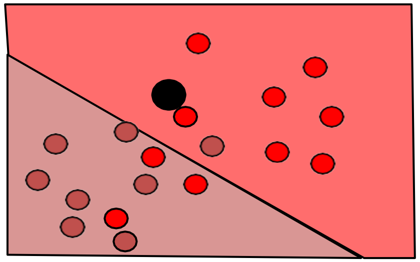
From: [François Chollet](#) **2019**

Readability Hierarchy

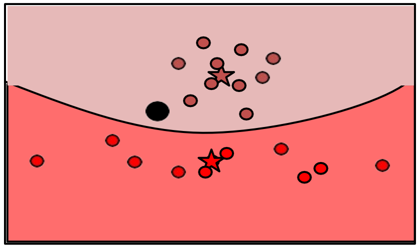
Readable

Decision Trees: Classifies based on a series of one-variable decisions.

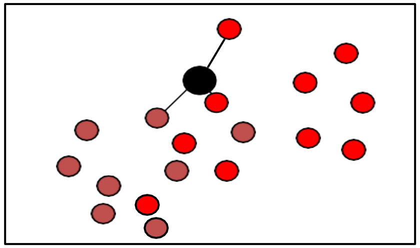
Linear Classifier: Weight vector w tells us how important each variable is for classification and in which direction it points.



Quadratic Classifier: Linear weights work as in linear classifier, with additional information coming from all products of variables.



k Nearest Neighbors: Classifies using the complete training set, no information about the nature of the class difference

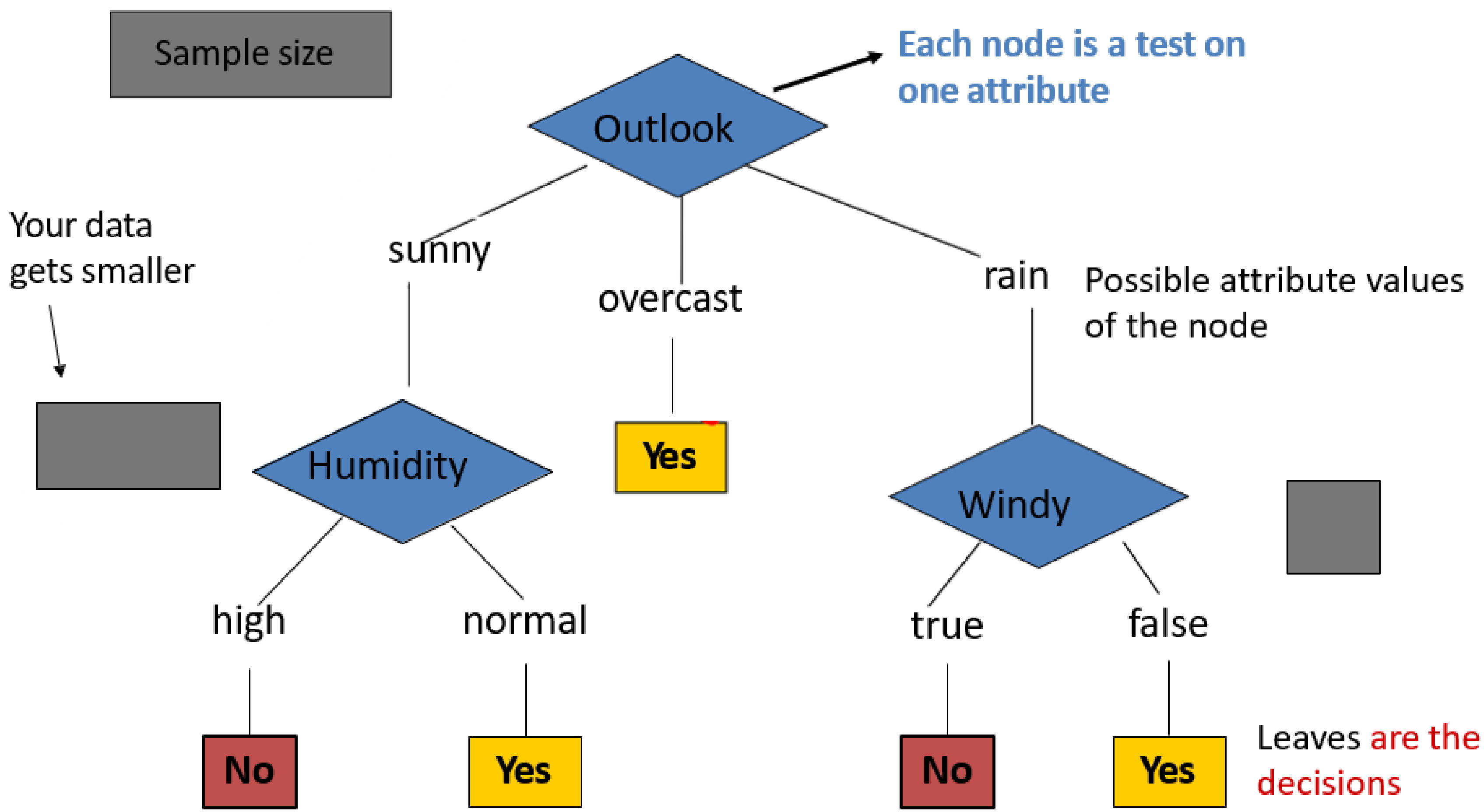


Example: Play Tennis

PlayTennis: training examples

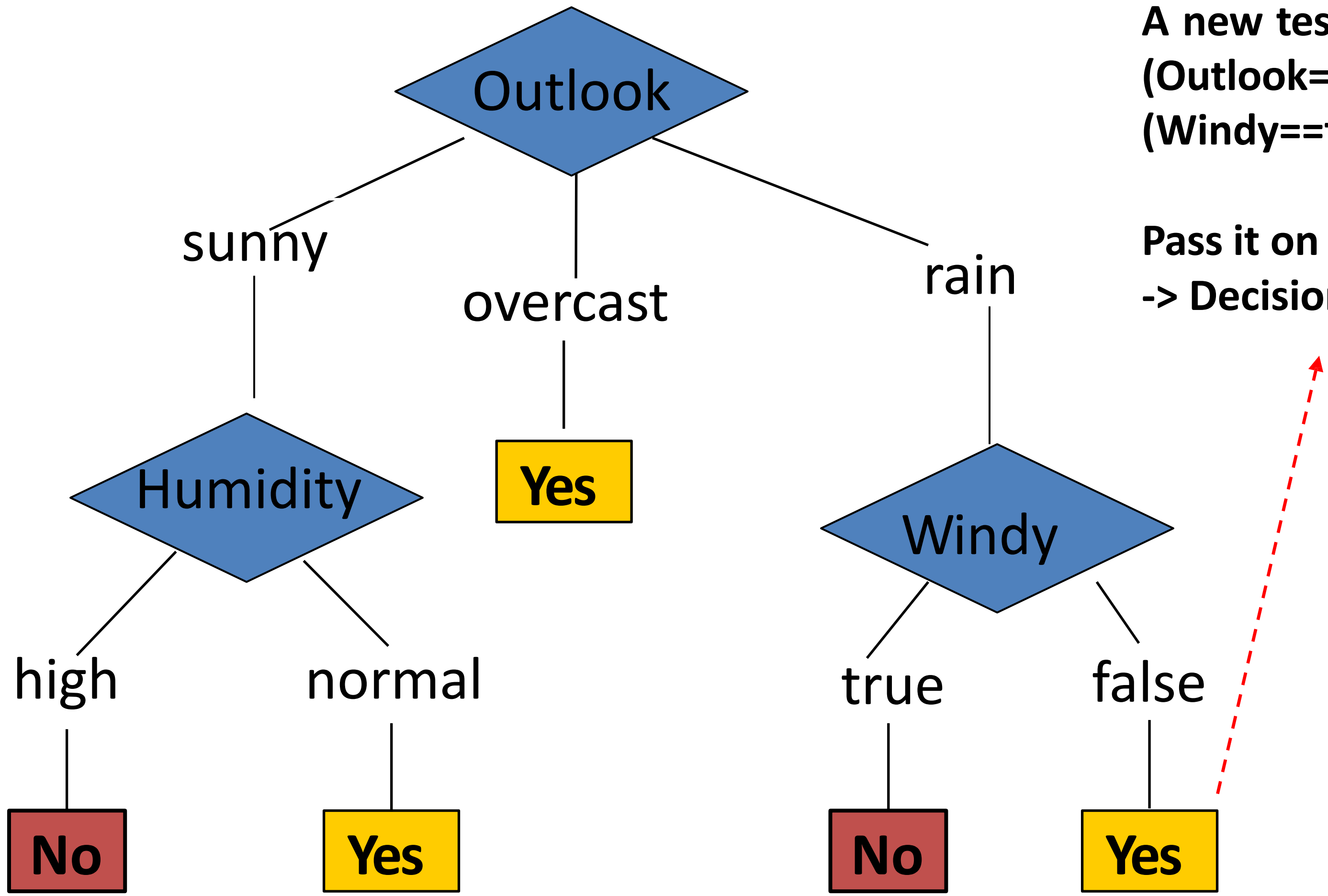
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Anatomy of a decision tree



Apply Model to Test Data:

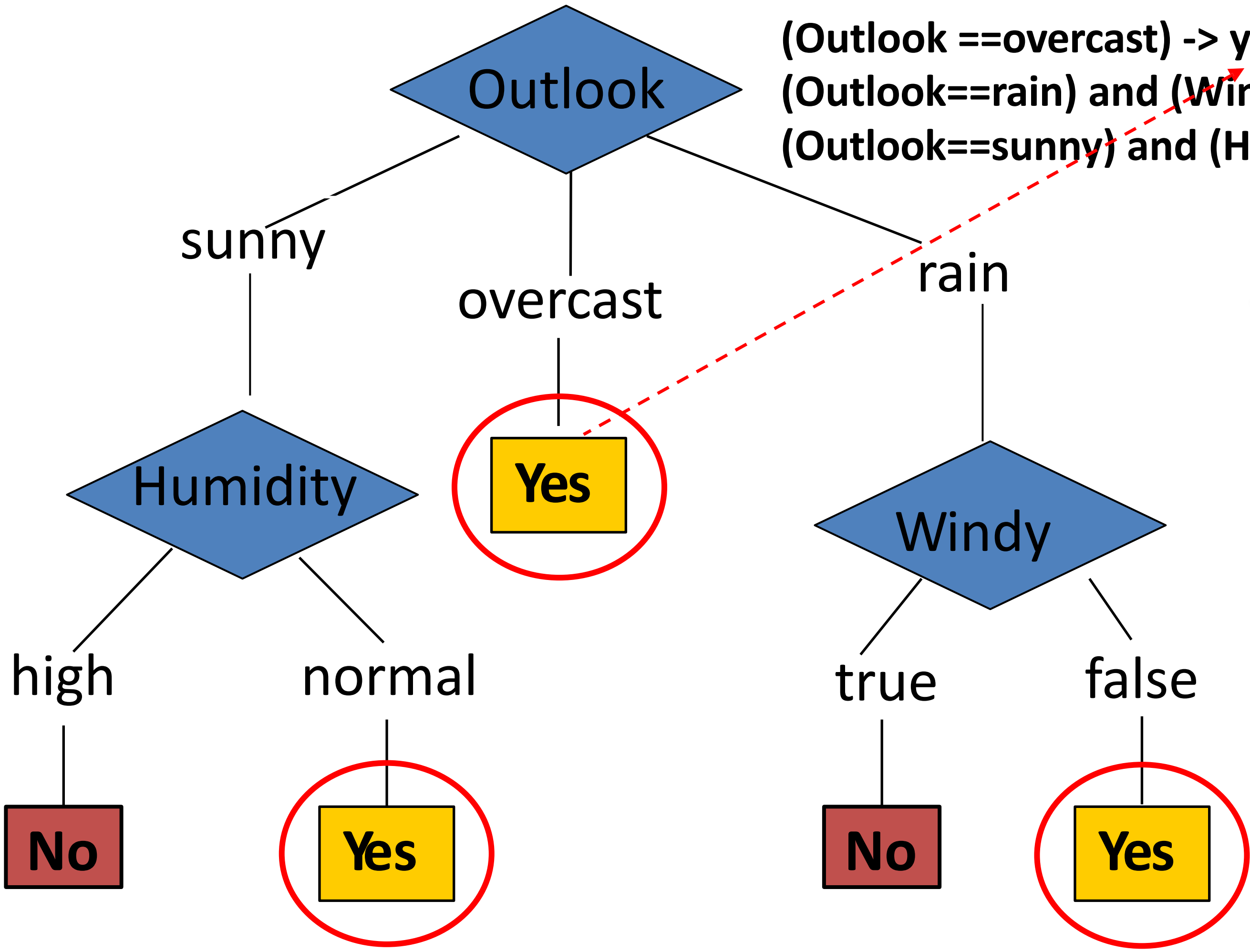
To ‘play tennis’ or not.



A new test example:
(Outlook==rain) and
(Windy==false)

Pass it on the tree
-> Decision is yes.

Apply Model to Test Data:
To ‘play tennis’ or not.



(Outlook ==overcast) -> yes
(Outlook==rain) and (Windy==false) ->yes
(Outlook==sunny) and (Humidity=normal) ->yes

Three cases of “YES”

Decision trees (on Discrete)

- Decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances.

(Outlook ==overcast)

OR

((Outlook==rain) and (Windy==false))

OR

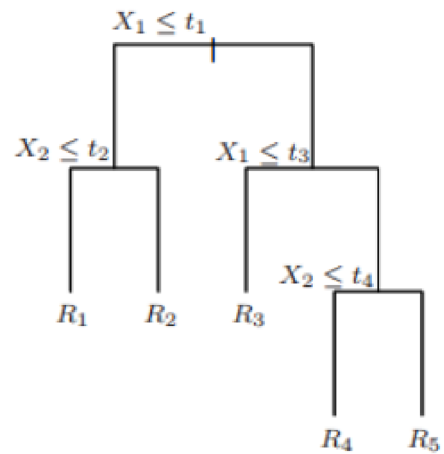
((Outlook==sunny) and (Humidity=normal))

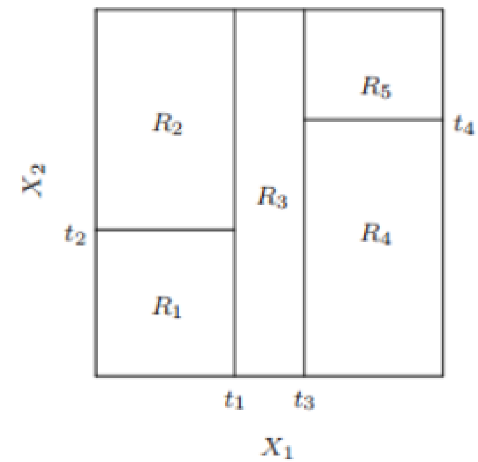
=> yes play tennis

Decision trees (on Continuous)

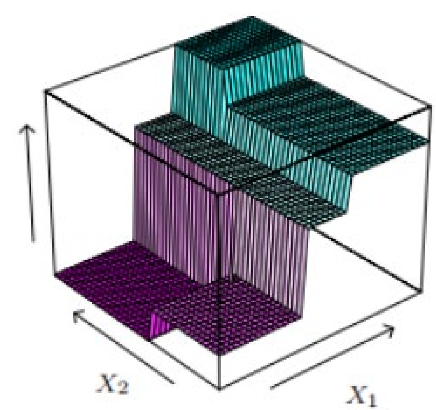
From ESL book Ch9 :

Classification **a**nd **R**egression **T**rees (CART)





- Partition feature space into set of rectangles
- Fit simple model in each partition



Decision Tree / Random Forest

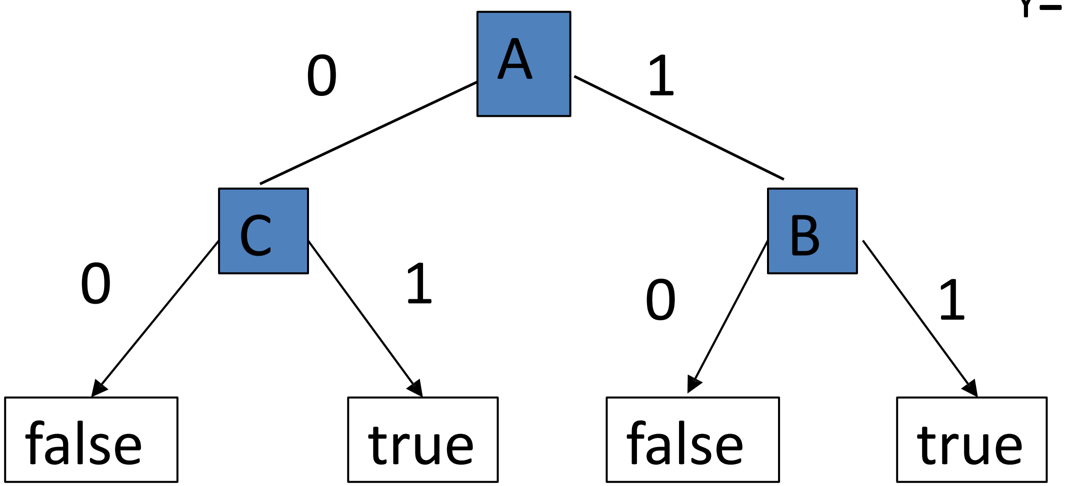
Task
↓
Representation
↓
Score Function
↓
Search/Optimization
↓
Models, Parameters

Classification
↓
Partition feature space into set of rectangles, local smoothness
↓
Greedy to find partitions
↓
Split with Purity measure / e.g. IG / cross-entropy / Gini /
↓
Tree Model (s), i.e. space partition

Today

- Decision Tree (DT):
  - Tree representation
- Brief information theory
- Learning decision trees
- Bagging
- Random forests: Ensemble of DT
- More about ensemble

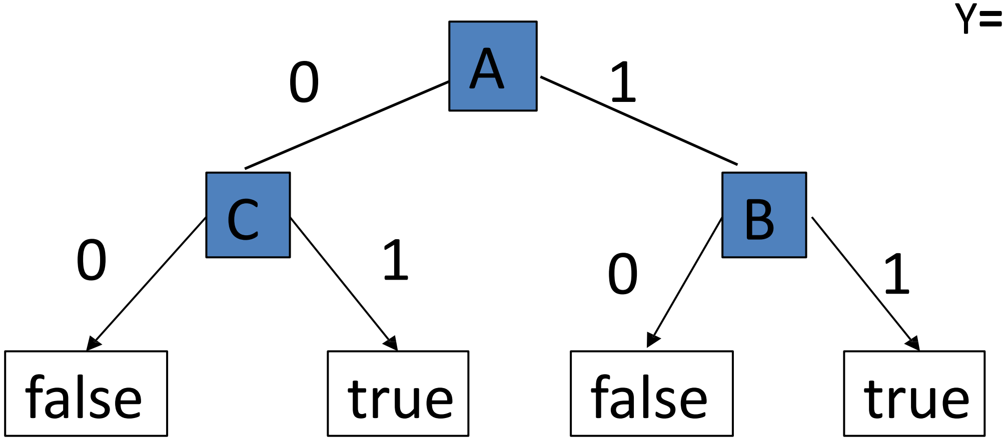
Challenge in Tree Representation



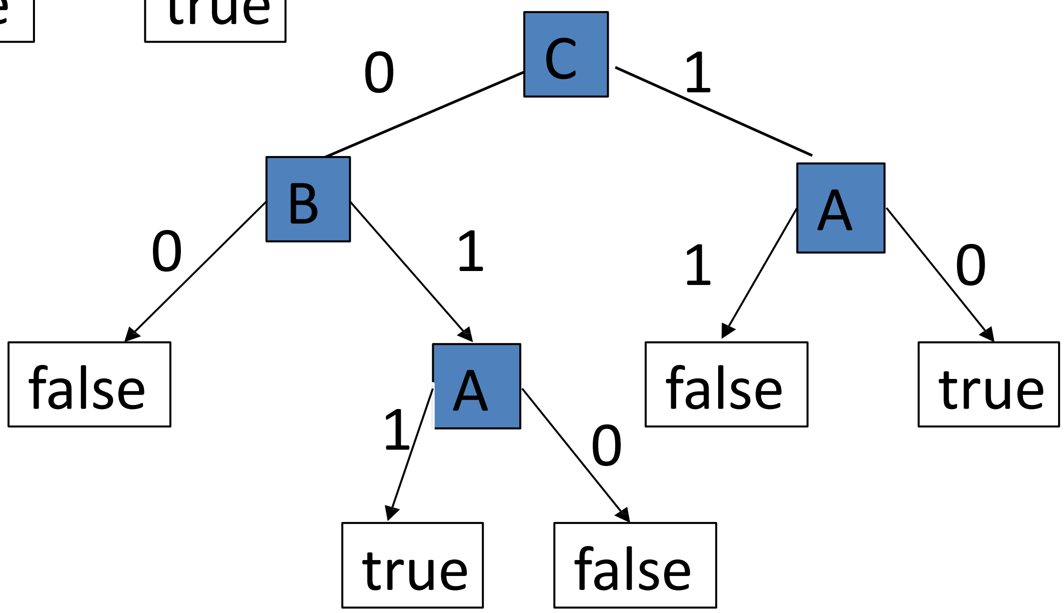
Y=((A and B) or ((not A) and C))

Same concept / different representation

$Y=((A \text{ and } B) \text{ or } ((\text{not } A) \text{ and } C))$



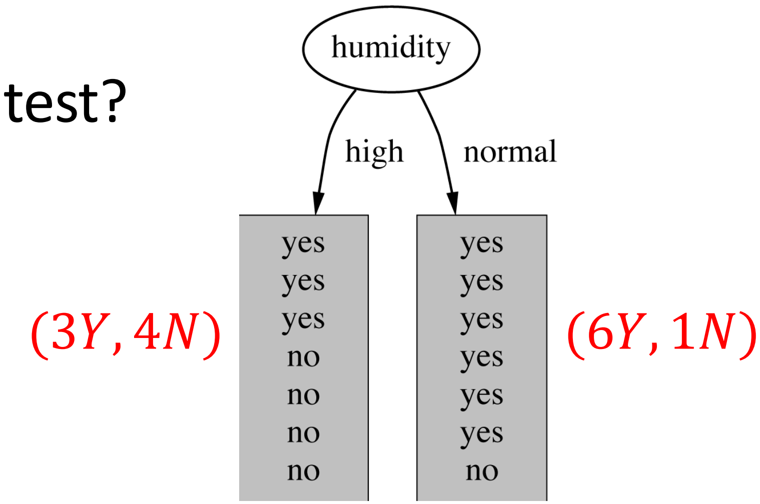
Not unique

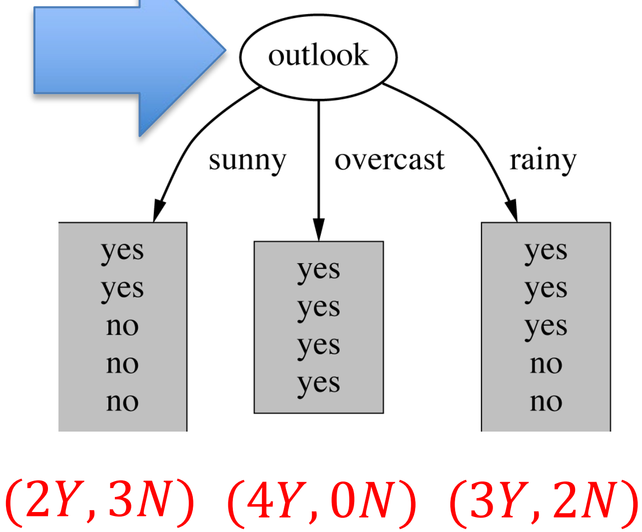


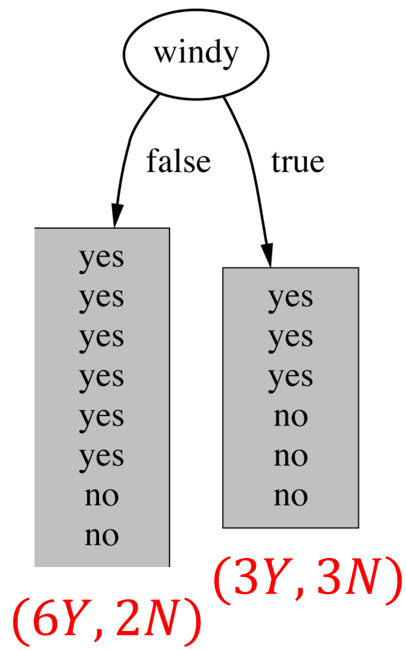
How do we choose which attribute to split ?

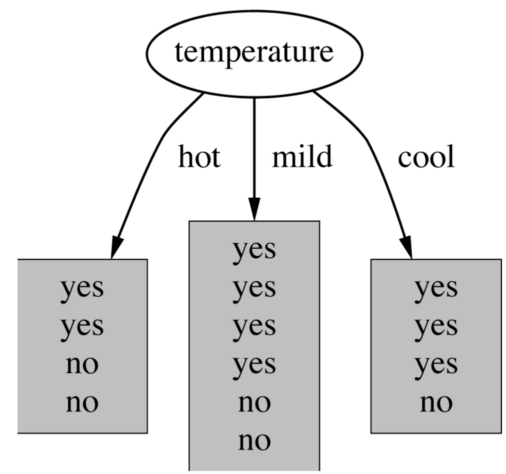
Which attribute should be used first to test?

Intuitively, you would prefer the one that *separates* the training examples as much as possible.









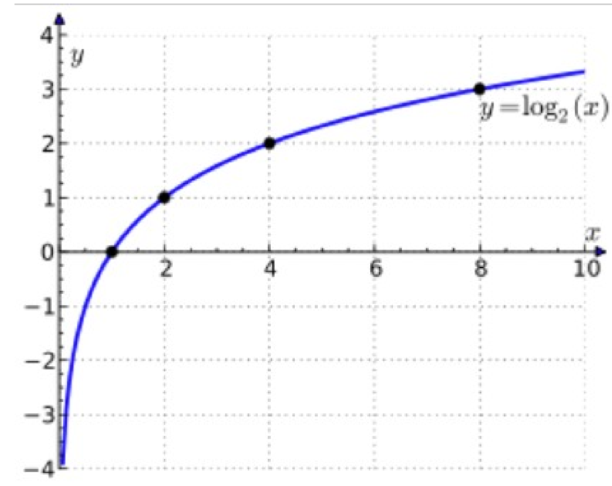
one criteria: Information gain

- Imagine:
  - Someone is about to tell you your own name
  - You are about to observe the outcome of a dice roll
  - You are about to observe the outcome of a coin flip
  - You are about to observe the outcome of a biased coin flip

- Each situation has a different amount of uncertainty as to what outcome you will observe.

Information

- Information: Reduction in uncertainty (amount of surprise in the outcome)

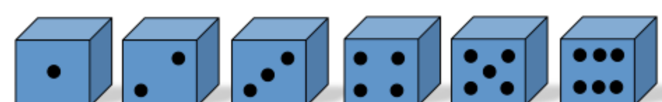


$$I(X) = \log_2 \frac{1}{p(x)} = -\log_2 p(x)$$

If the probability of this event happening is small and it happens, the information is large.

- ➢ Observing the outcome of a coin flip is head $\longrightarrow$ $I = -\log_2 \frac{1}{2} = 1$

- ➢ Observe the outcome of a dice is 6 $\longrightarrow$ $I = -\log_2 \frac{1}{6} = 2.58$



Entropy

- The expected amount of information when observing the output of a random variable X

$$H(X) = E\big(I(X)\big) = \sum_i p(x_i) I(x_i) = \sum_i p(x_i) \log_2 p(x_i)$$

- If the X can have 8 outcomes and all are equally likely

$$H(X) = -\sum_i \frac{1}{8} \log_2 \frac{1}{8} = 3$$

Entropy

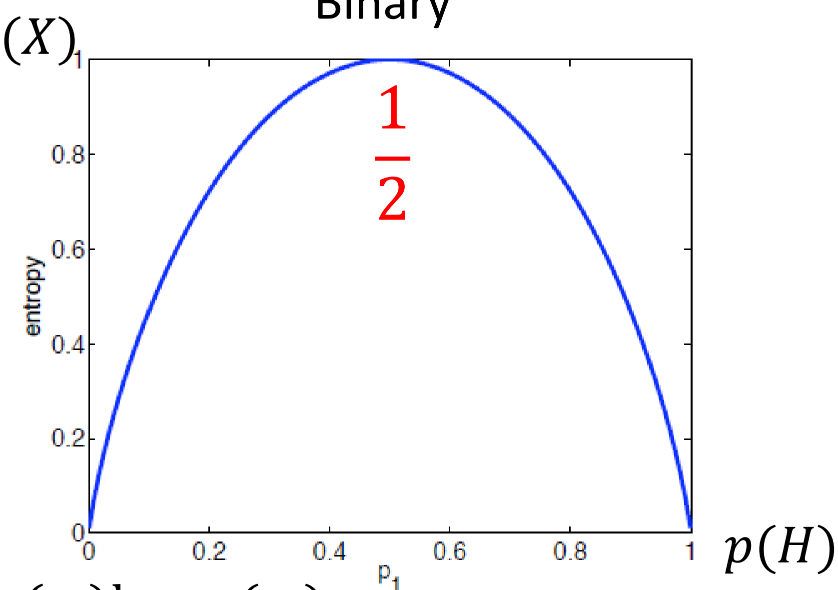
- If there are *k* possible outcomes

$$H(X) \leq \log_2 k$$

- Equality holds when all outcomes are equally likely

- The more the probability distribution that deviates from uniformity, the lower the entropy

Binary



$$H(X) = E(I(X)) = \sum_i p(x_i) I(x_i) = \sum_i p(x_i) \log_2 p(x_i)$$

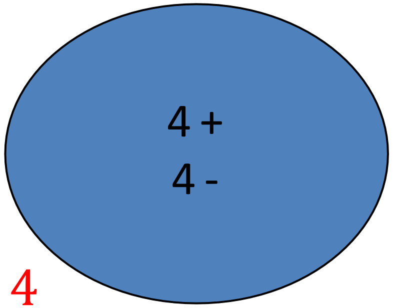
e.g. for a random binary variable

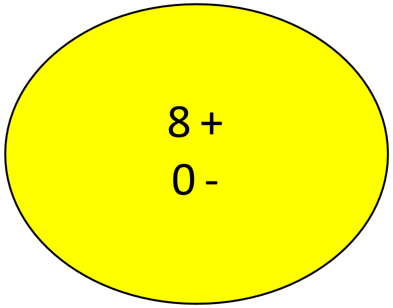
Entropy Lower → better purity

- Entropy measures the purity

$$P_{Yes} = P = \frac{4}{8}$$

$$P_{No} = 1 - P = \frac{4}{8}$$





$$P = \frac{8}{8} = 1 = P_{Yes}$$

$$1 - P = 0 = P_{No}$$

The distribution is less uniform

Entropy is lower

The node is purer

Information gain

- $IG(X,Y) = H(Y) - H(Y|X)$
- Reduction in uncertainty of Y by knowing a feature variable X

- Information gain:

  = (information before split) – (information after split)

  = entropy(parent) – [average entropy(children)]

Fixed – the lower, the better (children nodes are purer) = For IG, the higher, the better

Conditional entropy

$$H(Y) = -\sum_i p(y_i)\log_2 p(y_i)$$

$$H(Y \mid X = x_j) = -\sum_i p(y_i \mid x_j)\log_2 p(y_i \mid x_j)$$

$$H(Y \mid X) = \sum_j p(x_j) H(Y \mid X = x_j)$$

$$= -\sum_j p(x_j) \sum_i p(y_i \mid x_j)\log_2 p(y_i \mid x_j)$$

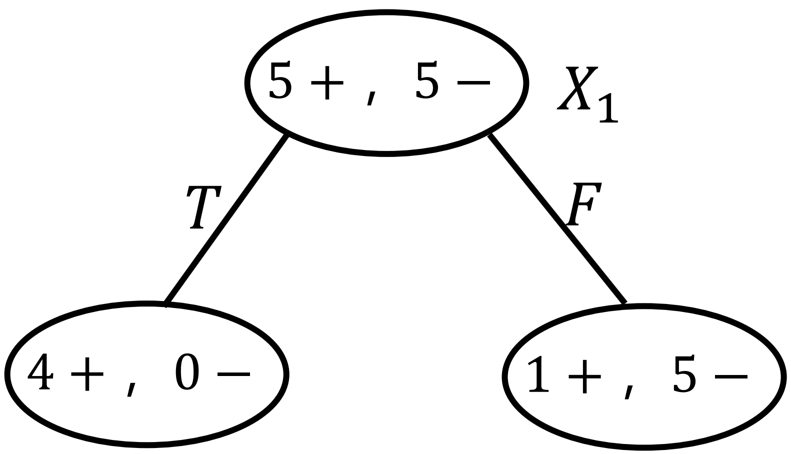
Example

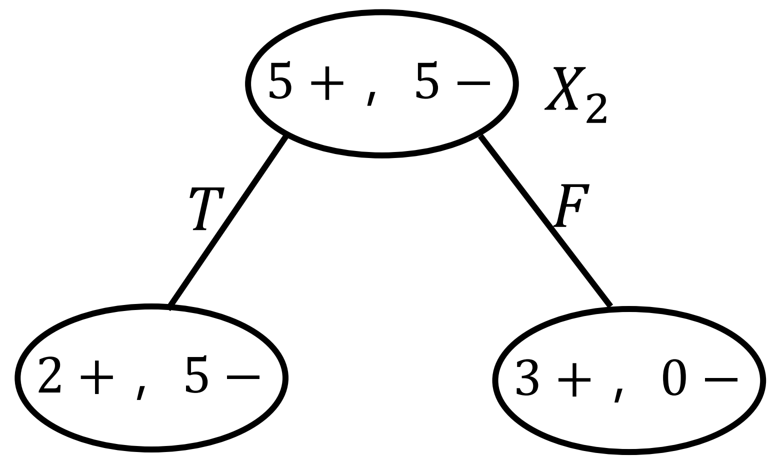
Attributes Labels

X1	X2	Y	Count
T	T	+	2
T	F	+	2
F	T	-	5
F	F	+	1

Which one do we choose?

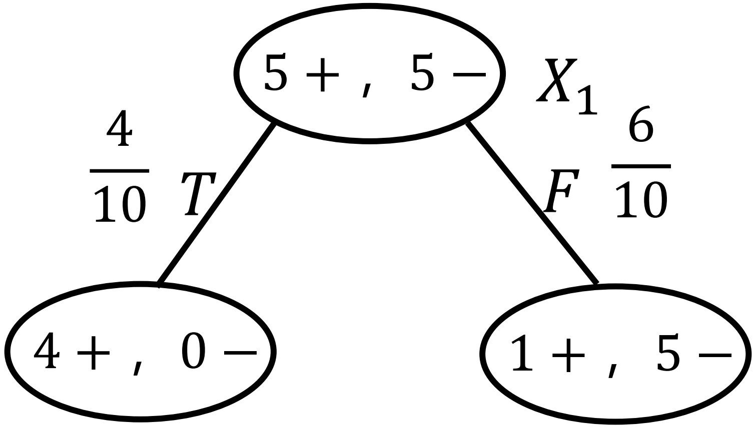
X_1 or X_2?





Example

X1	X2	Y	Count
T	T	+	2
T	F	+	2
F	T	-	5
F	F	+	1



$$H(Y|X_1 = T) = -\{P(Y = +|X_1 = T)\log P(Y = +|X_1 = T)$$
$$+P(Y = -|X_1 = T)\log P(Y = -|X_1 = T)\}$$

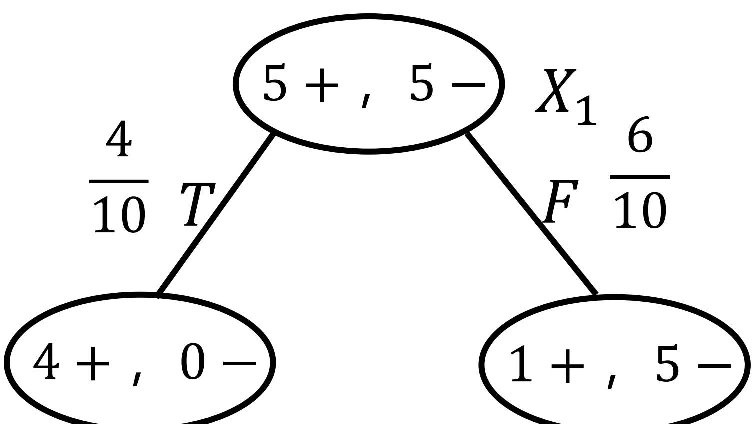
$4+,\ 0-$ $\Rightarrow$ $= 0$

$$H(Y|X_1 = T) = \begin{matrix}4+\\0-\end{matrix} \Rightarrow -\big(P(+)\log\big(P(+)\big) + P(-)\log\big(P(-)\big)\big)$$

$$= -(1\log 1 + 0\log 0) = 0$$

$$H(Y|X_1 = F) = \begin{matrix}1+\\5-\end{matrix} \Rightarrow -\big(P(+)\log\big(P(+)\big) + P(-)\log\big(P(-)\big)\big)$$

$$= -\left(\frac{1}{6}\log\frac{1}{6} + \frac{5}{6}\log\frac{5}{6}\right)$$



$$H(Y|X_1) = \frac{4}{10}H(Y|X_1 = T)$$

$$+\frac{6}{10}H(Y|X_1 = F)$$

Example

Attributes Labels

X1	X2	Y	Count
T	T	+	2
T	F	+	2
F	T	-	5
F	F	+	1

Which one do we choose?

X_1 or X_2?

IG(X1,Y) = H(Y) – H(Y|X1)

H(Y) = - (5/10) log(5/10) - 5/10log(5/10) = 1

H(Y|X1) = P(X1=T)H(Y|X1=T) + P(X1=F) H(Y|X1=F)

= 4/10 (1log 1 + 0 log 0) +6/10 (5/6log 5/6 +1/6log1/6)

= 0.39

Information gain (X1,Y) = 1 - 0.39 = 0.61

Which one do we choose?

Attributes Labels

X1	X2	Y	Count
T	T	+	2
T	F	+	2
F	T	-	5
F	F	+	1

Which one do we choose?

X_1 or X_2?

Information gain (X1,Y) = 0.61

Information gain (X2,Y) = 0.12

Pick the variable which provides the most information gain about Y

Pick X_1

Which one do we choose?

X1	X2	Y	Count
T	T	+	2
T	F	+	2
F	T	-	5
F	F	+	1

X1	X2	Y	Count
T	T	+	2
T	F	+	2
F	T	-	5
F	F	+	1

One branch

The other branch

Information gain (X1,Y)= 0.61

Information gain (X2,Y)= 0.12

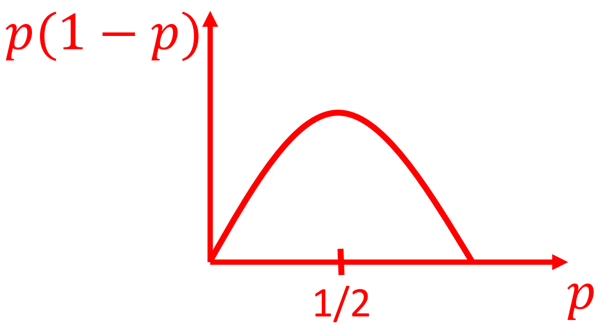
Pick the variable which provides the most information gain about Y

Pick X_1

Then recursively choose next Xi on branches

Decision Trees

- Caveats: The number of possible values influences the information gain.
  - The more possible values, the higher the gain (the more likely it is to form small, but pure partitions)



- Other Purity (diversity) measures
  - Information Gain
  - Gini (population impurity)
    - where is p_{mk} proportion of class k at node m

$$\sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$$

  - Chi-square Test

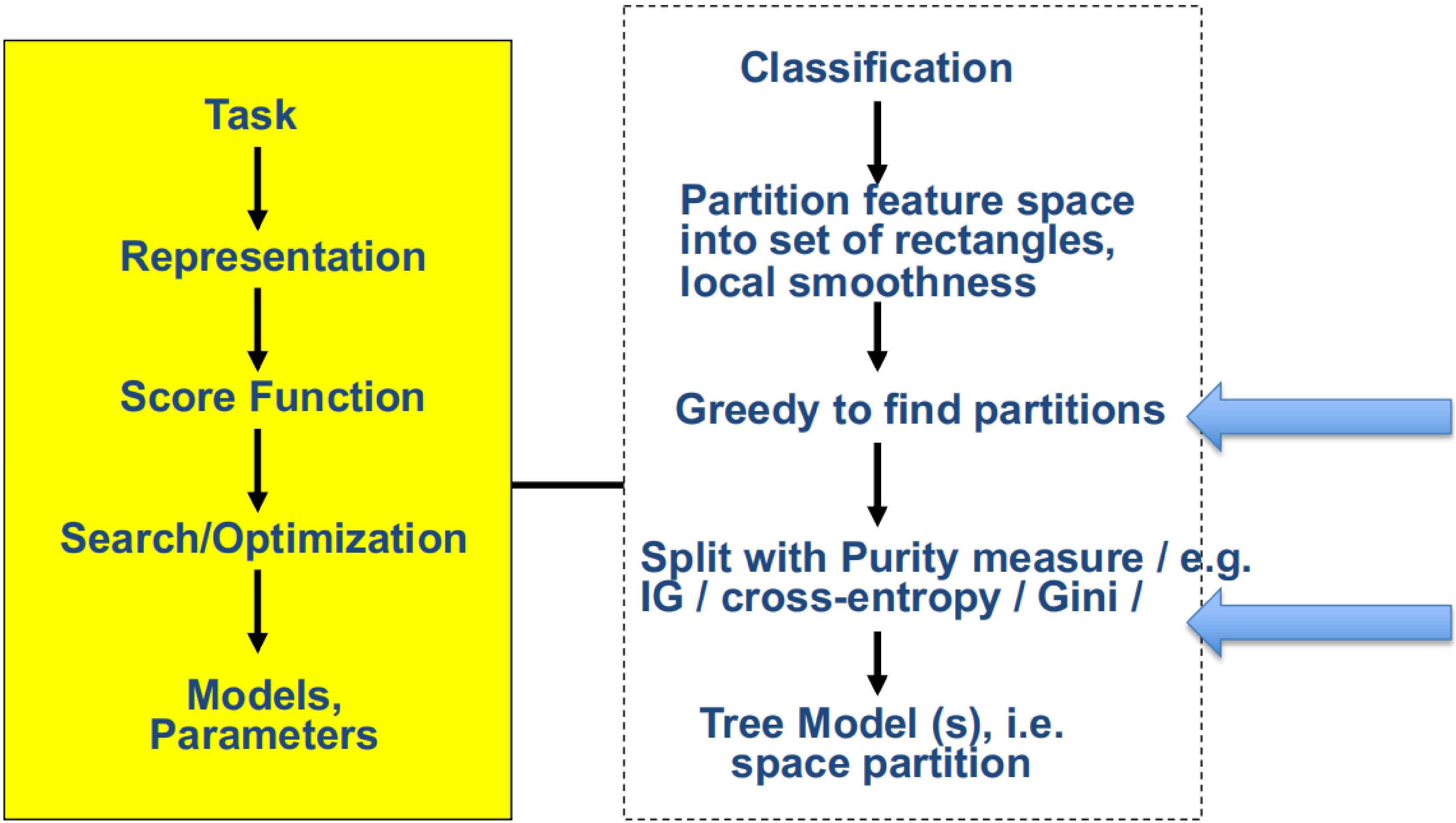
Overfitting

- You can perfectly fit DT to any training data
- Instability of Trees
  - High variance (small changes in training set will result in changes of tree model)
  - Hierarchical structure ➡ Error in top split propagates down
- Two approaches:
  - Stop growing the tree when further splitting the data does not yield an improvement
  - Grow a full tree, then prune the tree, by eliminating nodes.

Summary: Decision trees

- Non-linear classifier / regression
- Easy to use
- Easy to interpret
- Susceptible to overfitting but can be avoided.

Decision Tree / Random Forest



Today

- Decision Tree (DT):
  - Tree representation
- Brief information theory
- Learning decision trees
- Bagging
- Random forests: Ensemble of DT
- More about ensemble

Bagging

- Bagging or bootstrap aggregation
  - a technique for reducing the variance of an estimated prediction function.

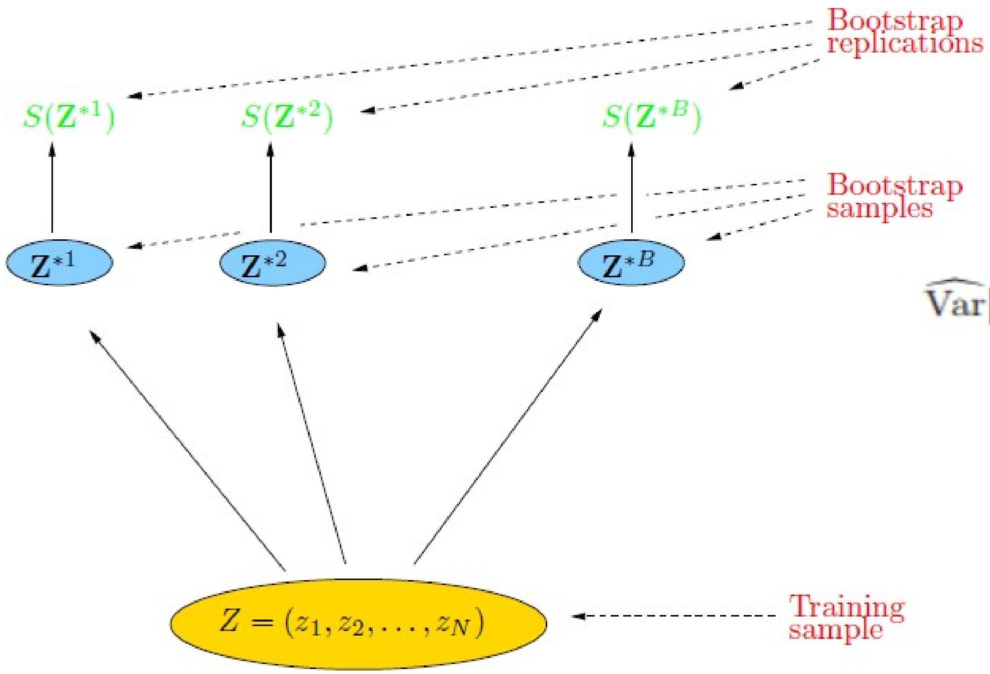
- For instance, for classification, a committee of trees
  - Each tree casts a vote for the predicted class.

Bootstrap

- The basic idea:
  - randomly draw datasets with replacement (i.e. allows duplicates) from the training data, each samples the same size as the original training set

Bootstrap

- The basic idea:
  - randomly draw datasets with replacement (i.e. allows duplicates) from the training data, each samples the same size as the original training set



$$\widehat{\mathrm{Var}}[S(\mathbf{Z})] = \frac{1}{B-1}\sum_{b=1}^{B}\left(S(\mathbf{Z}^{*b}) - \bar{S}^{*}\right)^2,$$

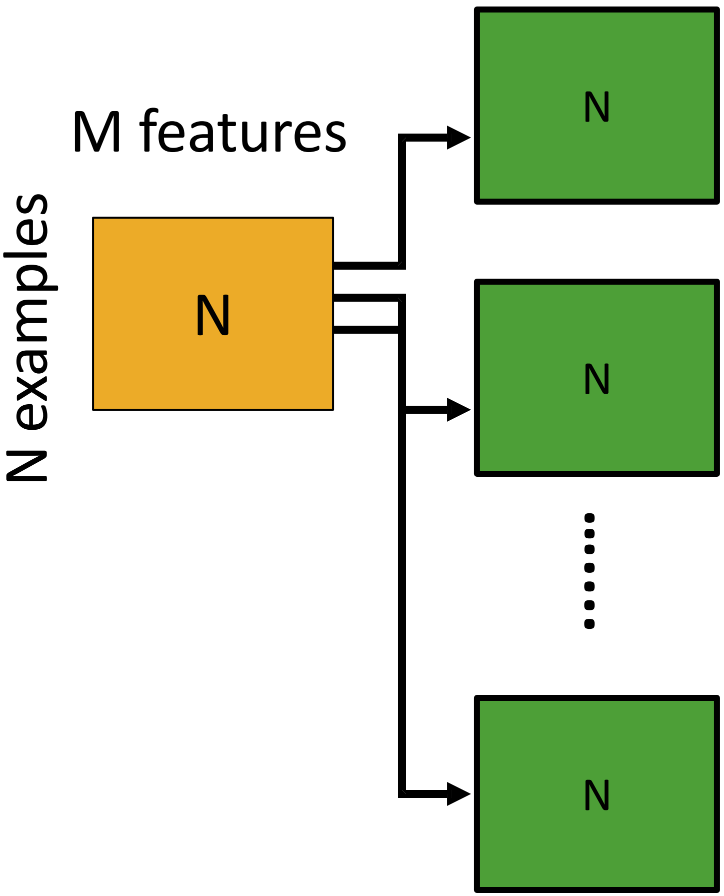
With vs Without Replacement

- Bootstrap with replacement can keep the sampling size the same as the original size for every repeated sampling. The sampled data groups are independent on each other.

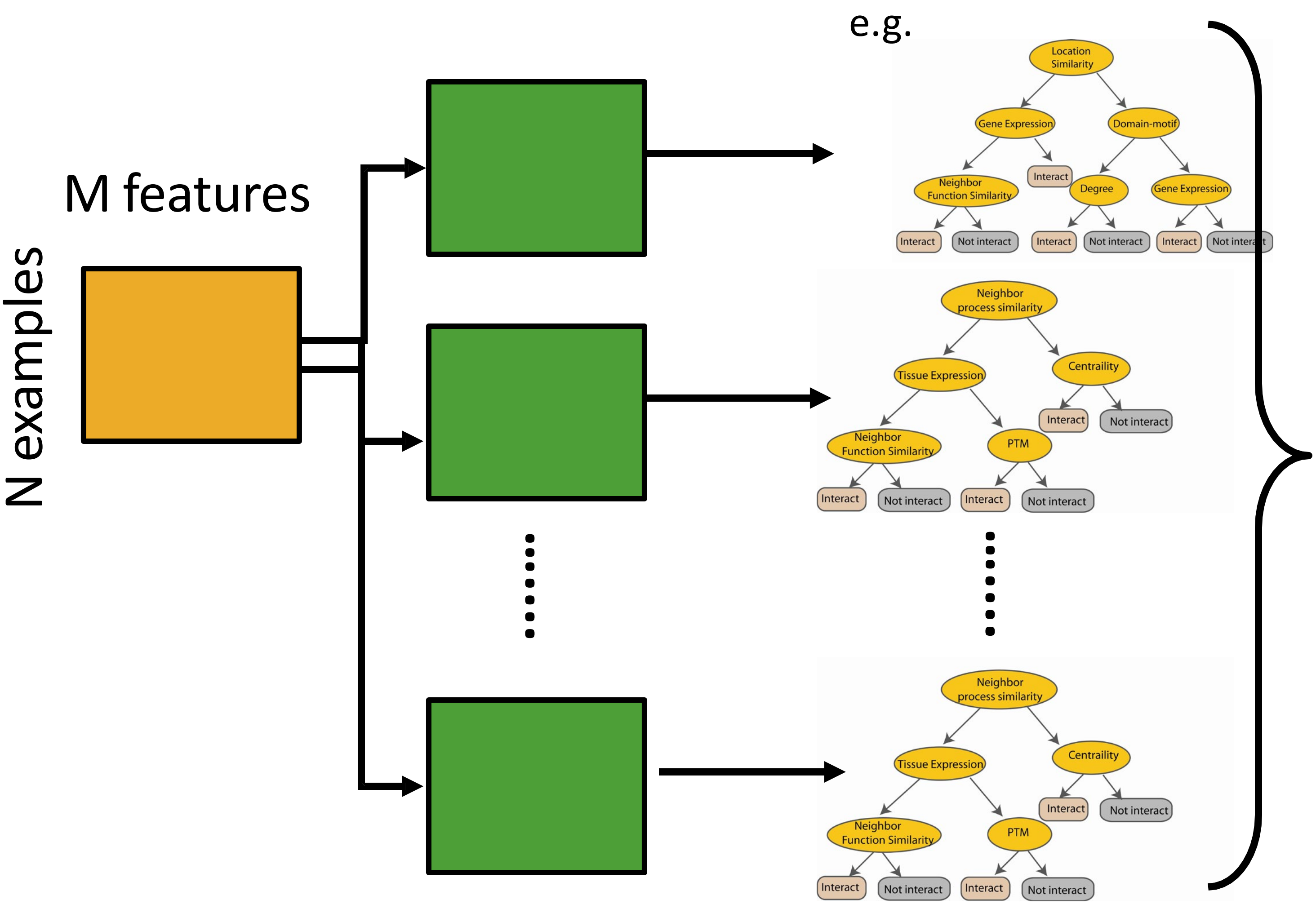
- Bootstrap without replacement cannot keep the sampling size the same as the original size for every repeated sampling. The sampled data groups are dependent on each other.

Bagging

Create bootstrap samples
from the training data



Bagging of DT Classifiers



Take the majority vote

i.e. Refit the model to each bootstrap dataset, and then examine the behavior over the B replications.

Peculiarities of Bagging

- Model Instability is good when bagging
  - The more variable (unstable) the basic model is, the more improvement can potentially be obtained
  - Low-Variability methods (e.g. SVM, LDA) improve less than High- Variability methods (e.g. decision trees)

Can understand the bagging effect in terms of a consensus of independent ***weak leaners*** and ***wisdom of crowds***

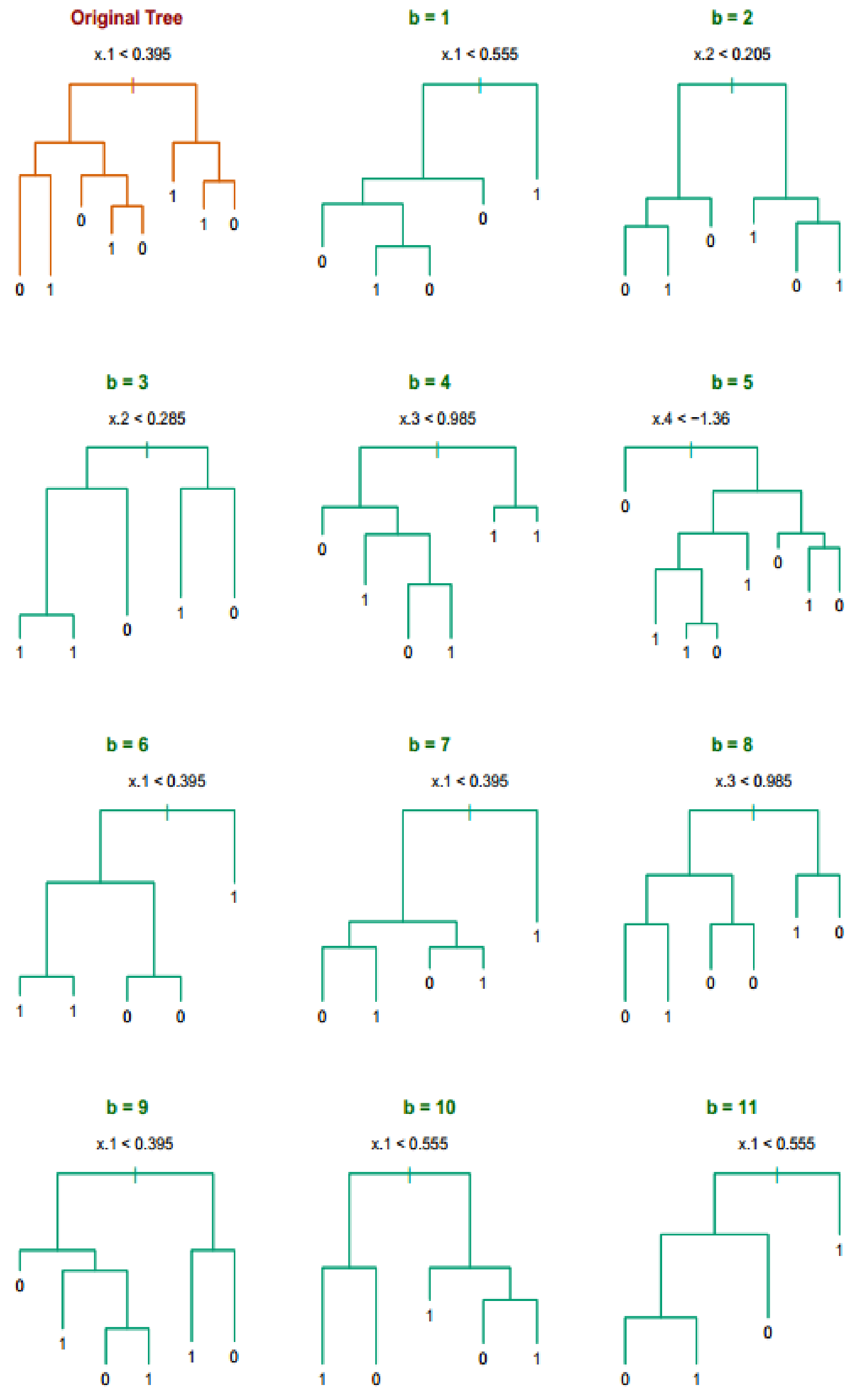
Bagging : an example with simulated data

- N = 30 training samples,
- two classes and p = 5 features,
- Each feature N(0, 1) distribution and pairwise correlation .95 Response Y generated according to:

$$\Pr(Y = 1 | x_1 \leq 0.5) = 0.2 \qquad \Pr(Y = 1 | x_1 > 0.5) = 0.8$$

- Test sample size of 2000
- Fit classification trees to training set and bootstrap samples B = 200

Notice the bootstrap trees are different than the original tree

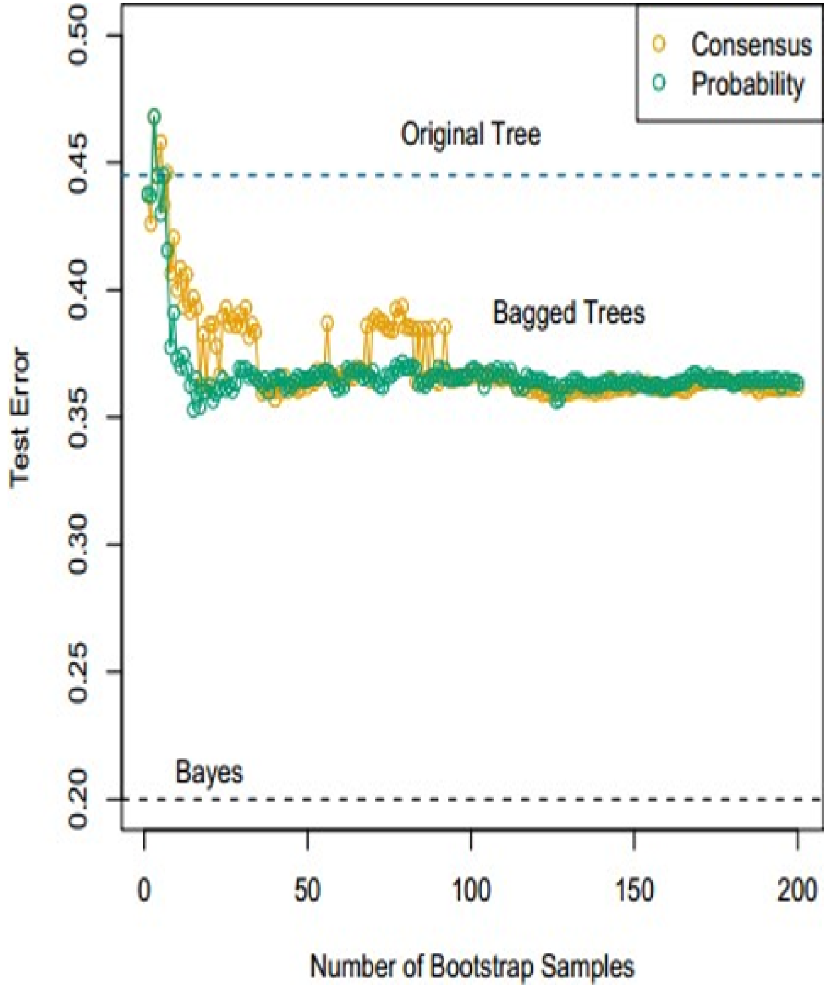


Five features highly correlated with each other

→ No clear difference with picking up which feature to split

→ Small changes in the training set will result in different tree

→ But these trees are actually quite similar for classification



B

- ➔ For B>30, more trees do not improve the bagging results
- ➔ Since the trees correlate highly to each other and give similar classifications

Consensus: Majority vote

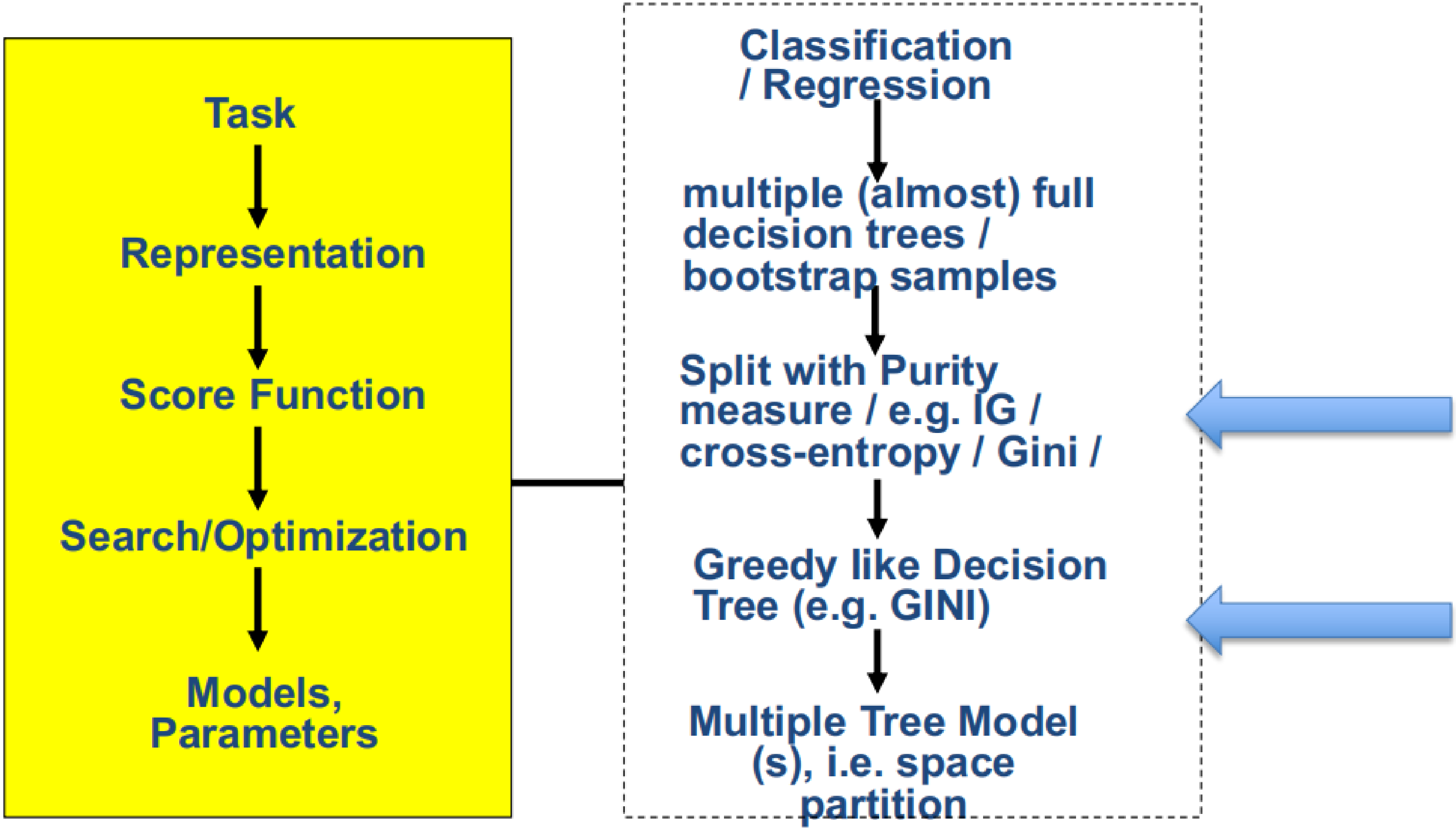
Probability: Average distribution at terminal nodes

Bagging

- Slightly increases model space
  - Cannot help where greater enlargement of space is needed

- Bagged trees are correlated
  - Use random forest to reduce correlation between trees

Bagged Decision Tree



References

- Prof. Tan, Steinbach, Kumar’s “Introduction to Data Mining” slide
- ESLbook : Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.
- Dr. Oznur Tastan’s slides about RF and DT
- Dr. Camilo Fosco’s slides

Thanks for listening