UVA CS 6316 – Fall 2015 Graduate: Machine Learning

Lecture 17: Decision Tree / Random Forest / Ensemble

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

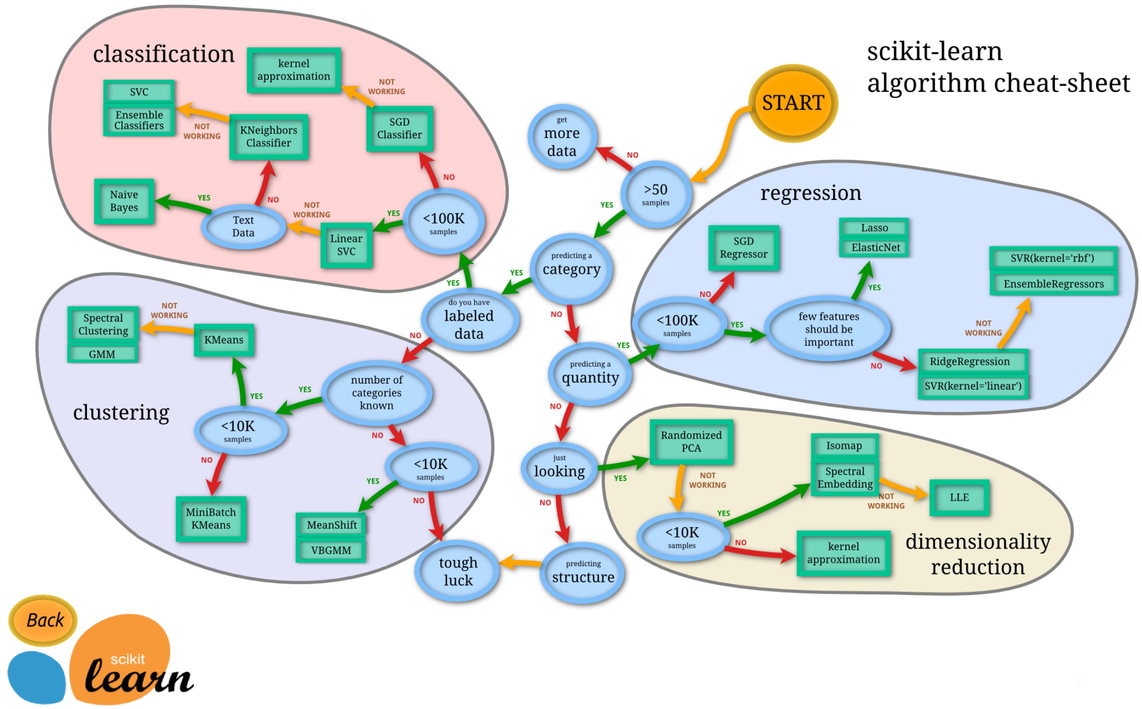
Dr. Yanjun Qi / UVA CS 6316 / f15

Where are we ? ➔ Five major sections of this course

- ❑ Regression (supervised)
- ❑ Classification (supervised)
- ❑ Unsupervised models
- ❑ Learning theory
- ❑ Graphical models

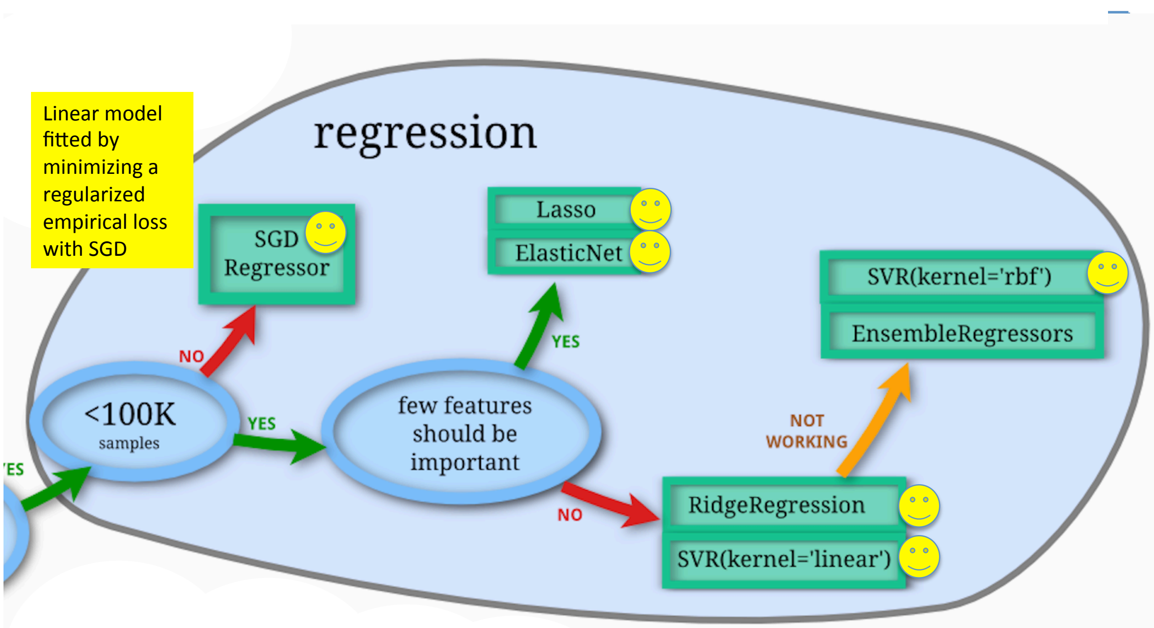
http://scikit-learn.org/stable/tutorial/machine_learning_map/

Choosing the right estimator

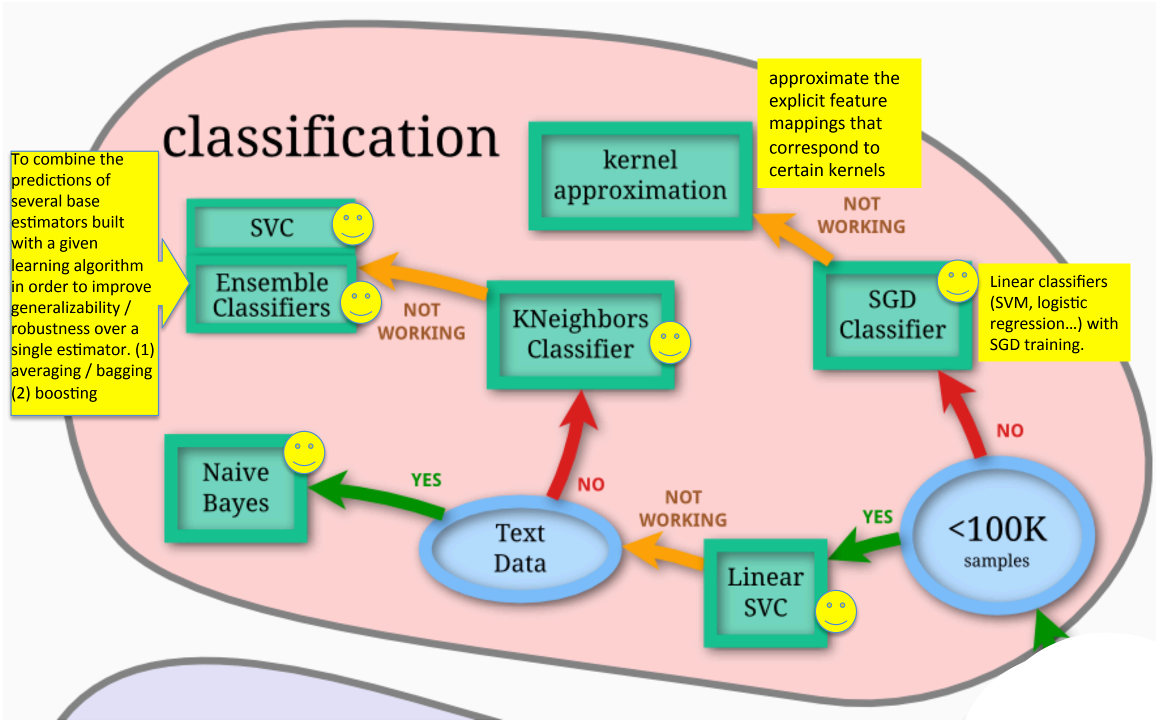


Scikit-learn : Regression

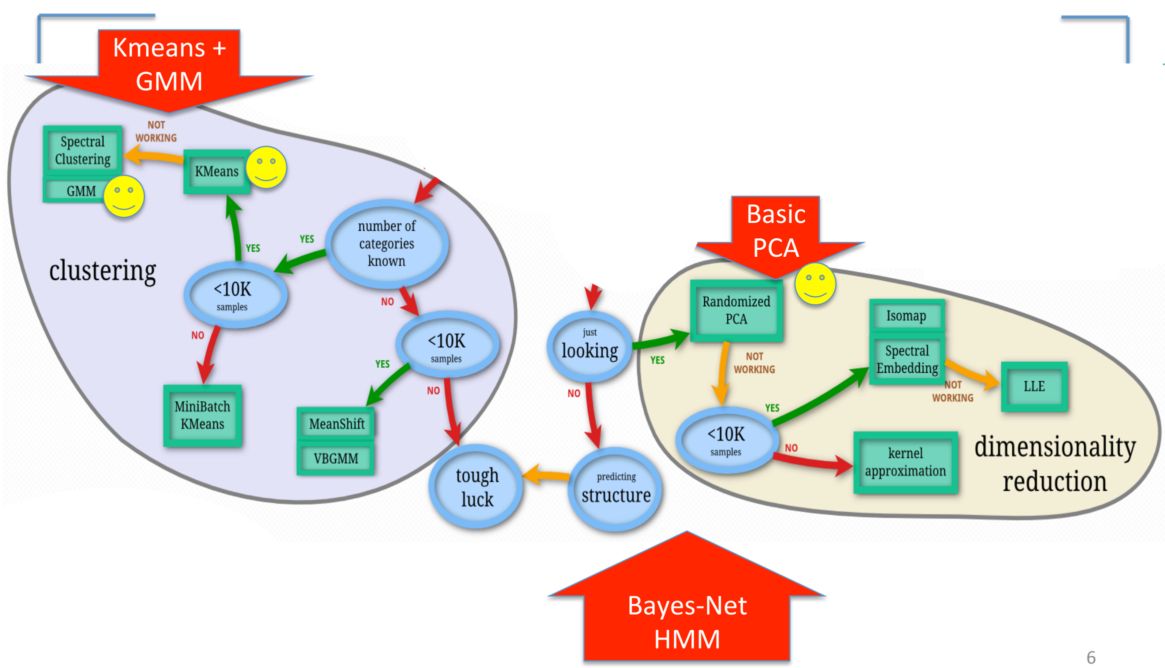
Linear model fitted by minimizing a regularized empirical loss with SGD



Scikit-learn : Classification



next after classification ?



Today

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

A study comparing Classifiers

An Empirical Comparison of Supervised Learning Algorithms

Rich Caruana CARUANA@CS.CORNELL.EDU
Alexandru Niculescu-Mizil ALEXN@CS.CORNELL.EDU
Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

Abstract

A number of supervised learning methods have been introduced in the last decade. Unfortunately, the last comprehensive empirical evaluation of supervised learning was the Statlog Project in the early 90's. We present a large-scale empirical comparison between ten supervised learning methods: SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps. We also examine the effect that calibrating the models via Platt Scaling and Isotonic Regression has on their performance. An important aspect of our study is

This paper presents results of a large-scale empirical comparison of ten supervised learning algorithms using eight performance criteria. We evaluate the performance of SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps on eleven binary classification problems using a variety of performance metrics: accuracy, F-score, Lift, ROC Area, average precision, precision/recall break-even point, squared error, and cross-entropy. For each algorithm we examine common variations, and thoroughly explore the space of parameters. For example, we compare ten decision tree styles, neural nets of many sizes, SVMs with many kernels, etc.

Because some of the performance metrics we examine

Proceedings of the 23rd International Conference on Machine Learning (ICML `06).

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

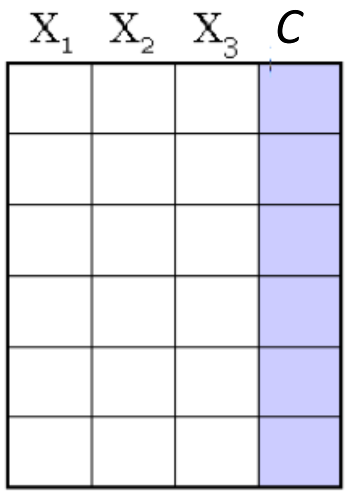
Where are we ? ➔
Three major sections for classification

- We can divide the large variety of classification approaches into roughly three major types

1. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., logistic regression, support vector machine, decisionTree

2. Generative:
   - build a generative statistical model
   - e.g., naïve bayes classifier, Bayesian networks

3. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors



A Dataset for classification

$$f : X \longrightarrow C$$

Output as Discrete Class Label $C_1, C_2, \ldots, C_L$

- **Data**/*points/instances/examples/samples/records*: [rows]
- **Features**/*attributes/dimensions/independent variables/covariates/predictors/regressors*: [columns, except the last]
- **Target**/*outcome/response/label/dependent variable*: special column to be predicted [last column]

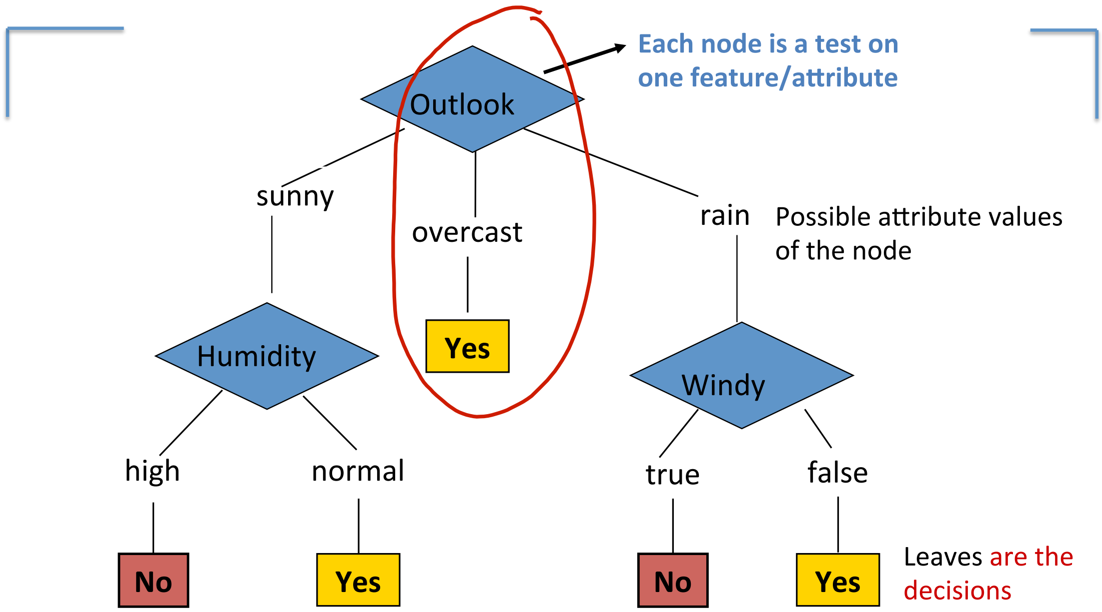
Example

- Example: Play Tennis

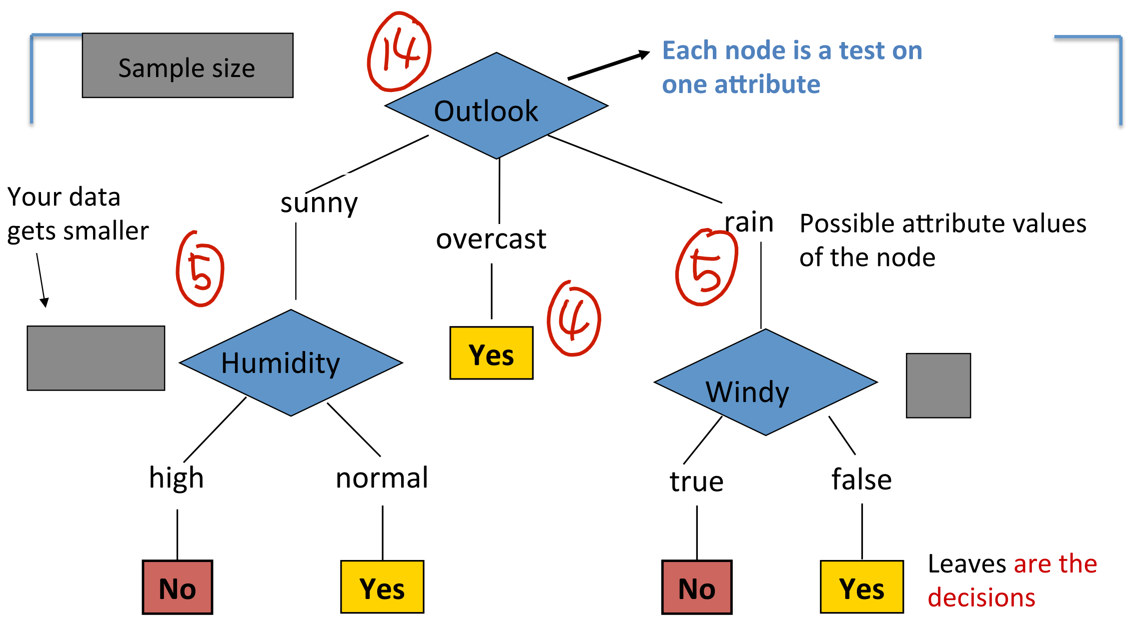
PlayTennis: training examples

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

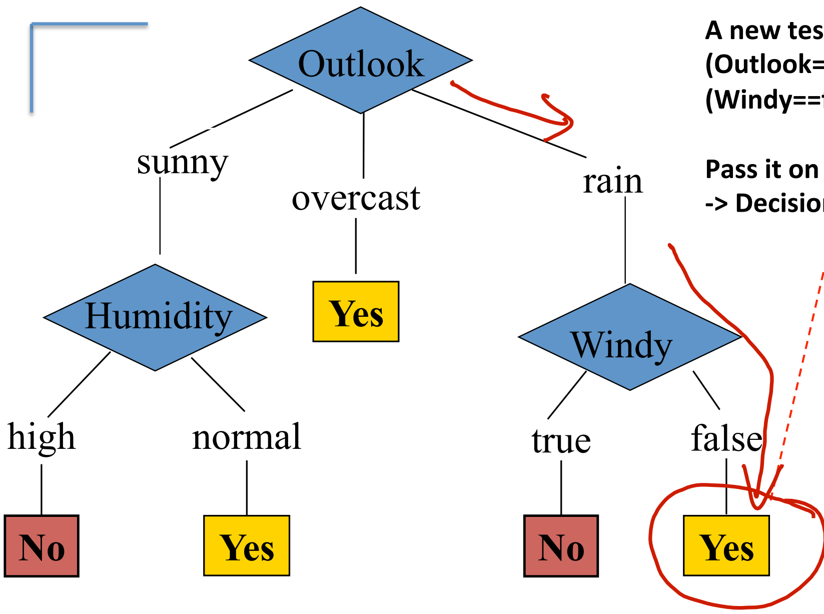


Anatomy of a decision tree



Apply Model to Test Data:

To ‘play tennis’ or not.

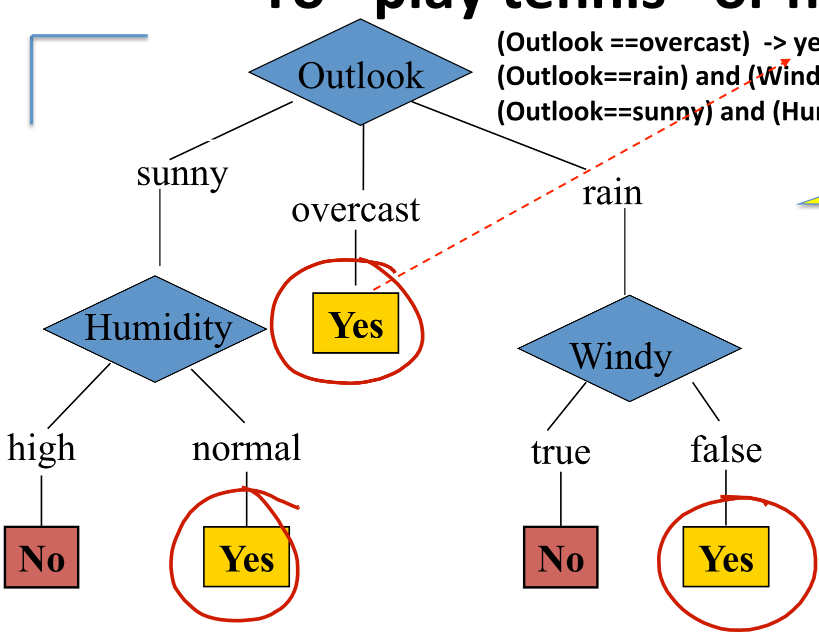


A new test example:
(Outlook==rain) and
(Windy==false)

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

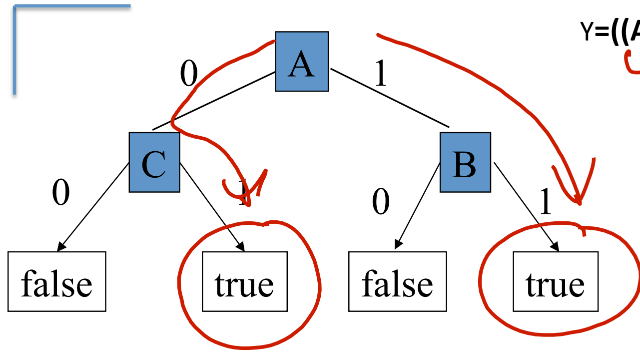
Three cases of “YES”

Decision trees

- 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
```


Representation

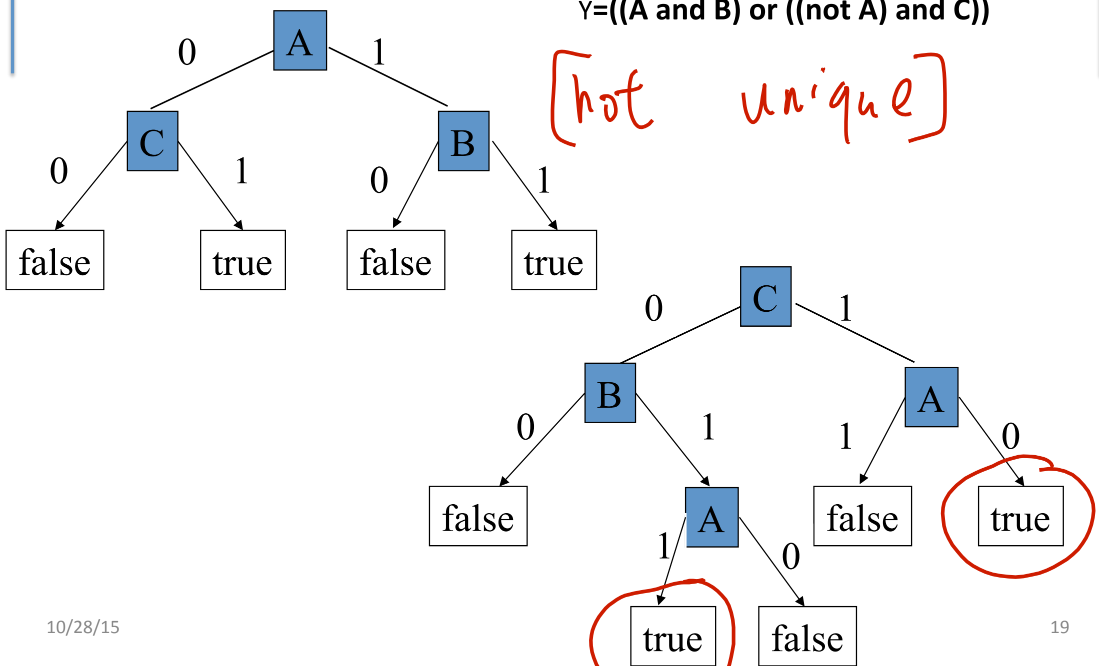


Y=((A and B) or ((not A) and C))

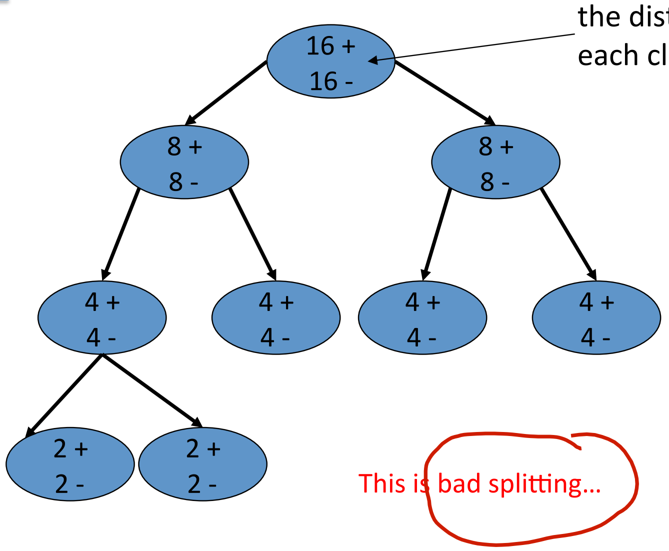
Same concept / different representation

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

[not unique]



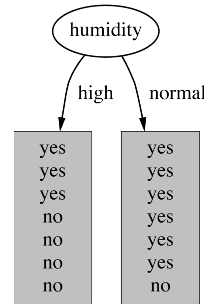
Which attribute to select for splitting?

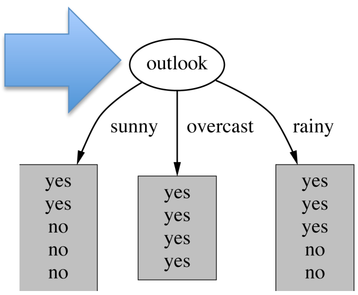


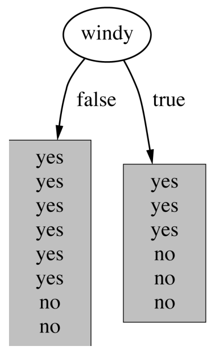
How do we choose which attribute to split ?

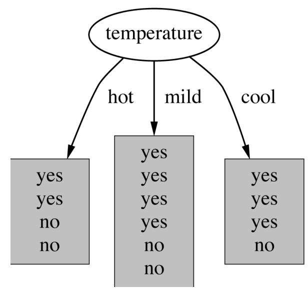
Which attribute should be used as the test?

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









Today

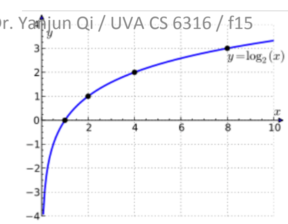
- Decision Tree (DT):
  - Tree representation
- Brief information theory
- Learning decision trees
- Bagging
- Random forests: Ensemble of DT
- More about ensemble

Information gain is one criteria to decide on which attribute for splitting

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

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

Information



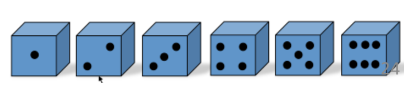
- Information:
- ➔ Reduction in uncertainty (amount of surprise in the outcome)

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

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 ⟶ $I = -\log_2 1/2 = 1$

- Observe the outcome of a dice is 6 ⟶ $I = -\log_2 1/6 = 2.58$



Entropy

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

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

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

$$H(X) = -\sum_i 1/8 \log_2 1/8 = 3$$

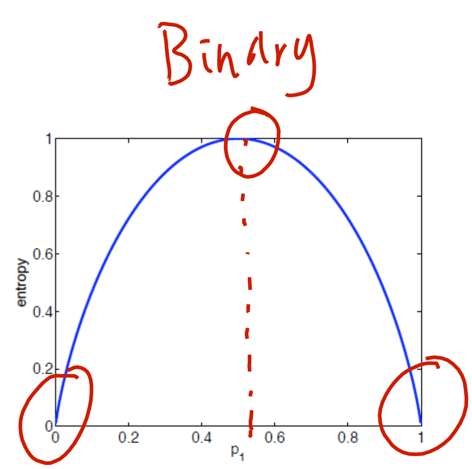
Entropy

- If there are *k* possible outcomes

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

- Equality holds when all outcomes are equally likely

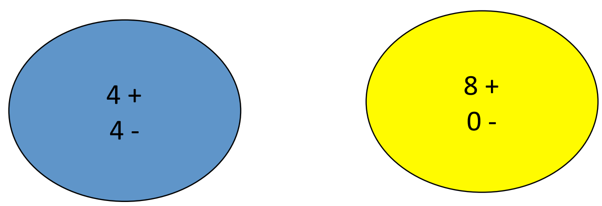
- The more the probability distribution the deviates from uniformity, the lower the entropy



e.g. for a random binary variable

Entropy Lower → better purity

- Entropy measures the purity



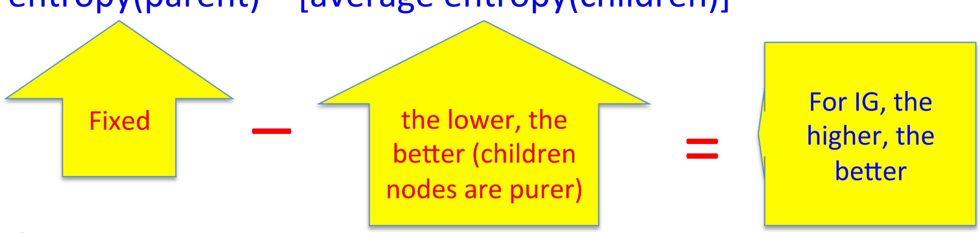
The distribution is less uniform
Entropy is lower
The node is purer

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)]



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

X1 or X2?

$H(Y) = -\sum p(y_i)\log p(y_i)$

$p(y=+) = 5/10$, $p(y=-) = 5/10$ (+ 5, − 5)

IG(X1,Y) = H(Y) – H(Y|X1)

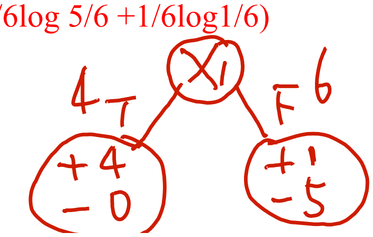
H(Y) = - (5/10) log(5/10) -5/10log(5/10) = 1

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?

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

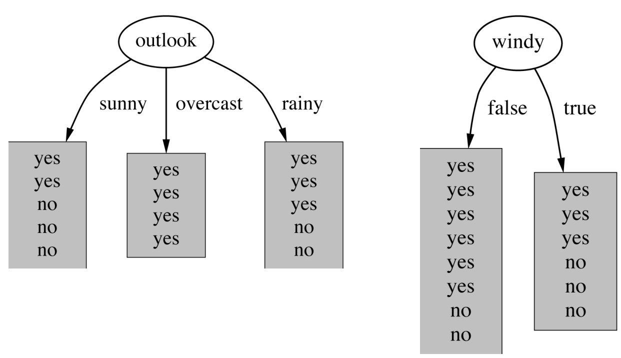
X1	X2	Y	Count	
T	T	+	2	One branch
T	F	+	2	
F	T	-	5	The other branch
F	F	+	1	

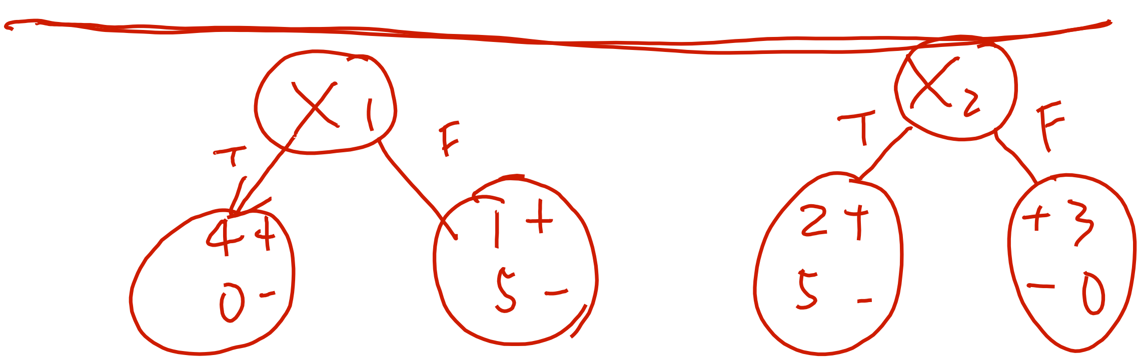
Information gain (X1,Y)= 0.61 $= H(Y) - H(Y|X_1)$ ⇒ smaller, purer ⇓ IG larger Better

Information gain (X2,Y)= 0.12 $= H(Y) - H(Y|X_2)$

Pick the variable which provides the most information gain about Y → Pick X1

➔ **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 diversity) $\sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$
    - where p_{mk} is proportion of class k at node m
  - Chi-square Test

Overfitting

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

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

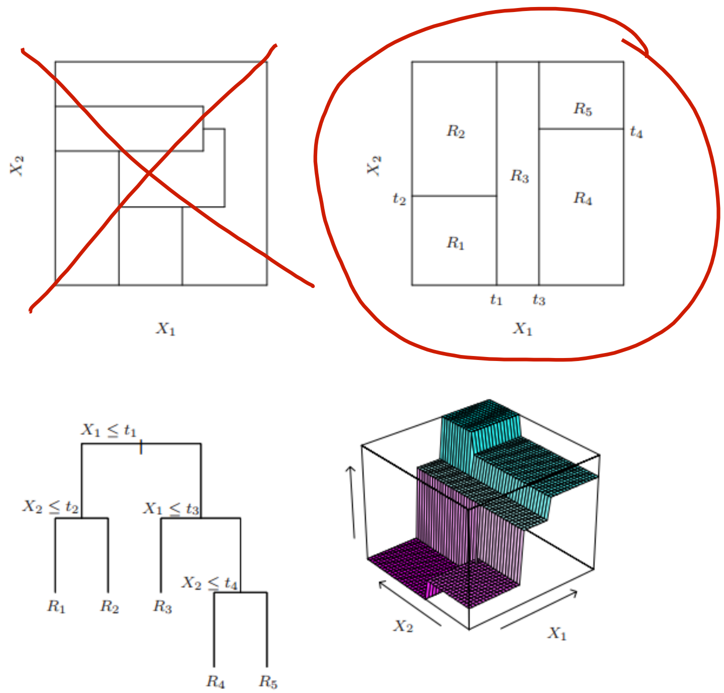
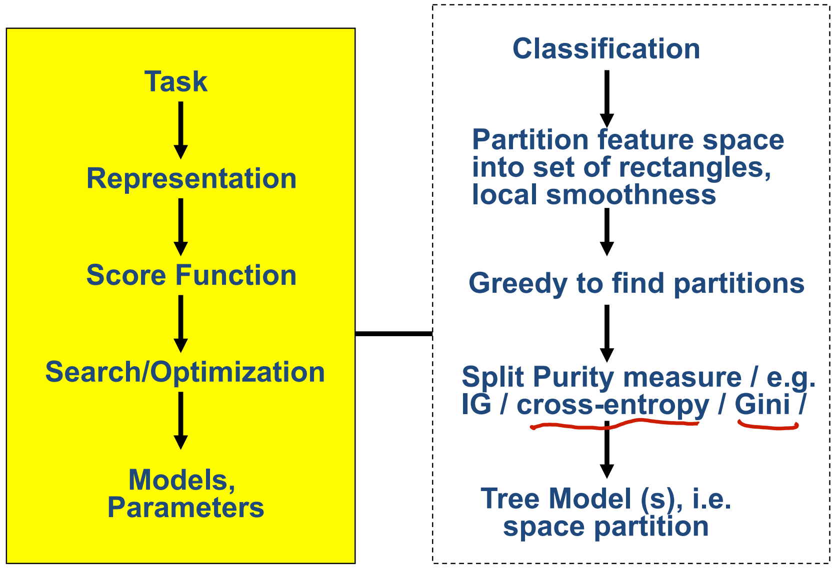


FIGURE 9.2. *Partitions and CART. Top right panel shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some fake data. Top left panel shows a general partition that cannot be obtained from recursive binary splitting. Bottom left panel shows the tree corresponding to the partition in the top right panel, and a perspective plot of the prediction surface appears in the bottom right panel.*

Summary: Decision trees

- Non-linear classifier
- 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

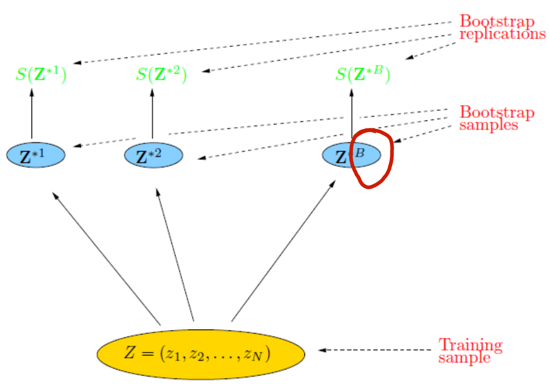
- Bagging or *bootstrap aggregation*
  - a technique for reducing the variance of an estimated prediction function.

- 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 sample *the same size as the original training set*



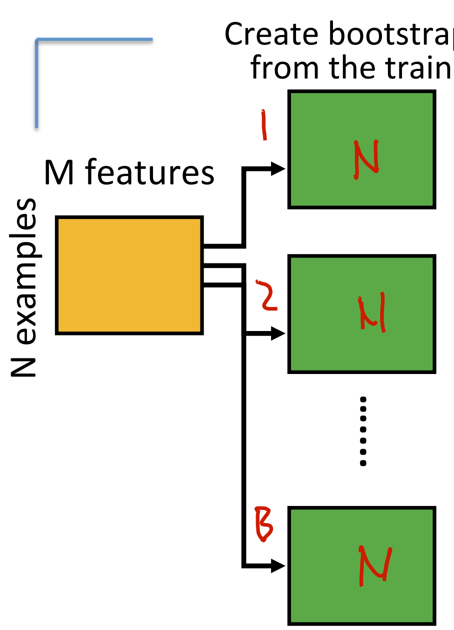
$$\widehat{\mathrm{Var}}[S(\mathbf{Z})] = \frac{1}{B-1}\sum_{b=1}^{B}(S(\mathbf{Z}^{*b}) - \bar{S}^{*})^2,$$

With vs Without Replacement

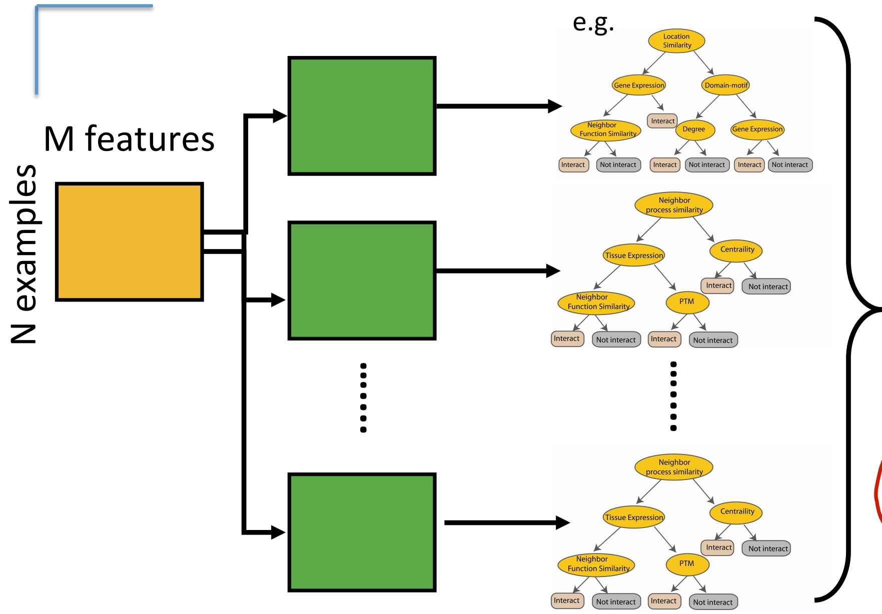
- Bootstrap with replacement can keep the sampling size the same as the original size for every repeated sampling. The sampled data groups are independent on each other.

- 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



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.

Bagging for Classification with 0,1 Loss

- Classification with 0, 1 loss
  - Bagging a **good** classifier can make it **better**.
  - Bagging a **bad** classifier can make it **worse**.
  - Can understand the bagging effect in terms of a consensus of independent ***weak leaners*** and ***wisdom of crowds***

Peculiarities

- 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. LDA) improve less than High-Variability methods (e.g. decision trees)
- Load of Redundancy
  - Most predictors do roughly "the same thing"

Bagging : an simulated example

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:

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

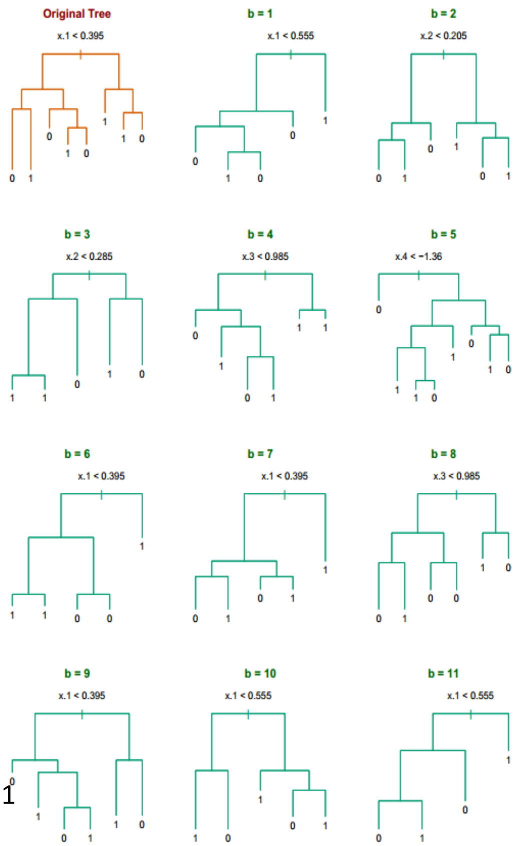
Test sample size of 2000

Fit classification trees to training set and bootstrap samples

B = 200

ESL book / Example 8.7.1

Notice the bootstrap trees are different than the original tree



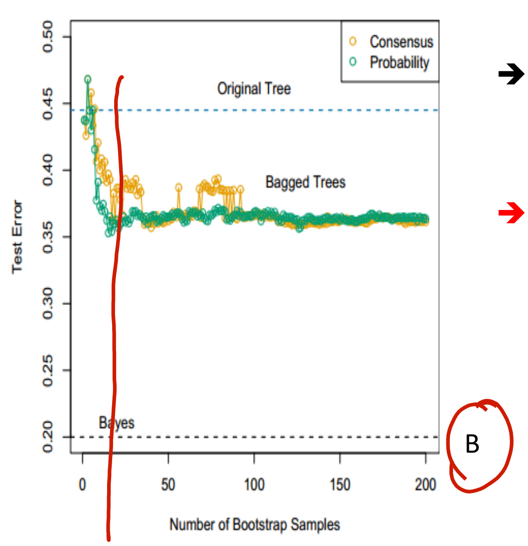
Five features highly correlated with each other

➔ 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

ESL book / Example 8.7.1



➔ 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

ESL book / Example 8.7.1

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

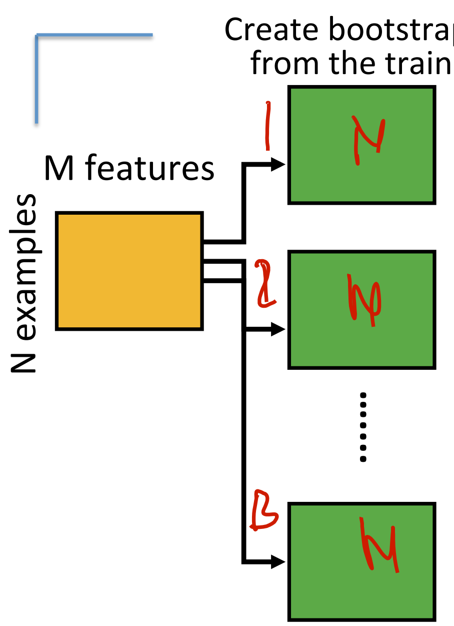
Today

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

Random forest classifier

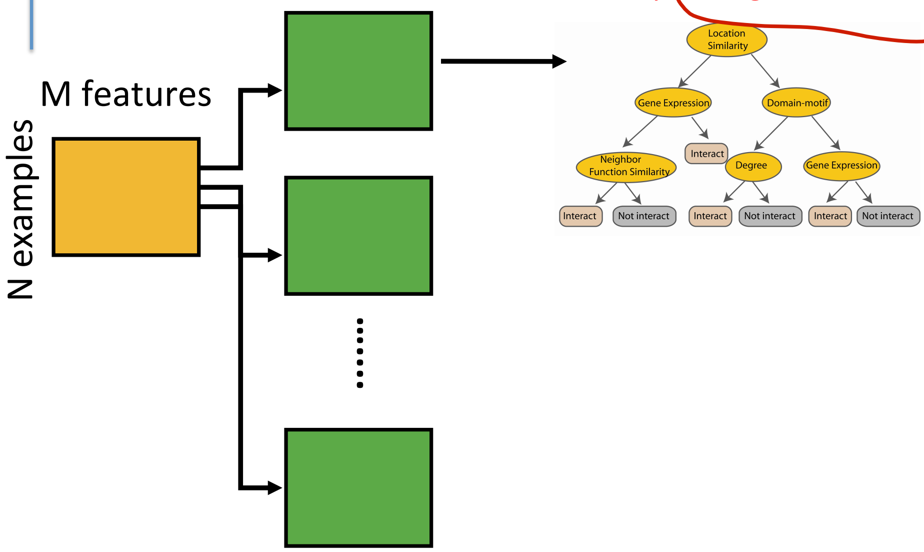
- Random forest classifier,
  - an extension to bagging
  - which uses *de-correlated* trees.

Random Forest Classifier



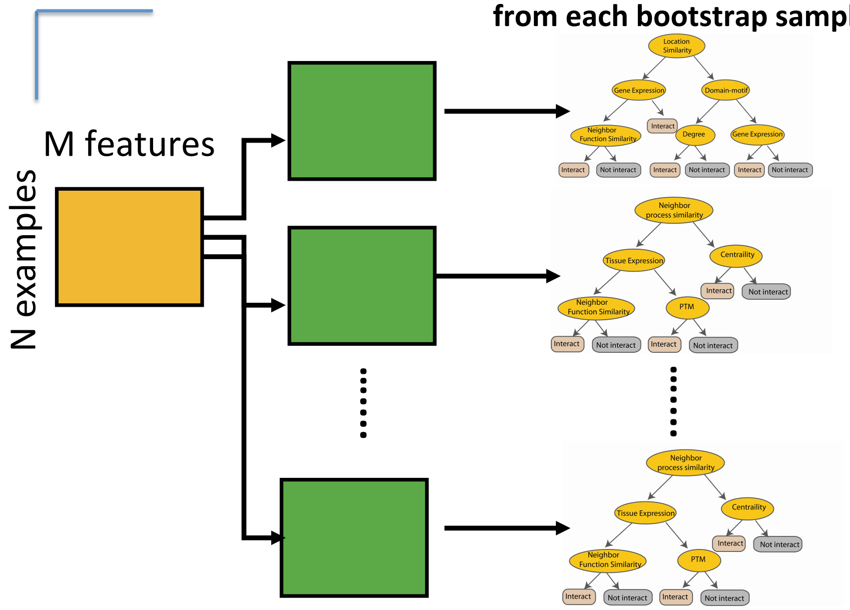
Random Forest Classifier

At each node when ~~choosing the split feature~~
choose only among *m*<*M* features

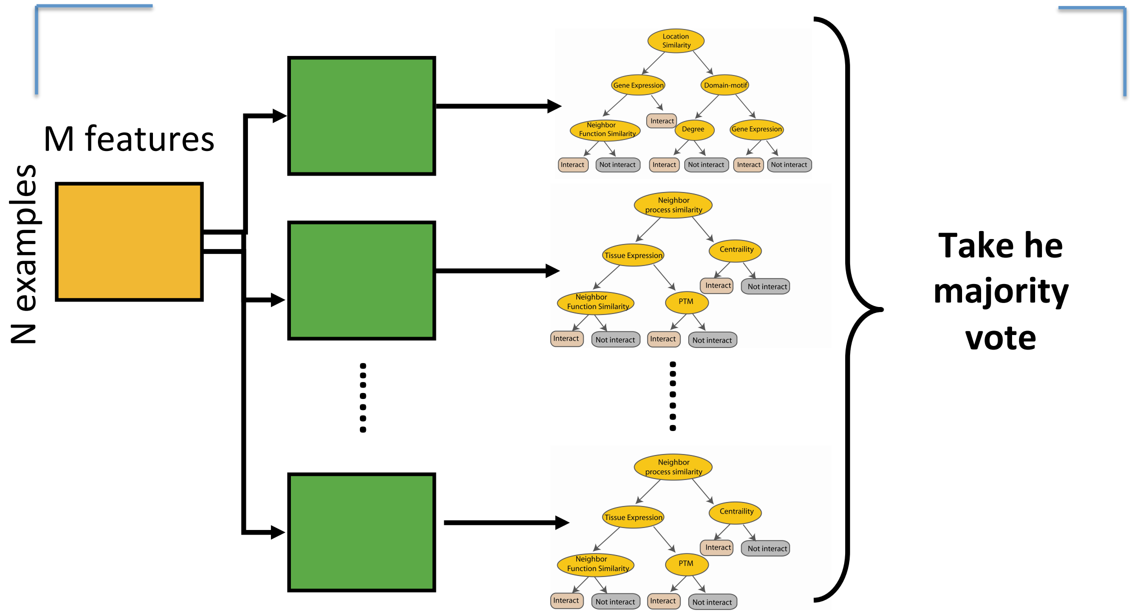


Random Forest Classifier

Create decision tree
from each bootstrap sample



Random Forest Classifier



Random Forests

For each of our *B* bootstrap samples

- Form a tree in the following manner
  - Given *p* dimensions, pick *m* of them
  - Split only according to these *m* dimensions
    - (we will NOT consider the other *p*-*m* dimensions)
  - Repeat the above steps i & ii for each split
    - Note: we pick a different set of *m* dimensions for each split on a single tree

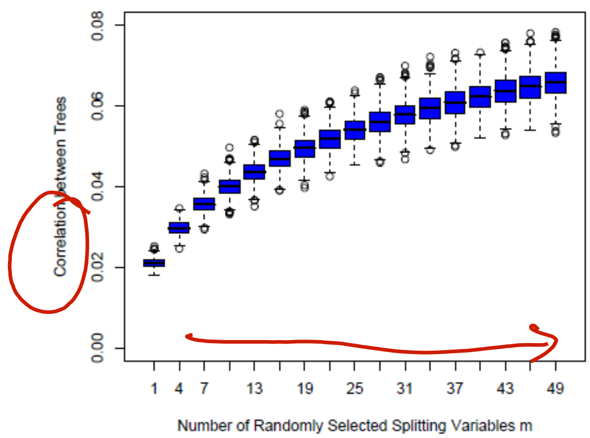


FIGURE 15.9. *Correlations between pairs of trees drawn by a random-forest regression algorithm, as a function of m. The boxplots represent the correlations at 600 randomly chosen prediction points x.*

Page 598-599 In ESL book

Random Forests

Random forest can be viewed as a refinement of bagging with a tweak of **decorrelating** the trees:

At each tree split, a random subset of **m** features out of all **p** features is drawn to be considered for splitting

Some guidelines provided by Breiman, but be careful to choose m based on specific problem:

m = p amounts to bagging

m = p/3 or log2(p) for regression

m = sqrt(p) for classification

Why correlated trees are not ideal ?

Random Forests try to reduce correlation between the trees.

Why?

Why correlated trees are not ideal ?

Assuming each tree has variance σ^2

If trees are independently identically distributed, then average variance is σ^2/B

Why correlated trees are not ideal ?

Assuming each tree has variance σ^2

If simply identically distributed, then average variance is

$$\Rightarrow \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

As B → ∞, second term → 0

Thus, the pairwise correlation always affects the variance

Why correlated trees are not ideal ?

How to deal?

If we reduce *m* (the number of dimensions we actually consider),

then we reduce the pairwise tree correlation

Thus, variance will be reduced.

Today

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

e.g. Ensembles in practice

Oct 2006 - 2009

Each rating/sample:
+ <user, movie, date of grade, grade>
Training set (100,480,507 ratings)
Qualifying set (2,817,131 ratings)→ winner

- Training data is a set of users and ratings (1,2,3,4,5 stars) those users have given to movies.
- Predict what rating a user would give to any movie

- $1 million prize for a 10% improvement over Netflix's current method (MSE = 0.9514)

Ensemble in practice

Team "Bellkor's Pragmatic Chaos" defeated the team "ensemble" by submitting just 20 minutes earlier! ➔ 1 million dollar !

Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11

The ensemble team ➔ blenders of multiple different methods

References

- Prof. Tan, Steinbach, Kumar's "Introduction to Data Mining" slide
- 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