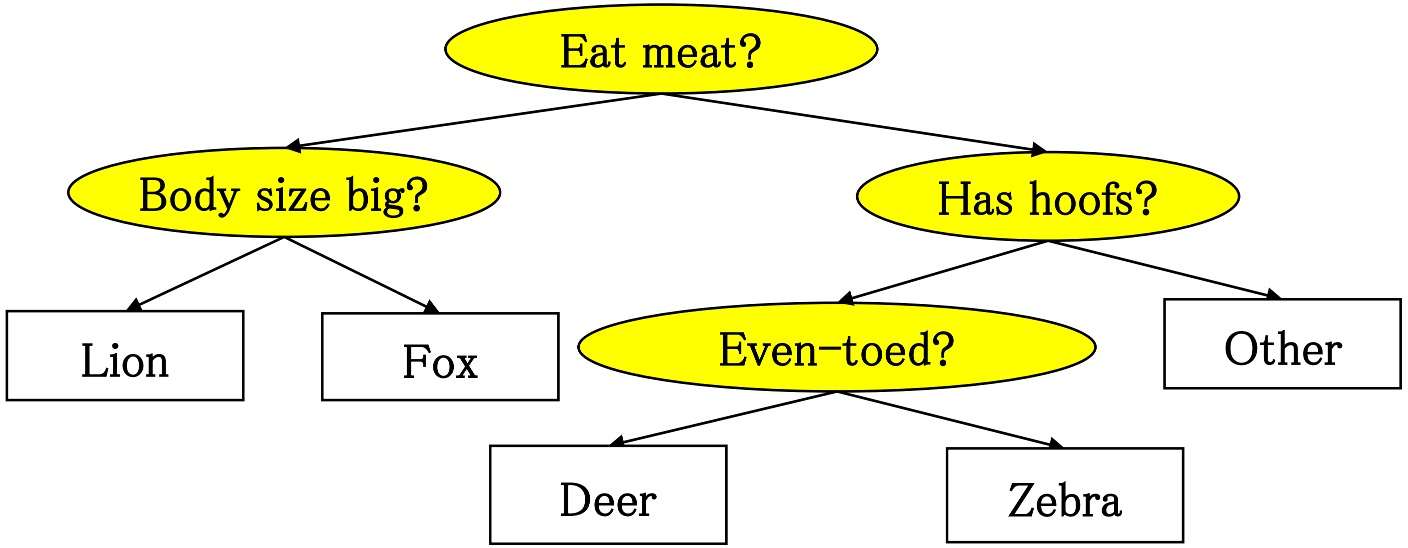
Lecture 10:
Trees, NNTree, and Ensembles

Topics of this lecture

- Definition of decision trees
- Inference with a DT
- Learning with a DT
- NNTree: combination of neural network and DT
- Ensemble learning: Basics
- Ensemble learning: Bagging
- Ensemble learning: AdaBoost
- Ensemble learning: Random forest

Definition of a decision tree

- In a decision tree, there two types of nodes: internal nodes and leaf nodes.
- The internal nodes are used to make local decisions based on the local information they possess; and the leaf nodes make the final decisions.



Definition of a decision tree

- Information used for local decision
  - Feature(s) to use, and a condition for visiting the next child.
  - In the internal node of a standard decision tree,

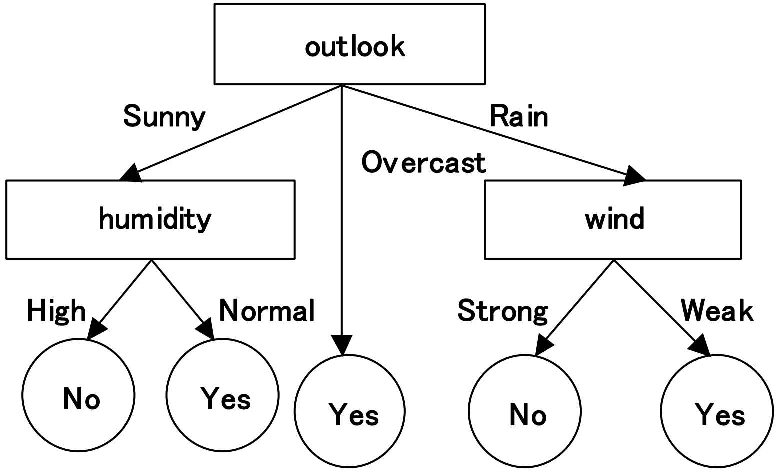
$$f(x) = x_i - a_i$$

is often used as a ***test function*** for making a local decision.

- Information used for final decision
  - Distribution of examples assigned to the leaf node by the tree.
  - Usually the "label" of a leaf node is determined via "majority voting".

Example: Shall I play tennis today ?
(from "Machine learning", written by T. M. Mitchell).

- Play tennis if (outlook is sunny & humidity is normal).
- Play tennis if (outlook is overcast).
- Play tennis if (outlook is rain & wind is weak).
- Otherwise not play.



A decision tree is a set of decision rules !

Inference using a DT

- Step 1: Set the root as the current node.
- Step 2: If the current node n is a leaf, return its class label and stop; otherwise, continue.
- Step 3: If f(x)<0, n=n->left; otherwise, n=n->right. Return to Step 2.

f(x) is the test function of node n

Learning with a DT

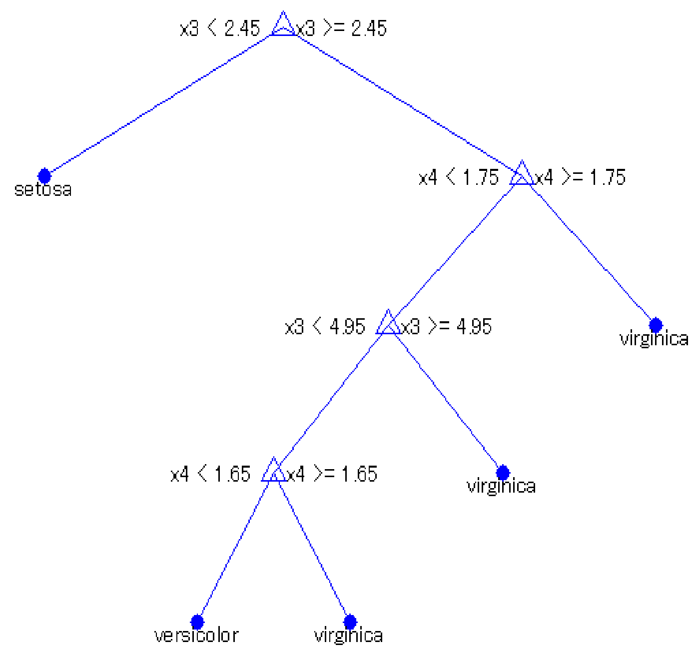
- At the beginning, assign all training examples to the root, and set the root as the current node.
- Do the following recursively:
  - If all training examples assigned to the current node belong to the same class, the current node is a leaf, and the common label of the examples is the label of this node.
  - Otherwise, the node is an internal node. Find a feature x_i and a threshold a_i, and divide all training examples assigned to this node into two groups. All examples in the first group satisfy $x_i < a_i$, and all examples in the second group do not satisfy this condition.
  - Assign the examples of each group to a child, and do the same thing recursively for each child.

Learning with a DT

- Splitting nodes:
  - How to determine the feature to use and the threshold ?
  - Usually we have a criterion (e.g. information gain ratio).
  - The feature and threshold are chosen so as to optimize the criterion.
- Determining which nodes are terminal:
  - The simplest way is to see if all examples are of the same class.
  - This simple way may result in large trees with less generalization ability.
  - An impure node can also be a terminal node.
- Assigning class label to the terminal nodes:
  - Majority voting is often used for classification.
  - Weighted sum is often used for regression.

Example: DT for Iris
(Results obtained using Matlab)

1 if x3<2.45 の場合はノード 2、elseif x3>=2.45 の場合はノード 3、else の場合は setosa
2 クラス = setosa
3 if x4<1.75 の場合はノード 4、elseif x4>=1.75 の場合はノード 5、else の場合は versicolor
4 if x3<4.95 の場合はノード 6、elseif x3>=4.95 の場合はノード 7、else の場合は versicolor
5 クラス = virginica
6 if x4<1.65 の場合はノード 8、elseif x4>=1.65 の場合はノード 9、else の場合は versicolor
7 クラス = virginica
8 クラス = versicolor
9 クラス = virginica



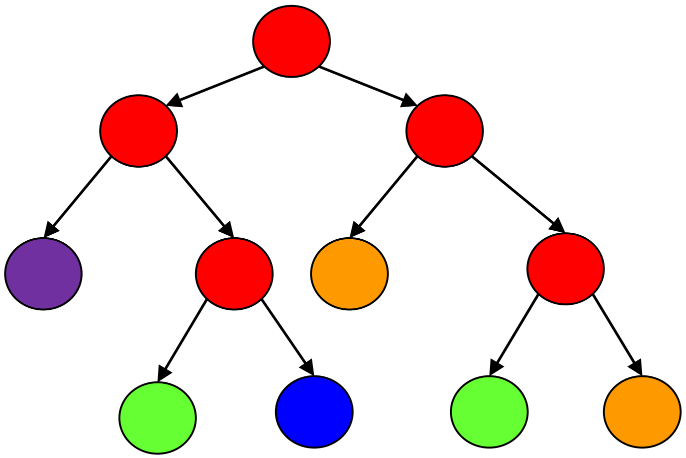
Pros and cons of DTs

- Pros:
  - Comprehensible.
  - Easy to design.
  - Easy to implement.
  - Good for structural learning.

- Cons
  - May become very large for complex problems.
  - Difficult to know the true concept.
  - Too many rules to be understood by human users.

Neural Network Tree (NNTree)

- NNTree is a multi-variate decision tree in which each internal node has a test function realized by an NN.
- Chicken and egg problem:
  - How to partition the data?
  - How to find the test function?
- Generation and test is not suitable for NNTree design.



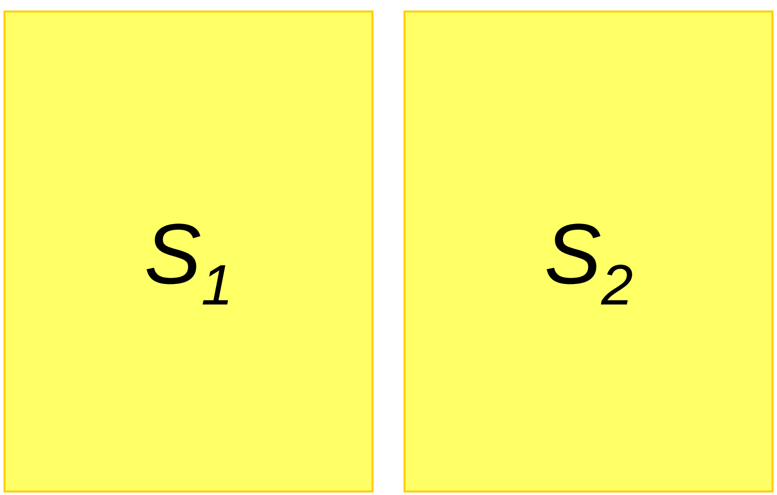
Q. F. Zhao, "Inducing NNC-Trees with the R4-Rule," IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics, Vol. 36, No. 3, pp. 520-533, 2006.

To induce NNTrees efficiently?

- Instead of generating many decision functions, we propose to generate only one decision function through supervised learning.
- The teacher signal $g(x)$ of a data is called the group label.
  - If $g(x) = i$, x is assigned to the i-th child of the current node.
- We use the following heuristics to find the group label for each datum.

> - **Put all data with the same class label to the same group.**
> - **Put data that are close to each other to the same group.**

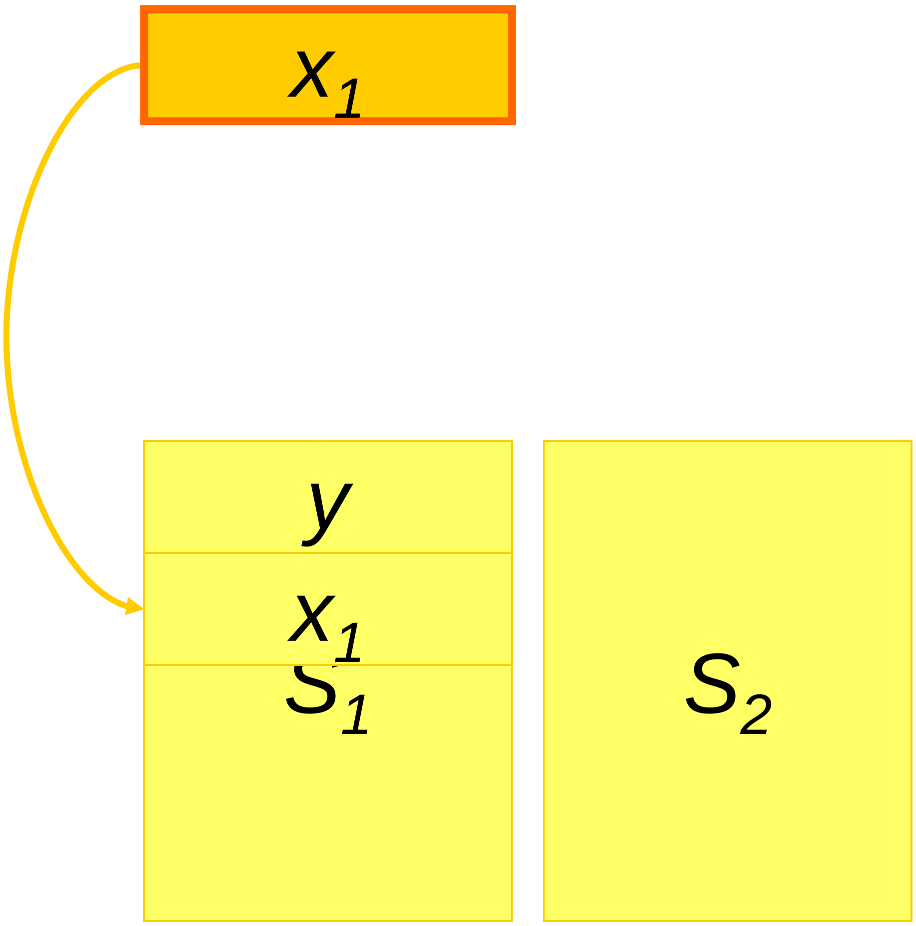
Definition the teacher signals



- Suppose that we want to partition S into N sub-sets $S_1, \ldots, S_N$.

1. If there is a $y \in S_i$, such that $label(x) = label(y)$, assign x to S_i.
2. Else if there is a S_i, such that S_i = *empty set*, assign x to S_i.
3. Else if find y, which is the nearest neighbor of x in S_i, assign x to same sub-set as y.

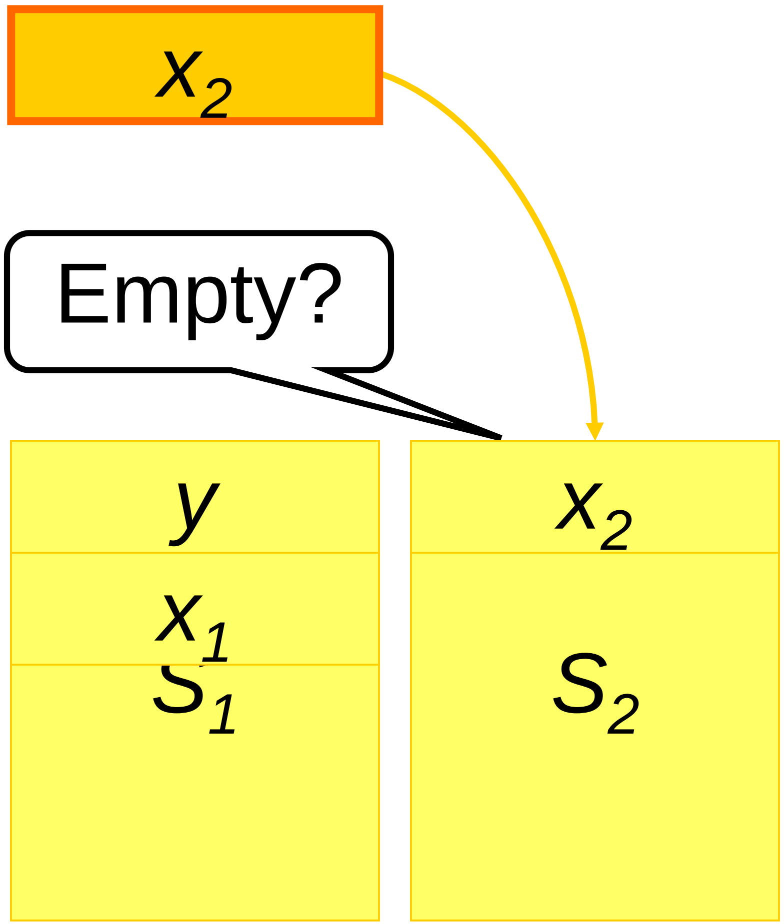
Definition the teacher signals



- Suppose that we want to partition S into N sub-sets $S_1, \ldots, S_N$.

1. If there is a $y \in S_i$, such that $label(x) = label(y)$, assign x to S_i.
2. Else if there is a S_i, such that S_i = *empty set*, assign x to S_i.
3. Else if find y, which is the nearest neighbor of x in S_i, assign x to same sub-set as y.

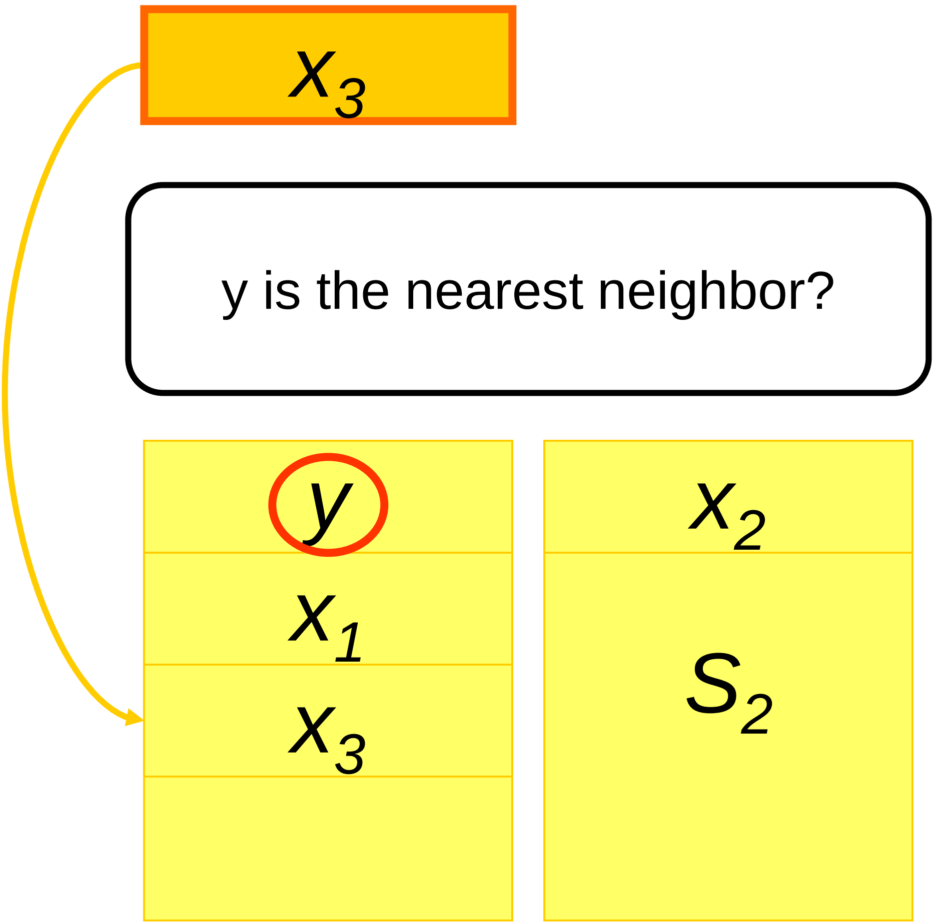
Definition the teacher signals



- Suppose that we want to partition S into N sub-sets $S_1, \ldots, S_N$.

1. If there is a $y \in S_i$, such that $label(x) = label(y)$, assign x to S_i.
2. Else if there is a S_i, such that S_i = *empty set*, assign x to S_i.
3. Else if find y, which is the nearest neighbor of x in S_i, assign x to same sub-set as y.

Definition the teacher signals



- Suppose that we want to partition S into N sub-sets $S_1, \ldots, S_N$.

1. If there is a $y \in S_i$, such that $label(x) = label(y)$, assign x to S_i.
2. Else if there is a S_i, such that S_i = *empty set*, assign x to S_i.
3. Else if find y, which is the nearest neighbor of x in S_i, assign x to same sub-set as y.

Method for inducing NNTrees

- Once the group labels are defined, we can find different kinds of decision functions using different learning algorithms.
- If we use a multilayer perceptron (MLP) in each internal node, we can use the back propagation (BP) algorithm.
- We can also use an NNC (nearest neighbor classifier) or SVM (support vector machine) in each internal node, and we may call the model NNC-Tree or SVM-Tree.

Advantages of NNTrees

- Adaptability
  - The NNs are learnable, and the tree can adapt to new data incrementally.
- Comprehensibility
  - Time complexity for interpreting is polynomial if the number of inputs for each NN is limited.
  - Or, if we consider each NN as a concept, the decision process is interpretable.
- Quicker decision
  - Since each internal node contains a multivariate decision function, long decision paths are not needed.

Qiangfu ZHAO, "Reasoning with Awareness and for Awareness," IEEE SMC Magazine, Vol. 3, No. 2, pp. 35-38, 2017.

Experimental results

Q. F. Zhao, "Inducing NNC-Trees with the R4-Rule," IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics, Vol. 36, No. 3, pp. 520-533, 2006.

Database		NNTree	NNC-Tree	APDT-See5	Oblique	NNC-m	NNC-M
cancer	Error	0.054 ± 0.027	0.047 ± 0.033	0.05 ± 0.032	0.039 ± 0.006	0.046 ± 0.026	**0.036 ± 0.034**
	Size	6.36 ±2.17	1.5 ± 1.39	7.34 ± 4.94	2.3 ± 0.6	10.7 ± 9.33	547
	Time	82 ± 28	1.08± 0.35	0 ± 0	28 ± 5	1.12 ± 0.7	0
diabetes	Error	0.321 ± 0.065	0.308 ± 0.07	0.266 ± 0.069	**0.259 ± 0.012**	0.26 ± 0.06	0.297 ± 0.041
	Size	35.9 ± 19.72	9.92 ± 4.30	21 ± 16.03	14.7 ± 1.24	9.4 ± 11.62	615
	Time	740 ± 224	4 ± 1	0.02 ± 0	33 ± 1	0.74 ± 0.36	0
glass	Error	0.359 ± 0.124	0.337 ± 0.128	0.296 ± 0.137	0.329 ± 0.009	0.390 ± 0.132	**0.295 ± 0.108**
	Size	18.36 ± 3.91	7.90 ± 2.14	17.8 ± 4.99	18.3 ± 4.1	15 ± 16.42	172
	Time	106 ± 28	0.99 ± 0.17	0.01 ± 0	0.1 ± 0.08	1.87 ± 0.22	0
iris	Error	0.039 ± 0.067	0.044 ± 0.048	0.057 ± 0.078	**0.037 ± 0.004**	0.04 ± 0.041	0.047 ± 0.081
	Size	2.88 ± 1.12	2.12 ± 0.85	3.04 ± 0.97	2.4 ± 0.3	4.1 ± 2.52	120
	Time	5 ± 6	0.08 ± 0.03	0 ± 0	0.9 ± 0.1	0.21 ± 0.03	0
vehicle	Error	0.263 ± 0.079	**0.220 ± 0.055**	0.276 ± 0.054	0.297 ± 0.007	0.225 ± 0.053	0.292 ± 0.056
	Size	40.12 ± 6.79	7.42 ± 4.10	57.76 ± 21.14	30.6 ± 4.8	18.9 ± 19.57	677
	Time	879±228	4 ± 1	0.04 ± 0	290 ±8	7 ± 5	0
optdigits	Error	0.055 ± 0.004	0.033 ± 0.003	0.104 ± 0.012	0.094 ± 0.006	0.035 ± 0.008	**0.014 ± 0.005**
	Size	43.18 ± 2.91	9 ± 0	156.84 ± 13.82	37.2 ± 10.0	19.5 ± 21.37	3823
	Time	5033 ± 421	51 ± 20	0.47 ± 0.03	1305 ± 33	389 ± 23	0
pen-based	Error	0.024 ± 0.003	0.017 ± 0.003	0.04 ± 0.006	0.15 ± 0.004	0.02 ± 0.007	**0.007 ± 0.002**
	Size	56.64 ± 3.64	14.3 ± 4.71	153.06 ± 14.25	49.7 ± 7	29.8 ± 24.11	7494
	Time	4322 ± 548	37 ± 3	0.38 ± 0.04	288 ± 7	348 ± 137	0
isolet	Error	0.135 ± 0.018	0.063 ± 0.016	0.161 ± 0.018	NA	**0.050 ± 0.014**	0.113 ± 0.027
	Size	156.06± 16.92	25 ± 0	306.46 ±15.32	NA	26 ± 0	6238
	Time	163346 ± 41234	822 ± 111	42 ± 0.79	NA	30973±362	0

Ensemble Learning: Basic concept

- Learning is the process for obtaining the best hypothesis from hypothesis space.
- The obtained hypothesis can be good for training data, but may not be good for testing data. That is, it may not generalize well.
- On effective way for solving this problem is to use a set of "weak" hypotheses to form a "strong" one.
- The idea is similar to committee-based decision making.
  - Even if each committee member may not be expert for making a certain decision, the whole committee can make good decisions for various problems.
- This method is commonly called "ensemble". It is useful not only for decision trees.

Ensemble Learning: Basic concept

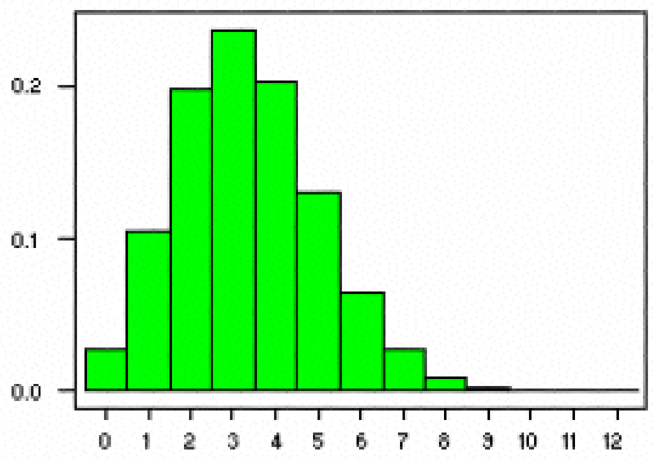
- The basic conditions for successful use of ensemble learning:
  - Each individual classifier should be good enough even if it is relatively weak (better than random guess).
  - The individual classifiers should be un-correlated. That is, the errors they produce are independent of each other.
- Under the above conditions, the error of the ensemble for newly observed data will be much smaller than that of each individual classifier.

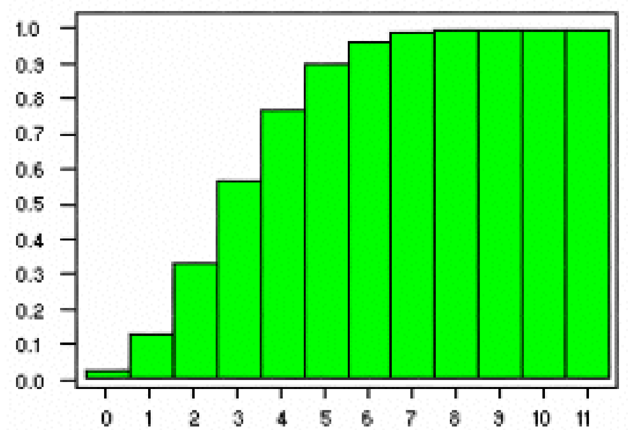
Ensemble Learning: Basic concept

- Binomial with n = 20 and p = 0.166667

x	P(X <= x)
0	0.0261
1	0.1304
2	0.3287
3	0.5665
4	0.7687
5	0.8982
6	0.9629
7	0.9887
8	0.9972
9	0.9994

- If there 20 "un-correlated" two-class classifiers, and each has an error rate p=0.1667.
- The error rate of the ensemble with majority voting is 1-0.9994=0.0006.





Ensemble Learning: Bagging

- Repeat for $t = 1,2,\ldots,T$
  - Make a data set Ω by copying randomly N data from the original training set Ω_0.
  - Obtain a weak classifier h_t.

> The data sets so obtained are different, and therefore, the classifiers can have different errors.

- Voting: For any given new datum x,

$$label(x) = 1 \text{ (or =-1) if } \sum_{i=1}^{T} h_t(x) > 0 \text{ (or } < 0)$$

- Bagging = **B**ootstrap **AGG**regat**ING**

Ensemble Learning: AdaBoost

See https://en.wikipedia.org/wiki/AdaBoost

- Repeat for t=1,2,…,T
  - Find a weak classifier h_t to minimize the weighted sum error

$$e_t = \sum_{\substack{i=1 \\ h_t(x_i) \neq y_i}}^{N} w_i^t$$

  - Update parameter

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right)$$

  - Update weights

$$w_i^{t+1} = w_i^t \exp\{-y_i \alpha_t h_t(x_i)\} \; for\ all\ i$$

- Weight is a "difficulty" measure of the each datum. Initially, all weights are 1/N. Should be normalized in each step.
- The parameter α_t is a "confidence" measure of the weak classifiers. Instead of equal voting, weighted voting is used for making a decision.

Ensemble Learning: Random forest

https://en.wikipedia.org/wiki/Random_forest

- Random forest is also ensemble learning.
- It is similar to Bagging, but, instead of using different data for obtaining the weak classifiers, *we select m features at random for node splitting* in the process of designing each individual DT.
- That is, for node splitting, we do not find the best test function based all features. We just find a relatively good test function based on part of the features.
- Here, m is much smaller than the total number of features.
- If N_f is the number of features, the recommended value for m is $\sqrt{N_f}$ for classification, or $N_f/3$ for regression.

Homework of Today

- Try to explain why ensemble is better than individual classifiers, using about 500 words.
- Try to provide theoretic support, as much as possible, for any conclusion you made.