Course ID 80250993 Dates: 9/16-12/23/2021 @ I-205
Tencent/VooV Meeting 521 4678 6257 (passwd 1205)

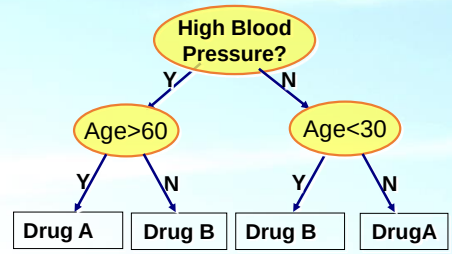
Chapter 9
Decision Trees, Random Forest and Ensemble Learning

Xuegong Zhang
October 28, 2021

9.1
Decision Tree

Decision trees in everyday life

- Many decisions are made stepwise in forms of trees
- e.g.
  - Credit risk assessment
  - Medical diagnosis
  - Market analysis
  - Production control
  - Policy making
  - BF/GF searching (?)
  - …



Slide from Vandi Verma, CMU

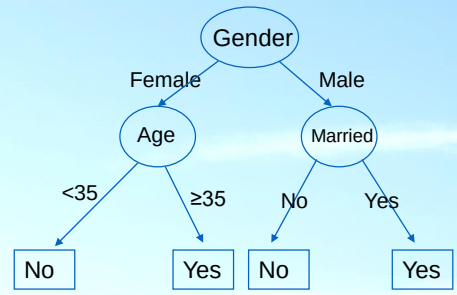
Examples of a Decision Tree

Data

編號	性別	年齡	婚姻	家庭人數	購買RV房車
A0001	Male	45	未婚	1	是
A0002	Male	52	已婚	7	是
A0003	Female	38	已婚	5	是
A0004	Male	25	已婚	5	否
A0005	Female	48	已婚	4	是
A0006	Male	32	未婚	3	是
A0007	Female	65	已婚	4	否
A0008	Male	33	已婚	3	是
A0009	Male	45	已婚	4	是
A0010	Female	52	未婚	1	是
A0011	Male	38	未婚	1	否
…	…	…	…	…	…
Z0099	Male	22	未婚	4	是

Class labels

Nominal Variables



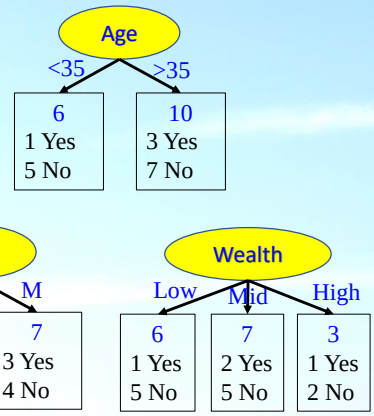
Can we make a machine to learn such trees?

Example from slides of 台湾致理技术学院 蘇志雄 教授

Building the tree

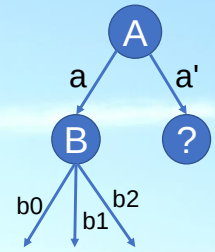
age	sex	wealth	buying
<35	M	Mid	No
>35	F	Mid	No
>35	F	Mid	No
>35	F	Low	No
<35	M	Hi	No
>35	F	Low	No
<35	F	Low	No
<35	F	Hi	Yes
>35	M	Mid	Yes
<35	M	Hi	No
>35	F	Mid	No
<35	M	Low	No
>35	F	Mid	No
>35	M	Low	Yes
>35	M	Mid	Yes
>35	F	Low	No

- Which one should be the root?
- Which should be the nodes in middle layers?



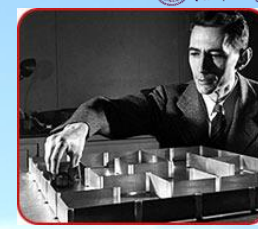
A basic algorithm to build DTs

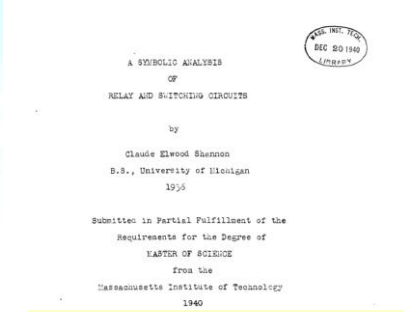
```
proc growtree (data)
        if (data not perfectly classified)
                find "best" splitting feature (attribute) A
                for each (a in A)
                        create child a
                        data_a=data restricted to A=a
                        growtree (data_a)
                endfor
        endif
endproc
```



Claude E. Shannon (1916-2001)

- University of Michigan (B.S.,1936)
- MIT (M.S.,1937; Ph.D.,1940)
- Bell Labs / MIT / Institute for Advanced Study



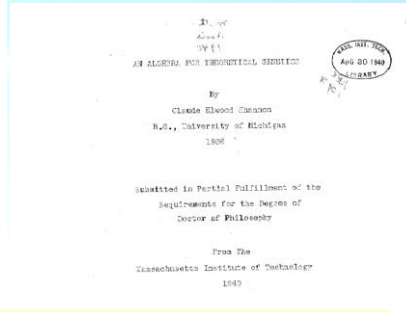


A SYMBOLIC ANALYSIS
OF
RELAY AND SWITCHING CIRCUITS

by

Claude Elwood Shannon
B.S., University of Michigan
1936

Submitted in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
from the
Massachusetts Institute of Technology
1940

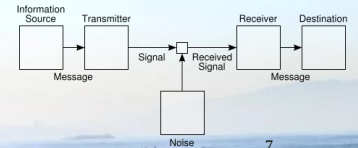


By
Claude Elwood Shannon
B.S., University of Michigan
1936

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
from the
Massachusetts Institute of Technology
1940

M.S. thesis: *A symbolic analysis of relay and switching circuits*

Ph.D. thesis: *An algebra for theoretical genetics*



ID3 (Iterative Dichotomiser 3) Algorithm

(Quinlan, 1979)

- Information Entropy (Shannon, 1949)

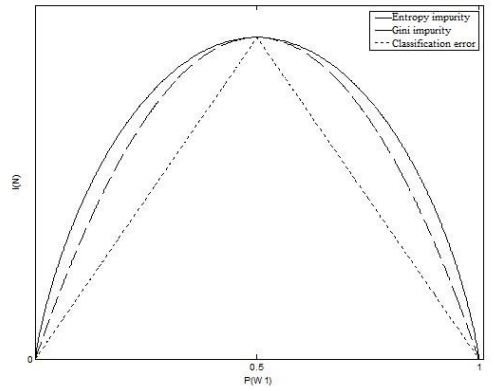
$$I = -(P_1 \log_2 P_1 + P_2 \log_2 P_2 + \cdots + P_k \log_2 P_k) = -\sum_{i=1}^{k} P_i \log_2 P_i$$

- Example 1: if $k = 4, P_1 = 0.25, P_2 = 0.25, P_3 = 0.25, P_4 = 0.25$
  $I = -(0.25 \times \log_2(0.25) \times 4) = 2$
- Example 2: if $k = 4, P_1 = 0, P_2 = 0.5, P_3 = 0, P_4 = 0.5$
  $I = -(0.5 \times \log_2(0.5) \times 2) = 1$
- Example 3: if $k = 4, P_1 = 1, P_2 = 0, P_3 = 0, P_4 = 0$
  $I = -(1 \times \log_2(1)) = 0$

- ID3: Use entropy as the measure of impurity of each feature (node)
  - Large entropy →high impurity → poor classification
  - Small entropy →low impurity → good classification

Other measures of impurity

- Entropy impurity
- Gini impurity
- Classification error



Q: For selecting the node of a tree, do we prefer features with large entropy or small entropy?

D

单选题 1分

设置

Shall we choose features with large entropy or small entropy?

A The larger the better

B The smaller the better

提交

Criterion for choosing node: *information gain*

- Upper node: Total samples n, samples in class 1: n_1, entropy $I(n, n_1)$
- Lower node: k values of feature X, m_i samples with value i, m_{i1} of class 1, entropy $E(X)$
- Information gain (*decrease of impurity*) if expanding the node by X

$$Gain\ (X)\ =\ I(n, n_1)\ -\ E(X),$$

where

$$I(n, n_1)\ =\ -\left(\left(\frac{n_1}{n}\right)\log_2\left(\frac{n_1}{n}\right)+\left(1-\frac{n_1}{n}\right)\log_2\left(1-\frac{n_1}{n}\right)\right)$$

$$E(X)=\left(\frac{m_1}{n}\right)I(m_1, m_{11})\ +\left(\frac{m_2}{n}\right)I(m_2, m_{21})\ +\cdots+\left(\frac{m_k}{n}\right)I(m_k, m_{k_1})$$

Equivalent forms:

$$\Delta I(N) = I(N) - \left(P_1 I(N_1) + P_2 I(N_2) + \cdots + P_k I(N_k)\right)$$

$$Gain(S, A) = Entropy(S) - \sum_{a\in A}\frac{|S_a|}{|S|}Entropy(S_a)$$

Example

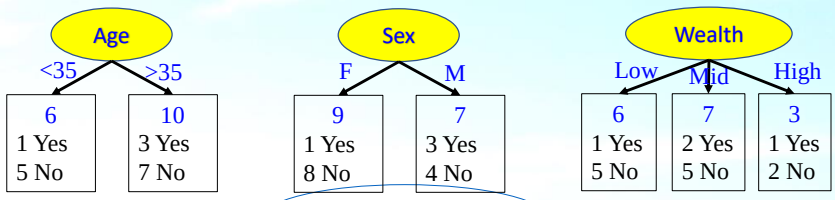
$n = 16$
$n_1 = 4$

age	sex	wealth	buying
<35	M	Mid	No
>35	F	Mid	No
>35	F	Mid	No
>35	F	Low	No
<35	M	Hi	No
>35	F	Low	No
<35	F	Low	No
<35	F	Hi	Yes
>35	M	Mid	Yes
<35	M	Hi	No
>35	F	Mid	No
<35	M	Low	No
>35	F	Mid	No
>35	M	Low	Yes
>35	M	Mid	Yes
>35	F	Low	No

$$I\ (16{,}4) = -\left(\left(\frac{4}{16}\right)\log_2\left(\frac{4}{16}\right)+\left(\frac{12}{16}\right)\log_2\left(\frac{12}{16}\right)\right) = 0.8113$$

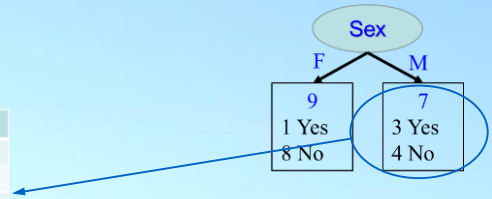
$$E\ (age) = \left(\frac{6}{16}\right)*I(6{,}1)+\left(\frac{10}{16}\right)*I(10{,}3) = 0.7946$$

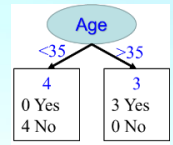
$$Gain(age) = I\ (16{,}4) - E\ (age)\ =\ 0.0167$$

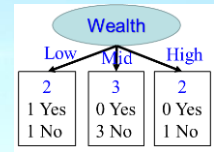


- Gain(age)=0.0167
- Gain(sex)=0.0972
- Gain(wealth)=0.0177
- Max: As the first node (root)

age	sex	wealth	buying
<35	M	Mid	No
<35	M	Hi	No
>35	M	Mid	Yes
<35	M	Hi	No
<35	M	Low	No
>35	M	Low	Yes
>35	M	Mid	Yes



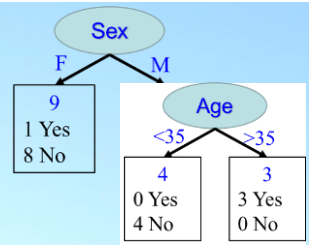




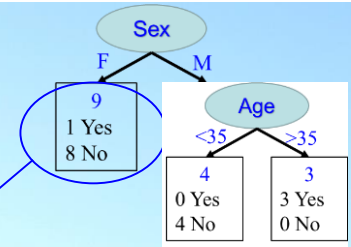
- Gain(age)=0.9852
- Gain(wealth)=0.688

$$I(7,3) = -((3/7) * \log_2(3/7) + (4/7) * \log_2(4/7)) = 0.9852$$

age	sex	wealth	buying
<35	M	Mid	No
<35	M	Hi	No
>35	M	Mid	Yes
<35	M	Hi	No
<35	M	Low	No
>35	M	Low	Yes
>35	M	Mid	Yes

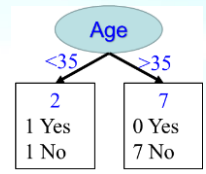


age	sex	wealth	buying
<35	M	Mid	No
<35	M	Hi	No
>35	M	Mid	Yes
<35	M	Hi	No
<35	M	Low	No
>35	M	Low	Yes
>35	M	Mid	Yes

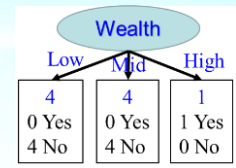


$$I(9,1) = -((1/9) * \log_2(1/9) + (8/9) * \log_2(8/9)) = 0.5032$$

age	sex	wealth	buying
>35	F	Mid	No
>35	F	Mid	No
>35	F	Low	No
>35	F	Low	No
<35	F	Low	No
<35	F	Hi	Yes
>35	F	Mid	No
>35	F	Mid	No
>35	F	Low	No

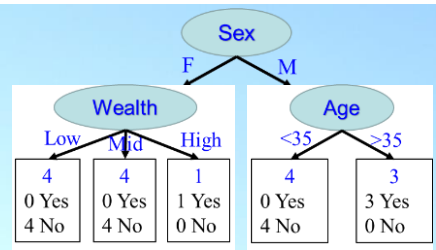


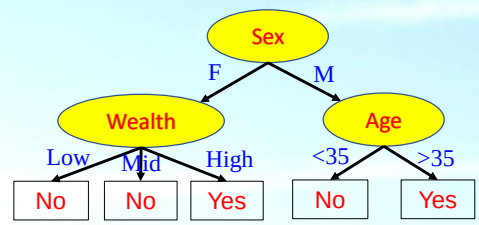
- Gain(age)=0.2222

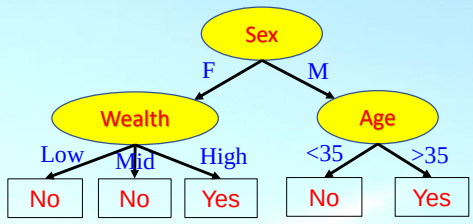


- Gain(wealth)=0.5032

age	sex	wealth	buying
<35	M	Mid	No
>35	F	Mid	No
>35	F	Mid	No
>35	F	Low	No
<35	M	Hi	No
>35	F	Low	No
<35	F	Low	No
<35	F	Hi	Yes
>35	M	Mid	Yes
<35	M	Hi	No
>35	F	Mid	No
<35	M	Low	No
>35	F	Mid	No
>35	M	Low	Yes
>35	M	Mid	Yes
>35	F	Low	No







- A set of inference rules:
  - If male and if not too young, Yes
  - If female and if not wealthy, No
  - …

3-second break

9.2
Overfitting and Tree Pruning

Training Examples *PlayTennis*

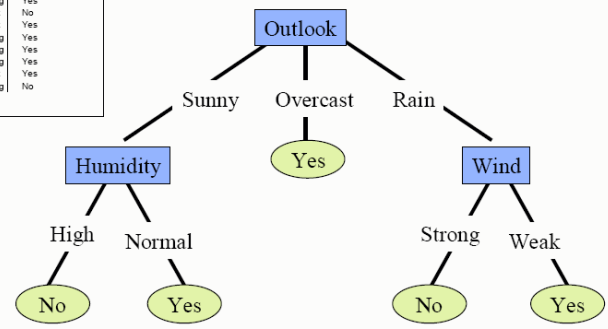
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
S2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Slide from Vandi Verma, CMU

Decision Tree for *PlayTennis*

Training Examples *PlayTennis*

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
S2	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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

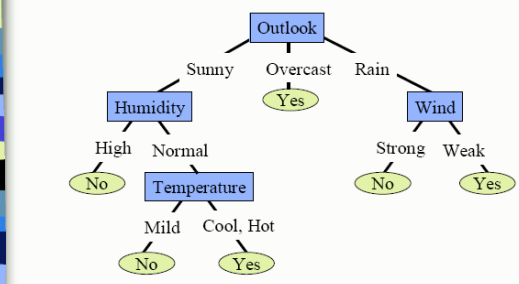


Slide from Vandi Verma, CMU

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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No
D15	Sunny	Mild	Normal	Strong	No

The risk of overfitting if we let the tree grow too deep.

Effect on Our Tree



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D15	Sunny	Mild	Normal	Strong	No

Slide from Vandi Verma, CMU

Overfitting vs. Size of the Tree

Overfitting

- Hypothesis h overfits iff ∃ h' with

$error_{train}(h) < error_{train}(h')$
$error_{true}(h) > error_{true}(h')$

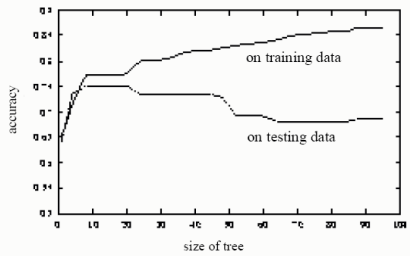
Bias Variance Trade-Off

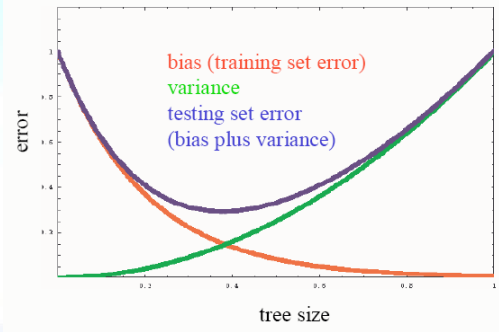
High Bias ⟺ **High Variance**

Small trees can't fit many data sets

Large trees sensitive to randomness in data selection

Overfitting in ID3





Xueqing Zhang

Slide from Vandi Verma, CMU

23

Ockham's Razor (Occam's Razor)

- William of Ockham (1285-1347):
  - "Non sunt multiplicanda entia praeter necessitatem."
  - Entities are not to be multiplied beyond necessity.

    ---- "law of parsimony"
- In machine learning (and science):
  - To prefer simpler hypotheses over more complex ones, when the sample size is limited.
- Albert Einstein (1879-1955):
  - "Everything should be made as simple as possible, but not simpler."

Xueqing Zhang

24

How to avoid overfitting in DT?

- Pruning
  - Stop growing in time
    - According to some threshold on value or significance of the information gain
  - Grow full tree, then prune
    - Pruning based on validation
    - MDL principle (Minimum Description Length)

→ Introducing democracy in decision making

9.3
Random Forest

- Making decision with many decision trees

➔ Random Forest

(Leo Breiman, *Machine Learning*, 45: 5-32, 2001)
(http://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm)

Leo Breiman (1928-2005)

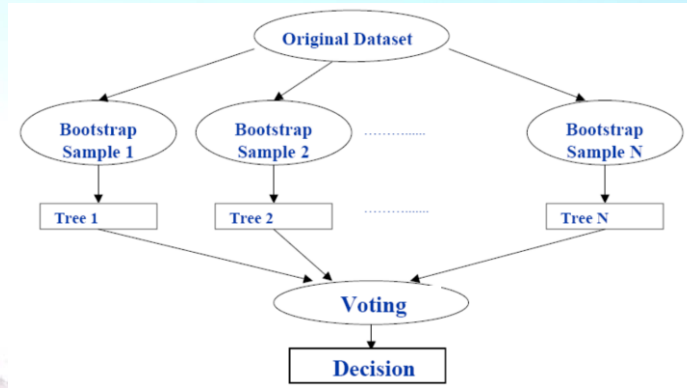
- The idea of democracy
  - Democracy (Greek: δημοκρατία dēmokratía, literally "rule by people")
  - To reduce/avoid the risk of a single decision maker
  - But unlikely to work if “people” are statistically not independent, or if sampling is biased, …
  - How can we generate multiple “democratic trees”?

- Bootstrapping
  - Getting (oneself or something) into or out of a situation using existing resources.
  - "To pull oneself up by one's bootstraps." (1834)
  - Random sampling with replacement
  - e.g., Resampling a given sample to estimate the mean height and its variance across samples in the population

Baron Munchausen pulls himself and his horse out of a swamp by his pigtail.

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

- Random Forest Method
  - Generate multiple trees by ***bootstrapping*** *the samples*
    - choosing *N* times with replacement from all *N* available training cases
  - For each node of the tree, ***randomly*** *choose m variables* on which to base the decision at that node.
    - Calculate *the best split* based on these *m* variables in the training set.
  - The RF classifier outputs the class that is *the* ***mode*** *of the classes* output by individual trees.



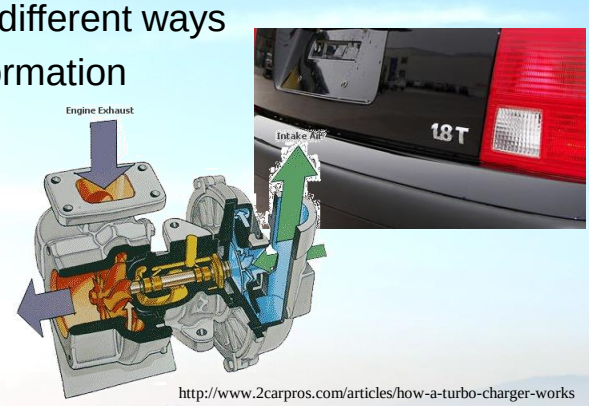
5-second break

9.4
Ensemble Learning

How to make a classification system better?

- By extracting better features
- By using better classification algorithms
- By transforming samples to a new space
- By using features & classifiers in different ways
- By using prior and contextual information
- By using more data
- ...
- **by boosting existing methods**

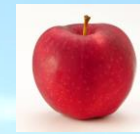
To help or encourage (something) to increase or improve.



http://www.2carpros.com/articles/how-a-turbo-charger-works

Basic Considerations

- Building a strong classifier (with high accuracy and generalization) is hard
- But building some weak classifier can be easier
  - Just need to be better than chance
  - E.g. call anything like a big red blob in an image as an “apple”
    or call any round object in an image as an “apple”
- Can we combine several weak classifiers to build a better one?
- Three humble shoemakers brainstorming will make a great statesman.
  “三个臭皮匠顶个诸葛亮”



=

How to combine classifiers?

- A similar idea: combine classifiers $h_1(\mathbf{x}), \ldots, h_m(\mathbf{x})$

$$H(\mathbf{x}) = \alpha_1 h_1(\mathbf{x}) + \ldots + \alpha_m h_m(\mathbf{x}),$$

- α_j is the *vote* assigned to classifier h_j.
  - Votes should be higher for more reliable classifiers.
- Prediction:

$$\hat{y}(\mathbf{x}) = \operatorname{sign} H(\mathbf{x}).$$

- Classifiers h_j can be simple (e.g., based on a single feature).

Many slides from Chen Yu of Indiana Univ, and Olga Veksler.

Ada Boost (Freund & Schapire, 1997)

- Basic Ideas
  - Treat every sample equally at the beginning and build a simple weak classifier
    - Maintain a distribution of weights over training samples according to whether they are easy to be classified
  - Misclassified examples will get larger weights in the next round
  - Adjust weights when a classifier is built, remember how well the classifier was, and start another round to build another weak classifier
    - Samples with larger weights will be paid more attention to
  - After multiple rounds, stop learning and collect all weak classifiers to vote

Ada Boost for 2 Classes

Initialization step: for each example x, set

$$D(x) = \frac{1}{N}, \text{ where } N \text{ is the number of examples}$$

Iteration step (for $t = 1 \ldots T$):

1. Find best weak classifier $h_t(x)$ using weights $D_t(x)$
2. Compute the error rate ε_t as $\varepsilon_t = \sum_{i=1}^{N} D(x_i) \cdot I[y_i \neq h_t(x_i)]$, where $I[y_i \neq h_t(x_i)] = \begin{cases} 1 & \text{if } y_i \neq h_t(x_i) \\ 0 & \text{otherwise} \end{cases}$
3. assign weight α_t to classifier h_t in the final hypothesis
$$\alpha_t = \log((1 - \varepsilon_t)/\varepsilon_t)$$
4. For each x_i, $D(x_i) = D(x_i) \cdot \exp(\alpha_t \cdot I[y_i \neq h_t(x_i)])$
5. Normalize $D(x_i)$ so that $\sum_{i=1}^{N} D(x_i) = 1$

$$f_{final}(x) = \text{sign}\left[\sum \alpha_t h_t(x)\right]$$

Many slides from Chen Yu of Indiana Univ, and Olga Veksler.

Ada Boost: step by step

1. Find best weak classifier $h_t(x)$ using weights $D(x)$

- Some classifiers accept weighted samples, but most don't
- If the classifier does not take weighted samples, this step is done by sampling from the training samples according to the distribution $D(x)$

1/16 1/4 1/16 1/16 1/4 1/16 1/4

- Draw k samples, each x with probability equal to $D(x)$:

Many slides from Chen Yu of Indiana Univ, and Olga Veksler.

Ada Boost: step by step

1. Find best weak classifier $h_t(x)$ using weights $D(x)$

- Give to the classifier the following re-sampled examples:

- To find the best weak classifier, go through ALL weak classifiers, and find the one that works best (gives smallest error) on the collection above

	$h_1(x)$	$h_2(x)$	$h_3(x)$	.....	$h_m(x)$
errors:	0.46	0.36	0.16		0.43

the best classifier $h_t(x)$ at iteration t

Ada Boost: step by step

2. Compute ε_t the error rate as

$$\varepsilon_t = \sum D(x_i) \cdot I[y_i \neq h_t(x_i)]$$

- where $I[y_i \neq h_t(x_i)] = \begin{cases} 1 & \text{if } y_i \neq h_t(x_i) \\ 0 & \text{otherwise} \end{cases}$

1/16	1/4	1/16	1/16	1/4	1/16	1/4
✓	✓	✓	✗	✗	✓	✓

$$\varepsilon_t = \frac{1}{4} + \frac{1}{16} = \frac{5}{16}$$

- ε_t is simply the weight of all misclassified examples added
  - notice that error rate is computed over original examples, not the re-sampled examples
- If a weak classifier is better than random, then $\varepsilon_t < ½$

Ada Boost: step by step

3. assign weight α_t to classifier h_t in the final hypothesis

$$\alpha_t = \log\left((1-\varepsilon_t)/\varepsilon_t\right)$$

Example from previous slide:

$$\varepsilon_t = \frac{5}{16} \Rightarrow \alpha_t = \log \frac{1-\frac{5}{16}}{\frac{5}{16}} = \log \frac{11}{5} \approx 0.8$$

- Recall that $\varepsilon_t < ½$
- Thus $(1-\varepsilon_t)/\varepsilon_t > 1 \Rightarrow \alpha_t > 0$
- The smaller is ε_t, the larger is α_t, and thus the more importance (weight) classifier $h_t(x)$ gets in the final classifier

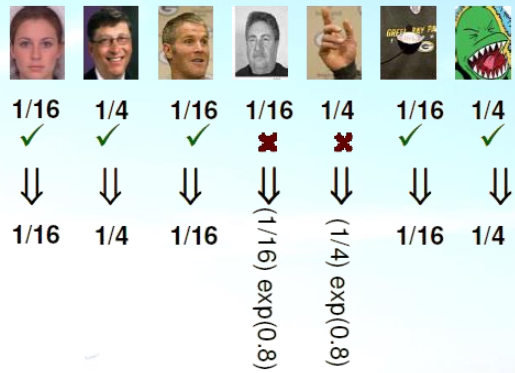
$$f_{final}(x) = \text{sign}\left[\sum \alpha_t h_t(x)\right]$$

Many slides from Chen Yu of Indiana Univ, and Olga Veksler.

Ada Boost: step by step

4. For each $\boldsymbol{x_i}$, $\boldsymbol{D(x_i)} = \boldsymbol{D(x_i)} \cdot \exp[\boldsymbol{\alpha_t} \cdot \boldsymbol{I}(\boldsymbol{y_i} \neq \boldsymbol{h_t(x_i)})]$

Example from previous slide: $\alpha_t = 0.8$



- Weight of misclassified examples is increased and the new $\boldsymbol{D(x_i)}$'s are normalized to be a distribution again

Many slides from Chen Yu of Indiana Univ, and Olga Veksler.

Ada Boost: step by step

5. Normalize $D(x_i)$ so that $\sum D(x_i) = 1$

Example from previous slide:

1/16 1/4 1/16 0.14 0.56 1/16 1/4

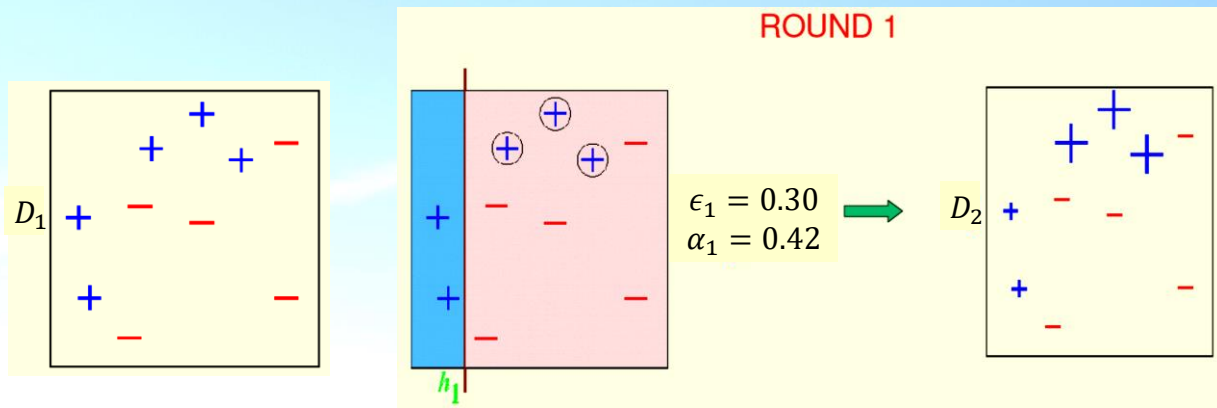
After normalization:

0.05 0.18 0.05 0.10 0.40 0.05 0.18

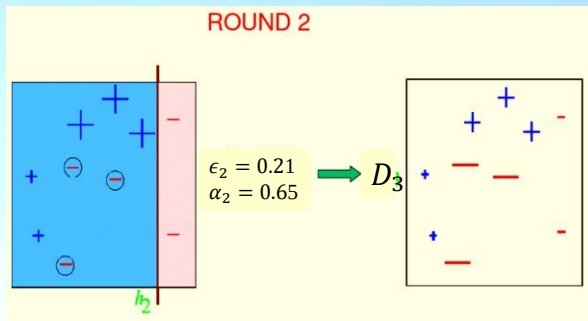
- In matlab, if D is a vector storing weights, D = D./sum(D)

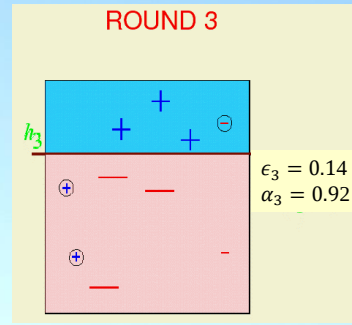
Many slides from Chen Yu of Indiana Univ, and Olga Veksler.

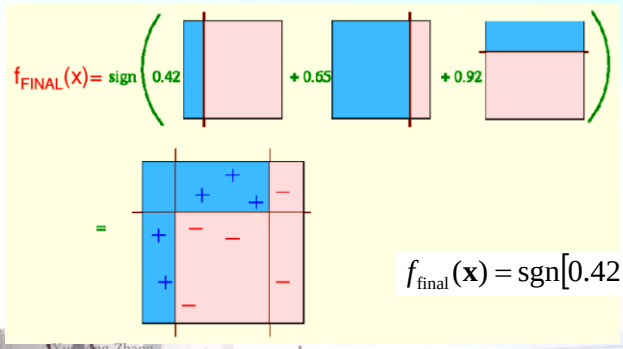
AdaBoost Example



Many slides from Chen Yu of Indiana Univ, and Olga Veksler.







$$f_{\text{final}}(\mathbf{x}) = \text{sgn}[0.42\,\text{sgn}(x_1 - 3) + 0.65\,\text{sgn}(x_1 - 7) + 0.92\,\text{sgn}(x_2 - 4)]$$

Pros & Cons

- Good generalization properties
  - Boosting increases the margins
- Fast, simple
- Only one parameter T
- Flexible: can be combined with many classifier
- Only requirement: weak learner, better than chance
- AdaBoost may not work if a weak learner is too complex (overfitting)
  - E.g., boosting an SVM or ANN may not work
- Sensitive to noises (e.g., wrong labels)
- May over-emphasize noisy samples

Homework

- Problems (Pr5)
  1. Bagging
  2. Decision Tree
- Deadline:
  - Nov. 3 (Wednesday), 23:00

- Computer exercises (Ex5b)
  - Random forest, to evaluate the contribution of features
- Deadline:
  - Nov. 10 (Wednesday), 23:00

About "balanced learning" of ML

- A key characteristic of this ML course: Balanced
  - Balanced between different schools of ML methods and theories
    - Traditional: regression, linear discriminant, k-NN, decision trees, …
    - ANN: perceptrons, MLP, deep learning, …
    - Statistical learning theory and SVMs
    - Probabilistic learning, Bayesian networks, Graphic models, …
    - …
  - Balanced between broadness and depth
  - Balanced between mathematical reasoning, mathematical skills and computer skills
  - Balanced between being a hard-core course and a popular course
- There is no single perfect way of learning ML. It depends on yourselves.
- Do not worry about what I (the exam) expect on you. Ask yourselves what you want to grasp on ML, based on your own past and future.


See you next week
for
Mid-term test & Probabilistic Learning
Xuegong Zhang
49