Introduction to Machine Learning

Decision Trees

Inas A. Yassine
Systems and Biomedical Engineering Department,
Faculty of Engineering - Cairo University
iyassine@eng.cu.edu.eg

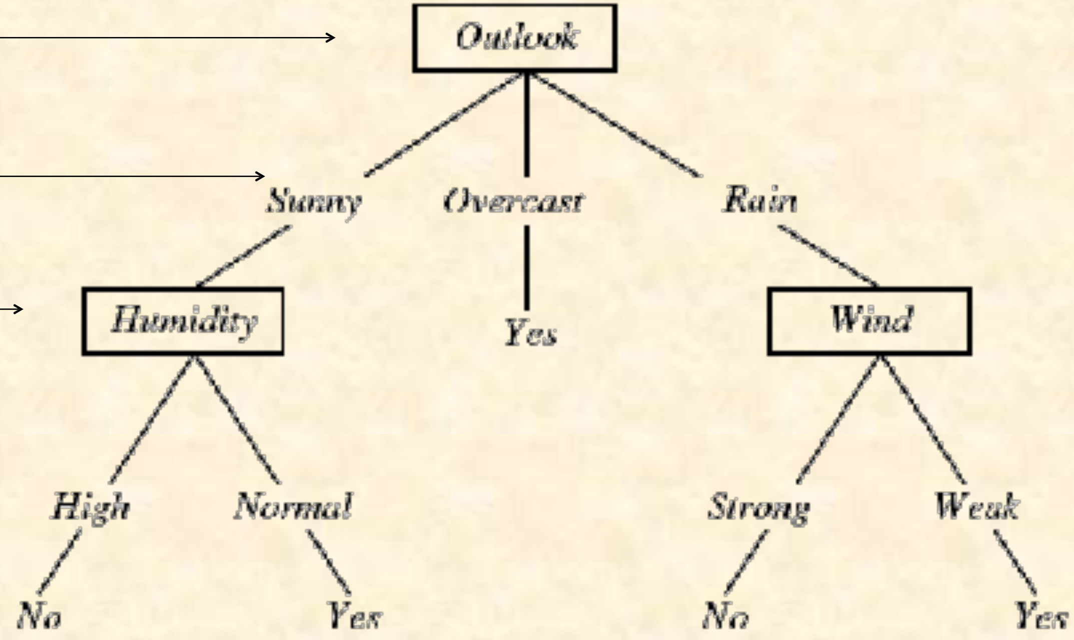
Decision Tree Representation

Root : Tree's first Node

Branch: Outcome of a test

Node: Decision Variable

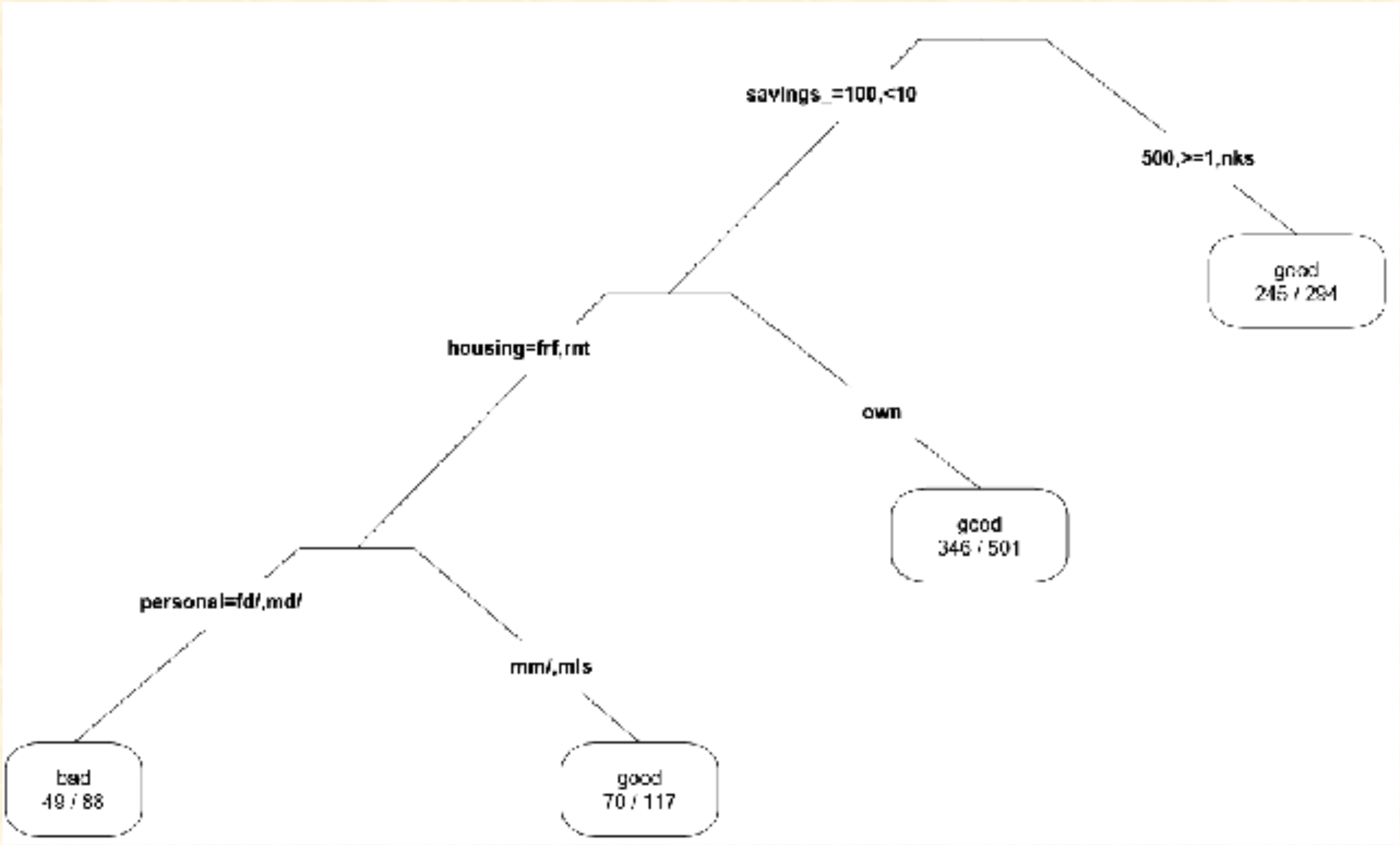
Leaf: Class Label



Decision Tree Classifier - Use Cases

- When a series of categorical questions are answered to arrive at a classification
  - Biological species classification
  - Checklist of symptoms during a doctor’s evaluation of a patient
- When “if-then” conditions are preferred to linear models.
  - Customer segmentation to predict response rates
  - Financial decisions such as loan approval
  - Fraud detection
- Short Decision Trees are the most popular "weak learner" in ensemble learning techniques

Example: The Credit Prediction Problem



Learning Data …

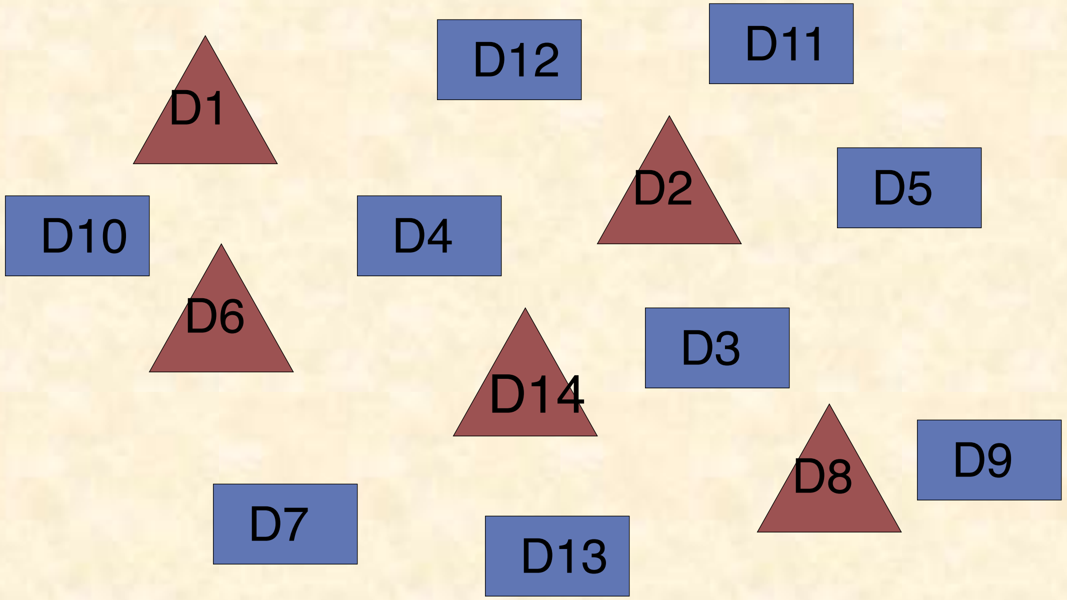
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Strong	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

- **Outlook: Sunny, Overcast, Rain**
- **Temperature: Hot, Mild Cool**
- **Humidity: High, Normal**
- **Wind: Weak, Strong**
- **Play Tennis: Yes, No**

Data to be Classified

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Overcast	Mild	High	Weak	?
2	Rain	Cool	Normal	Strong	?
3	Sunny	Hot	High	Strong	?

ID3: The Basic Decision Tree Learning Algorithm



What is the “best” attribute?

[“best” = with highest information gain]

Deciding whether a pattern is interesting

- Information Theory
  - A very large topic, originally used for compressing signals
  - But more recently used for data mining…

Information Gain

- The information gain of a feature *F* is the expected reduction in entropy resulting from splitting on this feature.
- $$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v) \quad , \qquad Entropy(decision) = P_+ \log_2 P_+ + P_- \log_2 P_-$$

  where S_v is the subset of S having value v for feature F.
- Entropy of each resulting subset weighted by its relative size.

<big, red, circle>: +

<small, red, circle>: +

<small, red, square>: −

<big, blue, circle>: −

2+, 2 −: E=1
size
big: 1+,1− E=1
small: 1+,1− E=1
Gain=1−(0.5·1 + 0.5·1) = 0

2+, 2 − : E=1
color
red: 2+,1− E=0.92
blue: 0+,1− E=0
Gain=1−(0.75·0.918 + 0.25·0) = 0.311

2+, 2 − : E=1
shape
circle: 2+,1− E=0.92
square: 0+,1− E=0
Gain=1−(0.75·0.918 + 0.25·0) = 0.311

Information Gain Calculation Example

- Entropy for a dataset
  - E(S)= $-(9/14)log(9/14) - (5/14)log(5/14) = 0.94$
- Outlook
  - Sunny[2+,3-], Overcast [4+,0-], Rain [3+, 2-]
  - $E[S_S]= -(2/5)log(2/5) - (3/5)log(3/5) = 0.4416$
  - $E[S_O]= -(4/4)log(4/4) - (0/4)log(0/4) = 0$
  - $E[S_R]=-(3/5)log(3/5) - (2/5)log(2/5) = 0.4416$
  - IF= *0.94- [(5/14)E[S_s]+ (4/14)E[S_N]+ (5/14)E[S_R]]=0.29*
- Temperature
  - Hot [2+,2-], Cool [4+,0-], Mild [4+, 2-]
  - $E[S_H]= -(2/4)log(2/4) - (2/4)log(2/4) = 1$
  - $E[S_C]= -(4/4)log(4/4) - (0/4)log(0/4) = 0$
  - $E[S_M]=-(4/6)log(4/6) - (2/6)log(2/6) = 0.92$
  - IF= 0.94- [(4/14)E[S_H]+ (4/14)E[S_C]+ 6/14 E[S_M]]= 0.26

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Strong	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

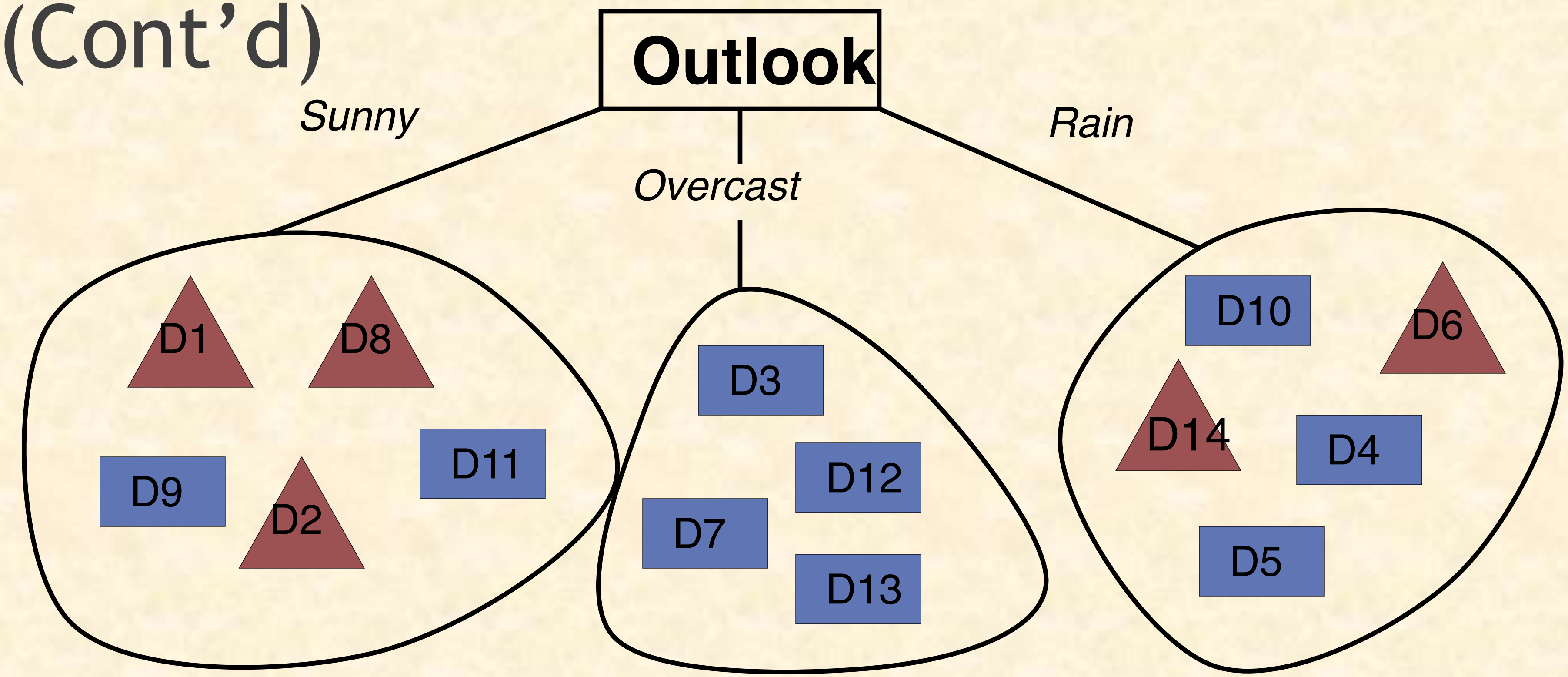
Information Gain Calculation Example

- Entropy for a dataset
  - $E(S)= \frac{-9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14} = 0.94$
- Humidity
  - High[3+,4-], Normal [6+,1-]
  - $E[S_H] = -(3/7)log(3/7) - (4/7log(4/7) = 0.984$
  - $E[S_N] = -(6/7)log(6/7) - (1/7)log(1/7) = 0.59$
  - IG= 0.94- [(7/14) $E[S_H]$+ (7/14)$E[S_N]$]=0.115
- Wind
  - Strong [6+,2-], Weak [3+,3-]
  - $E[S_s] = -(3/6)log(3/6) - (3/6)log(3/6) = 1$
  - $E[S_w] = -(6/8)log(6/8) - (2/8)log(2/8) = 0.8075$
  - IG= 0.94- [(8/14)$E[S_s]$ + (6/14)$E[S_w]$]=0.0225

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Strong	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

IG(S, Out)> IG(S, Temp)> IG(S, Hum)> IG(S, Wind) , **Outlook** is chosen as the root

ID3 (Cont’d)



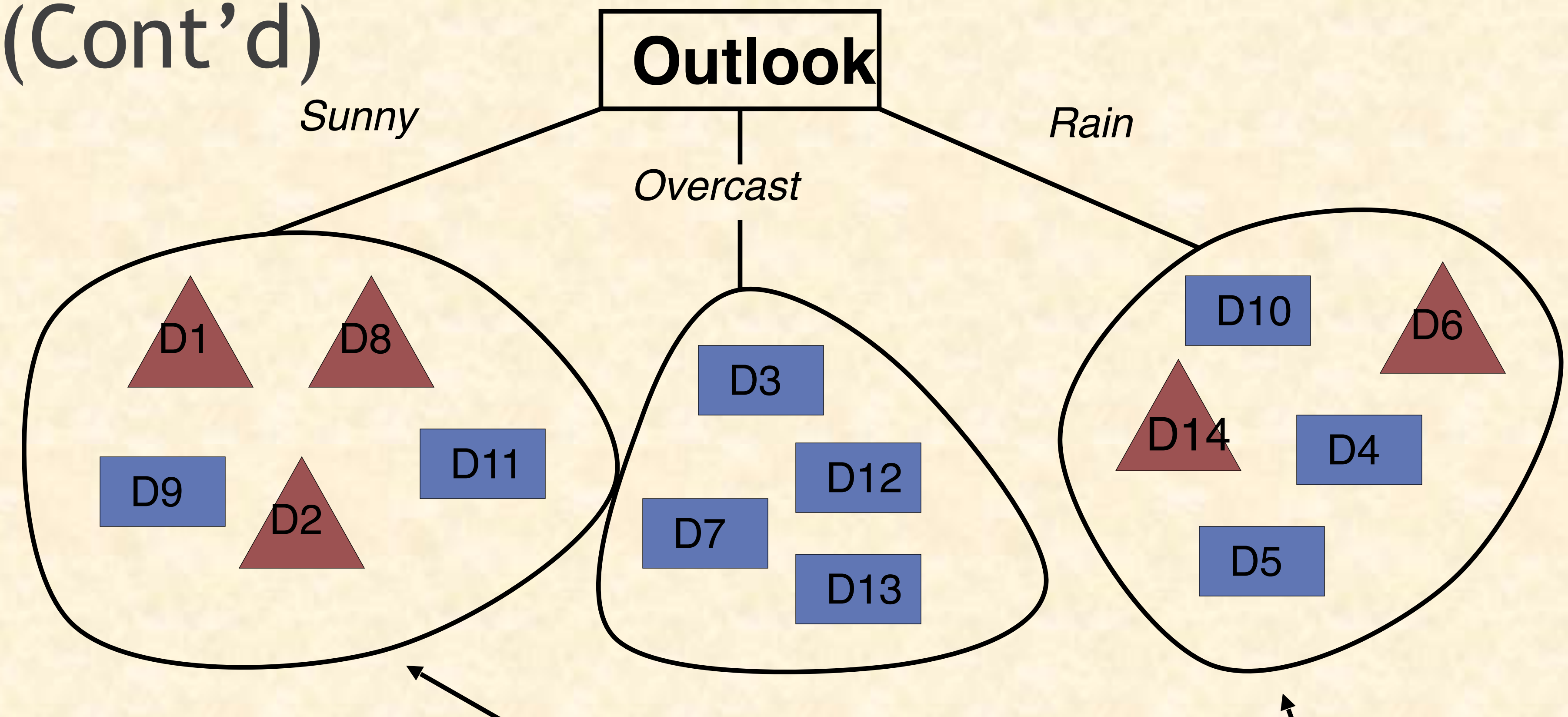
Outlook
Sunny
Overcast
Rain
D1
D8
D9
D2
D11
D3
D12
D7
D13
D10
D6
D14
D4
D5

Information Gain Calculation Example

- Entropy for outlook-(Sunny
  - E(S)= $-(3/5)log(3/5) - (2/5)log(2/5) = 0.4416$
- Temperature
  - Hot [0+,2-], Cool [1+,0-], Mild [1+, 1-]
  - $E[S_H]= -(2/2)log(2/2) - (0/2)log(0/2) = 0$
  - $E[S_C]= -(1/1)log(1/1) - (0/1)log(0/1) = 0$
  - $E[S_M]=-(1/2)log(1/2) - (1/2)log(1/2) = 1$
  - IF= 0.441- [(2/5)$E[S_H]$+ (1/5)$E[S_C]$+ 2/5 $E[S_M]$]= 0.041
- Humidity
  - High[0+,3-], Normal [2+,0-]
  - $E[S_H]= -(0/3)log(0/3) - (3/3log(3/3) = 0$
  - $E[S_N]= -(2/2)log(2/2) - (0/2)log(0/2) = 0$
  - IG= 0.441- [(3/5) $E[S_H]$+ (2/5)$E[S_N]$]=0.441
- Wind
  - Strong [1+,1-], Weak [1+,2-]
  - $E[S_W]=-(1/2)log(1/2) - (1/2)log(1/2) = 1$
  - $E[S_s]=-(6/8)log(6/8) - (2/8)log(2/8) = 0.9128$
  - IG= 0.441- [(2/5)$E[S_s]$ + (3/5)$E[S_w]$]=0.0225

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

ID3 (Cont'd)



Yes

What are the "best" attributes? **Humidity** and **Wind**

General Algorithm

- To construct tree T from training set S
  - If all examples in S belong to some class in C, or S is sufficiently "pure", then make a leaf labeled C.
  - Otherwise:
    - select the “most informative” attribute A
    - partition S according to A’s values
    - recursively construct sub-trees T1, T2, ..., for the subsets of S
- The details vary according to the specific algorithm – CART, ID3, C4.5 – but the general idea is the same

Decision Tree Classifier - Reasons to Choose (+) & Cautions (-)

Reasons to Choose (+)	Cautions (-)
Takes any input type (numeric, categorical) In principle, can handle categorical variables with many distinct values (ZIP code)	Decision surfaces can only be axis-aligned
Robust with redundant variables, correlated variables	Tree structure is sensitive to small changes in the training data
Naturally handles variable interaction	A "deep" tree is probably over-fit Because each split reduces the training data for subsequent splits
Handles variables that have non-linear effect on outcome	Not good for outcomes that are dependent on many variables Related to over-fit problem, above
Computationally efficient to build	Doesn't naturally handle missing values; However most implementations include a method for dealing with this
Easy to score data	In practice, decision rules can be fairly complex
Many algorithms can return a measure of variable importance	
In principle, decision rules are easy to understand	

Ensemble Learning

Random Forest

Motivation

- So far – learning methods that learn a single hypothesis, chosen form a hypothesis space that is used to make predictions.
- No Lunch Free Theorem: There is no algorithm that is always the most accurate
- Generate a group of base-learners which when combined have higher accuracy
- Build many models and combine them
- Ensemble model improves accuracy and robustness over single model methods
- Efficiency: a complex problem can be decomposed into multiple sub-problems that are easier to understand and solve (divide-and-conquer approach)
- Applications:
    - distributed computing
    - privacy-preserving applications
    - large-scale data with reusable models
    - multiple sources of data

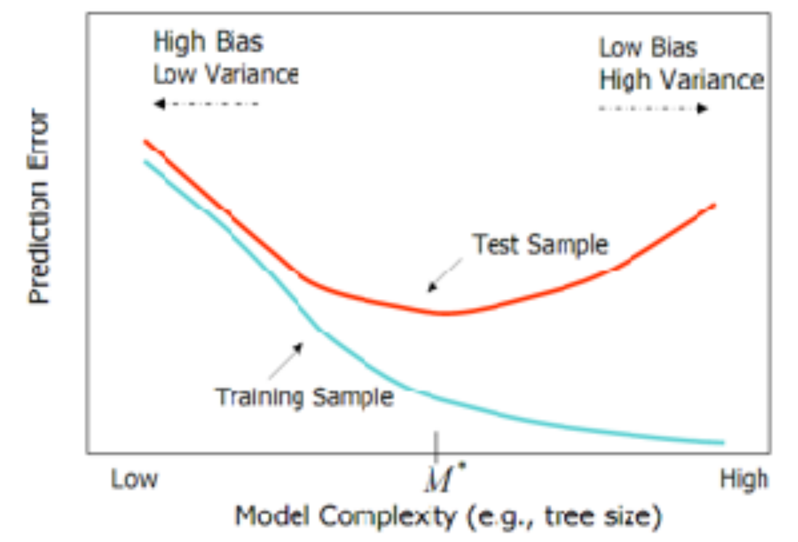
Strong versus Weak learner

- Strong Learner →Objective of machine learning
  - Take labeled data for training
  - Produce a classifier which can be *arbitrarily accurate*

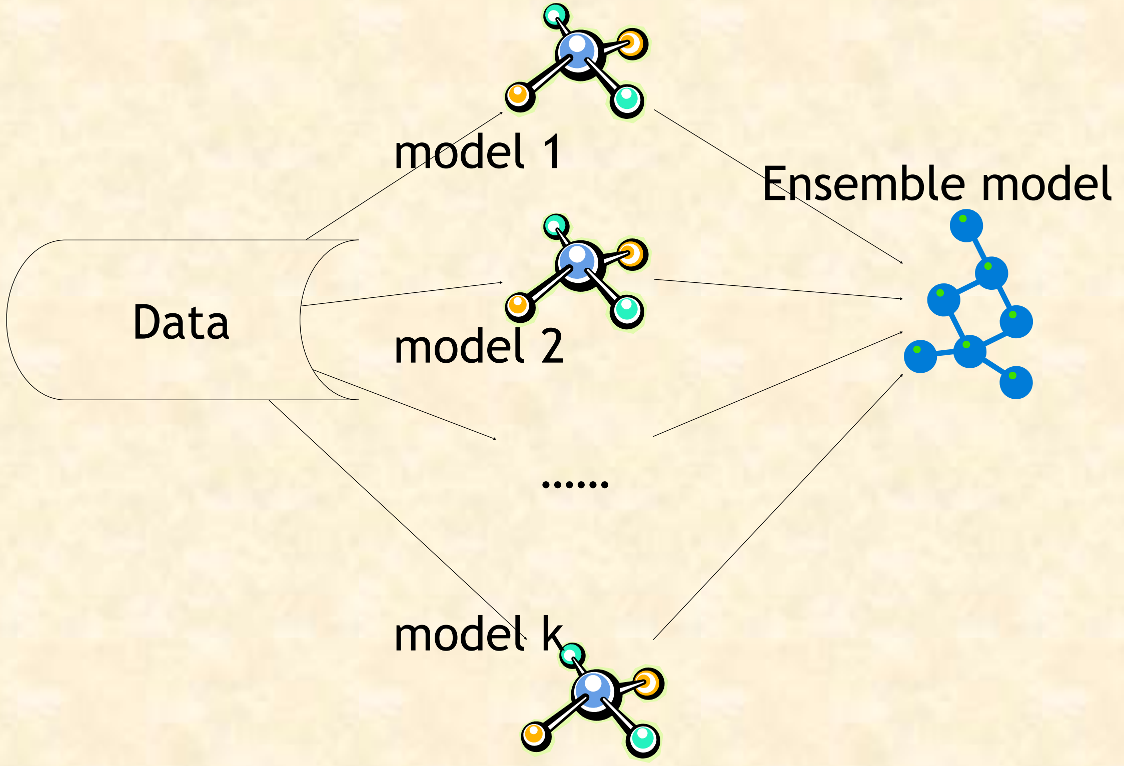
- Weak Learner
  - Take labeled data for training
  - Generate a hypothesis with a training accuracy greater than 0.5, i.e., < 50% error over any distribution ; more accurate than random guessing
- Strong learners are very difficult to construct
- Constructing weaker Learners is relatively easy

Bias versus Variance

- Bias is the persistent/systematic error of a learner independent of the training set.
  - Zero for a learner that always makes the optimal prediction
- Variance is the error incurred by fluctuations in response to different training sets.
  - Independent of the true value of the predicted variable and zero for a learner that always predicts the same class regardless of the training set

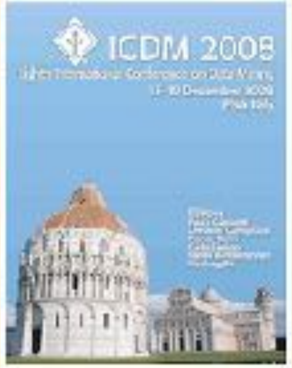


Ensemble Learning Block Diagram



Stories of Success

- Million-dollar prize
  - Improve the baseline movie recommendation approach of Netflix by 10% in accuracy
  - The top submissions all combine several teams and algorithms as an ensemble



- Data mining competitions
  – Classification problems
  – Winning teams employ an ensemble of classifiers

Cairo University

Netflix Prize

- Supervised learning task
  - Training dat ... ers have given to movies.
  - Construct a ... ectly classifies that movie as eit...
  - $1 million p ... e recommender
- Competition
  - At first, sing ... e improved
  - However, im...
  - Later, indivi... nprovements are observed

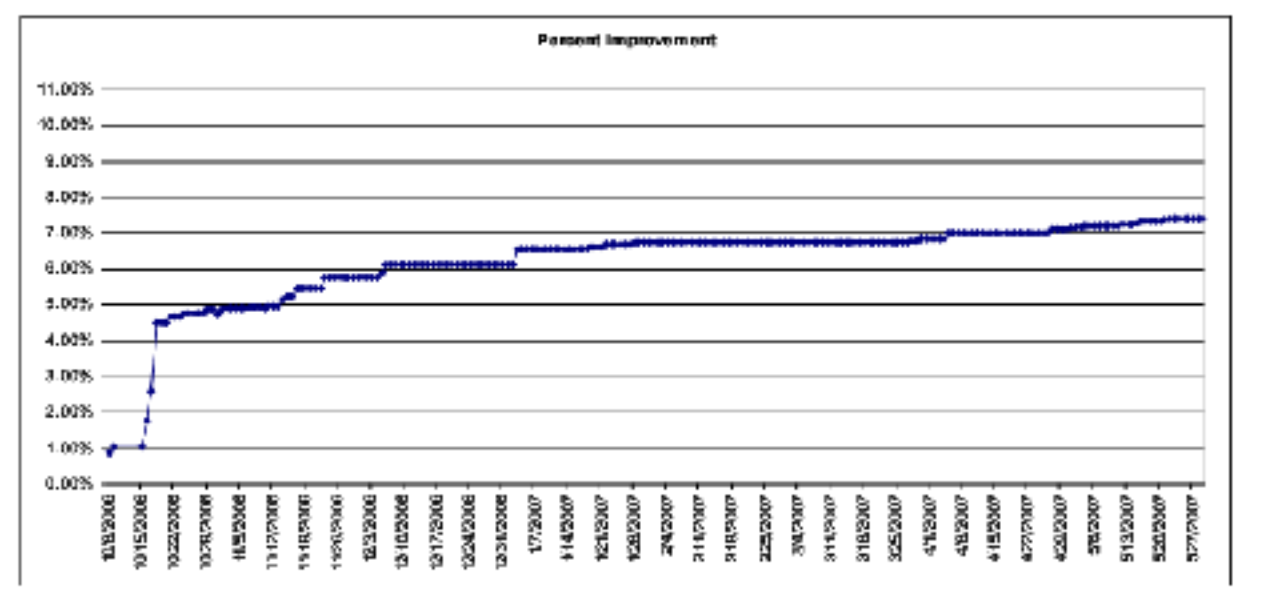


Figure 3: Aggregate improvement over Cinematch by time

Leaderb

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
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
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22
14	Gravity	0.8643	9.26	2009-04-22 18:31:32
15	Ces	0.8651	9.18	2009-06-21 19:24:53
Progress Prize 2007 - RMSE = 0.8723 - Winning Team: KorBell				
Cinematch score - RMSE = 0.9525				

“Our final solution (RMSE=0.8712) consists of blending 107 individual results. “

“Predictive accuracy is substantially improved when blending multiple predictors. Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a single technique. “

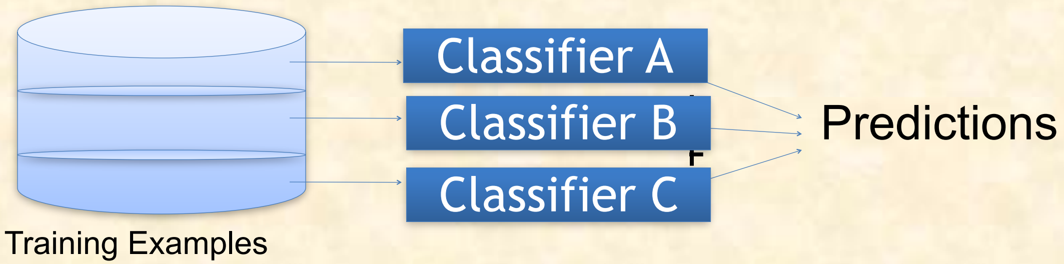
Different learners

- Subsampling training examples
- Manipulating input features
- Manipulating different Learning Algorithms
- Manipulating Different Algorithms parameters
- Injecting randomness

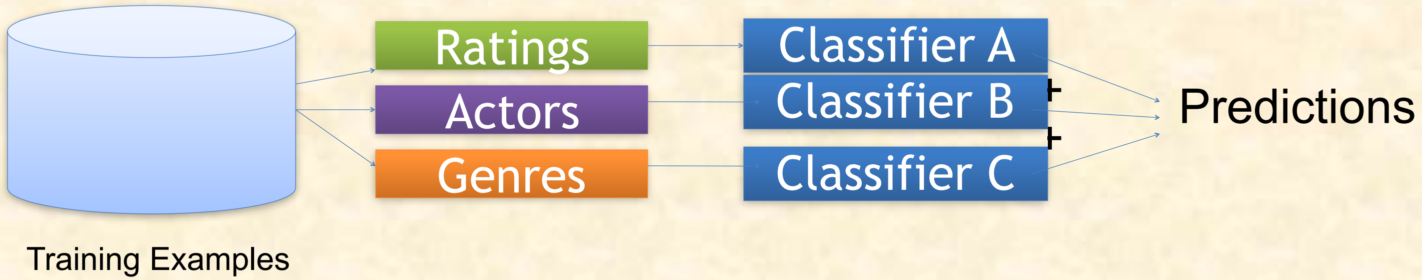
Achieving Diversity

Diversity from differences in inputs

1. Divide up training data among models



2. Different feature weightings



Ensemble Mechanisms - Combiners

- Voting
- Averaging (if predictions not 0,1)
- Weighted Averaging
  - base weights on confidence in component
- Learning combiner
  - Bagging
  - Boosting(Adaboost, Region Boost)
    - piecewise combiner
  - Gating, Stacking
    - general combiner

Random Forest

- Ensemble method specifically designed for decision tree classifiers
- Random Forests grows many trees
  - Ensemble of unpruned decision trees
  - Each base classifier classifies a "new" vector of attributes from the original data
  - Final result on classifying a new instance: voting. Forest chooses the classification result having the most votes (over all the trees in the forest)

Random Forests

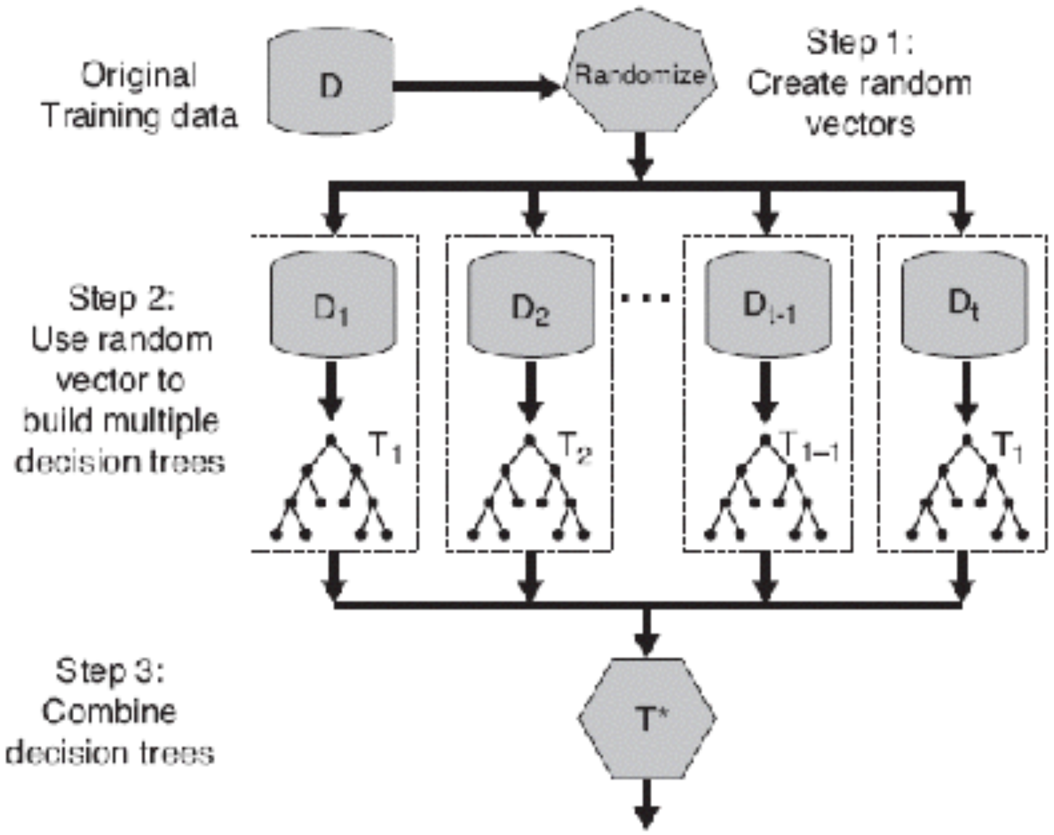


Figure 5.40. Random forests.

Bagging - Aggregate Bootstrapping

- Given a standard training set D of size n

- For i = 1 .. M
  - Draw a sample of size $n^{*}<n$ from D uniformly and with replacement
  - Learn classifier C_i
- Final classifier is a vote of $C_1 .. C_M$
- Increases classifier stability/reduces variance

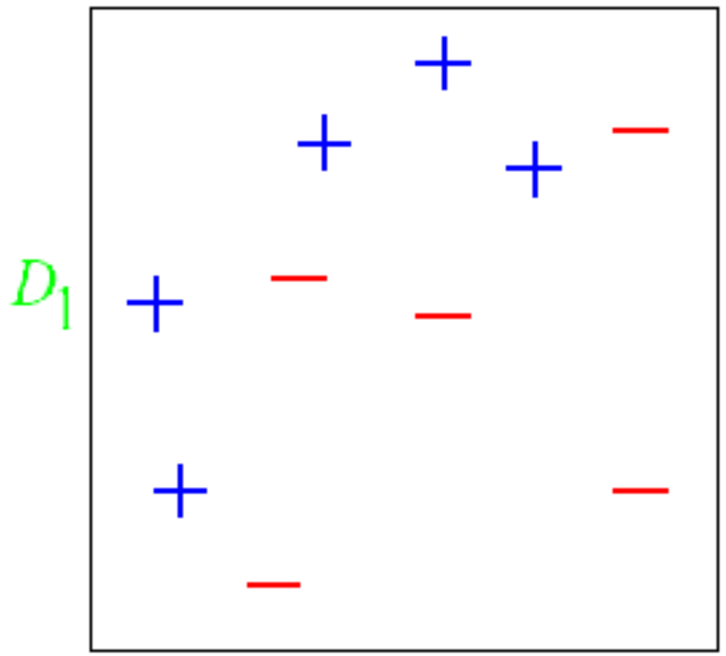
Boosting

- Developed to guarantee performance improvements on fitting training data for a ***weak learner*** (Schapire, 1990).
- Instead of sampling (as in bagging) re-weigh examples!
- Revised to be a practical algorithm, AdaBoost, for building ensembles that empirically improves generalization performance (Freund & Shapire, 1996).
- Examples are given weights. At each iteration, a new hypothesis is learned (weak learner) and the examples are reweighted to focus the system on examples that the most recently learned classifier got wrong.
- Final classification based on weighted vote of weak classifiers

Random Tree

- Pick N features at Random and build your tree
- Bootstrapping:
  - Draw points at random
  - Build the tree
  - Use the rest for testing
  - Decide how confident you are in this decision
- Paralelism:
  - Load data on different machines

AdaBoost Example*:

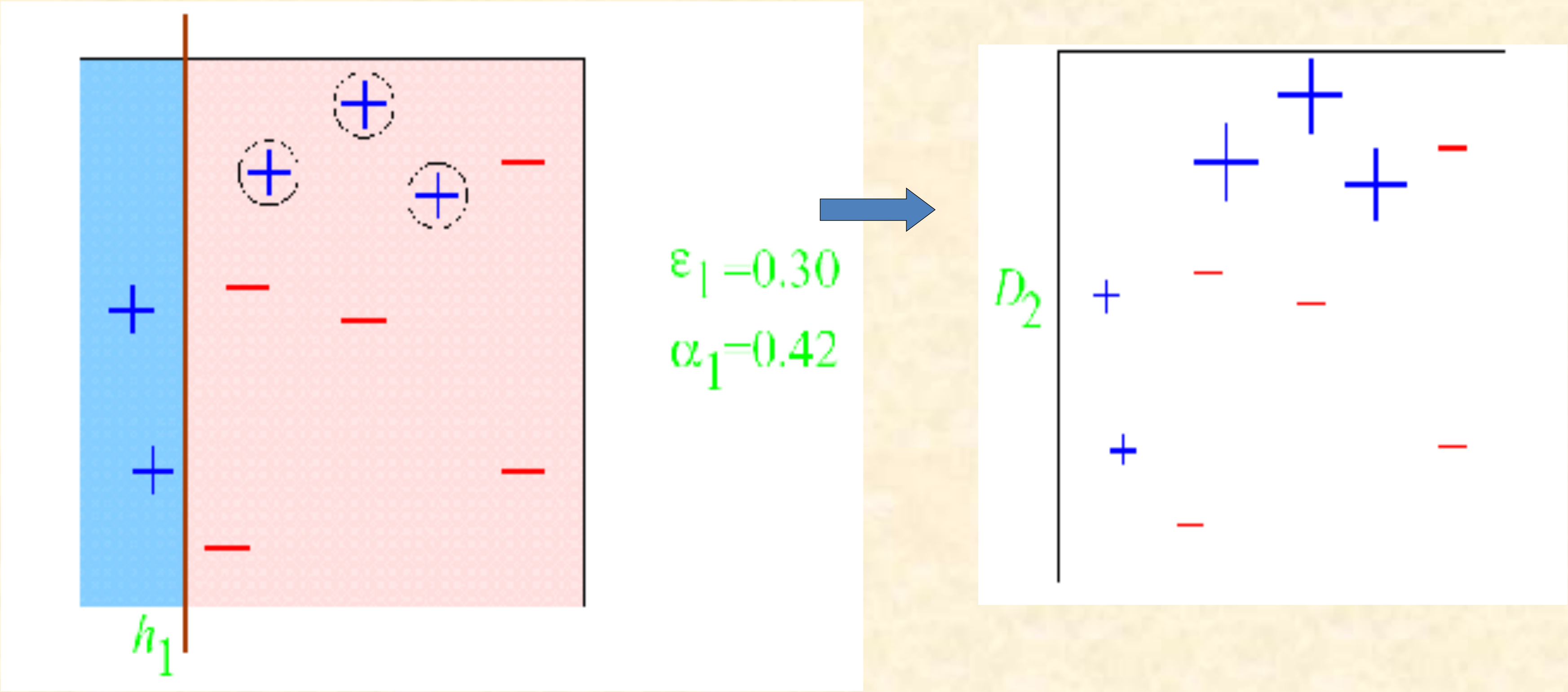


Original training set: equal weights to all training samples

AdaBoost Example:

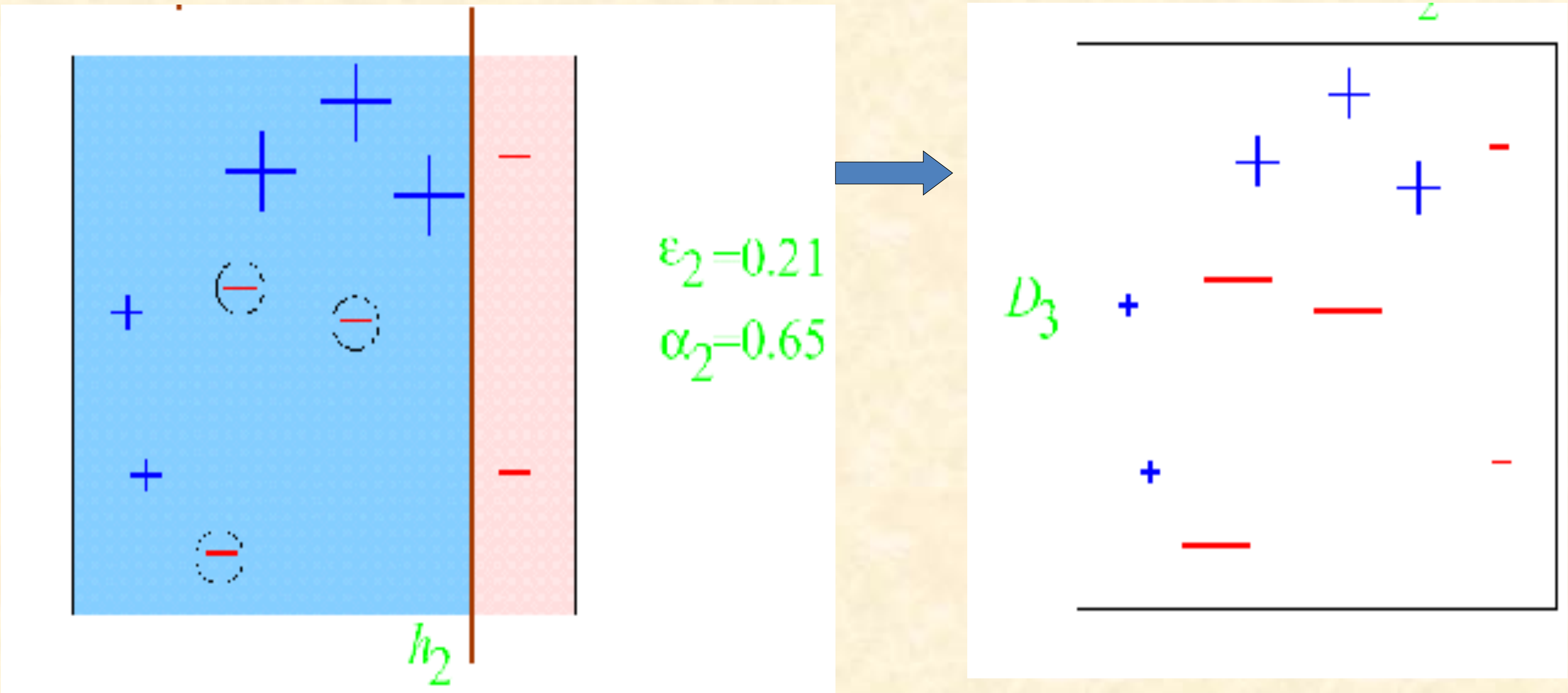
ε = error rate of classifier
α = weight of classifier

ROUND 1



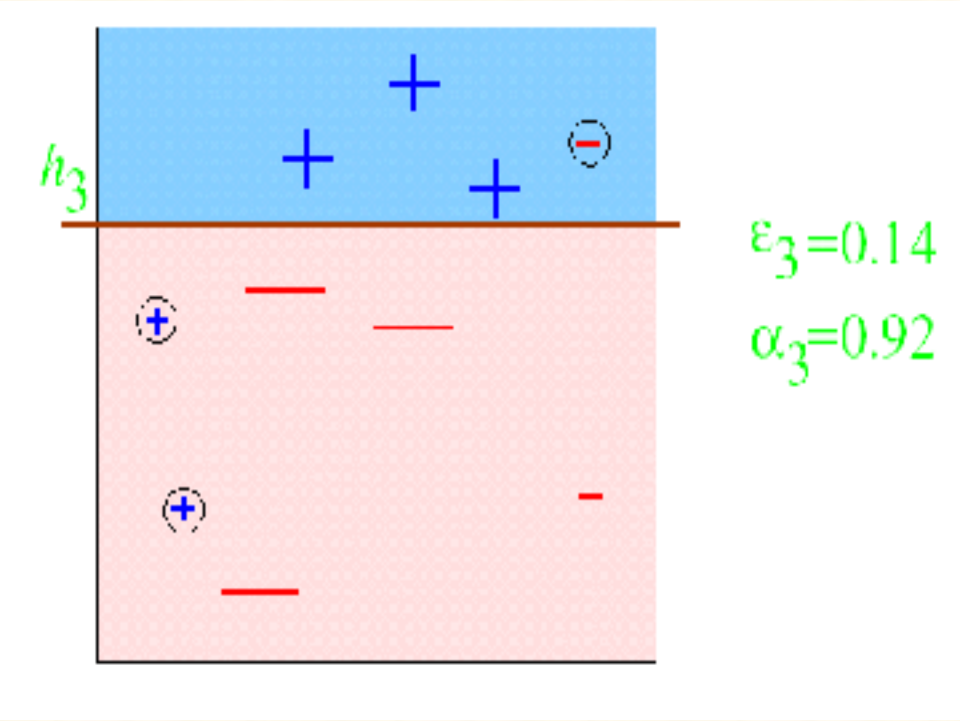
AdaBoost Example:

ROUND 2



AdaBoost Example:

ROUND 3



Random Forest

- Introduce two sources of randomness: “Bagging” and “Random input vectors”
  - Bagging method: each tree is grown using a bootstrap sample of training data
  - Random vector method: At each node, best split is chosen from a random sample of *m* attributes instead of all attributes

Issues in Ensembles

- Parallelism in Ensembles: Bagging is easily parallelized, Boosting is not.
- Variants of Boosting to handle noisy data.
- How “weak” should a base-learner for Boosting be?
- What is the theoretical explanation of boosting’s ability to improve generalization?
- Exactly how does the diversity of ensembles affect their generalization performance.
- Combining Boosting and Bagging.

Thank You …