For written notes on this lecture, please read chapter 3 of *The Practical Bioinformatician. Alternatively, please read* "Rule-Based Data Mining Methods for Classification Problems in Biomedical Domains", a tutorial at *PKDD04* by Jinyan Li and Limsoon Wong, September 2004. http://www.comp.nus.edu.sg/~wongls/talks/pkdd04/

Bioinformatics and Biomarker Discovery *Part 2: Tools*

Limsoon Wong

5 September 2012

NUS National University of Singapore

Outline

- **Overview of Supervised Learning**
- **Decision Trees Ensembles**
  – Bagging

Overview of Supervised Learning

National University of Singapore

Computational Supervised Learning

- **Also called classification**

- **Learn from past experience, and use the learned knowledge to classify new data**

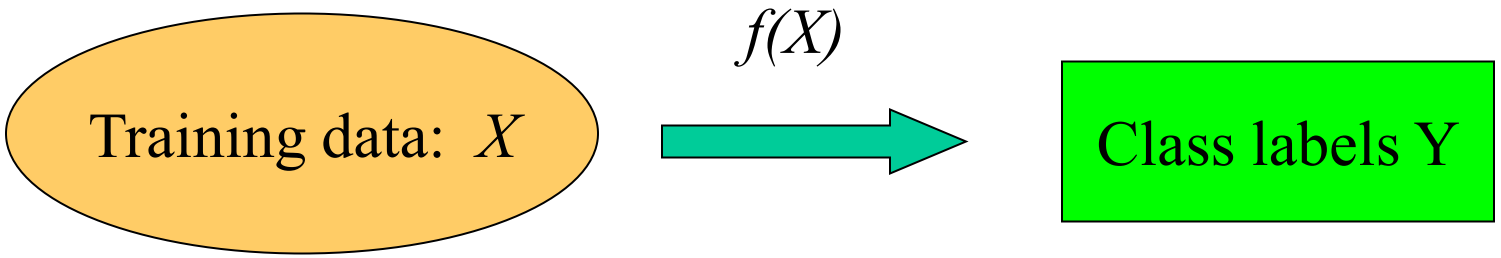
- **Knowledge learned by intelligent algorithms**

- **Examples:**
  - Clinical diagnosis for patients
  - Cell type classification

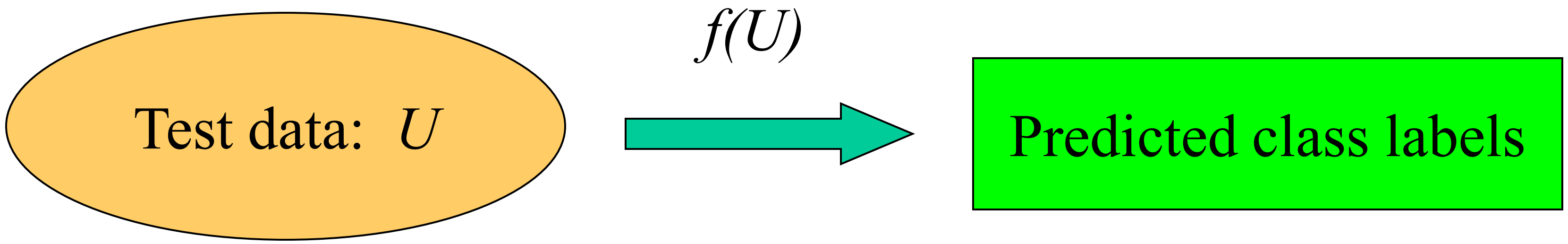
Data

- **Classification application involves > 1 class of data. E.g.,**
  - Normal vs disease cells for a diagnosis problem
- **Training data is a set of instances (samples, points) with known class labels**
- **Test data is a set of instances whose class labels are to be predicted**

Process



f(•): A classifier, a mapping, a hypothesis



Relational Representation of Patient Data

n features (order of 1000)

$$\begin{array}{c|cccccc|c}
 & \text{gene}_1 & \text{gene}_2 & \text{gene}_3 & \text{gene}_4 & \ldots & \text{gene}_n & \text{class} \\ \hline
 & x_{11} & x_{12} & x_{13} & x_{14} & \ldots & x_{1n} & \to P \\
 & x_{21} & x_{22} & x_{23} & x_{24} & \ldots & x_{2n} & \to N \\
m \text{ samples} & x_{31} & x_{32} & x_{33} & x_{34} & \ldots & x_{3n} & \to P \\
 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \\
 & x_{m1} & x_{m2} & x_{m3} & x_{m4} & \ldots & x_{mn} & \to N
\end{array}$$

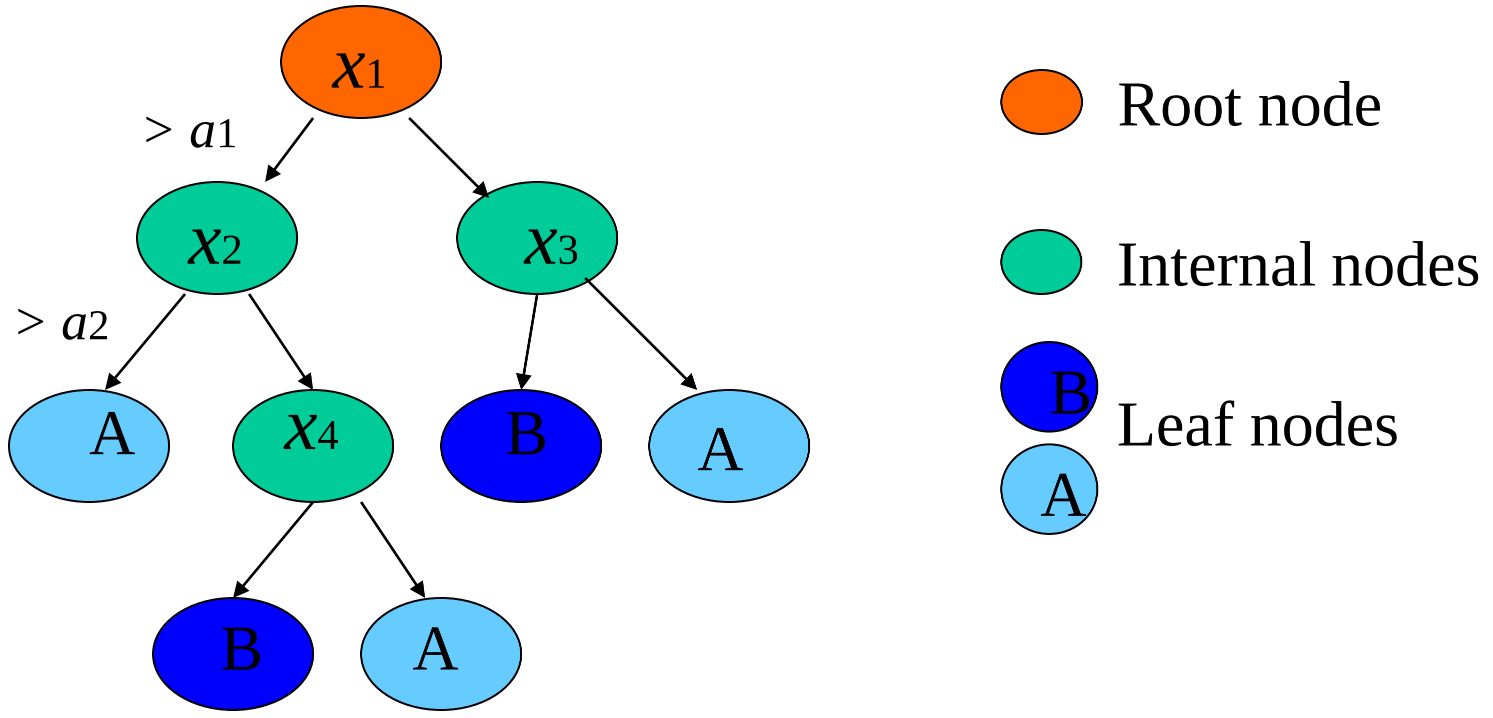
Importance of Rule-Based Methods

- **Systematic selection of a small number of features used for the decision making**

$\Rightarrow$ **Increase the comprehensibility of the knowledge patterns**

- **C4.5 and CART are two commonly used rule induction algorithms---a.k.a. decision tree induction algorithms**

Structure of Decision Trees



- **Every path from root to a leaf forms a decision rule**
  - If $x_1 > a_1$ & $x_2 > a_2$, then it's A class
- **C4.5, CART, two of the most widely used**
- **Easy interpretation, but accuracy generally unattractive**

A Simple Dataset

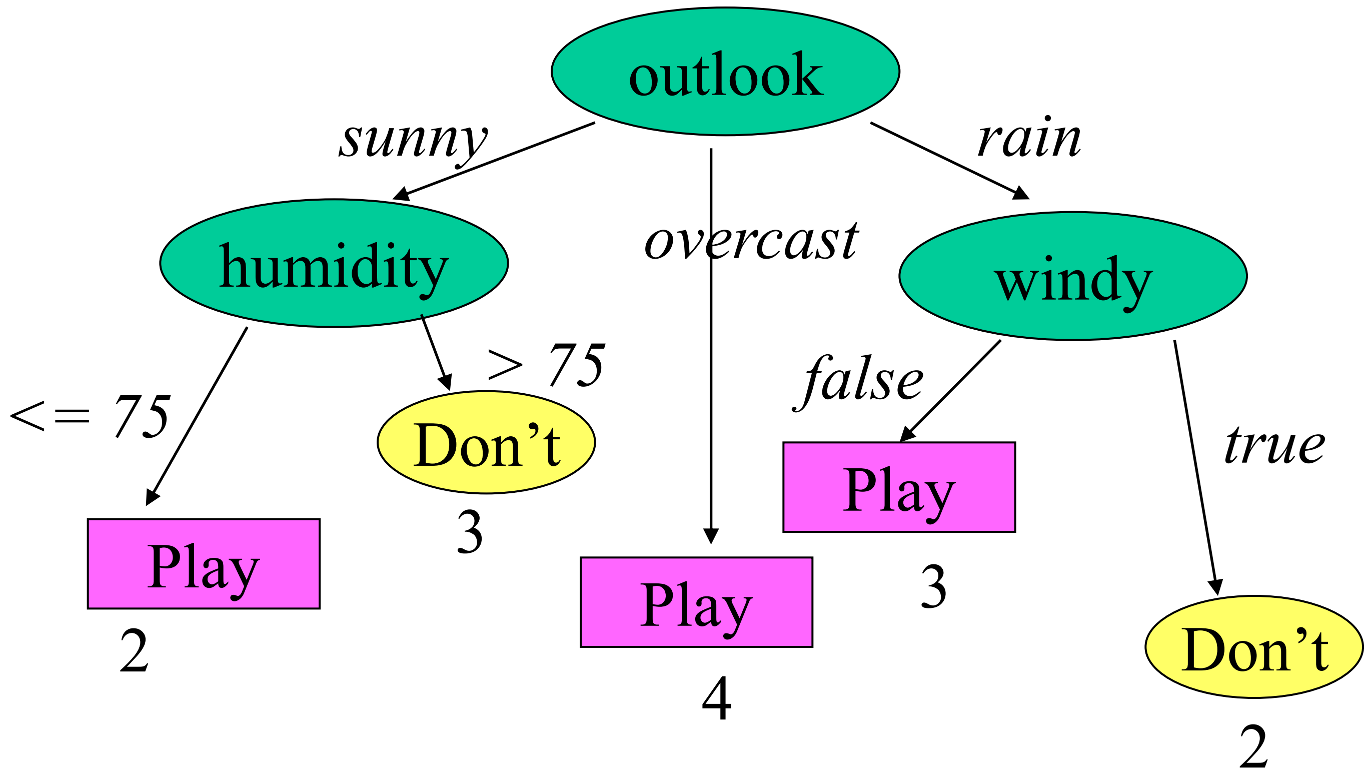
Outlook	Temp	Humidity	Windy	class
Sunny	75	70	true	Play
Sunny	80	90	true	Don’t
Sunny	85	85	false	Don’t
Sunny	72	95	true	Don’t
Sunny	69	70	false	Play
Overcast	72	90	true	Play
Overcast	83	78	false	Play
Overcast	64	65	true	Play
Overcast	81	75	false	Play
Rain	71	80	true	Don’t
Rain	65	70	true	Don’t
Rain	75	80	false	Play
Rain	68	80	false	Play
Rain	70	96	false	Play

9 Play samples

5 Don’t

A total of 14.

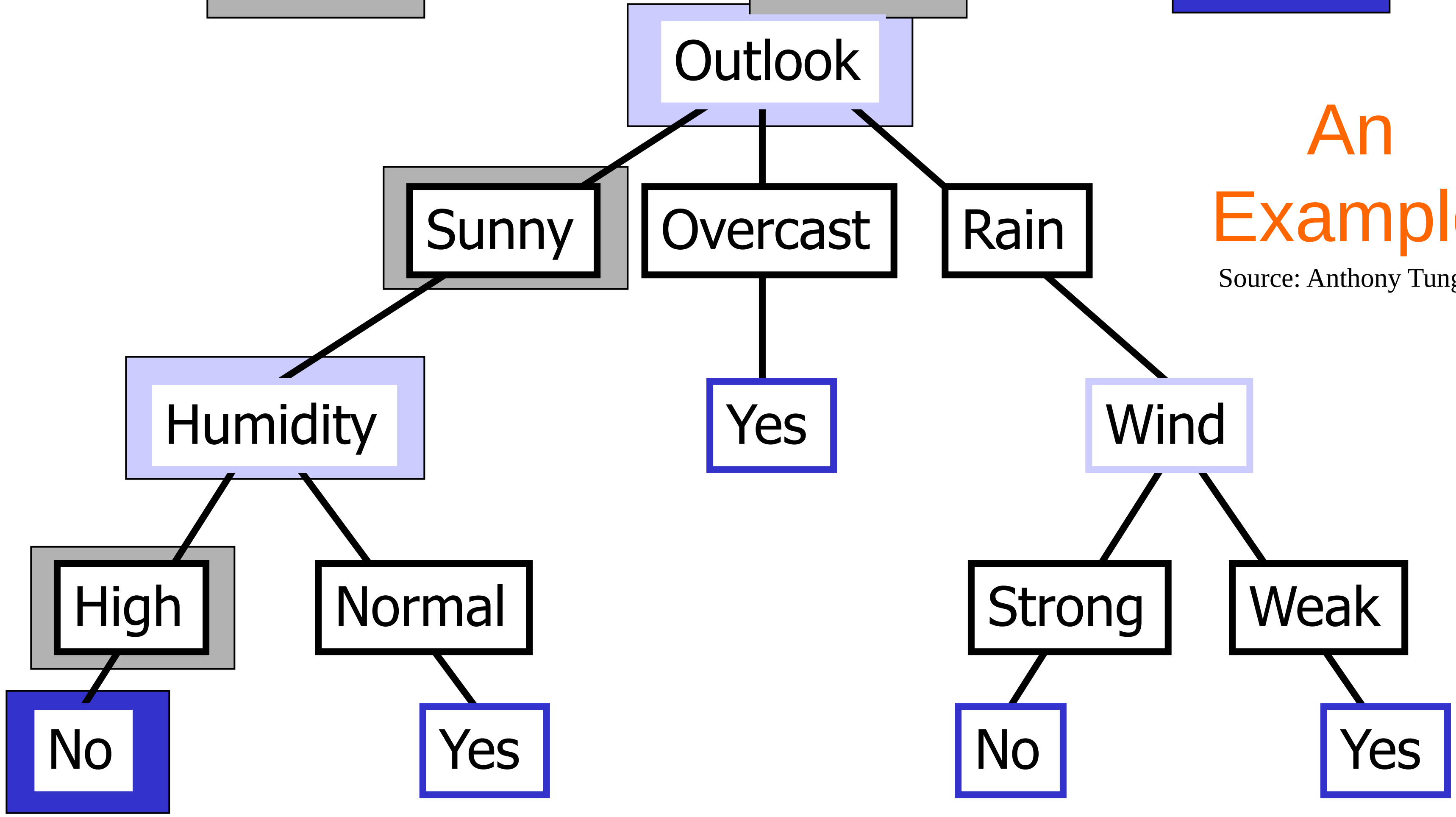
A Decision Tree



- **Construction of a tree is equivalent to determination of the root node of the tree and the root node of its sub-trees**

Exercise: What is the accuracy of this tree?

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	?No



An Example

Source: Anthony Tung

Most Discriminatory Feature

- **Every feature can be used to partition the training data**
- **If the partitions contain a pure class of training instances, then this feature is most discriminatory**

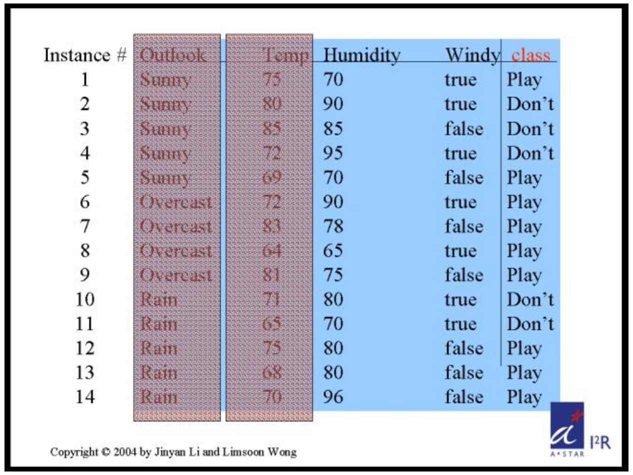
Example of Partitions

- **Categorical feature**
  - Number of partitions of the training data is equal to the number of values of this feature
- **Numerical feature**
  - Two partitions

Categorical feature

Numerical feature

Instance #	Outlook	Temp	Humidity	Windy	class
1	Sunny	75	70	true	Play
2	Sunny	80	90	true	Don't
3	Sunny	85	85	false	Don't
4	Sunny	72	95	true	Don't
5	Sunny	69	70	false	Play
6	Overcast	72	90	true	Play
7	Overcast	83	78	false	Play
8	Overcast	64	65	true	Play
9	Overcast	81	75	false	Play
10	Rain	71	80	true	Don't
11	Rain	65	70	true	Don't
12	Rain	75	80	false	Play
13	Rain	68	80	false	Play
14	Rain	70	96	false	Play



Instance #	Outlook	Temp	Humidity	Windy	class
1	Sunny	75	70	true	Play
2	Sunny	80	90	true	Don't
3	Sunny	85	85	false	Don't
4	Sunny	72	95	true	Don't
5	Sunny	69	70	false	Play
6	Overcast	72	90	true	Play
7	Overcast	83	78	false	Play
8	Overcast	64	65	true	Play
9	Overcast	81	75	false	Play
10	Rain	71	80	true	Don't
11	Rain	65	70	true	Don't
12	Rain	75	80	false	Play
13	Rain	68	80	false	Play
14	Rain	70	96	false	Play

Total 14 training instances

Outlook = sunny

1,2,3,4,5
P,D,D,D,P

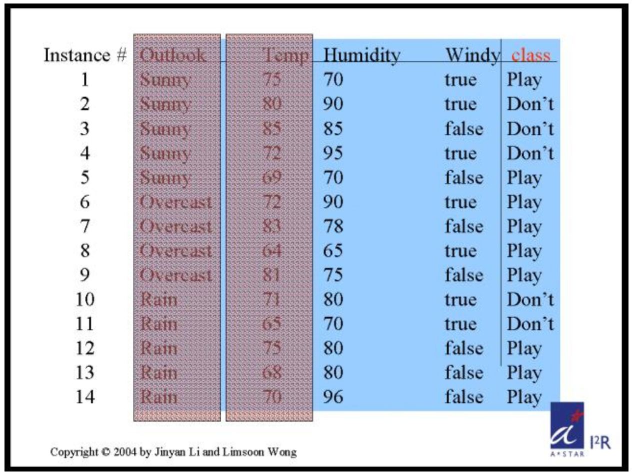
Outlook = overcast

6,7,8,9
P,P,P,P

Outlook = rain

10,11,12,13,14
D, D, P, P, P

A categorical feature is partitioned based on its number of possible values



Instance #	Outlook	Temp	Humidity	Windy	class
1	Sunny	75	70	true	Play
2	Sunny	80	90	true	Don't
3	Sunny	85	85	false	Don't
4	Sunny	72	95	true	Don't
5	Sunny	69	70	false	Play
6	Overcast	72	90	true	Play
7	Overcast	83	78	false	Play
8	Overcast	64	65	true	Play
9	Overcast	81	75	false	Play
10	Rain	71	80	true	Don't
11	Rain	65	70	true	Don't
12	Rain	75	80	false	Play
13	Rain	68	80	false	Play
14	Rain	70	96	false	Play

Total 14 training instances

Temperature <= 70

5,8,11,13,14
P,P, D, P, P

Temperature > 70

1,2,3,4,6,7,9,10,12
P,D,D,D,P,P,P,D,P

A numerical feature is generally partitioned by choosing a "cutting point"

Steps of Decision Tree Construction

- **Select the "best" feature as the root node of the whole tree**
- **Partition the dataset into subsets using this feature so that the subsets are as "pure" as possible**
- **After partition by this feature, select the best feature (wrt the subset of training data) as the root node of this sub-tree**
- **Recursively, until the partitions become pure or almost pure**

Measures to Evaluate Which Feature is Best

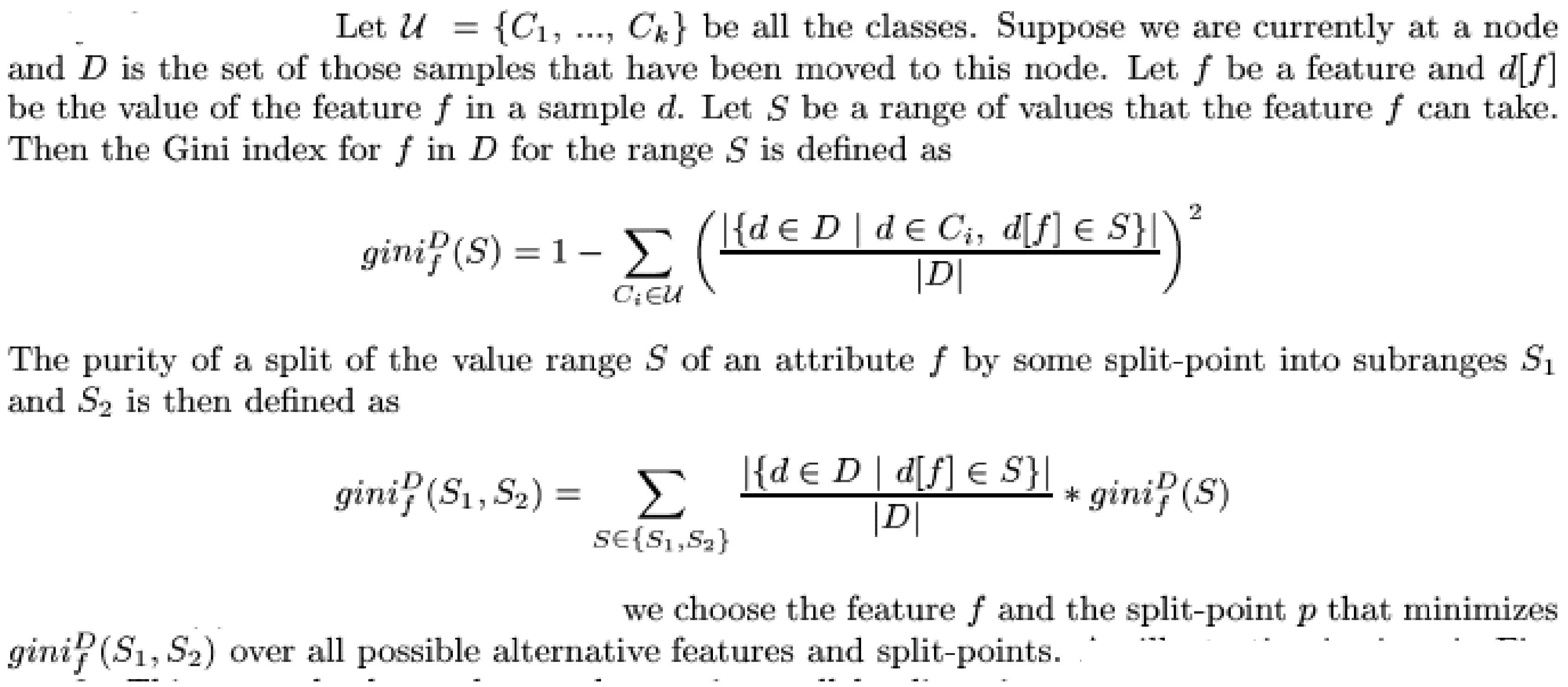
- **Gini index**
- **Information gain**
- **Information gain ratio**
- **T-statistics**
- **$\chi 2$**
- **…**

Gini Index

$$
\begin{aligned}
\text{gini}(S) &= \frac{\text{diff of two arbitrary specimen in } S}{\text{mean specimen in } S} \\
&= \text{prob(getting two specimen of diff class in } S) \\
&= \text{1 - prob(getting two specimen of same class in } S) \\
&= 1 - \textstyle\sum_i \text{prob(getting specimen of class } i \text{ in } S)^2
\end{aligned}
$$

- **Gini index is the expected value of the ratio of the diff of two arbitrary specimens to the mean value of all specimens**
- **Closer to 0, means the samples are “pure”**

Gini Index



Let $\mathcal{U} = \{C_1, ..., C_k\}$ be all the classes. Suppose we are currently at a node and D is the set of those samples that have been moved to this node. Let f be a feature and $d[f]$ be the value of the feature f in a sample d. Let S be a range of values that the feature f can take. Then the Gini index for f in D for the range S is defined as

$$gini_f^D(S) = 1 - \sum_{C_i \in \mathcal{U}} \left(\frac{|\{d \in D \mid d \in C_i,\ d[f] \in S\}|}{|D|} \right)^2$$

The purity of a split of the value range S of an attribute f by some split-point into subranges S_1 and S_2 is then defined as

$$gini_f^D(S_1, S_2) = \sum_{S \in \{S_1, S_2\}} \frac{|\{d \in D \mid d[f] \in S\}|}{|D|} * gini_f^D(S)$$

we choose the feature f and the split-point p that minimizes $gini_f^D(S_1, S_2)$ over all possible alternative features and split-points.

- **Gini index of a node: the weighted average of the purity (measured by Gini) of subtrees at the node**

$\Rightarrow$ **If each subtree is "pure", this node is good**

Decision Tree Ensembles

Motivating Example

- h_1, h_2, h_3 are indep classifiers w/ accuracy = 60%
- C_1, C_2 are the only classes
- t is a test instance in C_1
- $h(t) = \text{argmax}_{C \in \{C1,C2\}} |\{h_j \in \{h_1, h_2, h_3\} \mid h_j(t) = C\}|$
- Then $\text{prob}(h(t) = C_1)$

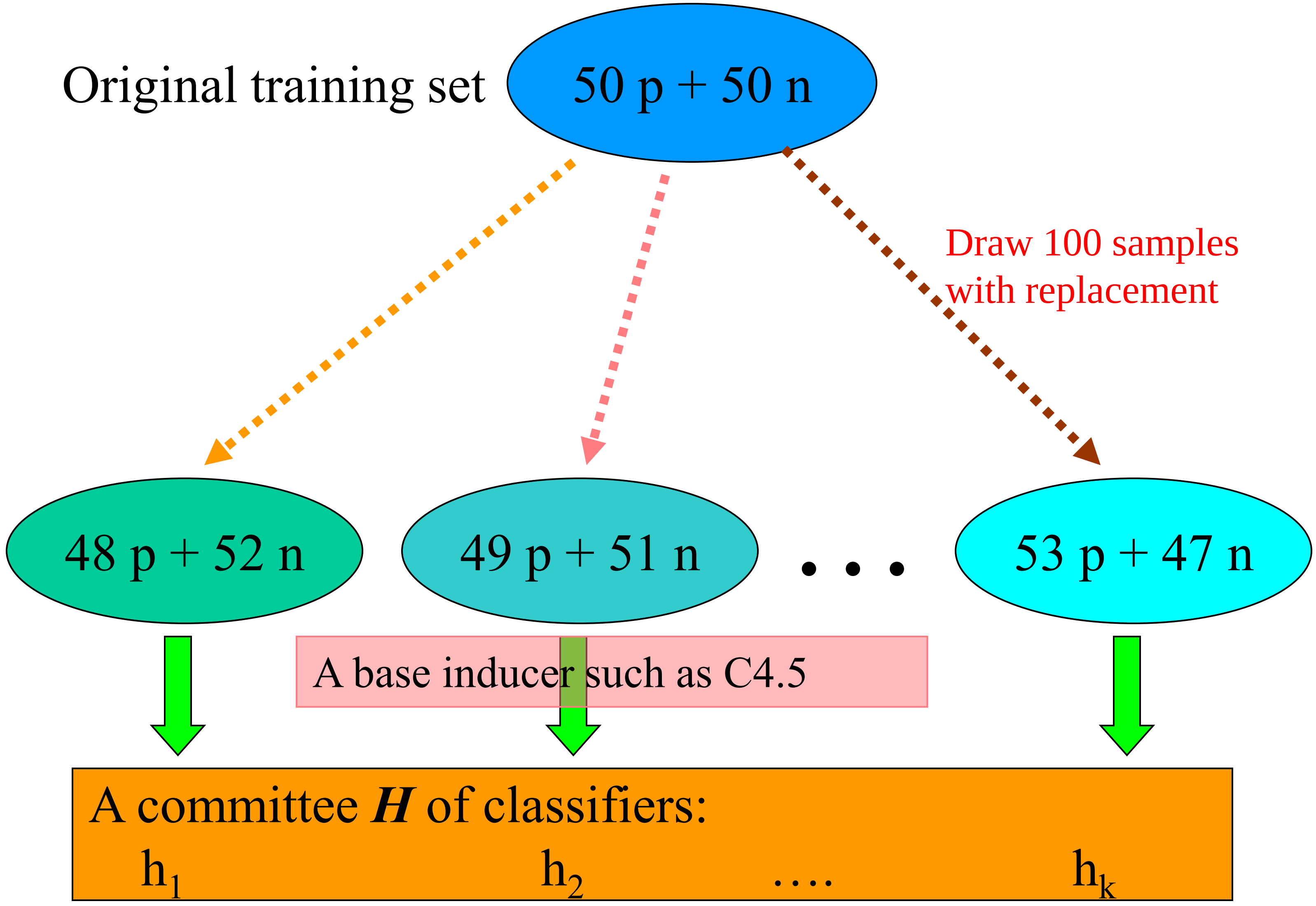
$$= \text{prob}(h_1(t)=C_1 \,\&\, h_2(t)=C_1 \,\&\, h_3(t)=C_1) + \text{prob}(h_1(t)=C_1 \,\&\, h_2(t)=C_1 \,\&\, h_3(t)=C_2) + \text{prob}(h_1(t)=C_1 \,\&\, h_2(t)=C_2 \,\&\, h_3(t)=C_1) + \text{prob}(h_1(t)=C_2 \,\&\, h_2(t)=C_1 \,\&\, h_3(t)=C_1)$$

$$= 60\% * 60\% * 60\% + 60\% * 60\% * 40\% + 60\% * 40\% * 60\% + 40\% * 60\% * 60\% = 64.8\%$$

Bagging

- **Proposed by Breiman (1996)**
- **Also called Bootstrap aggregating**
- **Make use of randomness injected to training data**

Main Ideas


Original training set
50 p + 50 n
Draw 100 samples with replacement
48 p + 52 n
49 p + 51 n
. . .
53 p + 47 n
A base inducer such as C4.5
A committee *H* of classifiers:
h_1
h_2
....
h_k

Decision Making by Bagging

Given a new test sample T

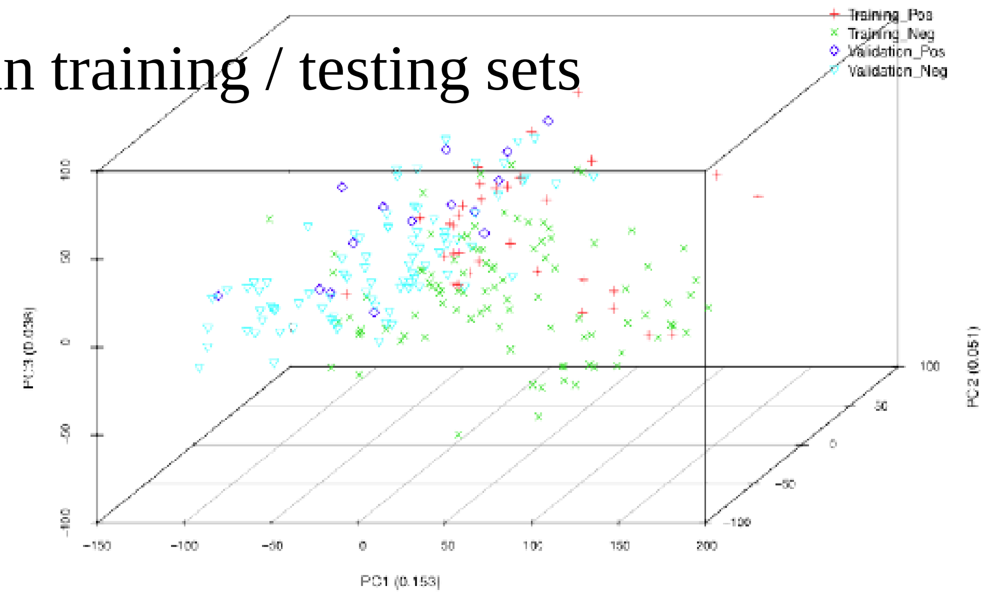
$$bagged(T) = \text{argmax}_{C_j \in \mathcal{U}} |\{h_i \in \mathcal{H} \mid h_i(T) = C_j\}|$$

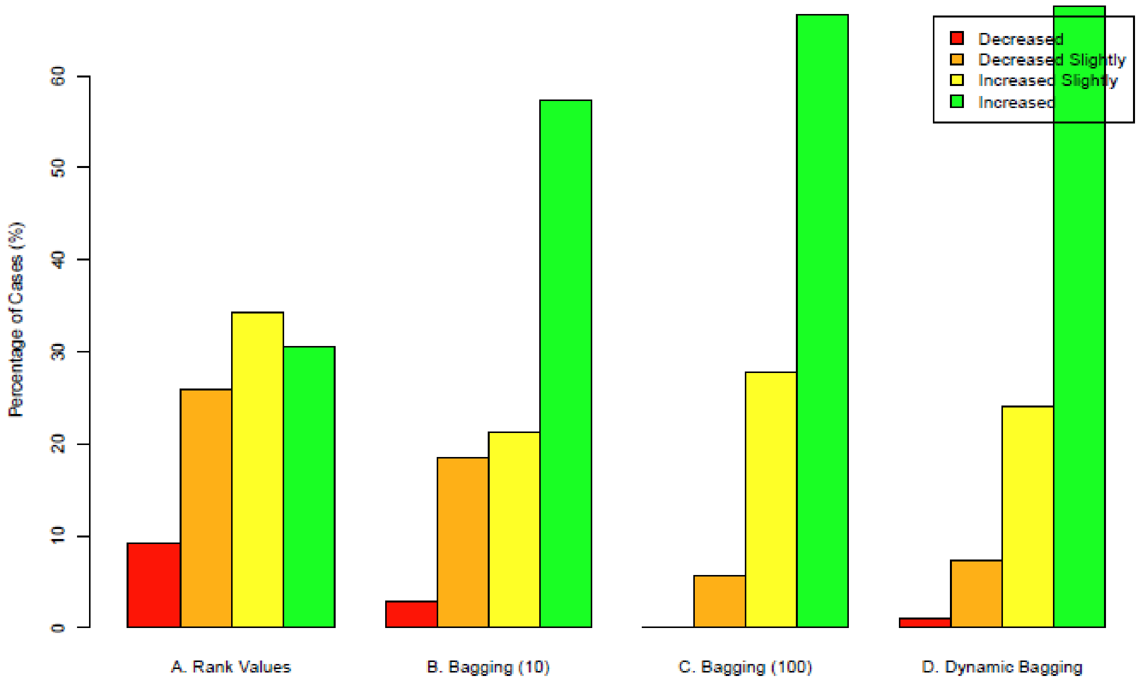
$$\text{where } \mathcal{U} = \{C_1, ..., C_r\}$$

Exercise: What does the above formula mean?

Batch effect in training / testing sets

Significantly improves cross-batch prediction accuracy in gene expression profile analyses





Concluding Remarks…

NUS
National University of Singapore

What have we learned?

- **Decision Trees**

- **Decision Trees Ensembles**
  - Bagging

- **There are many other approaches of interest, but no time to cover here …**
  - Support vector machine (SVM)
  - Nearest neighbour (kNN)
  - Naïve Bayes

References

- L. Breiman, et al. *Classification and Regression Trees*. Wadsworth and Brooks, 1984
- L. Breiman, Bagging predictors, *Machine Learning*, 24:123--140, 1996
- L. Breiman, Random forests, *Machine Learning*, 45:5-32, 2001
- J. R. Quinlan, Induction of decision trees, *Machine Learning*, 1:81--106, 1986
- J. R. Quinlan, *C4.5: Program for Machine Learning*. Morgan Kaufmann, 1993
- C. Gini, Measurement of inequality of incomes, *Economic J*, 31:124--126, 1921
- Jinyan Li et al., Data Mining Techniques for the Practical Bioinformatician, *The Practical Bioinformatician*, Chapter 3, pages 35—70, WSPC, 2004
- Y. Freund, et al. Experiments with a new boosting algorithm, *ICML 1996*, pages 148--156
- T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine Learning*, 40:139--157, 2000
- J. Li, et al. Ensembles of cascading trees, *ICDM 2003*, pages 585—588