COMP5121
Data Mining and Data Warehousing Applications

Week 6: Classification – Basic Concepts
(Chapter 8 in textbook)

Dr. Fengmei Jin

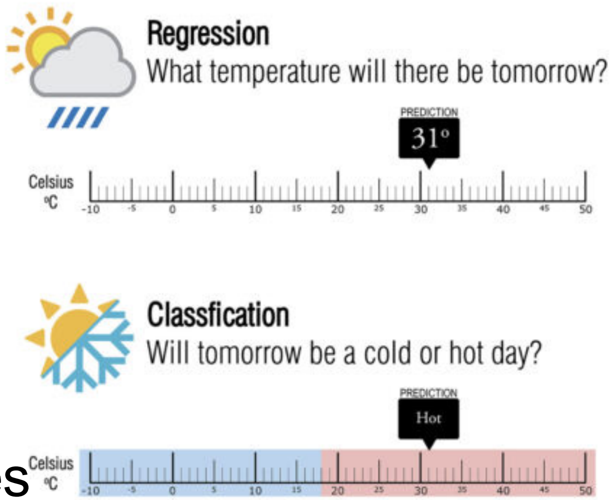
- Email: fengmei.jin@polyu.edu.hk
- Office: PQ747 (+852 3400 3327)
- Consultation Hours: 2.30-4.30pm every Thursday

Outline

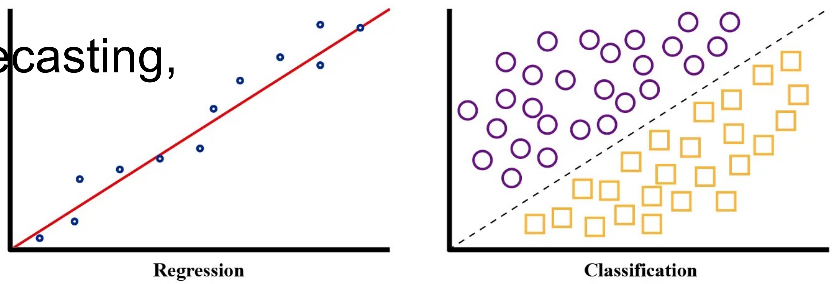
- Classification: Basic Concepts
- Decision Tree Induction – Entropy-based ID3 Algorithm
- Bayes Classification Methods
- Model Evaluation and Selection

Prediction Analysis: Classification vs. Regression

- **Prediction**: a general process of forecasting or estimating an unknown outcome



- **Classification**: categorize data objects into predefined classes
  - Output: categorical/discrete labels
  - Applications: spam detection, medical diagnosis, credit/loan approval
- **Regression**: predict continuous numerical values
  - Output: numbers on a continuous scale
  - Applications: stock price, temperature forecasting, traffic flow control



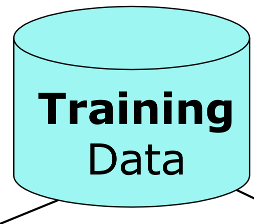
Supervised vs. Unsupervised Learning

- Supervised Learning: Classification
  - **Supervision**: The training data are accompanied by labels, indicating classes of observations.
- Unsupervised Learning: Clustering
  - The class labels of training data is unknown.
  - Given a set of observations, it aims to establish clusters in the data through a *self-discovery* process.
  - Applications: customer segmentation, topic modeling

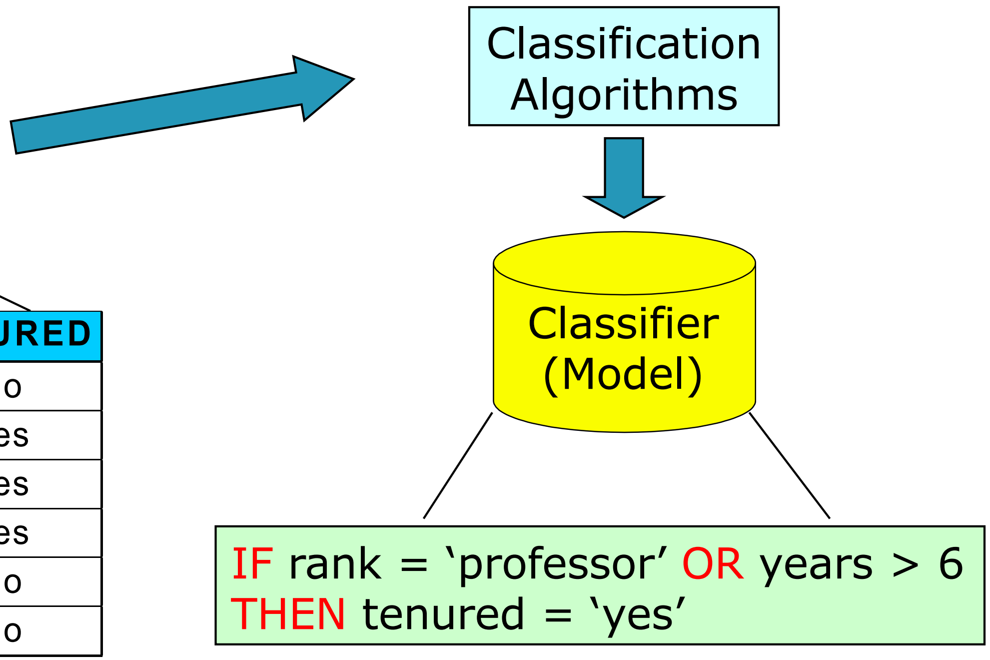
Classification – A Two-Step Process

1. **Model Construction**
   - To learn a model using a training set (data with known labels), assuming each tuple belongs to a predefined class
   - **Model**: represented as classification rules, decision trees, or mathematical formulas
2. **Model Usage**
   - To classify future or unknown objects
   - Evaluation by comparing predicted labels with actual ones
     - **Accuracy**: % of samples correctly classified by the models
     - Avoid overfitting by testing on independent data (testing/validating)
     - If the accuracy is acceptable, use the model for future predictions

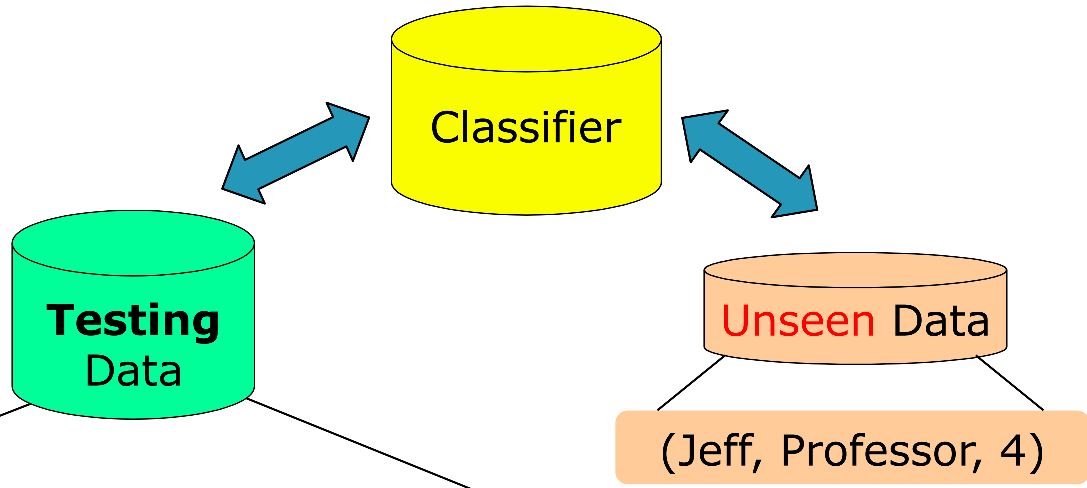
Process (1): Model Construction



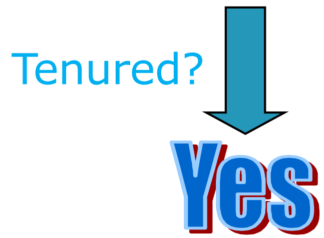
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no



Process (2): Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes



Issues in Classification and Prediction

- **Data Preparation**
  - Data cleaning: reduce noises and handle missing values
  - Correlation analysis: remove irrelevant or redundant attributes
  - Data transformation: generalize and normalize data
- **Model Evaluation**
  - Accuracy: how well the model performs
  - Speed: time to construct and use the model
  - Scalability: efficiency when handling large-scale DBs
  - Robustness: ability to handle noise and missing values
  - Interpretability: how easily the model's insights can be understood
  - Goodness of rules: 1) decision tree size; 2) compactness of rules

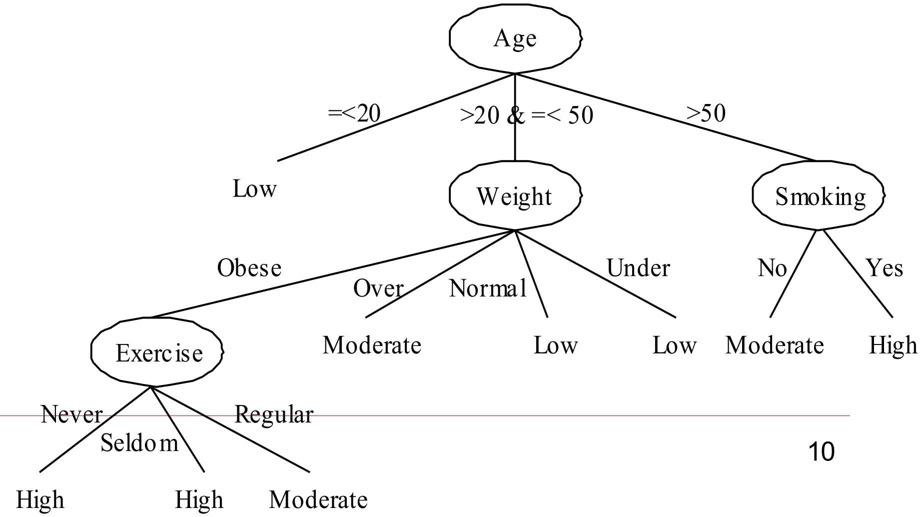
Iterative Dichotomiser (ID3), CART, C4.5, …

CLASSIFICATION BY DECISION TREE INDUCTION

Decision Tree Structure

- A flow-chart-like structure used for classification
  - **Internal node**: a test on an attribute (e.g., age, exercise, weight, smoking)
  - **Branch**: an outcome of the test
  - **Leaf nodes**: class labels (e.g., high-, moderate-, and low-risk)

How it works:
An object is classified by traversing the tree from its root to a leaf.



Decision Tree Induction

- **Decision Tree Generation**
  1. Tree Construction
     - Initially, all training examples are at the root (a single node).
     - The data is partitioned recursively based on selected attributes.
     - Data-driven: requires NO domain knowledge or parameter settings.
  2. Tree Pruning
     - Identify and remove branches that reflect noise or outliers.
     - Improve accuracy on unseen data by preventing overfitting.

- **Use of Decision Tree**: Classify an unknown sample by testing its attribute values against the decision tree

Decision Tree Construction

- A decision tree is said to represent the classification if it correctly classifies all training instances → consistent with the training data.

- Simple Idea, Complex Problem
  - Can be built in many possible ways to split the data and represent the classification → *finding the best one is a challenge!*
  - Does a tree consistent with training data have the highest likelihood of accurately classifying unseen instances of the population?

- The goal is not just consistency with the training data, but also generalization to unseen new data.

Example: Training Set

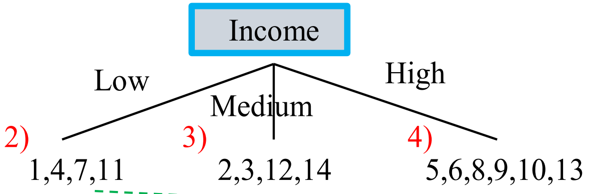
Risk Assessment for Loan Applications

Client #	Credit History	Debt Level	Collateral	Income Level	RISK LEVEL
1	Bad	High	None	Low	**HIGH**
2	Unknown	High	None	Medium	**HIGH**
3	Unknown	Low	None	Medium	**MODERATE**
4	Unknown	Low	None	Low	**HIGH**
5	Unknown	Low	None	High	**LOW**
6	Unknown	Low	Adequate	High	**LOW**
7	Bad	Low	None	Low	**HIGH**
8	Bad	Low	Adequate	High	**MODERATE**
9	Good	Low	None	High	**LOW**
10	Good	High	Adequate	High	**LOW**
11	Good	High	None	Low	**HIGH**
12	Good	High	None	Medium	**MODERATE**
13	Good	High	None	High	**LOW**
14	Bad	High	None	Medium	**HIGH**

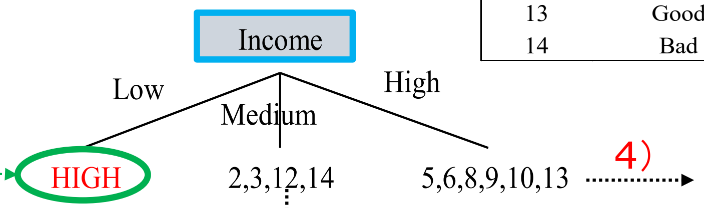
Decision Tree Generation

Client #	Credit History	Debt Level	Collateral	Income Level	RISK LEVEL
1	Bad	High	None	Low	**HIGH**
2	Unknown	High	None	Medium	**HIGH**
3	Unknown	Low	None	Medium	**MODERATE**
4	Unknown	Low	None	Low	**HIGH**
5	Unknown	Low	None	High	**LOW**
6	Unknown	Low	Adequate	High	**LOW**
7	Bad	Low	None	Low	**HIGH**
8	Bad	Low	Adequate	High	**MODERATE**
9	Good	Low	None	High	**LOW**
10	Good	High	Adequate	High	**LOW**
11	Good	High	None	Low	**HIGH**
12	Good	High	None	Medium	**MODERATE**
13	Good	High	None	High	**LOW**
14	Bad	High	None	Medium	**HIGH**

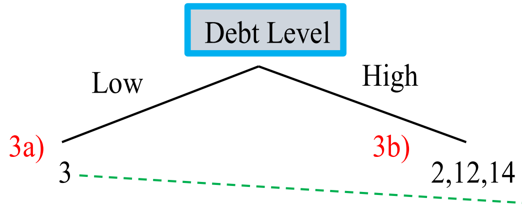
1) Choose Income as root of tree.



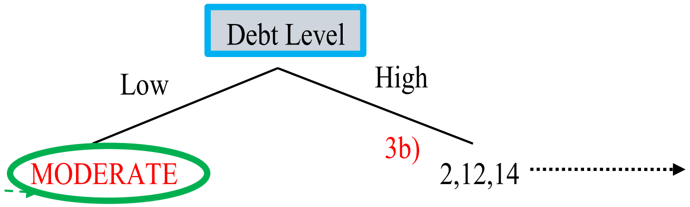
2) All examples are in the same class, HIGH.
Return Leaf Node.



3) Choose Debt Level as root of subtree.



3a) All examples are in the same class, MODERATE.
Return Leaf node.

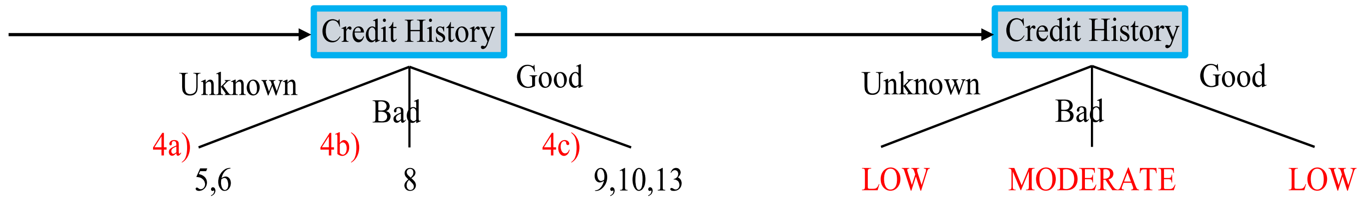


Decision Tree Generation

Client #	Credit History	Debt Level	Collateral	Income Level	RISK LEVEL
1	Bad	High	None	Low	**HIGH**
2	Unknown	High	None	Medium	**HIGH**
3	Unknown	Low	None	Medium	**MODERATE**
4	Unknown	Low	None	Low	**HIGH**
5	Unknown	Low	None	High	**LOW**
6	Unknown	Low	Adequate	High	**LOW**
7	Bad	Low	None	Low	**HIGH**
8	Bad	Low	Adequate	High	**MODERATE**
9	Good	Low	None	High	**LOW**
10	Good	High	Adequate	High	**LOW**
11	Good	High	None	Low	**HIGH**
12	Good	High	None	Medium	**MODERATE**
13	Good	High	None	High	**LOW**
14	Bad	High	None	Medium	**HIGH**

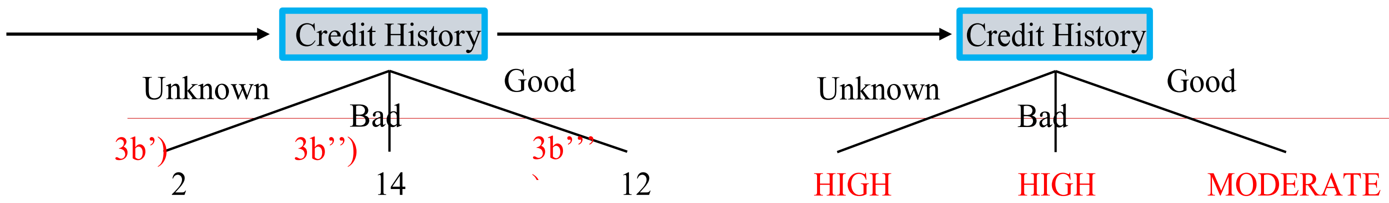
4) Choose Credit History as root of subtree.

4a-4c) All examples are in the same class.
Return Leaf nodes.



3b) Choose Credit History as root of subtree.

3b'-3b''') All examples are in the same class.
Return Leaf nodes.



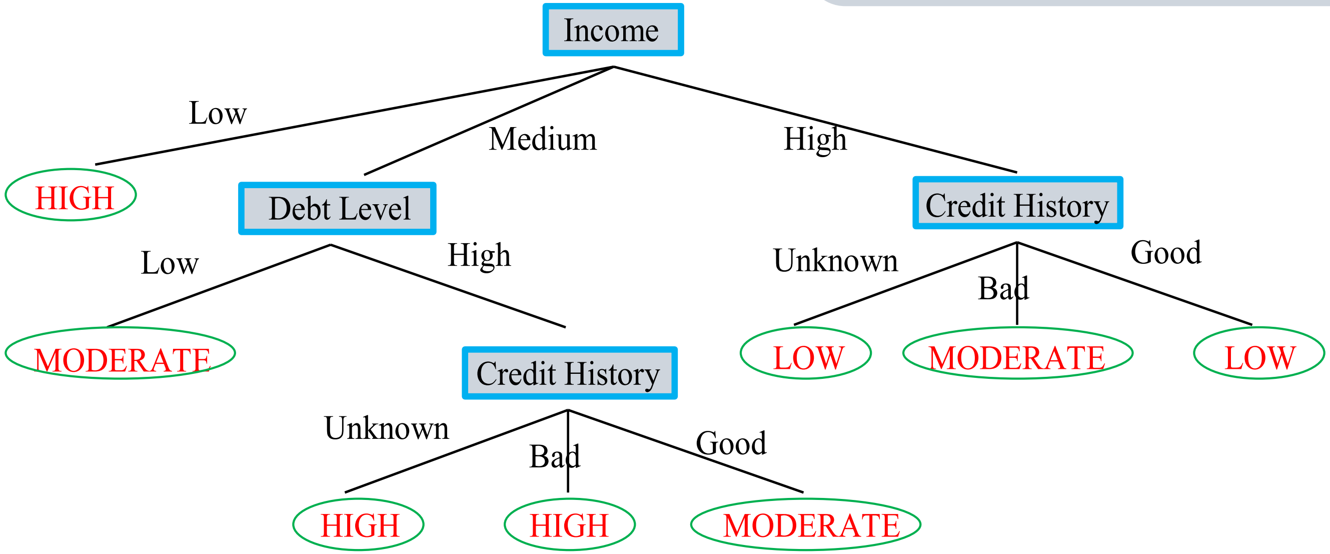
Final Decision Tree

Attribute Selection:

based on heuristic or statistical measures

(e.g., information gain in ID3).

Attach subtrees at appropriate places.



Entropy

- A measure of randomness, uncertainty, and disorder in a system with probability distributions of outcome.
- Entropy is formulated as a *function* that measures disorder.
  - *"The higher the entropy, the greater the disorder."*
  - For classification, it tells how diverse the classes are in a set.
- Let $\boldsymbol{D}$ be a set of examples from $\boldsymbol{m}$ classes.

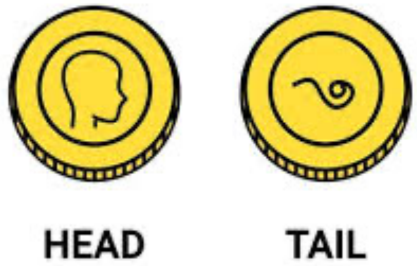
$$Info(D) = -\sum_{i=1}^{m} p_i \cdot \log_2(p_i)$$

- **Input**: Distribution of outcomes
- **Output** : A value indicating how disordered the outcomes are
- p_i: The proportion of examples observed in D that belong to i-th class within [0,1].

Example: Tossing Coins in Casino

- **Casino A** with real coins (50/50 chances):

$$Info(Coin\ Toss) = -p(head)\log_2 p(head) - p(tail)\log_2 p(tail)$$
$$= -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$



- **Casino B** with fake coins (75/25 chances):

$$Info(Coin\ Toss) = -p(head)\log_2 p(head) - p(tail)\log_2 p(tail)$$
$$= -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.811$$

Entropy is a measure of randomness and disorder.
Higher entropy means higher uncertainty.

Information Gain and Iterative Dichotomiser (ID3)

- **Classification Goal**: To split the dataset in a way that reduces entropy the most.
- **Information Gain:** To measure the reduction in entropy after splitting the dataset on an attribute A

$$Gain(D, A) = Info(D) - Info_A(D)$$

  - Weighted entropy after split: $Info_A(D) = \sum_{j=1}^{n} p(D_j|A) Info(D_j)$
    - D_j: subsets of D created by splitting on A

ID3 Algorithm: Repeatedly selects the attribute with the highest information gain at each step to build the decision tree.

ID3 Example (Decision: buy computer or not)

- **Class P**: buys_computer = 'yes' → 9
- **Class N**: buys_computer = 'no' → 5

- **Info**$(D) = \sum -p_i \times \log_2 p_i$
- **Info**$_A(D) = \sum [p(D_j|A) \times \mathbf{Info}(D_j)]$
- **Gain**$(D, A) = \mathbf{Info}(D) - \mathbf{Info}_A(D)$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\boxed{\frac{5}{14}I(2,3)}$ means 'age <=30' has 5 out of 14 samples, with 2 'yes' and 3 'no'.

Hence,

$$\boxed{Gain(age) = Info(D) - Info_{age}(D) = 0.246}$$

Similarly, $Gain(income) = 0.029$

$Gain(student) = 0.151$

$Gain(credit\_rating) = 0.048$

ID3 Example (Decision: buy computer or not)

Detailed Calculations

income	total	p	n	I(p,n)
low	4	3	1	0.8113
medium	6	4	2	0.918
high	4	2	2	1

INFO(D)= -9/14 $\log_2(9/14)$- 5/14$\log_2(5/14)$ = 0.940 bits

$Info_{INCOME}$(D)=4/14 I(3,1)+6/14 I(4,2)+4/14I(2,2)

$Info_{INCOME}$(D)=4/14 *0.811+6/14 *0.918)+4/14*1= 0.231+0.393+0.285 =0.909

GAIN(INCOME) = INFO(D)- $INFO_{income}$(D)= 0.940-0.909=0.029

student	total	p	n	I(p,n)
yes	7	6	1	0.5917
no	7	3	4	0.9852

INFO(D)= -9/14 $\log_2(9/14)$- 5/14$\log_2(5/14)$ = 0.940 bits

INFO(student)= 7/14 *I(6,1)+7/14* I(3,4)

INFO(student)= 7/14 *0.597 +7/14* 0.9852 =0.791

GAIN(STUDENT) = INFO(D)- $INFO_{STUDENT}$(D)= 0.940-0.791= 0.151 bits

creditrating	total	p	n	I(p,n)
fair	8	6	2	0.8113
excellent	6	3	3	1

INFO(D)= -9/14 $\log_2(9/14)$- 5/14$\log_2(5/14)$ = 0.940 bits

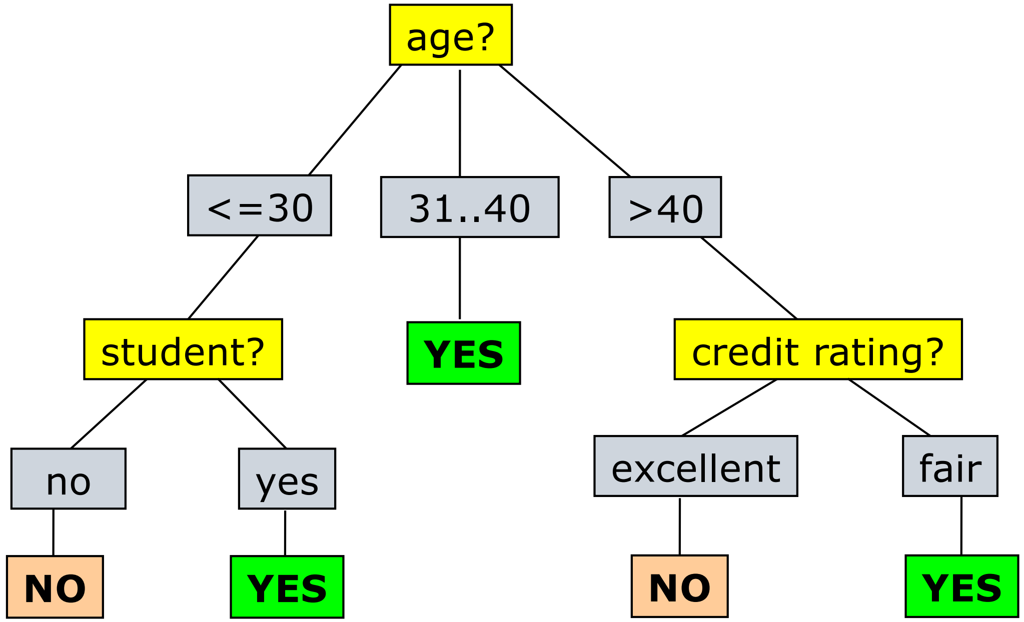
INFO(creditrating)= 8/14 *I(6,2)+7/14* I(3,3)

INFO(creditrating)= 7/14 *0.811 +7/14* 1 =0.9055

GAIN(credit rating) = INFO(D)- $INFO_{creditrating}$(D)= 0.940-0. 9055= 0.048 bits

Final Decision Tree based on ID3

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



ID3 Example (Decision: play outside or not)

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	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

- $\mathbf{Info}(D) = \sum -p_i \times \log_2 p_i$
- $\mathbf{Info}_A(D) = \sum\left[p\left(D_j \middle| A\right) \times \mathbf{Info}\left(D_j\right)\right]$
- $\mathbf{Gain}(D, A) = \mathbf{Info}(D) - \mathbf{Info}_A(D)$

Step 1: Entropy based on "Decision"

$$\mathbf{Info}(\mathit{Decision}) = -p(Yes)\times\log_2 p(Yes) - p(No)\times\log_2 p(No)$$
$$= -\frac{9}{14}\times\log_2\frac{9}{14} - \frac{5}{14}\times\log_2\frac{5}{14} = 0.940$$

ID3 Example (Decision: play outside or not)

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

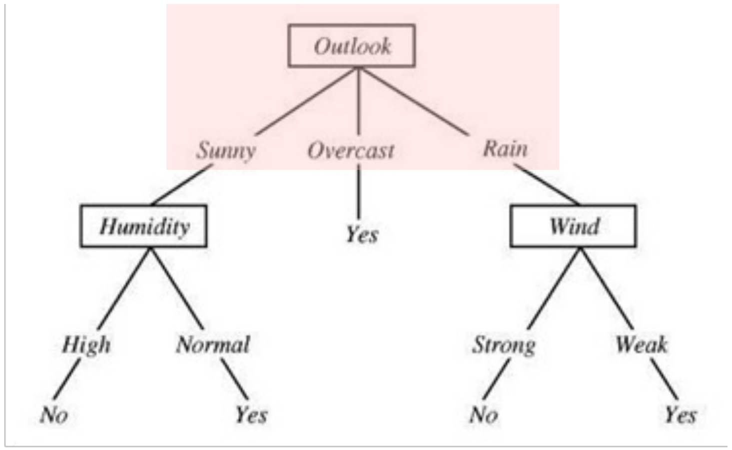
Wind on Decision

Gain(*Decision*, *Wind*)
= **Info**(*Decision*)
– [p(*Decision* | *Wind*=*Weak*) x **Info**(*Decision* | *Wind*=*Weak*)]
– [p(*Decision* | *Wind*=*Strong*) x **Info**(*Decision* | *Wind*=*Strong*)]

- **Info**(*Decision* | *Wind*=*Weak*) $= -\frac{2}{8} \times \log_2 \frac{2}{8} - \frac{6}{8} \times \log_2 \frac{6}{8} = 0.811$
- **Info**(*Decision* | *Wind*=*Strong*) $= -\frac{3}{6} \times \log_2 \frac{3}{6} - \frac{3}{6} \times \log_2 \frac{3}{6} = 1$
- **Gain**(Decision, Wind)
  = 0.940 – (8/14) x 0.811 – (6/14) x 1 = 0.048

ID3 Example (Decision: play outside or not)

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	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, **Temperature** and **Humidity** on Decision

- **Gain**(*Decision*, *Outlook*) = 0.246 (highest gain)
- **Gain**(*Decision*, *Humidity*) = 0.151
- **Gain**(*Decision*, *Wind*) = 0.048
- **Gain**(*Decision*, *Temperature*) = 0.029

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
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

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

(1) Outlook = *Overcast* on Decision

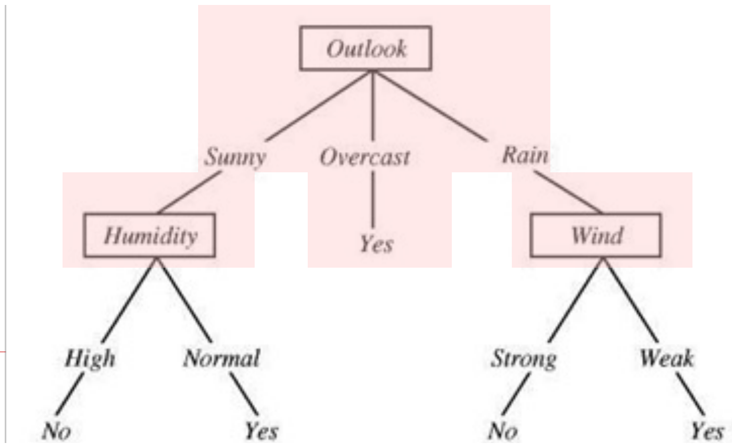
Decision will always be 'Yes', no need to calculate this branch, it is leaf node already.

(2) Outlook = *Sunny* on Decision

- For 5 sunny instances: 3/5 'No', and 2/5 'Yes'.
  - Gain(Outlook = *Sunny*, Temp.) = 0.570
  - Gain(Outlook = *Sunny*, Humidity) = 0.971
  - Gain (Outlook = *Sunny*, Wind) = 0.020

(3) Outlook = *Rain* on Decision

- For 5 rain instances: 2/5 "No", and 3/5 "Yes".
  - Gain(Outlook = *Rain*, Temp.) = 0.020
  - Gain(Outlook = *Rain*, Humidity) = 0.020
  - Gain (Outlook = *Rain*, Wind) = 0.971



Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data.
  - **Too many branches**: some may reflect noise or outliers, rather than meaningful patterns → Poor accuracy for unseen samples

- Avoid Overfitting
  - **Pre-pruning**: stop tree construction early – do not split a node if it would result in the *goodness* measure falling below a threshold
    - Challenge: choosing an appropriate threshold can be difficult.
  - **Post-pruning**: start with a fully grown tree, then remove branches progressively to simplify the tree
    - Use a validation dataset different from training sets to decide the best-pruned tree

statistical, probabilistic, efficient

BAYES CLASSIFICATION METHODS

Bayesian Classifier

- **Decision Tree**: data-driven, rule-based reasoning; generating highly interpretable and explainable decisions

- Real-world problems involve **uncertainty**.
  - How likely is this email a spam based on the words it contains?
  - What is the likelihood of a patient having flu based on symptoms?
- Probability-based Bayesian Classifiers:
  - prior knowledge (what we already know from training data) + observed evidence (new data) + **Bayesian Theorem**

    → Make informed decisions

Bayesian Theorem

- $P(H|E)$: Posterior probability, the probability of H holds given E
  - E: Evidences (e.g., a data tuple) with attribute description
  - H: Hypothesis to be verified (e.g., a class label that E belongs to)

$$\boldsymbol{P(H|E) = \frac{P(E|H)\,P(H)}{P(E)}}$$

- $P(H)$: prior probability, i.e., the initial probability of hypothesis H before observing evidence E
- $P(E)$: marginal probability, i.e., the total probability of observing evidence E under all possible hypotheses
- $P(E|H)$: likelihood, i.e., the probability of observing evidence E given that the hypothesis $H = true$

Bayesian Classification

- A data tuple: $X = (A_1 = x_1, A_2 = x_2, A_3 = x_3, \dots, A_n = x_n)$
- To classify X, we need to estimate $P(C_i \mid X)$
  - C_i represents the **hypothesis** that X belongs to C_i.
  - We say X belongs to C_i iff: $P(C_i|X) > P(C_j|X), for\ all\ j \neq i$

- How to estimate $P(C_i \mid X)$ for classifying X?
  - **Bayesian theorem**: $P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$
  - The problem becomes → estimating $P(X|C_i)$ and $P(C_i)$

Bayesian Classification

- Estimate the priori probability of the i-th class C_i from the training set D: $P(C_i) = \frac{|C_i|}{|D|}$

- Independence Assumption: For $P(X \mid C_i)$, we assume that the effect of each attribute A_j is independent to others:

$$P(X = (A_1 = x_1, A_2 = x_2, \ldots, A_n = x_n) \mid C_i) \\ = P(A_1 = x_1 | C_i) \times P(A_2 = x_2 | C_i) \times \cdots \times P(A_n = x_n | C_i)$$

where $P\left(A_j = x_j \middle| C_i\right)$ can also be estimated from the training set D.

Example

Bayesian: $P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$

- Given a training set, predict if a person X will buy a computer
  - $\boldsymbol{X}$: {*age* = youth, *income* = medium, *student* = yes, *credit_rating* = fair}
  - Yes or No? $P(buy\_computer|X)$

Priori Probability in training Data:

- $P(buy\_computer = yes) = 9/14 = 0.643$
- $P(buy\_computer = no) = 5/14 = 0.357$

age	*buys_computer* yes	*buys_computer* no
youth	2	3
middle_aged	4	0
senior	3	2

income	*buys_computer* yes	*buys_computer* no
low	3	1
medium	4	2
high	2	2

student	*buys_computer* yes	*buys_computer* no
yes	6	1
no	3	4

credit_ratting	*buys_computer* yes	*buys_computer* no
fair	6	2
excellent	3	3

To calculate $\boldsymbol{P(X \mid buy\_computer = yes)}$:

- $P(age = youth \mid yes) = 2/9 = 0.222$
- $P(income = medium \mid yes) = 4/9 = 0.444$
- $P(student = yes \mid yes) = 6/9 = 0.667$
- $P(credit\_rating = fair \mid yes) = 6/9 = 0.667$

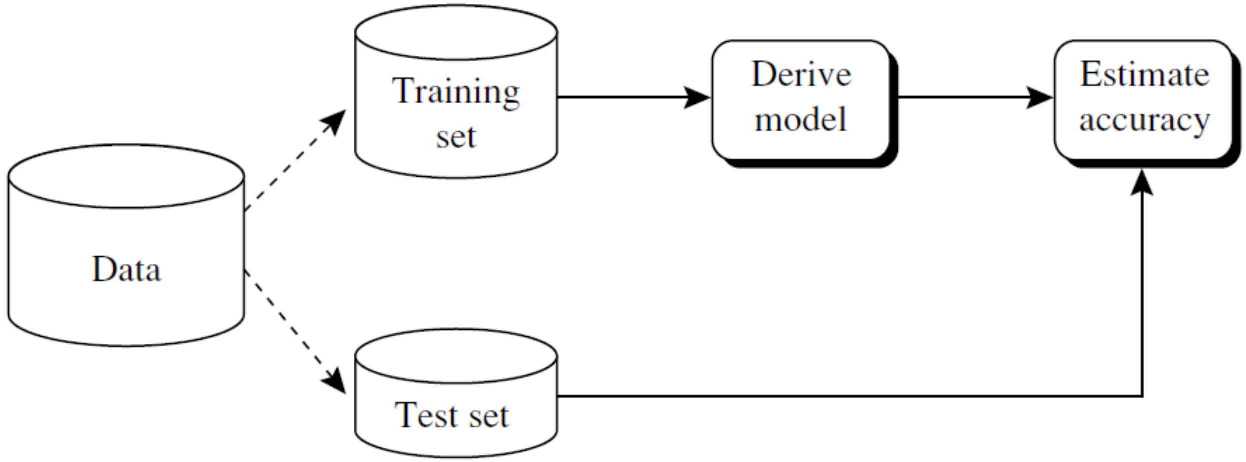
→ $P(X \mid buy\_computer = yes) = 0.044$

→ Similarly, $P(X \mid buy\_computer = no) = 0.019$

Through Bayesian:

- $P(X \mid yes) \times P(buy\_computer = yes) = 0.028$
- $P(X \mid no) \times P(buy\_computer = no) = 0.007$

Conclusion: X will buy a computer.



MODEL EVALUATION AND SELECTION

Evaluation Measures

- To assess how "accurate" your classifier is at predicting the class label of tuples compared to actual labels
  - **True Positives TP**: positive tuples that were correctly labeled
    - Positive tuples: tuples of the main class of interest
  - **True Negatives TN**: negative tuples that were correctly labeled
  - **False Positives FP**: negative tuples that were incorrectly labeled as positive (e.g., people who do not buy computers but are labeled as $buys\_computer = yes$)
  - **False Negatives FN**: positive tuples that were mislabeled as negative (e.g., people who really buy computers but are labeled as $buys\_computer = no$)

		Predicted class		
Actual class		*yes*	*no*	Total
	yes	TP	FN	P
	no	FP	TN	N
	Total	P'	N'	$P + N$

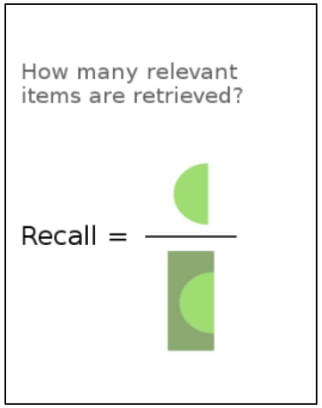
Evaluation Measures

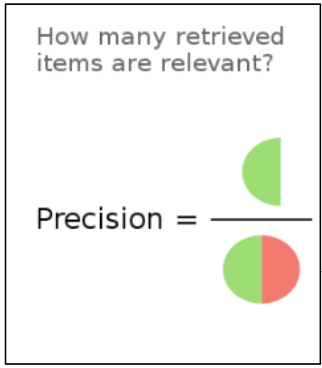
Measure	Formula
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
F, F_1, F-score, harmonic mean of precision and recall	$\frac{2 \times precision \times recall}{precision + recall}$
F_β, where β is a non-negative real number	$\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$

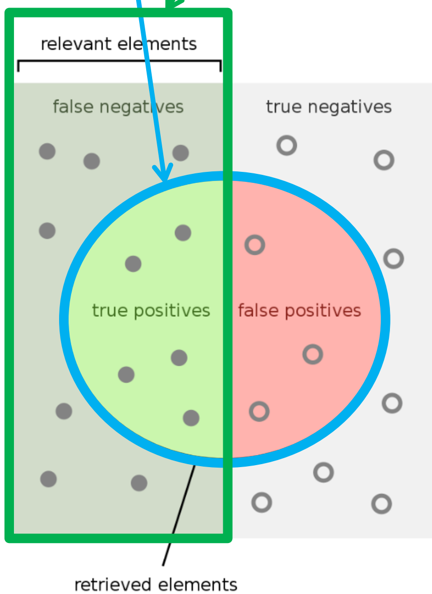
Predicted class

Actual class

	yes	*no*	Total
yes	TP	FN	P
no	FP	TN	N
Total	P'	N'	$P+N$







K-Fold Cross-Validation

Key Concepts

- Partitioning: splits data into k equal-sized folds
- Rotation: each fold is a test set once, others form the training set
- Iteration: k rounds of training/testing

Process

- 1st Iteration: Train on D2 to Dk, test on D1.
- 2nd Iteration: Train on D1, D3 to Dk, test on D2.
- ...
- kth Iteration: Train on D1 to D(k-1), test on Dk.

Advantages

- Bias Reduction: Each data point is used for training and testing.
- Robustness: Accuracy is averaged over k iterations.

Summary

- **Classification**: a form of data analysis that extracts models describing data classes. A classifier predicts categorical labels.
- **Decision tree induction**: a top-down recursive tree induction model, using an attribute selection measure to select the attribute tested for each *non-leaf* node in the tree – ID3 as the example algorithm
  - Tree pruning: to improve accuracy by removing tree branches reflecting noise in the data.
- **Bayesian** classifier: based on Bayes' theorem of posterior probability
  - the effect of an attribute value on a given class is independent of the values of the other attributes
- **Evaluation**: accuracy, precision, recall, F1, …

Email: fengmei.jin@polyu.edu.hk

Office: PQ747

THANK YOU!