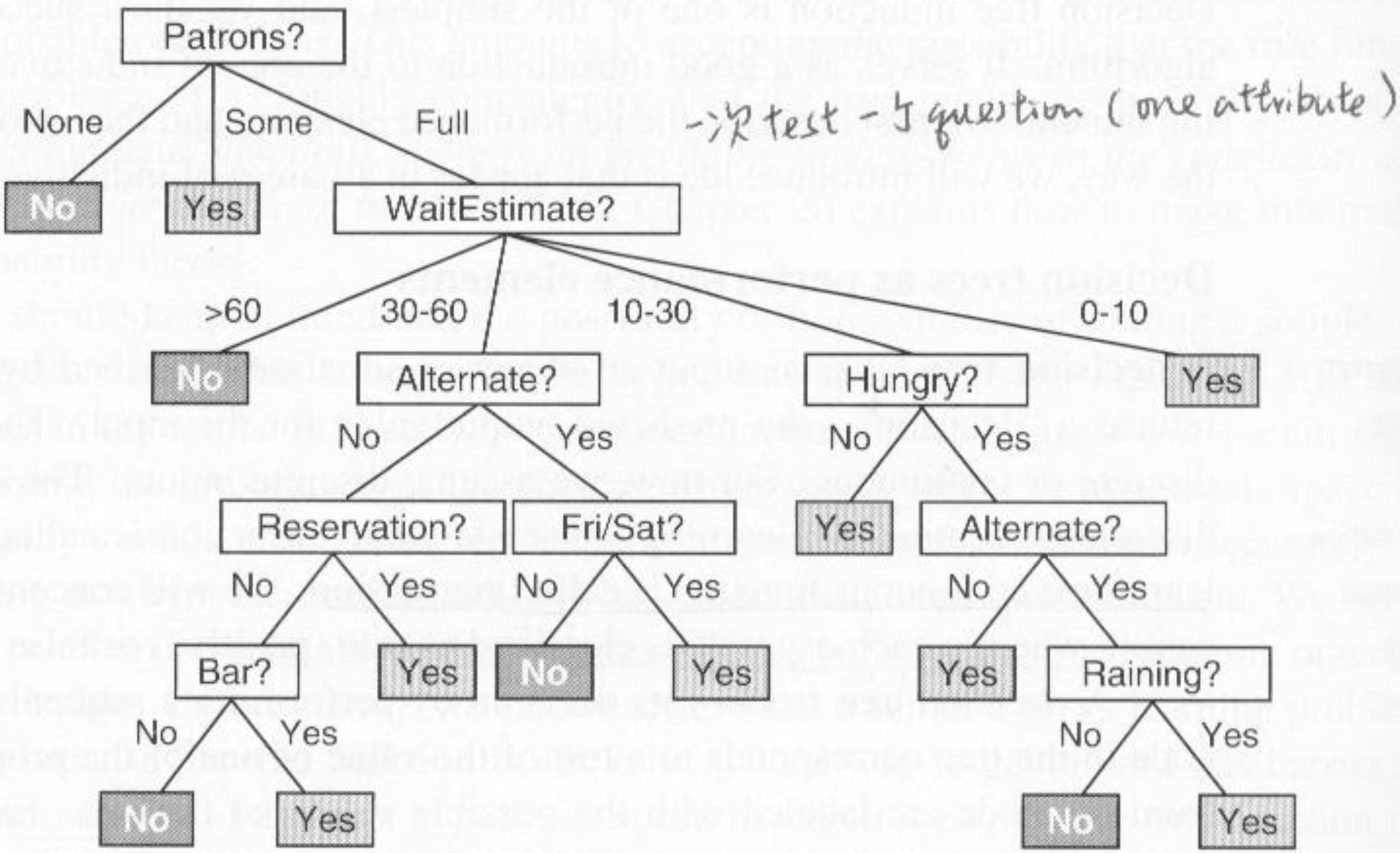
Inductive Learning - Classification and Regression Tree

Bor-shen Lin
bslin@cs.ntust.edu.tw
http:www.cs.ntust.edu.tw/~bslin

DECISION TREE

Cited from AI, A modern approach, Russel Norvig



Decision Tree

- Every non-leaf node is attached with a question based on some property
- Every leaf node is assigned a class
  - Yes/No here
- Every branch onto a node is associated with a condition for the question applied at its parent node
- Every instance can be classified to specific class after going through the decision tree
- Is helpful for decision making
- It can be generated manually or automatically

Inductive Methods

- What is induction?
  - Find common rules from cases/experiences
- Human is able to use inductive methods
  - Find classification rules for things
  - Construct ontologies for things (classification trees)
- Induction can help to make decision
  - Apply a school: reputation, site, fee, gender, ….
  - Find mates: economy, character, look, shape, …
  - Find jobs: money, load, distance, prospect, …
  - Invest: fund, risk, reward, value, …
  - Buy a house/car or choose restaurants …
- Can computers perform inductive learning?

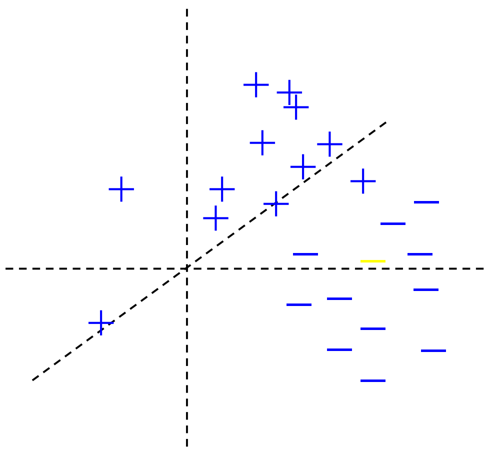
Cases for Decision

Input X

Output Y

	income	height	weight	look	school	PASS
A	32k	172	66	A	X Univ.	YES
B	35k	166	56	B	C Univ.	NO
C	29k	180	79	B	T Univ.	YES
D	25k	175	65	A	U Univ.	NO
E	40k	169	75	C	Y Univ.	NO
…	…	…	…	…	…	…

Vector Space Model



- Input X (domain): high dimensional vector
  - $\underline{X}$ = [32k, 172, 66, A+, X-Univ]
- Output Y(range): YES(+)/NO(-)
  - Y = [YES]
- Y = f($\underline{X}$) f: classifier function

Practice: Vector Space Model

- Represent a decision issue as vector space model
  - Input $\underline{X}$, output Y
  - Question, Property
- Example
  - Apply a school
  - Find a mate
  - Look for a job
  - Perform investment
  - Buy a car
  - Buy a house

	Y	X_1	X_2	X_3	X_4
	RISK	Credit history	Debt	Collateral	Income
E1	**High**	Bad	High	None	<15k
E2	**High**	Unknown	High	None	15k-35k
E3	**Moderate**	Unknown	Low	None	15k-35k
E4	**High**	Unknown	Low	None	<15k
E5	**Low**	Unknown	Low	None	>35k
E6	**Low**	Unknown	Low	Adequate	>35k
E7	**High**	Bad	Low	None	<15k
E8	**Moderate**	Bad	Low	Adequate	>35k
E9	**Low**	Good	Low	None	>35k
E10	**Low**	Good	High	Adequate	>35k
E11	**High**	Good	High	None	<15k
E12	**Moderate**	Good	High	None	15k-35k
E13	**Low**	Good	High	None	>35k
E14	**High**	Bad	High	None	15k-35k

Practice: Partition of Data

- Please partition the data E1~E14 according to the property X1 (credit history)
- Please partition the data E1~E14 according to property X4 (income)

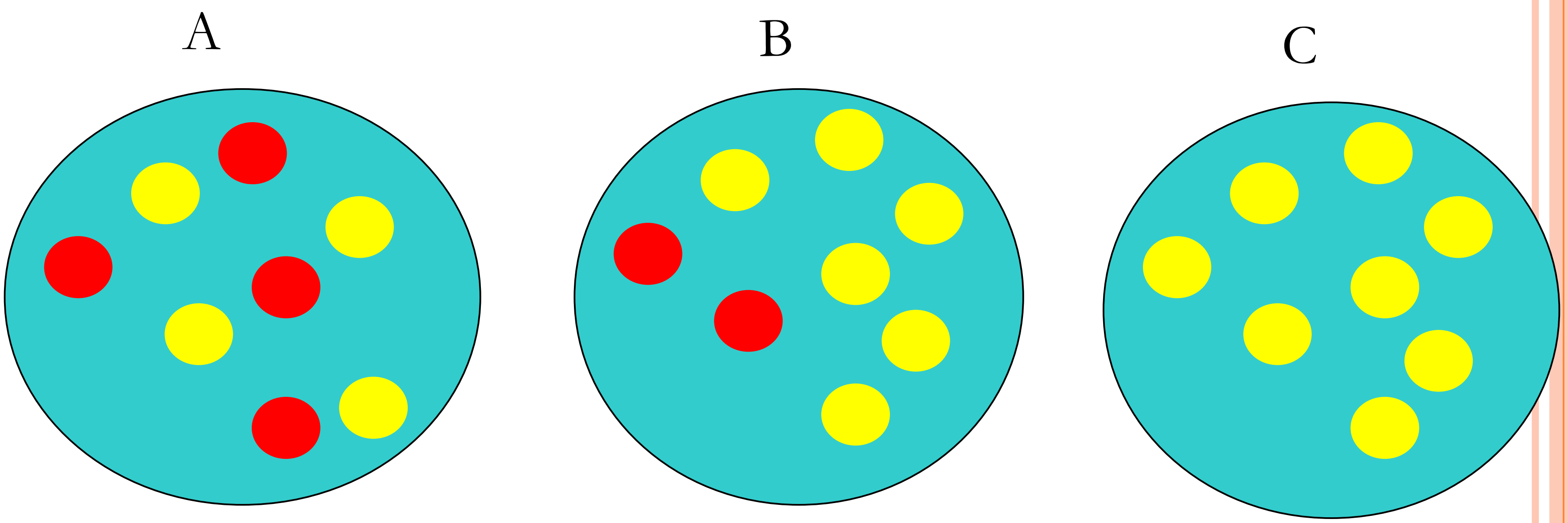
ID3 Inductive Learning

- Generate decision tree according to the training data automatically
- How?
  - Apply a question about some property for a tree node
    - e.g.: "What is the income of the customer?"
  - The training data could be partitioned based on some question
    - e.g. < 15k, 15k-35k, > 35k
  - Which question should be asked first ?

    → according to entropy!

ENTROPY: MEASURE OF UNCERTAINTY

- Degree of chaos for a distribution
- Uncertainty about the observation (picking a ball)



Entropy high ~
Uncertainty high

Entropy low ~
Uncertainty low

Entropy

- Definition(Information Theory, Shannon)
  - For distribution $p(m_i)$ of $M = \{ m_1, m_2, \dots, m_n \}$

$$I(M) \equiv \Sigma_i\, p(m_i)\log_2(1/p(m_i)) = -\Sigma_i\, p(m_i) \cdot \log_2(p(m_i))$$

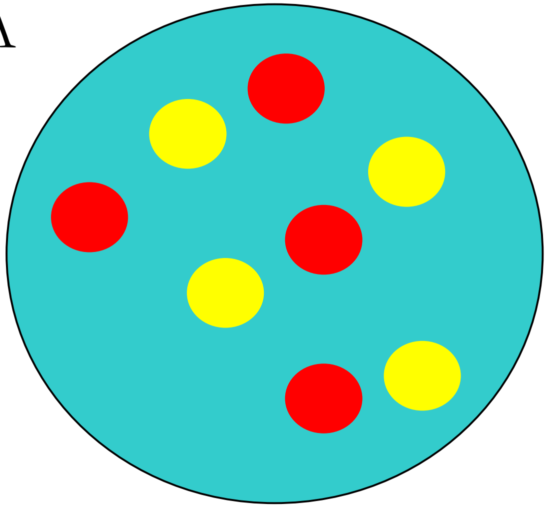
- Measure of average amount of information
  - Information for the outcome m_i: $\log_2(1/p(m_i))$
  - The lower $p(m_i)$ is, the more the information
    - e.g. "finding living creature on Mars"
  - probability=1 → no information (entropy = 0)
    - A MUST has no news-value

      e.g. "the sun rises from the east

Entropy: Measuring Degrees of Chaos / Purification

- the lower the entropy for a set is, the higher the homogeneity

A

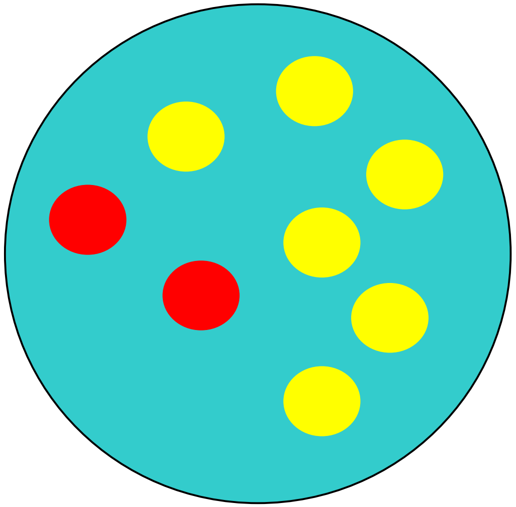


0.5

0.5

$I(A) = -0.5 \cdot \log_2(0.5) - 0.5 \cdot \log_2(0.5) =$ **1 bit**

B

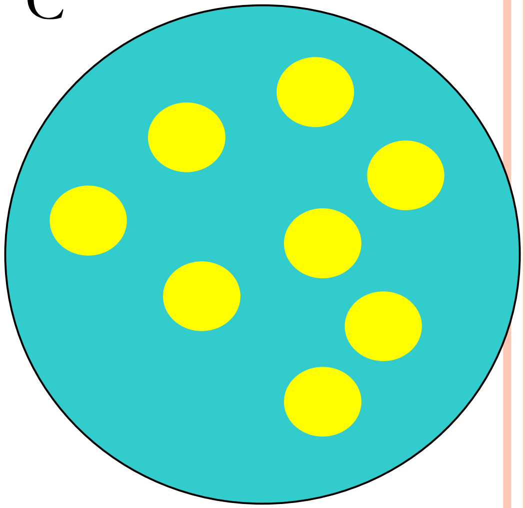


0.25

0.75

$I(B) = -0.75 \cdot \log_2(0.75) - 0.25 \cdot \log_2(0.25)$ $=$ **0.811 bit**

C



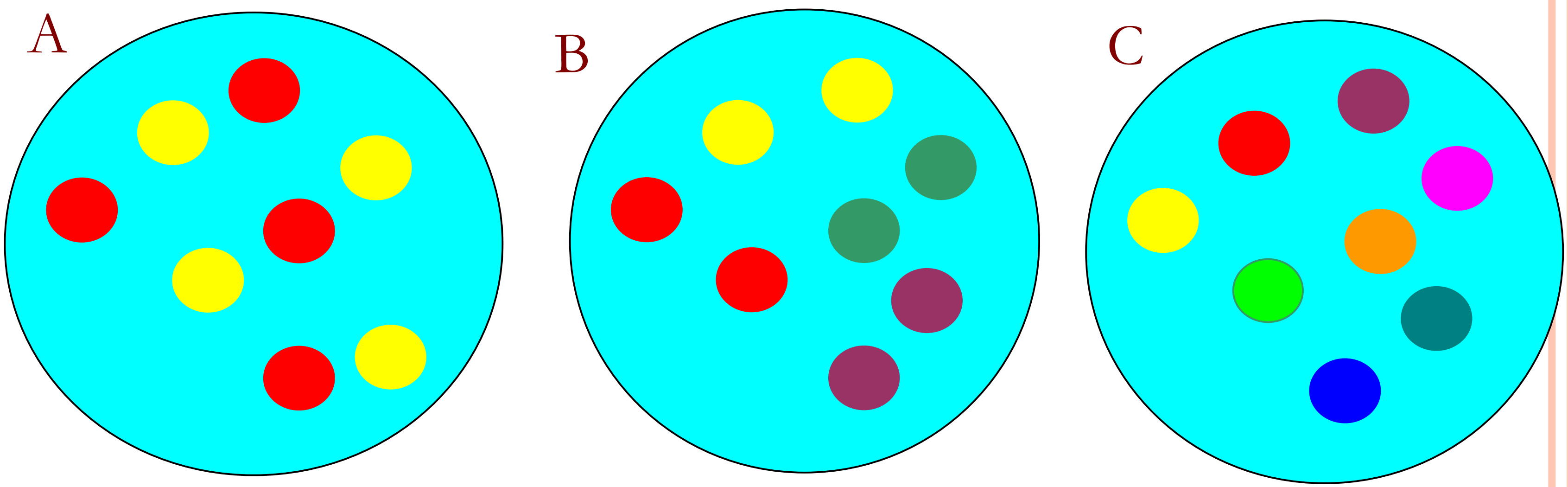
0

1

$I(C) = -1 \cdot \log_2(1)$ $=$ **0 bit**

Practice: Computation of Entropy

- The higher the entropy, the higher the chaos



$-(1/2) * \log_2(1/2) - (1/2) * \log_2(1/2)$

$= 1$ bit

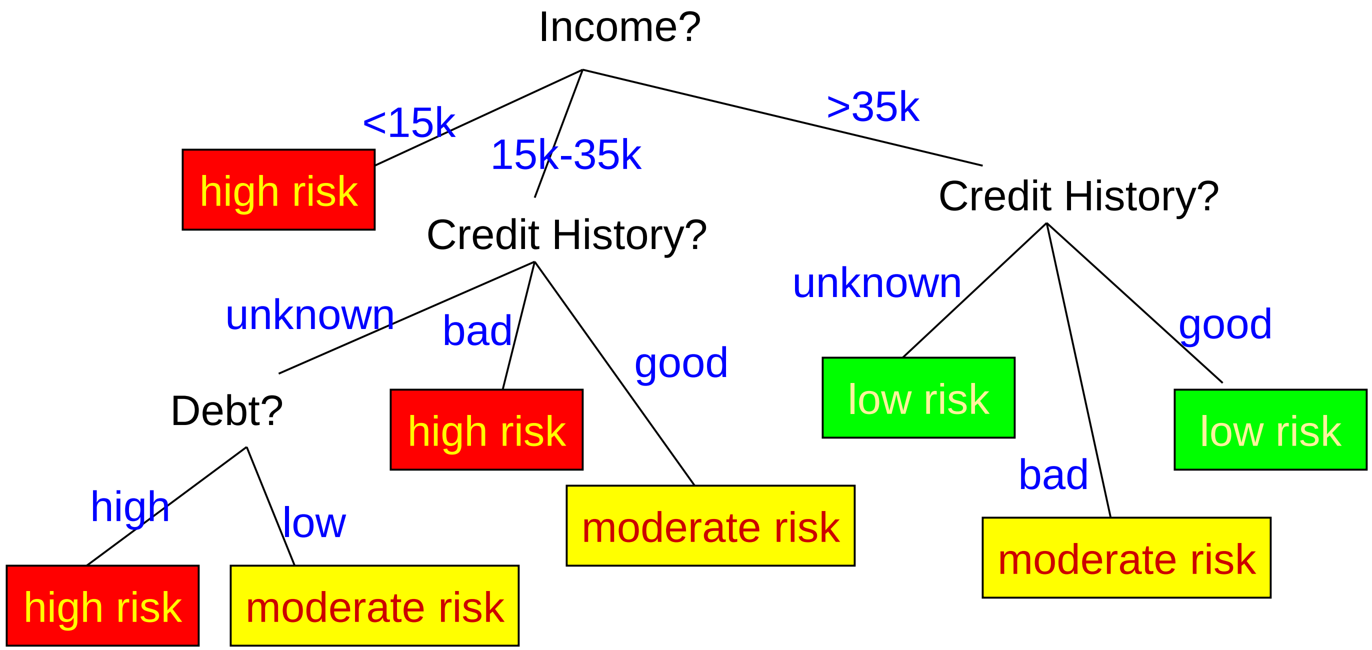
$-[(1/4) * \log_2(1/4)]*4$

$= 2$ bits

$-[(1/8) * \log_2(1/8)]*8$

$= 3$ bits

Learned Decision Tree



Questions: Income? Credit History? Debt?

Classes: high risk, moderate risk, low risk

ID3 Inductive Learning

- Produce decision tree according to the training data
- Training set C
  - C = { E1,E2, …, E14} for root node of tree
  - P(high) = 6/14, P(moderate)=3/14, P(low)=5/14
  - Entropy for set C

$$I(C) = -(6/14)\cdot\log_2(6/14) - (3/14)\cdot\log_2(3/14)$$

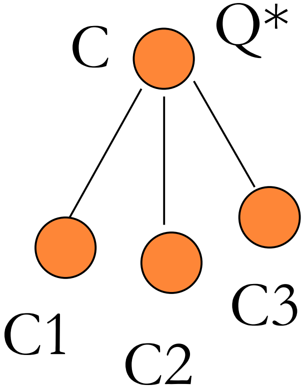
$$-(5/14)\cdot\log_2(5/14) = 1.531 \text{ bits}$$

ID3 Inductive Learning

- Goal of Learning Algorithm
  - The data set will be partitioned into smaller sets by applying a question.
  - In the leaf nodes, all data are of the same output category, and the partition is stopped.
    - Entropy is 0 for leaf nodes
  - Good decision tree → reduce the entropy to 0 more quickly

ID3 Inductive Learning

- For current node, a question (property) Q in a set of questions is asked (e.g. income? Collateral? Debt? 或Credit history?)
- The question Q can partition the set C into smaller sets (C → C1, C2, …, C_n)
  - E.g. income? → <15k, 15k-35k, >35k
- Which question is selected among questions?
  - Q*: maximum entropy reduction

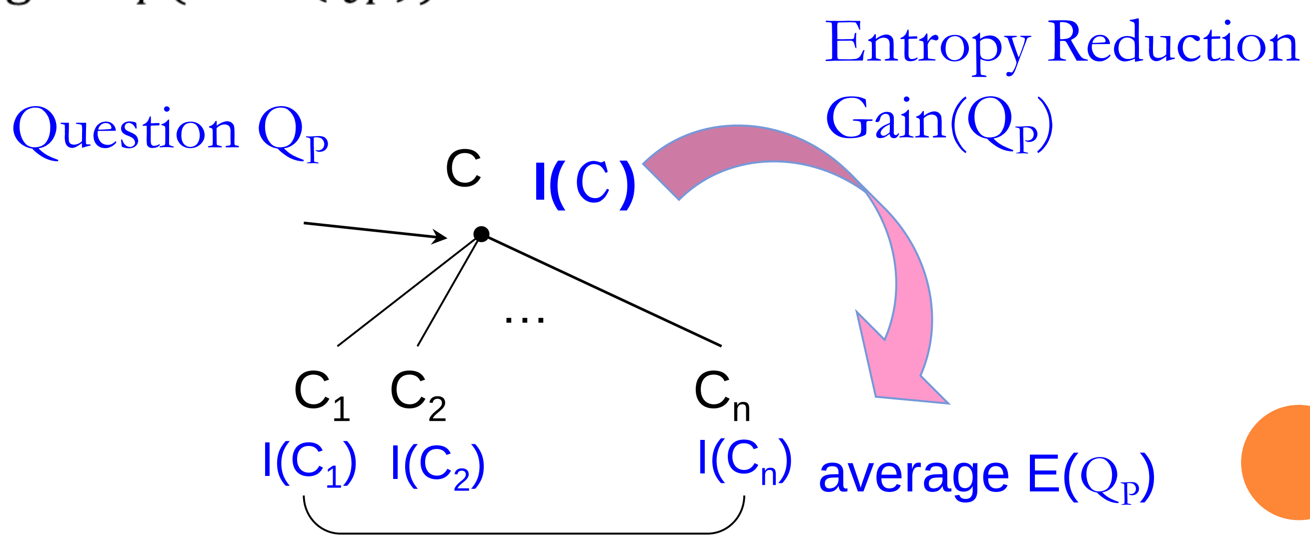


- Once Q* is determined, apply ID3 for children (C_i's)
  - If set C_i has the entropy of 0, this set is not further processed.
  - Otherwise, call ID3 recursively for C_i with the rest of the questions.

Maximum Entropy Reduction

- If S is divided into subsets $C_1 \sim C_n$ through applying the question Q_P for property P
  - $E(Q_P) = \Sigma_1^n (|C_i|/|C|) \cdot I(C_i)$ average entropy
  - $Gain(Q_P) = I(C) - E(Q_P)$ entropy reduction
  - Choose P with maximum Gain among all properties

    $Q^* = \text{argmax}_P(Gain(Q_P))$



Maximum Entropy Reduction

- For root node, I(C) = 1.531 bits
- Apply Q="income?" at root node, C can be partitioned
  - C1= {E1,E4,E7,E11} C2={E2,E3,E12,E14}
    C3= {E5,E6,D8,E9,E10,E13}
  - E(income) = (4/14)*I(C1)+(4/14) * I(C2)+(6/14)*I(C3)
    = (4/14)*0 +(4/14)*1 +(6/14)*0.650= 0.564 bits
  - Entropy reduction : *Gain(income)*=1.531-0.564 = 0.967 bits
- Similarly, other gains can be obtained
  - *Gain(credit history)*= 1.531-1.265 = 0.266
  - *Gain(debt)*=1.531-0.95 = 0.581
  - *Gain(collateral)* = 1.531 - 0.775 = 0.756
- Optimal question Q*="income?" (highest gain)

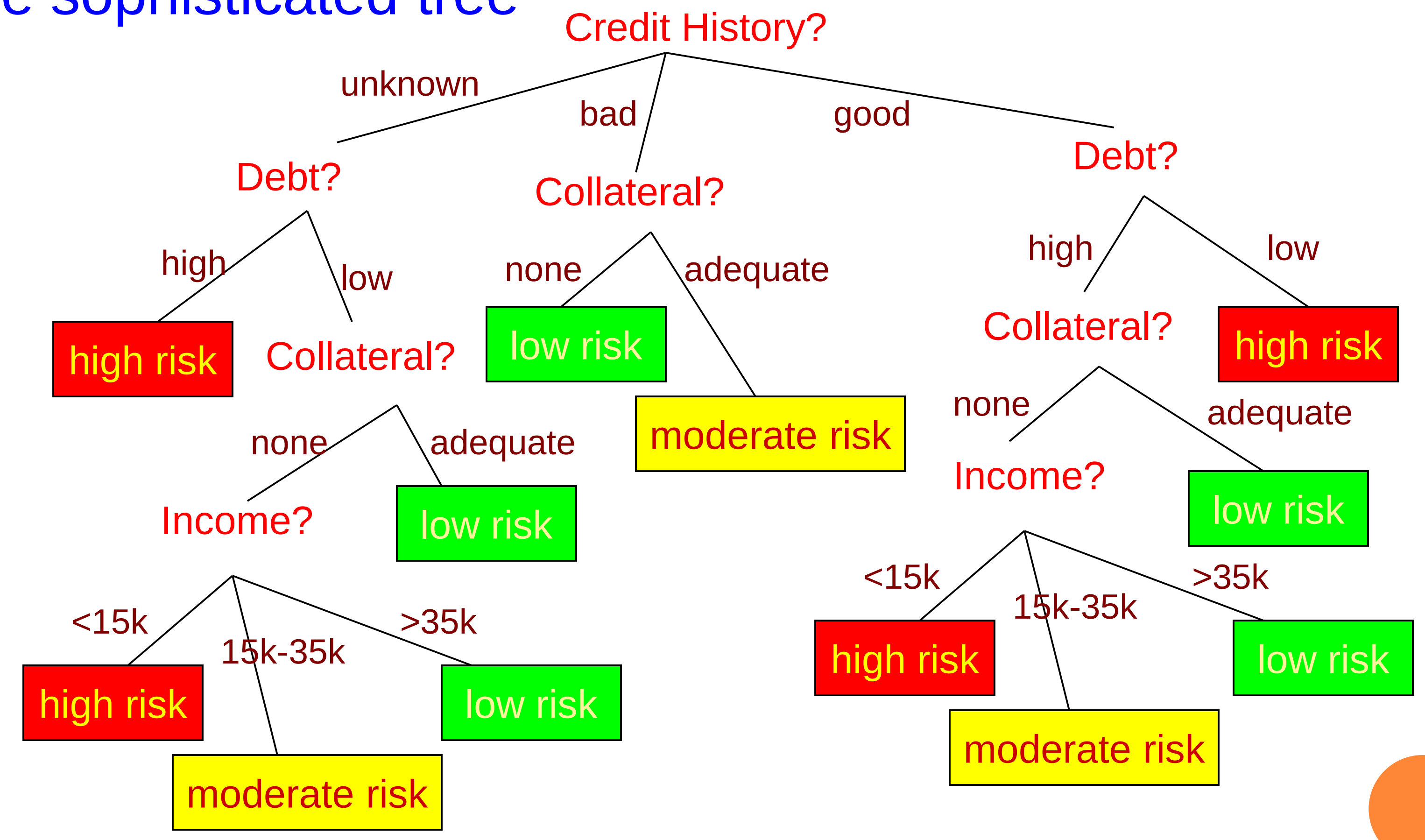
Inductive Learning Algorithm

- Training data $\{ d_i \} = \{ (\underline{x}_i, y_i) \}$, $\underline{x}_i$ properties, y_i class
- A set of questions $Q = \{ Q_P \}$ based on the properties

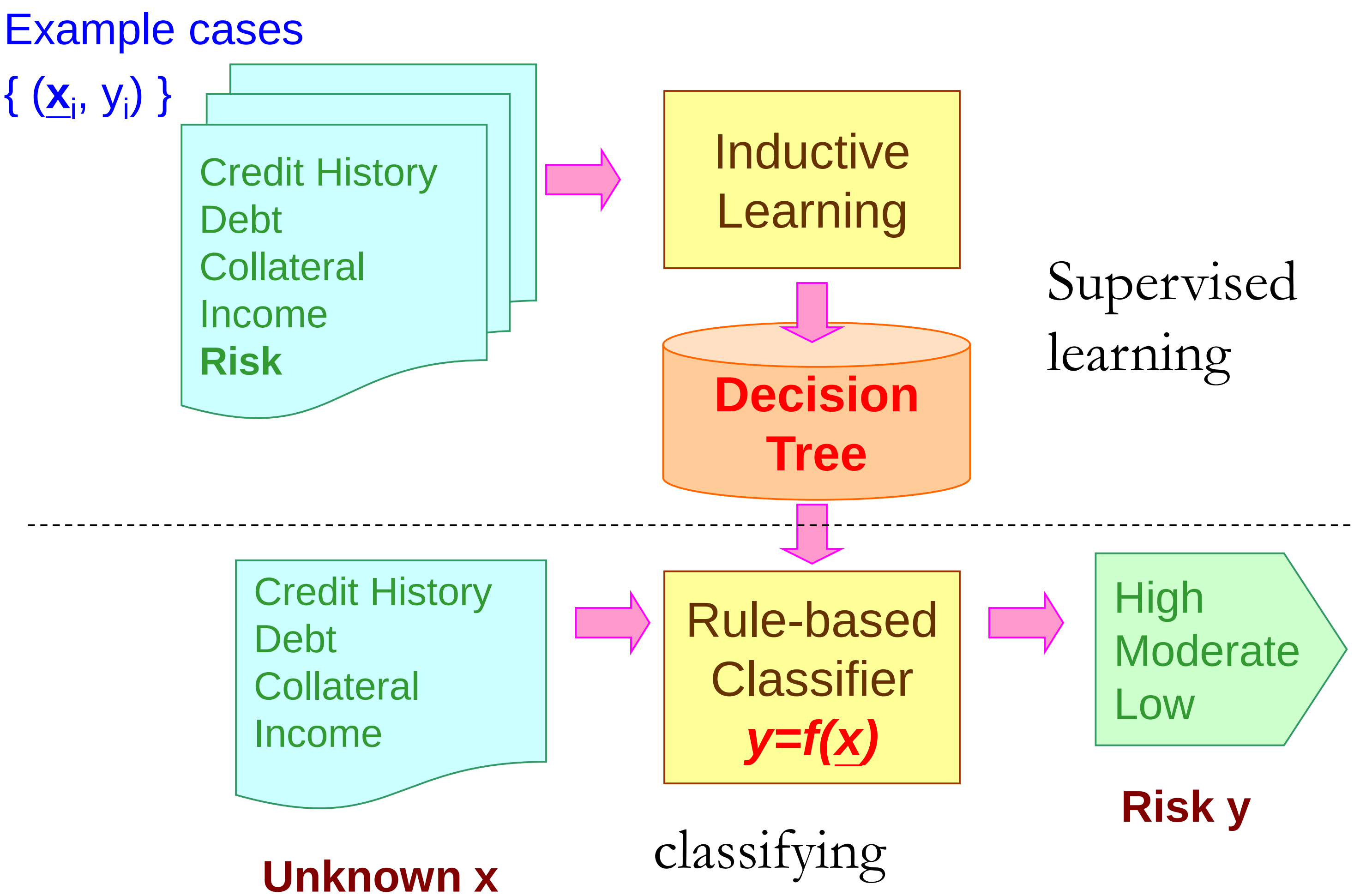
1. Initially, current set C includes all data $\{ d_i \}$
2. For each d_i in C, first calculate the entropy of C, $I(C)$. If the entropy is small, stop spanning from current set.
3. For each question Q_P, generate sub-sets of C by applying question Q_P. Find the entropies for these sets respectively, and calculate the average entropy $E(Q_P)$.
4. Choose question Q^* with maximum entropy reduction as question of current set C, and apply the partition of C by Q^* to obtain the sub-sets.
5. Perform step 2-4 for the sub-sets respectively.

Decision Tree Is Not Unique

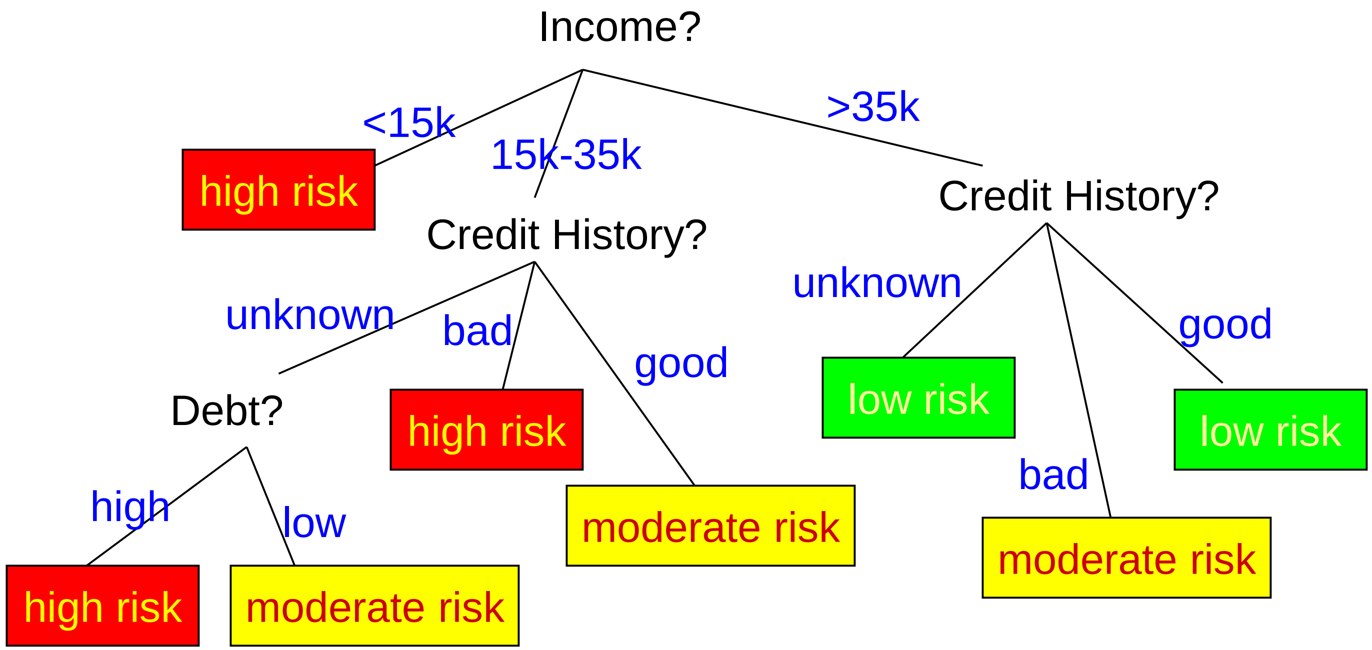
More sophisticated tree



System Architecture



Learned Decision Tree



Questions: Income? Credit History? Debt?

Classes: high risk, moderate risk, low risk

Discussions

- A simple tree is obtained
  - Minimum depth tree
  - Greedy algorithm (select optimal question locally)
- There may be irrelevant properties
  - e.g. “Collateral” is not an relevant property and is ignored automatically because it is ineffective in entropy reduction
  - Some properties may be noises that interfere the classification if the training data is insufficient

DISCUSSIONS

- There may be classification errors
  - There might be noises (inconsistencies) for a large amount of training data (e.g. different outputs for the same input).
  - The learning is stopped if entropy is lower than a threshold (need not be 0)
  - Use majority vote to make final decision

Discussions

- The property need to be categorical data for ID3
  - If property P is numeric, binary test can be achieved by applying a threshold Z (such as the *income* in previous example)

    P? → $P \leqq Z$ or $P > Z$
  - Optimal Z for property P can be obtained (maximum entropy reduction for all possible Z's)
- A question may combine several properties
  - Question: (income < 15k & debt == 'none')?
- Entropy is not the only indication of purification
  - Could be used for clustering (unsupervised learning)
  - Use distance or similarity as indicator

BN and CART

- BN
  - The information for classification is stored in the probabilities
  - Decision function based on *numerical computation*
    - Decision is made from probabilistic point of view
  - Statistical dependency among variables may be assumed to simplify the computation of probabilities.
- CART
  - The information for classification is stored in the decision rules that might be *meaningful* for humans
  - Decision function based on *classification rules*
    - Explicit rules associated with the decision can be given.
    - Can give a reason & deduct rules
    - Automatic learning for *decision rules*
  - Statistical dependency among variables can be learned.

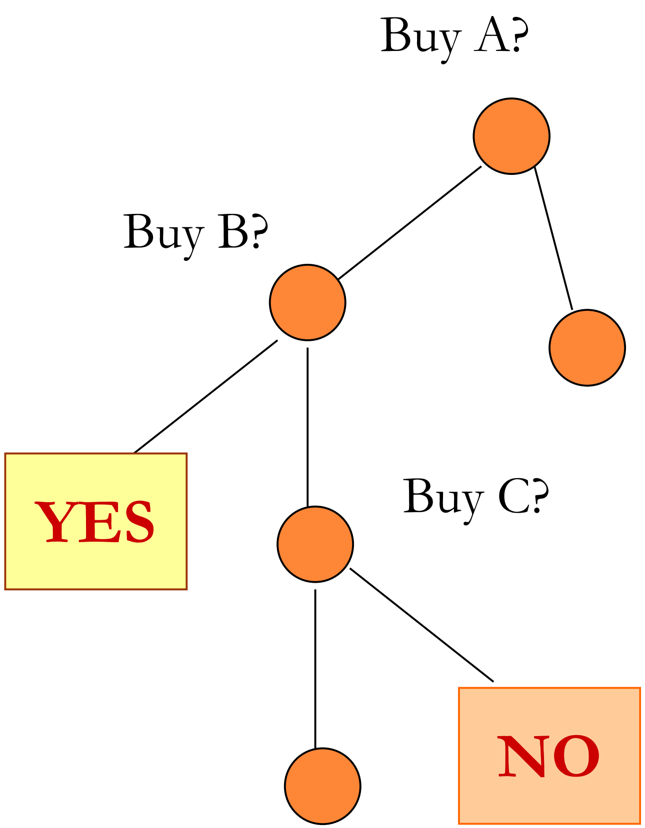
Discussions

- In ID3, there is a strong bias in favor of properties with many outcomes (more branches)
  - It is not fair to compare directly the properties with different number of outcomes
  - Those properties with more outcomes tends to have lower average entropy (and thus have higher chance to defeat others) because of finer partition.
- Normalization on entropy
  - Split-info(P)=$-\Sigma_i((|C_i|/|C|)\log_2(|C_i|/|C|))$
  - Gain-ratio(P) = Gain(P) / Split-info(P)
  - Maximum gain ratio instead of maximum gain
- Tools: Weka, R, Matlab, Python

Application Domains of CART

- Classification
- Clustering
  - Example: clustering models (such as GMM or HMM) according to their attributes
  - Similarity/distance of models can be used as the indicator of closeness
- Regression
  - Input are categorical(discrete) features while output is continuous variable
  - Example: prediction of speed based on attributes (date, time of day…)
  - Variance of speed as indicator of closeness

Potential Applications



Marketing :

Who will buy product F?

→ Those who buy A & B!

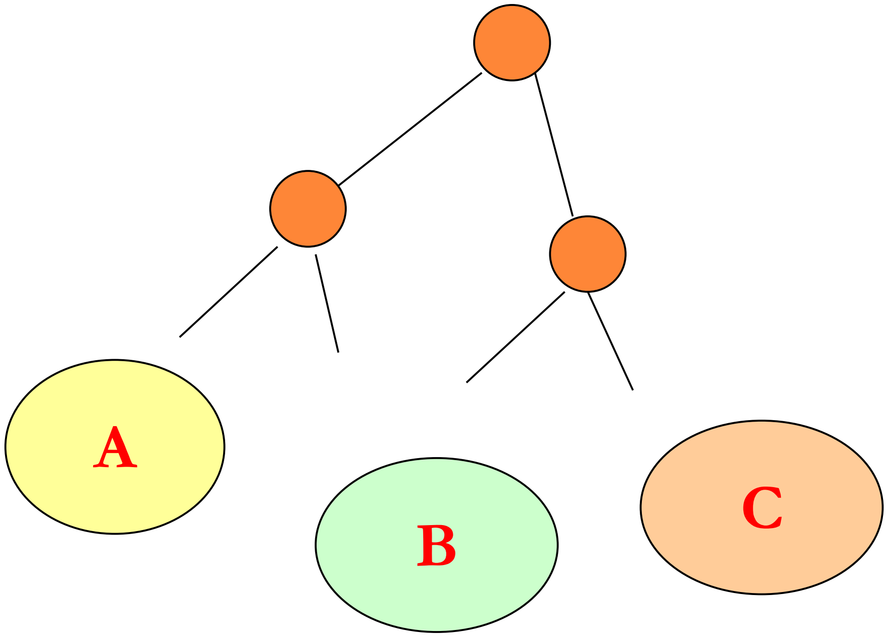
Transactions

$\underline{x}$ (Buy A … …), y (Buy F)

	Buy A	Buy B	…	Buy F
1	YES	YES	…	YES
2	YES	NO	…	NO
3	NO	YES	…	YES
…	…	…	…	…

Potential Applications

- Classifying major customers with different ranks (e.g. according to sales/profits/clicks)



APPLICATIONS

- Suitable for classifying *categorical data*
- If the input features contain numerical data, they need first be converted into categorical data
  - Regions of value
  - Vector quantization
- The knowledge learned is stored in the *decision tree* (or *decision rules*)
  - e.g. if(income > 100k && look == A) accept = true

References

- *Data Mining: concepts, models, methods, and algorithms*

  Mehmed Kantardzic

  Wiley Inter-Science