IS422P - DATA MINING CLASSIFICATION

(PART 1)

AMIRA REZK

INFORMATION SYSTEM DEPARTMENT

AGENDA

The Basics
- What is Classification?
- General Approach

Decision Tree Induction
- The Algorithm
- Attribute Selection Measures
- Tree Pruning
- Extracting Rules from Decision Trees

Bayes Classification
- Bayes' Theorem
- Naïve Bayesian Classification

Lazy Learners
- K-Nearest Neighbor Classifiers

Regression analysis
- Linear regression

Model Evaluation
- Metrics for Evaluating Classifiers Performance
- Cross-Validation
- Bootstrap

Improving Classification Accuracy
- Bagging
- Boost and AdaBoost

THE BASICS
WHAT IS CLASSIFICATION

- Motivation: Prediction
  - Is a bank loan applicant "safe" or "risky"?
  - Which treatment is better for patient, "treatmentX" or "treatmentY"?
- Classification is a data analysis task where a model is constructed to predict class labels (categories)

THE BASICS
GENERAL APPROACH

- A two-step process:
- Learning (training) step → construct classification model
  - Build classifier for a predetermined set of classes
  - Learn from a training dataset (data tuples + their associated classes) → Supervised Learning
- Classification step → model is used to predict class labels for given data (test set)

THE BASICS
GENERAL APPROACH

Attribute vector

Class label

Training data

name	*age*	*income*	*loan_decision*
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

Classification Algorithm

Classification rules

IF *age* = *youth* THEN *loan_decision* = *risky*
IF *income* = *high* THEN *loan_decision* = *safe*
IF *age* = *middle_aged* AND *income* = *low*
THEN *loan_decision* = *risky*
...

THE BASICS
GENERAL APPROACH

Classification rules

IF $age = youth$ THEN $loan\_decision = risky$
IF $income = high$ THEN $loan\_decision = safe$
IF $age = middle\_aged$ AND $income = low$
THEN $loan\_decision = risky$
...

Estimate classifier accuracy (to avoid overfitting)

Test data

name	*age*	*income*	*loan_decision*
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

% test set tuples correctly classified

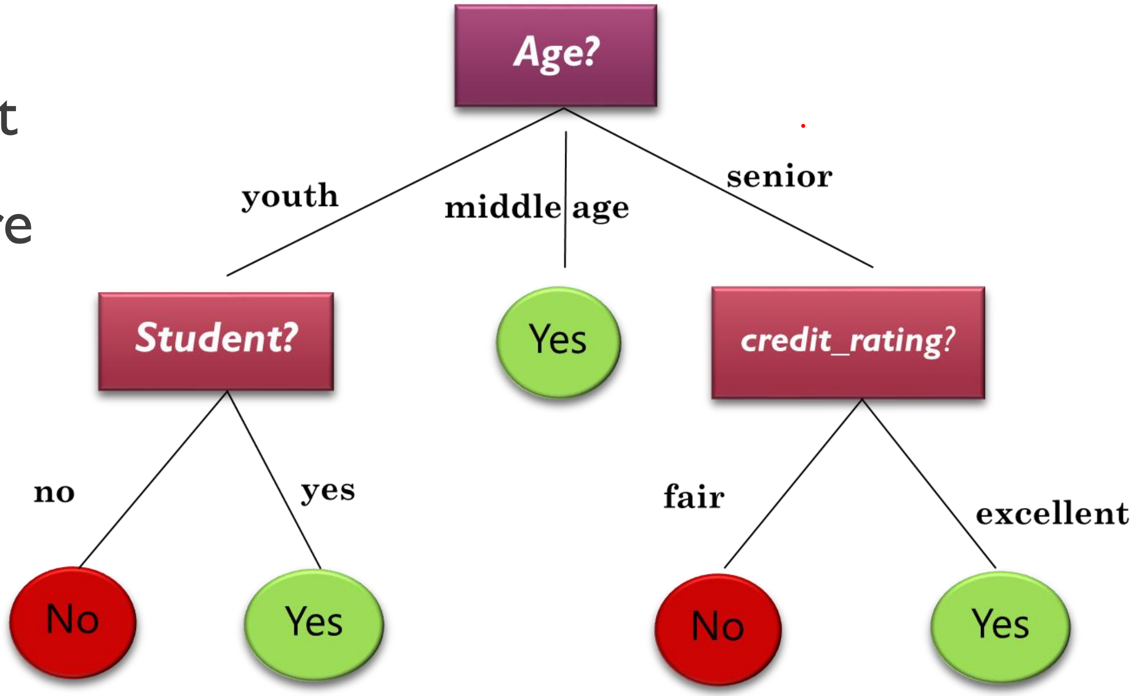
Predict classification of new data

(Mohammed, youth, medium)
Loan decision?

Risky

DECISION TREE INDUCTION

- Learning of decision trees from training dataset
- Decision tree → A flowchart-like tree structure
  - Internal node → a test on an attribute
  - Branch → a test outcome
  - Leaf node → a class label
- Constructed tree can be binary or otherwise

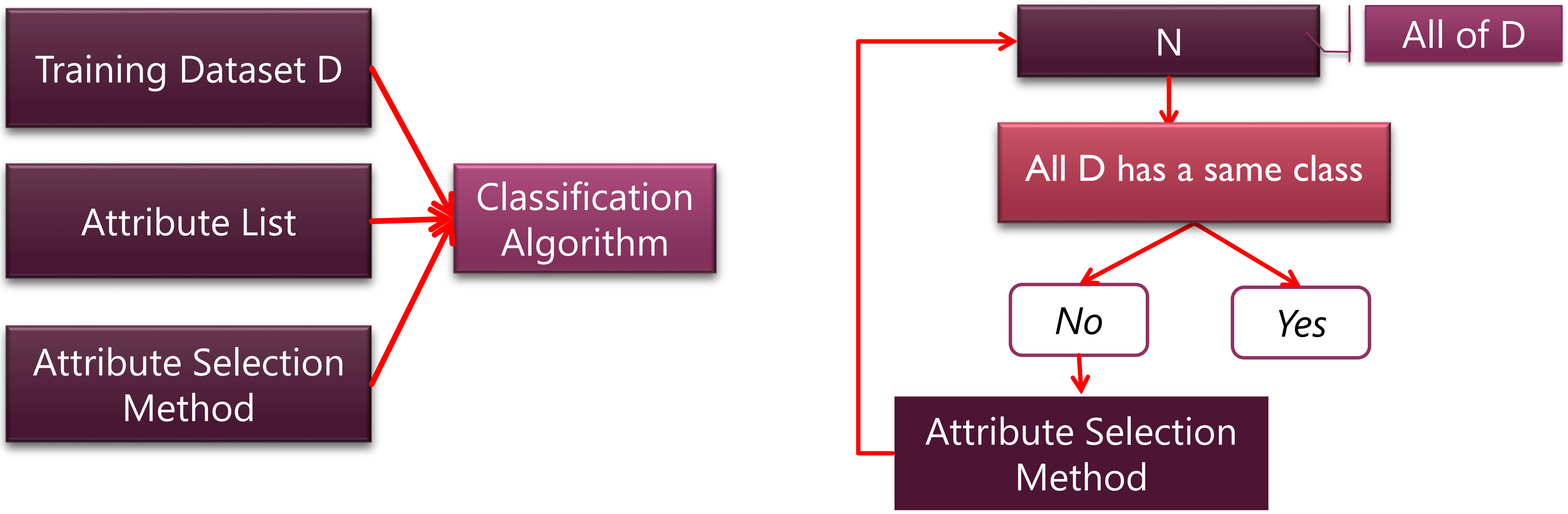


DECISION TREE INDUCTION

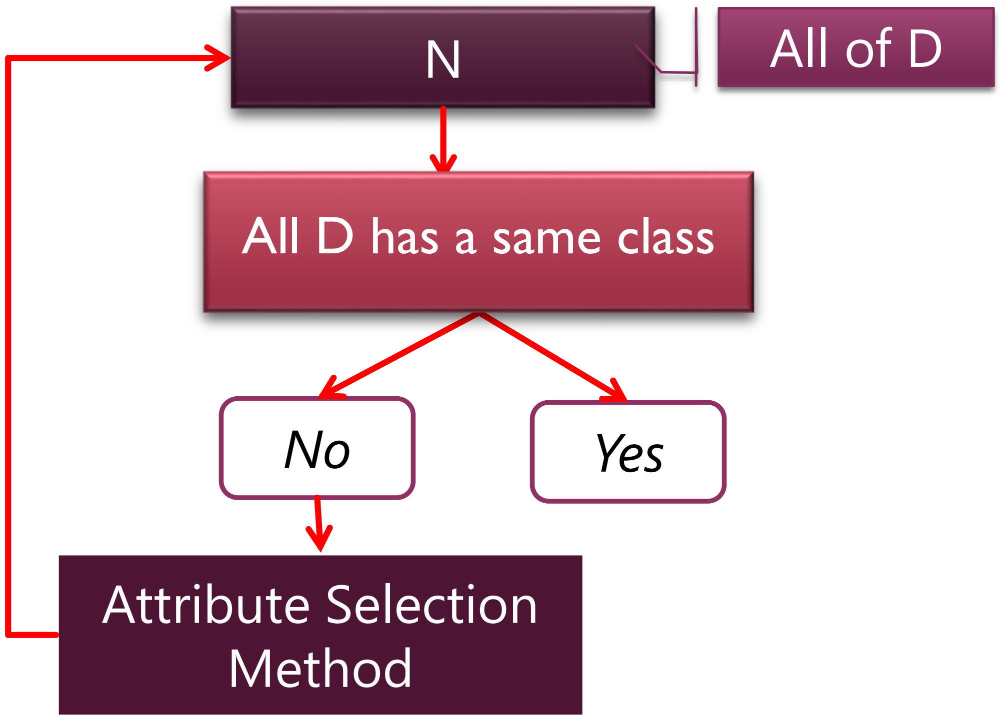
Benefits

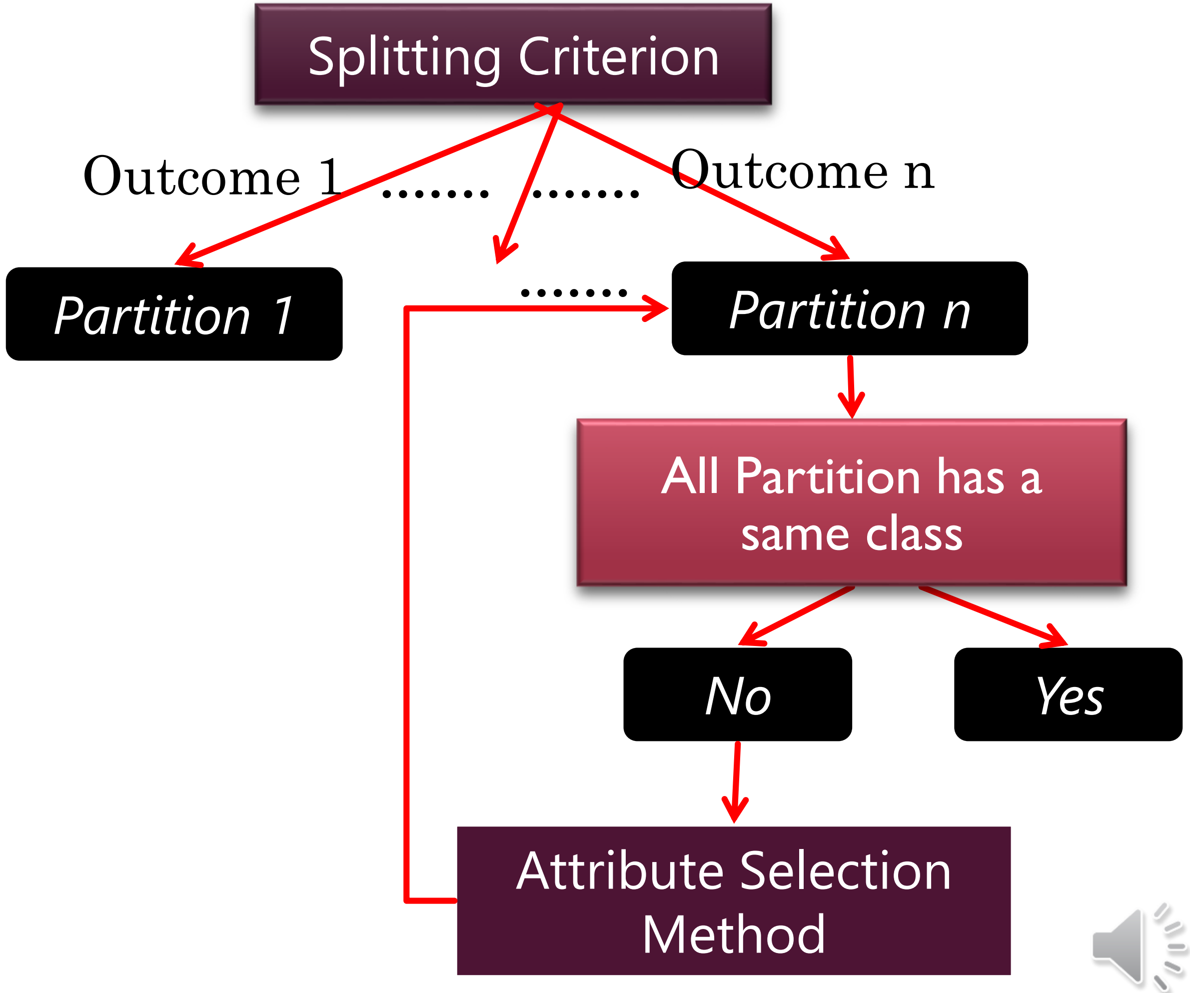
- No domain knowledge required
- No parameter setting
- Can handle multidimensional data
- Easy-to-understand representation
- Simple and fast

DECISION TREE INDUCTION
THE ALGORITHM

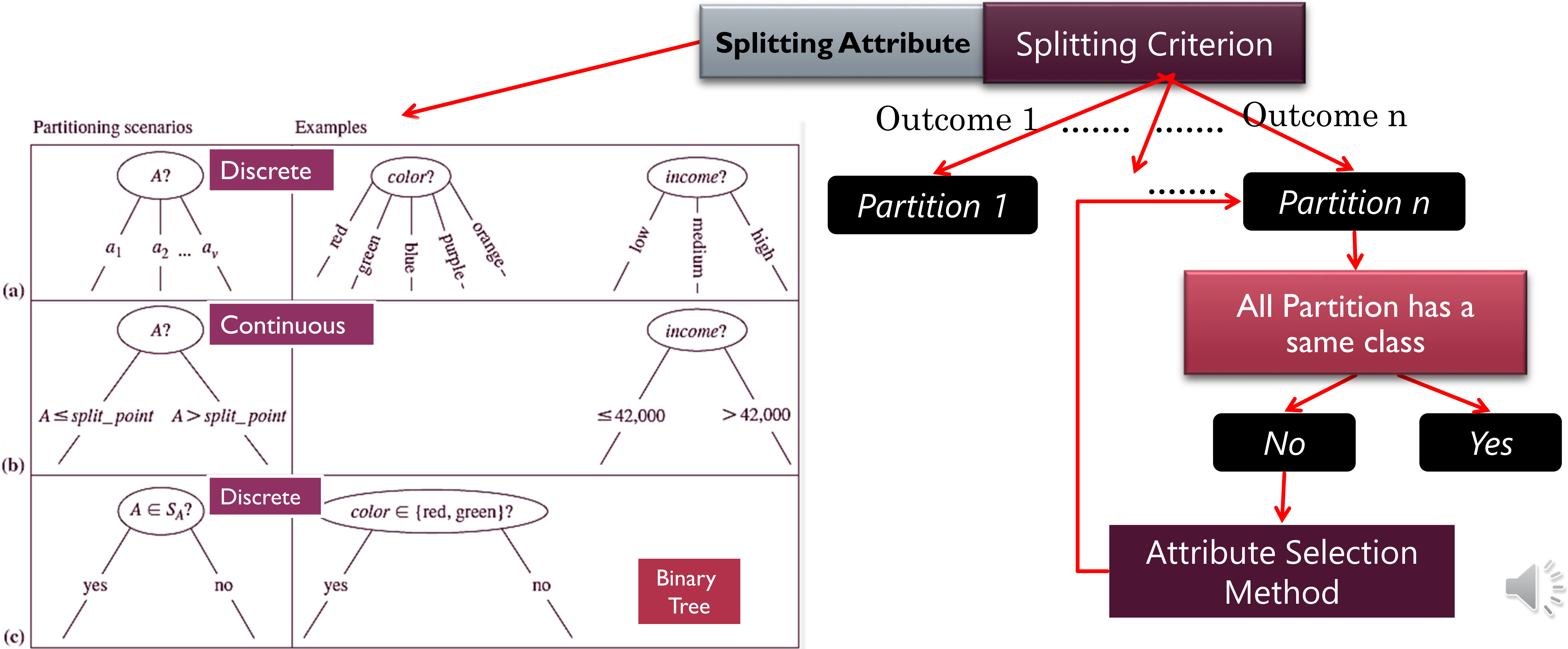


DECISION TREE INDUCTION
THE ALGORITHM





DECISION TREE INDUCTION
THE ALGORITHM



DECISION TREE INDUCTION
THE ALGORITHM

- **Splitting Criterion is a test:**
  - Which attribute to test at node N → What is the "best" way to partition D into mutually exclusive classes
  - which (and how many) branches to grow from node N to represent the test outcomes
- Resulting partitions at each branch should be as "pure" as possible
  - A partition is "pure" if all its tuples belong to the same class
- When attribute is chosen to split training data set, it's removed from attribute list

DECISION TREE INDUCTION
THE ALGORITHM

- **Terminating conditions**
  - All the tuples in D (represented at node N) belong to the same class
  - There are no remaining attributes on which the tuples may be further partitioned
    - majority voting is employed → convert node into a leaf and label it with the most common class in data partition
  - There are no tuples for a given branch
    - a leaf is created with the majority class in data partition

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

- **Attribute selection measure** → a heuristic for selecting the splitting criterion that "best" splits a given data partition into smaller mutually exclusive classes
- Attributes are ranked according to a measure
  - attribute having the best score is chosen as the splitting attribute
  - split-point for continuous attributes
  - splitting subset for discrete attributes with binary trees
- Measures: Information Gain, Gain Ratio, Gini Index

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

Information Gain

- Based on *Shannon's information theory*
- Goal is to **minimize the expected number of tests needed to classify a tuple**
  - guarantee that a <u>simple tree</u> is found
- Attribute with the *<u>highest information gain</u>* is chosen as the splitting attribute
  - minimizes information needed to classify tuples in resulting partitions
  - reflects least "impurity" in resulting partitions

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

- Given m class labels (C_i, i =1 to m)
- Expected Information needed to classify a tuple in D
- Info (D)= **entropy** = $-\sum_{i=1}^{m} p_i \log_2(pi)$
- p_i → probability that an arbitrary tuple in D belong to class Ci

$$p_i = \frac{|C_{i,D}|}{|D|}$$

- $C_{i,D}$ → set of tuples having class label C_i in partition D

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

- How much more Information would be needed after Partitioning to arrive at a "pure" classification"
  - Expected information required to classify a tuple from D based on the partitioning by attribute A:
  - $info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{D} \times info(Dj)$
  - The smaller the expected information still required, the greater the purity of the partitions

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

- **Information gain** is the different between the original information required (based on proportion of classes) and the new requirement (after partitioning on A)
- *Gain(A)= info(D) – info$_A$(D)*
- *Gain(A) tells you how much would be gained by branching on A*
  - *Expected reduction in the information requirement caused by knowing the values of A*
  - *Attributes A with the highest Gain(A) is chosen as the splitting attribute at node N*

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

C1 (Senior) = 80 , C2(Junior) =120

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

$$= -\frac{80}{200}\log_2\frac{80}{200} - \frac{120}{200}\log_2\frac{120}{200} = 0.97$$

info $_A$ (D) = $-\sum_{j=1}^{n}\frac{D_j}{D}\ info\ (D_j)$

Department: Sales = 100, system= 50, marketing= 30, secretary = 20

Info $_{\text{department}}$ =

$$\frac{100}{200}\left(-\frac{30}{100}\log\frac{30}{100} - \frac{70}{100}\log\frac{70}{100}\right) + \frac{50}{200}\left(-\frac{30}{50}\log\frac{30}{50} - \frac{20}{50}\log\frac{20}{50}\right)$$

$$+\ \frac{30}{200}\left(-\frac{10}{30}\log\frac{10}{30} - \frac{20}{30}\log\frac{20}{30}\right) + \frac{20}{200}\left(-\frac{10}{20}\log\frac{10}{20} - \frac{10}{20}\log\frac{10}{20}\right)$$

=0.92

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

C1 (Senior) = 80 , C2(Junior) =120

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

$$= -\frac{80}{200}\log_2\frac{80}{200} - \frac{120}{200}\log_2\frac{120}{200} = 0.97$$

info $_A$ **(D) =** $-\sum_{j=1}^{n}\frac{D_j}{D}\, info\,(D_j)$

Department: Sales = 100, system= 50, marketing= 30, secretary = 20

Info $_{\text{department}}$ =

$$\frac{100}{200}\left(-\frac{30}{100}\log\frac{30}{100}-\frac{70}{100}\log\frac{70}{100}\right)+\frac{50}{200}\left(\boxed{-\frac{30}{50}\log\frac{30}{50}}\boxed{-\frac{20}{50}\log\frac{20}{50}}\right)$$

$$+\frac{30}{200}\left(-\frac{10}{30}\log\frac{10}{30}-\frac{20}{30}\log\frac{20}{30}\right)+\frac{20}{200}\left(-\frac{10}{20}\log\frac{10}{20}-\frac{10}{20}\log\frac{10}{20}\right)$$

=0.92

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

C1 (Senior) = 80 , C2(Junior) =120

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

$= -\frac{80}{200}\log_2\frac{80}{200} - \frac{120}{200}\log_2\frac{120}{200}$ = 0.97

info $_A$ (D) = $-\sum_{j=1}^{n}\frac{D_j}{D}\ info\,(D_j)$

Department: Sales = 100, system= 50, marketing= 30, secretary = 20

Info $_{department}$ =

$$\frac{100}{200}\left(-\frac{30}{100}\log\frac{30}{100}-\frac{70}{100}\log\frac{70}{100}\right)+\frac{50}{200}\left(-\frac{30}{50}\log\frac{30}{50}-\frac{20}{50}\log\frac{20}{50}\right)$$

$$+\ \frac{30}{200}\left(\boxed{-\frac{10}{30}\log\frac{10}{30}}\boxed{-\frac{20}{30}\log\frac{20}{30}}\right)+\frac{20}{200}\left(-\frac{10}{20}\log\frac{10}{20}-\frac{10}{20}\log\frac{10}{20}\right)$$

=0.92

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

C1 (Senior) = 80 , C2(Junior) =120

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

$$= -\frac{80}{200}\log_2\frac{80}{200} - \frac{120}{200}\log_2\frac{120}{200} = 0.97$$

info $_A$ (D) = $-\sum_{j=1}^{n} \frac{D_j}{D}\ info\,(D_j)$

Department: Sales = 100, system= 50, marketing= 30, secretary = 20

Info $_{\text{department}}$ =

$$\frac{100}{200}\left(-\frac{30}{100}\log\frac{30}{100}-\frac{70}{100}\log\frac{70}{100}\right)+\frac{50}{200}\left(-\frac{30}{50}\log\frac{30}{50}-\frac{20}{50}\log\frac{20}{50}\right)$$

$$+\frac{30}{200}\left(-\frac{10}{30}\log\frac{10}{30}-\frac{20}{30}\log\frac{20}{30}\right)+\frac{20}{200}\left(\boxed{-\frac{10}{20}\log\frac{10}{20}}\boxed{-\frac{10}{20}\log\frac{10}{20}}\right)$$

=0.92

Gain $_{\text{department}}$ = 0.97 - 0.92 = 0.05

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

C1 (Senior) = 80 , C2(Junior) =120

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

$= -\frac{80}{200}\log_2\frac{80}{200} - \frac{120}{200}\log_2\frac{120}{200} = 0.97$

info $_A$ (D) = $-\sum_{j=1}^{n} \frac{D_j}{D}\ info\,(D_j)$

Department: Sales = 100, system= 50, marketing= 30, secretary = 20

Info $_{\text{department}}$ =

$\frac{100}{200}\left(-\frac{30}{100}\log\frac{30}{100}-\frac{70}{100}\log\frac{70}{100}\right)+\frac{50}{200}\left(-\frac{30}{50}\log\frac{30}{50}-\frac{20}{50}\log\frac{20}{50}\right)$

$+\ \frac{30}{200}\left(-\frac{10}{30}\log\frac{10}{30}-\frac{20}{30}\log\frac{20}{30}\right)+\frac{20}{200}\left(-\frac{10}{20}\log\frac{10}{20}-\frac{10}{20}\log\frac{10}{20}\right)$

=0.92

Gain $_{\text{department}}$ = 0.97 - 0.92 = 0.05 bits

Gain $_{\text{Age}}$ = 0.97 -0.55 = 0.42 bits

Gain $_{\text{Salary}}$ = 0.97 – 0.45 = 0.52 bits

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

Salary?

Low

department	age	status
sales	youth	junior
sales	Middle aged	junior
secretary	youth	junior

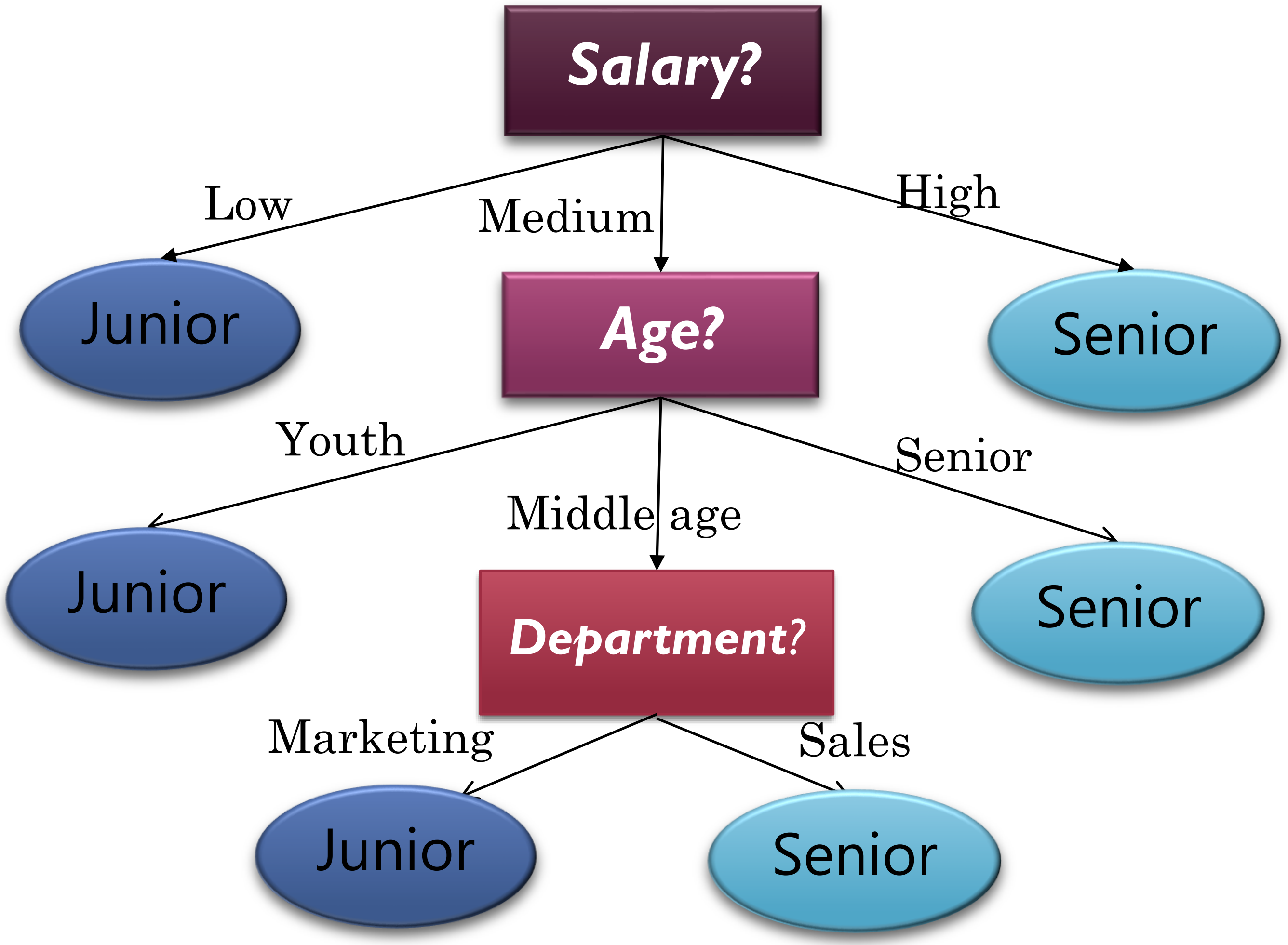
medium

department	age	status
sales	Middle aged	senior
systems	youth	junior
marketing	senior	senior
marketing	Middle aged	junior
secretary	senior	senior

high

department	age	status
systems	Middle aged	senior
systems	senior	senior

DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES



DECISION TREE INDUCTION
ATTRIBUTE SELECTION MEASURES

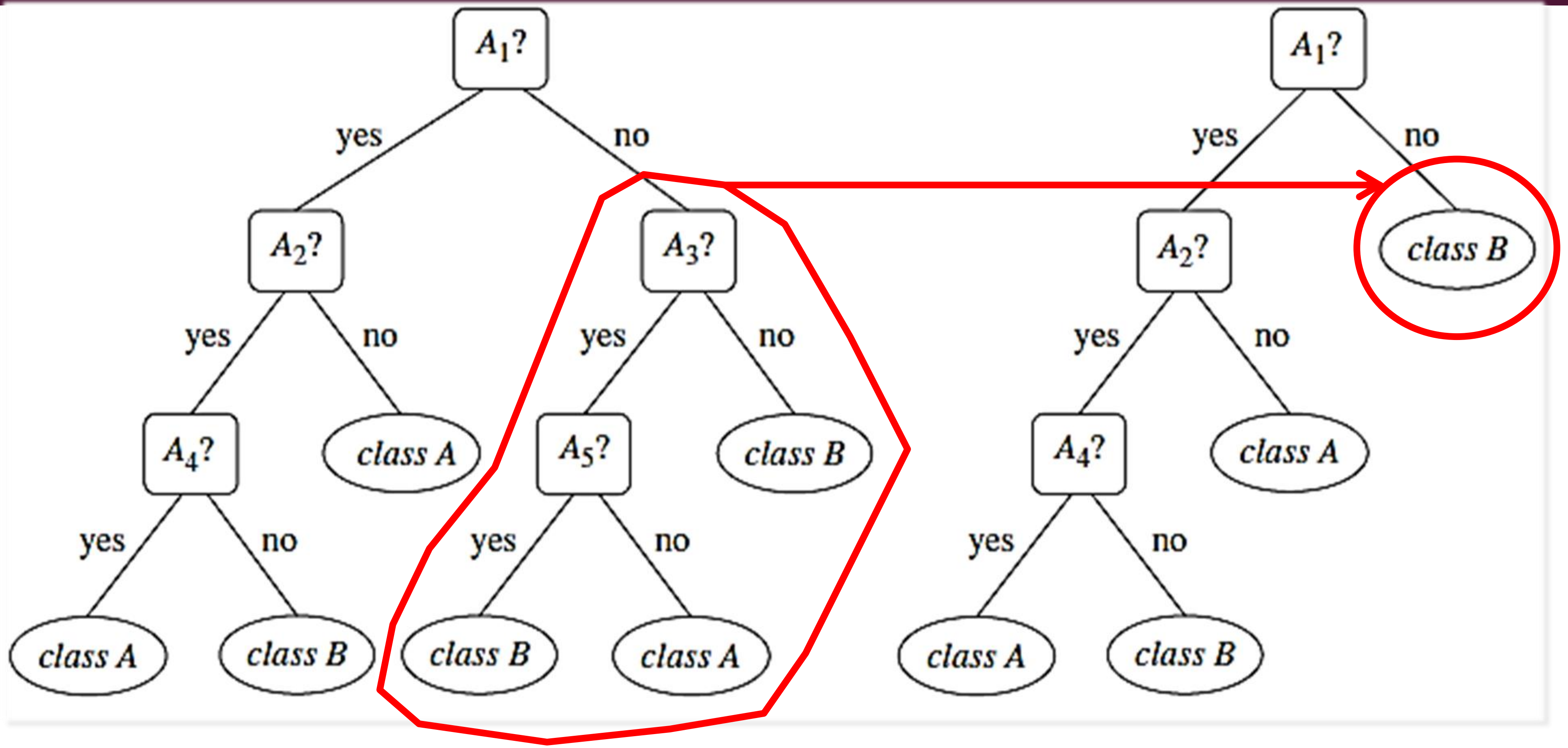
Information gain for **continuous attributes**

1. Sort values in increasing order
2. Each *midpoint* between two adjacent values can serve as *split-point*
3. Split-point between two values v_i *and* $v_{i+1} = \frac{v_i+v_{i+1}}{2}$
4. For each split-point, evaluate $info_A(D)$ with the number of partitions $= 2$ ($A \leq split\text{-}point$ & $A > split\text{-}point$)

DECISION TREE INDUCTION
TREE PRUNING

- Data may be *overfitted* to dataset anomalies and outliers
- **Pruning** removes the least reliable branches
  - DT becomes less complex
- Prepruning → statistically assess the *goodness of a split* before it takes place
  - hard to choose *thresholds* for statistical significance
- Postpruning → remove sub-trees from already constructed trees
  1. remove sub-tree branches and replace with leaf node
  2. leaf is labeled with most frequent class in sub-tree

DECISION TREE INDUCTION
TREE PRUNING



DECISION TREE INDUCTION

RULE EXTRACTION FROM A DECISION TREE – WHAT ARE RULES?

department	age	salary	status	count
sales	Middle aged	medium	senior	30
sales	youth	low	junior	30
sales	Middle aged	low	junior	40
systems	youth	medium	junior	20
systems	Middle aged	high	senior	20
systems	senior	high	senior	10
marketing	senior	medium	senior	10
marketing	Middle aged	medium	junior	20
secretary	senior	medium	senior	10
secretary	youth	low	junior	10

R1: IF salary =medium AND age = youth THEN Status = Junior

$$coverage(R) = \frac{n_{covers}}{|D|}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}.$$

- coverage(R1) = 20/200=10% and
- accuracy(R1)= 20/20 = 100%.

X: (Department = system *age* = *youth*, *salary* = *low*)

?

DECISION TREE INDUCTION
RULE EXTRACTION FROM A DT – RESOLVING RULES CONFLICTS

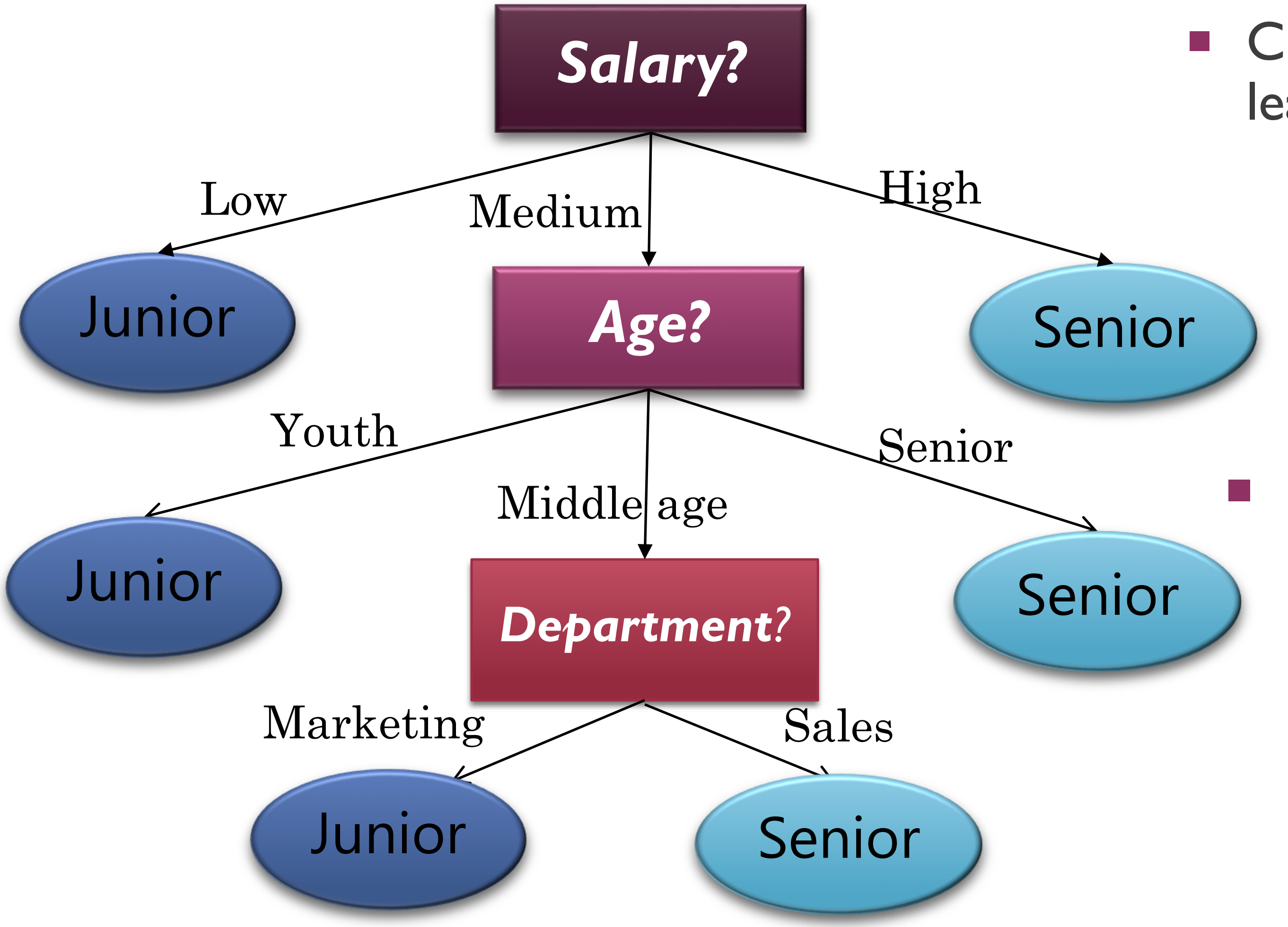
Rules conflicts are the result of a tuple firing more than one rule with different class predictions

Two resolution strategies

- Size Ordering →rule with **largest antecedent** (toughest) has highest priority fires and returns class prediction
- Rule Ordering →rules **prioritized apriori** according to
  - Class-based ordering → decreasing importance (most frequent are highest – order of prevalence)
  - Rule-based ordering → measures of rule quality (e.g. accuracy, size, domain expertise)

Fallback (default) rule when *no rules* are triggered

DECISION TREE INDUCTION
RULE EXTRACTION FROM A DECISION TREE



- Create one rule for each path from root to leaf in the decision tree
  1. Each <u>splitting criterion</u> is <u>ANDed</u> to form rule antecedent (IF)
  2. <u>Leaf</u> node holds <u>class prediction</u> (THEN)
- R1: IF salary =medium AND age = youth THEN Status = Junior

Can the rules resulting from decision trees have conflicts?

QUESTION?

NEXT …….