University of Colorado
Boulder

CSCI 4502/5502
Data Mining

Fall 2020
Lecture 11 (Sep 29)

Reminders: Course Project Proposal

- Due at 9:30am, Tuesday, September 29
  - meeting availability survey
- Due at 9:30am, Thursday, October 1
  - Course project announcement & proposal slides
- Due at 9:30m, Thursday, October 8
  - Course project proposal report

Course Project Grading

- 40% of final grade
- Proposal: 10% (announcement, meeting, report)
- Checkpoint: 10% (meeting, report)
- Final report: 20% (meeting, report, code & results)
- Projects evolve over time, changes are fine

Project Proposal Meetings

- Thursday, Oct 1 to Tuesday, Oct 6
- Public to the whole class, not recorded
- 5~10 minute presentation + ~5 minute discussion
- No regular lectures or office hours

Review

- Chapter 7: Advanced Pattern Mining
  - kinds of patterns, completeness, level of abstraction
  - number of data dimensions, types of value, types of rules
  - multi-level, multi-dimension, quantitative associations
  - constraint-based mining, metarule

Anti-Monotonicity

- Anti-monotonicity
  - if itemset S violates the constraint, so does any of its superset
- Example
  - sum(S.price) ≤ 100: yes
  - sum(S.price) ≥ 100: no
  - range(S.profit) ≤ 15: yes

Monotonicity

- Monotonicity
  - if itemset S satisfies the constraint, so does any of its superset
- Example
  - sum(S.price) $\geq$ 100: yes
  - min(S.price) $\leq$ 100: yes
  - range(S.profit) $\geq$ 15: yes

Succinctness

- Succinctness
  - enumerate all and only those sets that are guaranteed to satisfy the constraint
- Example
  - min(S.price) $\leq$ v: yes
  - sum(S.price) $\geq$ v: no
- Pre-counting prunable: no need for support counting

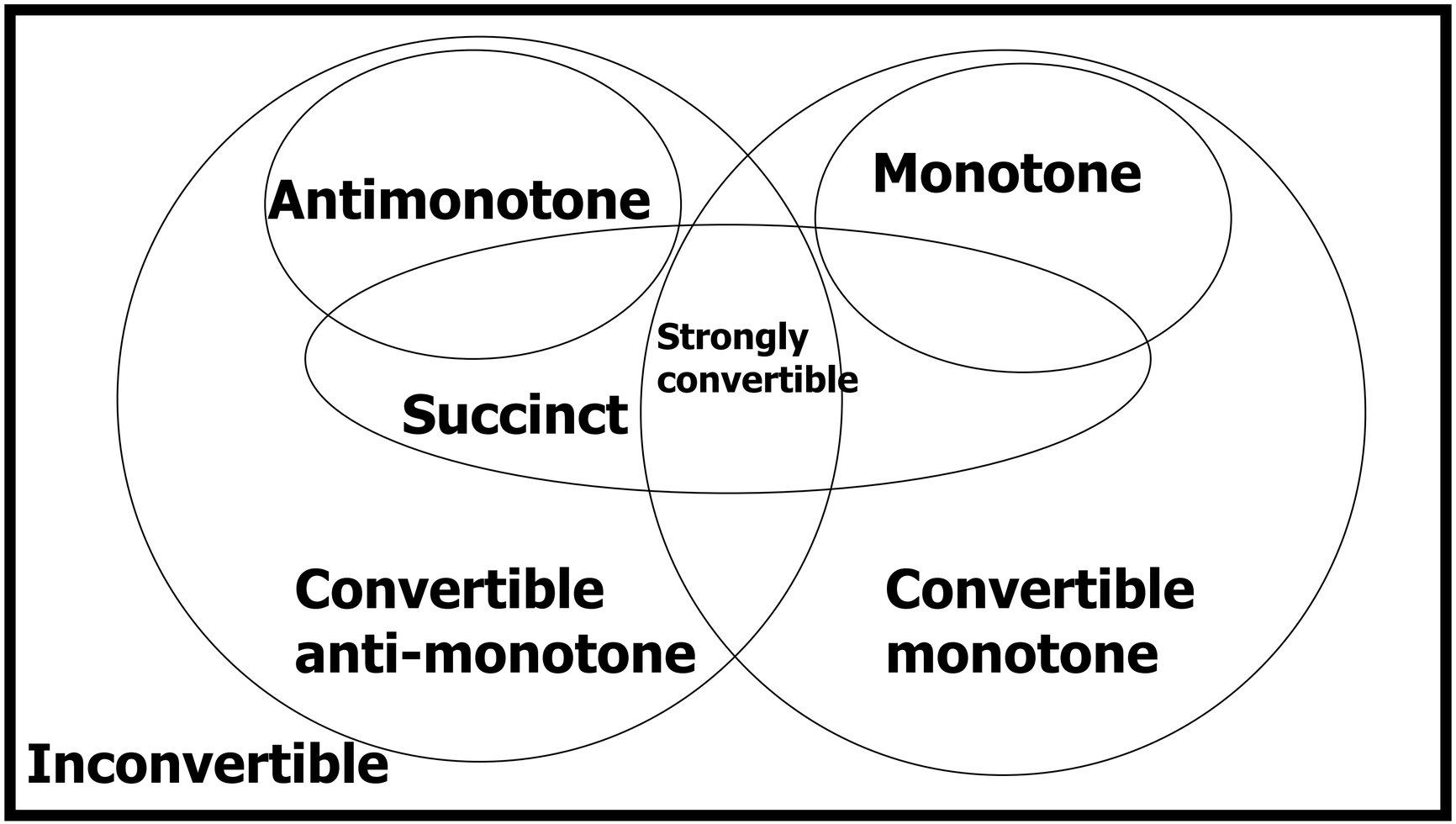
Convertible Constraints

- Convert tough constraints into anti-monotonic or monotonic by properly ordering items
- Example
  - avg(S.profit) ≥25
  - ordering items in descending order: <a, f, g, d, b, h, c, e>
  - if afb violates C, so does afb*
  - it becomes anti-monotonic

Strongly Convertible

- avg(S.profit) ≥ 25 is convertible anti-monotonic w.r.t. descending order
- avg(S.profit) ≥ 25 is convertible monotonic w.r.t. ascending order
- avg(S.profit) ≥ 25 is strongly convertible

Classification of Constraints



Example: Apriori Algorithm

tid	items
1	1, 3, 4
2	2, 3, 5
3	1, 2, 3, 5
4	2, 5

min_sup = 2

itemset	sup
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

itemset	sup
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

itemset	sup
{2, 3, 5}	2

Apriori + Constraint

tid	items
1	1, 3, 4
2	2, 3, 5
3	1, 2, 3, 5
4	2, 5

min_sup = 2

constraint:
sum(S.price)<5

itemset	sup
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

itemset	sup
{2, 3, 5}	2

itemset	sup
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

Apriori + Constraint

tid	items
1	1, 3, 4
2	2, 3, 5
3	1, 2, 3, 5
4	2, 5

min_sup = 2

constraint:
sum(S.price)<5

itemset	sup
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

itemset	sup
{2, 3, 5}	2

itemset	sup
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

Apriori + Constraint

tid	items
1	1, 3, 4
2	2, 3, 5
3	1, 2, 3, 5
4	2, 5

min_sup = 2

constraint:
min(S.price) ≤ 1

itemset	sup
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

itemset	sup
{2, 3, 5}	2

itemset	sup
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

Apriori + Constraint

tid	items
1	1, 3, 4
2	2, 3, 5
3	1, 2, 3, 5
4	2, 5

min_sup = 2

constraint:
min(S.price) ≤ 1

itemset	sup
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

itemset	sup
{2, 3, 5}	2

itemset	sup
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

Summary (1)

- Frequent patterns
  - itemset, subsequence, substruture
  - closed patterns, max-patterns
  - kinds of patterns, completeness, levels & dimensions, types of values, kinds of rules
- Mining frequent itemsets
  - Apriori, FP-growth, vertical data format
  - partition, sampling, hash-tree

Summary (2)

- Correlation analysis
  - support, confidence, correlation
- Mining various association rules
  - multi-level, multi-dimensional, quantitative
- Constraint-based mining
  - metarule
  - anti-monotonic, monotonic, succinct, convertible, inconvertible

University of Colorado
Boulder

Chapter 8
Classification: Basic Concepts

Chap 8: Classification

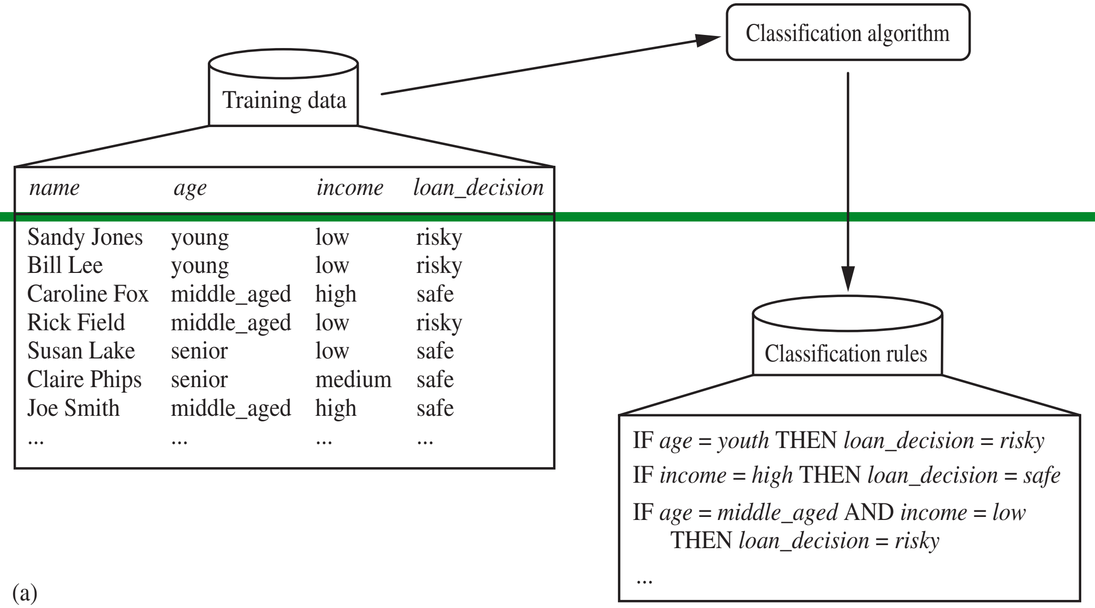
- Basic concepts
- Decision tree induction
- Bayesian classification
- Rule-based classification
- Model evaluation and selection
- Improve classification accuracy
- Summary

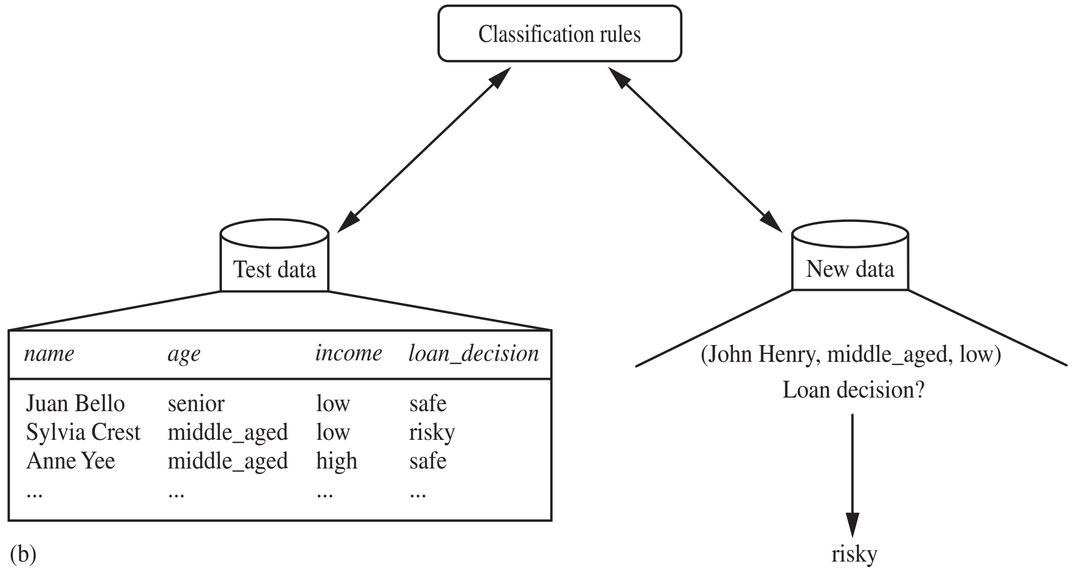
Classification vs. Prediction

- Classification
  - determines categorical class labels
  - e.g., safe vs. risky; weather condition
- Prediction
  - models continuous-valued functions
- Typical applications
  - loan approval, target marketing, medical diagnosis, fraud detection, etc.

Classification

- Step 1: Learning
  - model construction
  - training set, class labels
- Step 2: Classification
  - test set, accuracy





Supervised vs. Unsupervised

- Supervised learning (classification)
  - supervision: training data accompanied by class labels
  - new data is classified based on training set
- Unsupervised learning (clustering)
  - class labels of training data is unknown
  - aims to establish the existence of classes or clusters in the data

Issues: Evaluation Criteria

- Accuracy: classification vs. prediction
- Speed: time to construct / use the model
- Robustness: handling noise & missing values
- Scalability: large amounts of data
- Interpretability: understanding and insight
- Goodness of rules: e.g., decision tree size, compactness of classification rules

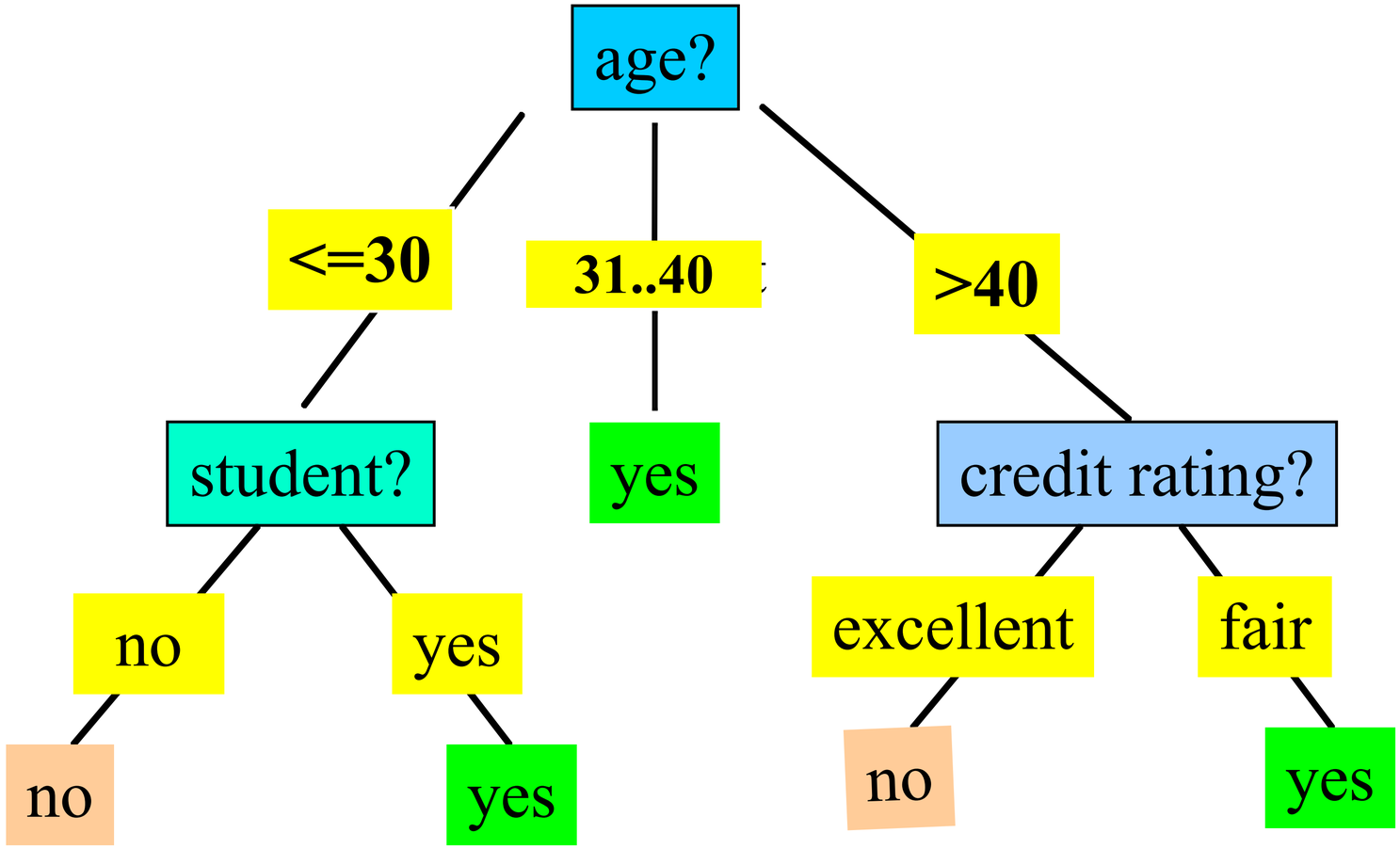
Chap 8: Classification

- Basic concepts
- Decision tree induction
- Bayesian classification
- Rule-based classification
- Model evaluation and selection
- Improve classification accuracy
- Summary

Example: Training Set

CID	age	income	student	credit_rating	buys_computer
1	<= 30	high	no	fair	no
2	<= 30	high	no	excellent	no
3	31-40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31-40	low	yes	excellent	yes
8	<= 30	medium	no	fair	no
9	<= 30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<= 30	medium	yes	excellent	yes
12	31-40	medium	no	excellent	yes
13	31-40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Example: Decision Tree

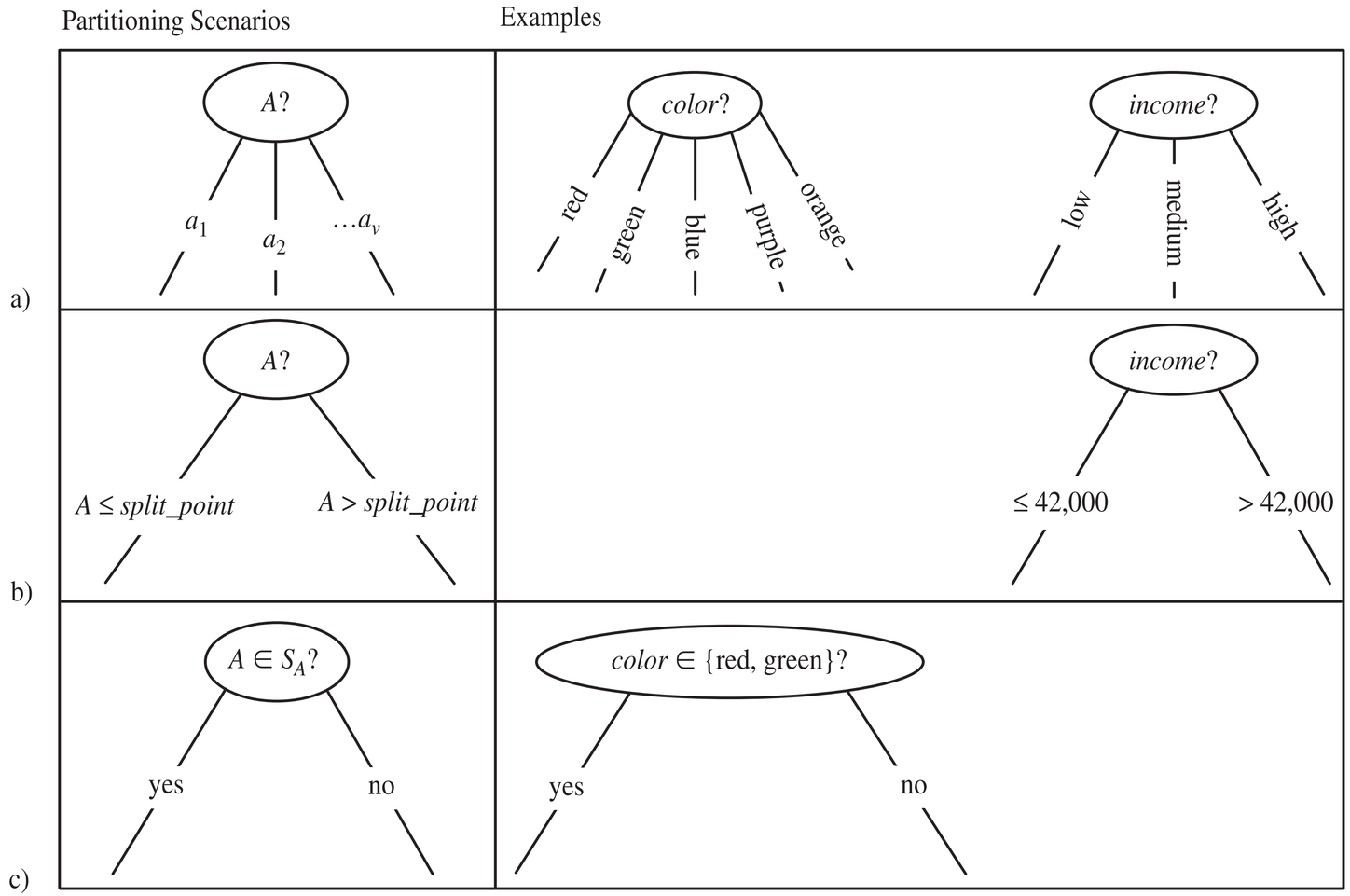


Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - top-down, recursive, divide-and-conquer
  - attribute selection
  - attribute split

Splitting Attributes

- Discrete-valued
- Continuous-valued: split_point
- Discrete-valued: binary tree, splitting_ subset



Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - top-down, recursive, divide-and-conquer
  - attribute selection
  - attribute split
- Stopping conditions
  - all samples belong to the same class
  - no remaining attributes: majority voting
  - no samples left