

Lecture 3: 13 January, 2026

Madhavan Mukund

<https://www.cmi.ac.in/~madhavan>

Data Mining and Machine Learning
January–April 2026

Market-basket analysis

- Set of items $I = \{i_1, i_2, \dots, i_N\}$
- A transaction is a set $t \subseteq I$ of items
- Set of transactions $T = \{t_1, t_2, \dots, t_M\}$
- Identify association rules $X \rightarrow Y$
 - $X, Y \subseteq I, X \cap Y = \emptyset$
 - If $X \subseteq t_j$ then it is likely that $Y \subseteq t_j$

Setting thresholds

- For $Z \subseteq I$, $Z.\text{count} = |\{t_j \mid Z \subseteq t_j\}|$
- How frequently does $X \subseteq t_j$ imply $Y \subseteq t_j$?
 - Fix a **confidence level** χ
 - Want $\frac{(X \cup Y).\text{count}}{X.\text{count}} \geq \chi$
- How significant is this pattern overall?
 - Fix a **support level** σ
 - Want $\frac{(X \cup Y).\text{count}}{M} \geq \sigma$
- Given sets of items I and transactions T , with confidence χ and support σ , find all valid association rules $X \rightarrow Y$

- If Z is frequent, so is every subset $Y \subseteq Z$
- We exploit the contrapositive

Apriori observation

If Z is not a frequent itemset, no superset $Y \supseteq Z$ can be frequent

- For any frequent pair $\{x, y\}$, both $\{x\}$ and $\{y\}$ must be frequent
- Build frequent itemsets bottom up, size 1,2,...

Apriori algorithm

- F_i : frequent itemsets of size i — **Level i**
- F_1 : Scan T , maintain a counter for each $x \in I$
- $C_2 = \{\{x, y\} \mid x, y \in F_1\}$: **Candidates** in level 2
- F_2 : Scan T , maintain a counter for each $X \in C_2$
- $C_3 = \{\{x, y, z\} \mid \{x, y\}, \{x, z\}, \{y, z\} \in F_2\}$
- F_3 : Scan T , maintain a counter for each $X \in C_3$
- ...
- C_k = subsets of size k , every $(k-1)$ -subset is in F_{k-1}
- F_k : Scan T , maintain a counter for each $X \in C_k$
- ...

Association rules

- Given sets of items I and transactions T , with confidence χ and support σ , find all valid association rules $X \rightarrow Y$
 - $X, Y \subseteq I, X \cap Y = \emptyset$
 - $\frac{(X \cup Y).count}{X.count} \geq \chi$
 - $\frac{(X \cup Y).count}{M} \geq \sigma$
- For a rule $X \rightarrow Y$ to be valid, $X \cup Y$ should be a frequent itemset
- Apriori algorithm finds all $Z \subseteq I$ such that $Z.count \geq \sigma \cdot M$

Association rules

Naïve strategy

- For every frequent itemset Z
 - Enumerate all pairs $X, Y \subseteq Z, X \cap Y = \emptyset$
 - Check $\frac{(X \cup Y).count}{X.count} \geq \chi$
- Can we do better?
- Sufficient to check all partitions of Z
 - If $X, Y \subseteq Z, X \cup Y$ is also a frequent itemset

Association rules

- Sufficient to check all partitions of Z
- Suppose $Z = X \uplus Y$, $X \rightarrow Y$ is a valid rule and $y \in Y$
- What about $(X \cup \{y\}) \rightarrow Y \setminus \{y\}$?
 - Know $\frac{(X \cup Y).count}{X.count} \geq \chi$
 - Check $\frac{(X \cup Y).count}{(X \cup \{y\}).count} \geq \chi$
 - $X.count \geq (X \cup \{y\}).count$, always
 - Second fraction has smaller denominator, so $(X \cup \{y\}) \rightarrow Y \setminus \{y\}$ is also a valid rule

Observation: Can use apriori principle again!

Apriori for association rules

- If $X \rightarrow Y$ is a valid rule, and $y \in Y$,
 $(X \cup \{y\}) \rightarrow Y \setminus \{y\}$ must also be a valid rule
- If $X \rightarrow Y$ is not a valid rule, and $x \in X$,
 $(X \setminus \{x\}) \rightarrow Y \cup \{x\}$ cannot be a valid rule
- Start by checking rules with single element on the right
 - $Z \setminus z \rightarrow \{z\}$
- For $X \rightarrow \{x, y\}$ to be a valid rule, both
 $(X \cup \{x\}) \rightarrow \{y\}$ and $(X \cup \{y\}) \rightarrow \{x\}$ must be valid
- Explore partitions of each frequent itemset “level by level”

Association rules for classification

- Classify documents by topic
- Consider the table on the right
- Items are regular words and topics
- Documents are transactions — set of words and one topic
- Look for association rules of a special form
 - $\{\text{student, school}\} \rightarrow \{\text{Education}\}$
 - $\{\text{game, team}\} \rightarrow \{\text{Sports}\}$
- Right hand side always a single topic
- Class Association Rules

Words in document	Topic
student, teach, school	Education
student, school	Education
teach, school, city, game	Education
cricket, football	Sports
football, player, spectator	Sports
cricket, coach, game, team	Sports
football, team, city, game	Sports

Summary

- Market-basket analysis searches for correlated items across transactions
- Formalized as association rules
- Apriori principle helps us to efficiently
 - identify frequent itemsets, and
 - split these itemsets into valid rules
- Class association rules — simple supervised learning model

Supervised learning

- A set of items
 - Each item is characterized by attributes (a_1, a_2, \dots, a_k)
 - Each item is assigned a class or category c
- Given a set of examples, predict c for a new item with attributes $(a'_1, a'_2, \dots, a'_k)$
- Examples provided are called **training data**
- Aim is to **learn** a mathematical model that **generalizes** the training data
 - Model built from training data should extend to previously unseen inputs
- **Classification** problem
 - Usually assumed to binary — two classes

Example: Loan application data set

ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Basic assumptions

Fundamental assumption of machine learning

- Distribution of training examples is identical to distribution of unseen data

What does it mean to learn from the data?

- Build a model that does better than random guessing
 - In the loan data set, always saying **Yes** would be correct about $9/15$ of the time
- Performance should ideally improve with more training data

How do we evaluate the performance of a model?

- Model is optimized for the training data. How well does it work for unseen data?
- Don't know the correct answers in advance to compare — different from normal software verification

The road ahead

Many different models

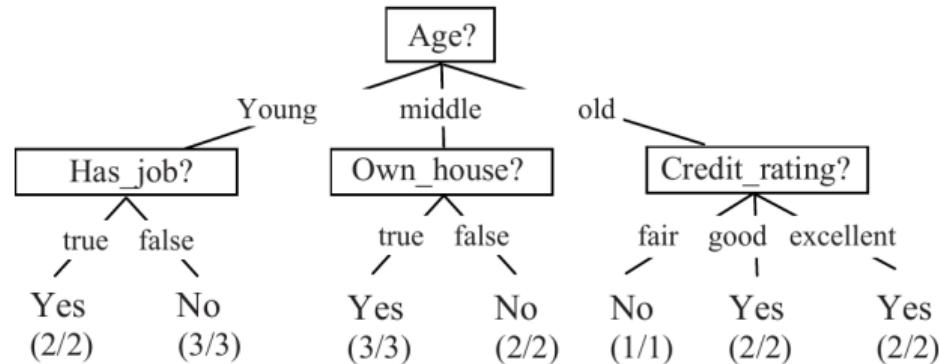
- Decision trees
- Probabilistic models — naïve Bayes classifiers
- Models based on geometric separators
 - Support vector machines (SVM)
 - Neural networks

Important issues related to supervised learning

- Evaluating models
- Ensuring that models generalize well to unseen data
 - A theoretical framework to provide some guarantees
- Strategies to deal with the training data bottleneck

Decision trees

- Play “20 Questions” with the training data
- Query an attribute
 - Partition the training data based on the answer
- Repeat until you reach a partition with a uniform category
- Queries are **adaptive**
 - Different along each path, depends on history



ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Decision tree algorithm

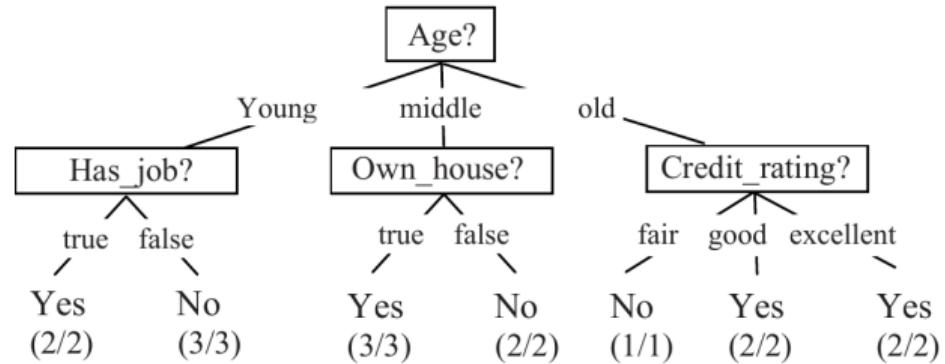
A : current set of attributes

Pick $a \in A$, create children corresponding to resulting partition with attributes $A \setminus \{a\}$

Stopping criterion:

- Current node has uniform class label
- A is empty — no more attributes to query

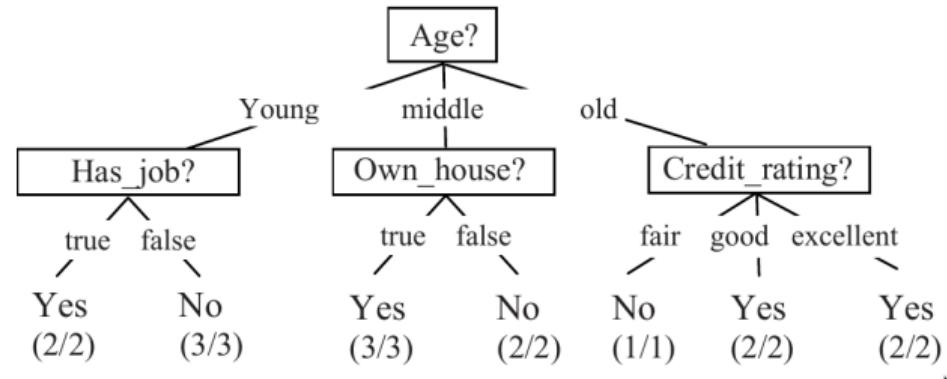
If a leaf node is not uniform, use majority class as prediction



- Non-uniform leaf node — identical combination of attributes, but different classes
- Attributes do not capture all criteria used for classification

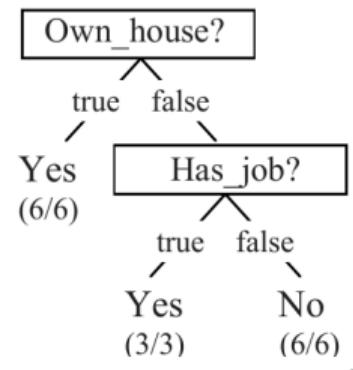
Decision trees

- Tree is not unique
- Which tree is better?
- Prefer small trees
 - Explainability
 - Generalize better (see later)



Unfortunately

- Finding smallest tree is NP-complete — for any definition of “smallest”
- Instead, greedy heuristic



Greedy heuristic

- Goal: partition with uniform category — **pure leaf**
- Impure node — best prediction is majority value
- Minority ratio is **impurity**
- Heuristic: reduce impurity as much as possible

