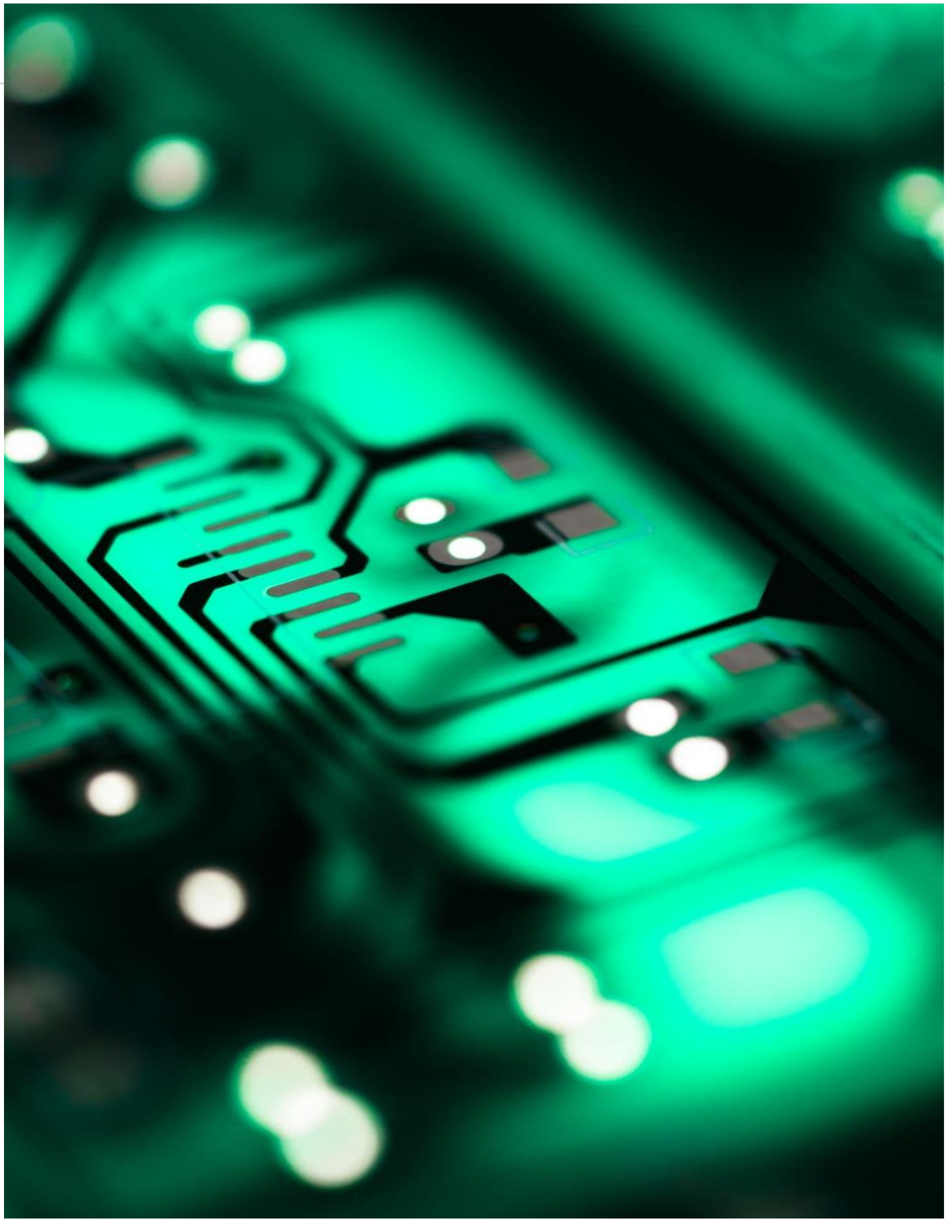
Introduction
Artificial Intelligence



Inference mechanism of Rule Based Expert System
&
Association Rules Mining

Dr. Muhammad Shuaib Qureshi

Outline

- Inference mechanism
- Market basket analysis
- Association Rules
- Apriori Algorithm
- Generating Rules

Rule-based expert systems

Mostly expert systems are rule-based.

Knowledge is represented as multiple rules that specify what should (or shouldn't) be concluded from different situations.

The inference engine determines which rules to fire

Ideally, the inference mechanism should deal with inexact and incomplete information.

Inference mechanism

Two approaches are common for inference

Forward chaining: primarily data-driven

Starts with facts and uses rules to draw conclusions (or to take actions)

Compare data/facts in the working memory against the condition in the IF parts of the rules to determine which rule fires

Ideal for prediction, monitoring, and control [X% People will die of disease Y]

Example:

CLIPS : "C Language Integrated Production System”

CLIPS is probably the most widely used expert system tool.

Inference mechanism

Backward chaining: primarily goal driven

Starts with hypothesis/query and look for rules that allow hypothesis to be proven true

Look for the action in the THEN clause of the rule that matches the goal

Ideal for diagnostic processes

Example: MYCIN

What about prolog inference engine?

Inference: example

Let the facts in the working memory be

A is true

B is true

Let the rules in the KB be

1. If A & C then F
2. If A & E then G
3. If B then E
4. If G then D

Let the query/hypothesis/goal be

Prove that If A & B are true then D is true

Inference: example [forward chaining]

Start with the given input: A & B are true

Loop1: Search in the KB top to bottom to find a match

- Rule 3 fires
  - Conclusion: E is true
  - New knowledge is found (E)
- Going further down, no other rule fires; goal not found yet

Loop2: Search in the KB top to bottom to find a match

- Rule 2 fires
  - Conclusion: G is true
  - New knowledge is found (G)
- Going further down, rule 4 fires
  - Conclusion: D is true; Goal is found

A is true
B is true

Let the rules in the KB be
If A & C then F
If A & E then G
If B then E
If G then D

Prove that If A & B are true then D is true

Inference: example [backward chaining]

Start with the goal (D)

Loop1: Search in the KB top to bottom to find a match

- Rule 4 fires
  - New sub-goal to prove: G is true
- No other rule fires; goal not found yet

Loop2: Search in the KB top to bottom to find a match

- Rule 2 fires
  - New sub-goal to prove: A & E is true
  - A is true, given, we need to prove E is true
- Going further down, rule 3 fires
  - New sub-goal to prove: B is true
  - B is also true, given
  - Both inputs A & B ascertained, so hypothesis is proved

A is true
B is true

Let the rules in the KB be
If A & C then F
If A & E then G
If B then E
If G then D

Prove that If A & B are true then D is true

Market basket analysis

- Market Basket Analysis is a **modelling technique**.
- It is based on, if you buy a certain group of items, you are more (or less) likely to buy another group of items.
- For example, if you are in a store and you buy a car then you are more likely to buy insurance at the same time than somebody who didn't buy insurance.
- The **set of items** a customer buys is referred to as an **itemset**.
- Market basket analysis seeks to **find relationships between purchases** (Items).
  - **E.g.** IF {Car, Accessories} THEN {Insurance}

$$\boxed{\{\text{Car, Accessories}\} \rightarrow \{\text{Insurance}\}}$$

Market basket analysis (Cont..)

$\boxed{\{Car, Accessories\} \rightarrow \{Insurance\}}$

- The **probability** that a customer will buy car **without** an accessories is referred as the **support** for **rule**.

- The **conditional probability** that a customer will purchase Insurance is referred to as the **confidence**.

Association rule mining

- Given a set of transactions, we need rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

- **Market-Basket transactions**

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

Example of Association Rules

{Chocolate} → {Pepsi},
{Milk, Bread} → {Eggs, Coke},
{Pepsi, Bread} → {Milk}

Association rule mining (Cont..)

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

- **Itemset**
  - A collection of **one or more items**
    - **E.g.** : {Milk, Bread, Chocolate}
  - k-itemset

    An itemset that contains **k** items
- **Support count (σ)**
  - **Frequency** of occurrence of an **itemset**
    - **E.g.** σ({Milk, Bread, Chocolate}) = 2
- **Support**
  - **Fraction of transactions** that **contain an itemset**
    - **E.g.** s({Milk, Bread, Chocolate}) = 2/5
- **Frequent Itemset**
  - An itemset whose **support is greater than or equal to a minimum support** threshold

Association rule mining (Cont..)

- **Association Rule**
  - An **implication expression** of the form **X → Y**, where X and Y are itemsets
    - **E.g.**: {Milk, Chocolate} → {Pepsi}
- **Rule Evaluation**
  - Support (s)
    - **Fraction of transactions** that **contain both X and Y**
  - Confidence (c)
    - Measures **how often items in Y appear** in **transactions that contain X**

Association rule mining (Cont..)

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

Example:

Find support & confidence for **{Milk, Chocolate}** $\Rightarrow$ **Pepsi**

$$S = \frac{\sigma\,(Milk, Chocolate, Pepsi)}{|T|} = \frac{2}{5} = 0.4$$

$$C = \frac{\sigma\,(Milk, Chocolate, Pepsi)}{\sigma\,(Milk, Chocolate)} = \frac{2}{3} = 0.67$$

Association rule mining - example

TID	Items
1	Bread, Milk
2	Bread, Chocolate, Pepsi, Eggs
3	Milk, Chocolate, Pepsi, Coke
4	Bread, Milk, Chocolate, Pepsi
5	Bread, Milk, Chocolate, Coke

Calculate **Support** & **Confidence:**

{Milk, Chocolate} → {Pepsi}
{Milk, Pepsi} → {Chocolate}
{Chocolate, Pepsi} → {Milk}
{Pepsi} → {Milk, Chocolate}
{Chocolate} → {Milk, Pepsi}
{Milk} → {Chocolate, Pepsi}

Answer

Support (s) : **0.4**

{Milk, Chocolate} → {Pepsi} c = **0.67**
{Milk, Pepsi} → {Chocolate} c = **1.0**
{Chocolate, Pepsi} → {Milk} c = **0.67**
{Pepsi} → {Milk, Chocolate} c = **0.67**
{Chocolate} → {Milk, Pepsi} c = **0.5**
{Milk} → {Chocolate, Pepsi} c = **0.5**

Association rule mining (Cont..)

- A common strategy adopted by many association rule mining algorithms is to decompose the problem into two major subtasks:

1. **Frequent Itemset Generation**
   - The objective is to find all the item-sets that satisfy the minimum support threshold.
   - These itemsets are called **frequent itemsets**.
2. **Rule Generation**
   - The objective is to extract all the high-confidence rules from the frequent itemsets found in the previous step.
   - These rules are called **strong rules**.

Apriori algorithm

- **Purpose**: The Apriori Algorithm is an influential algorithm for mining **frequent itemsets** for Boolean **association rules**.
- **Key Concepts**:
  - **Frequent Itemsets**:

    The sets of item which has **minimum support** (denoted by L_i for ith-Itemset).
  - **Apriori Property**:

    Any **subset of frequent itemset must** be **frequent**.
  - **Join Operation**:

    To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} itself.

Apriori algorithm (Cont..)

- **Find the frequent itemsets**
  - The **sets of items that have minimum support** and a **subset of a frequent itemset must** also be a **frequent itemset (Apriori Property).**
  - **E.g.** if **{AB}** is a frequent itemset, both {A} and {B} should be a frequent itemset.
  - Use the frequent item sets to generate association rules.
- **The Apriori Algorithm : Pseudo code**
  - **Join Step**: C_k is generated by joining L_{k-1} with itself
  - **Prune Step**: Any (k-1) itemset that is not frequent cannot be a subset of a frequent k-itemset

Apriori algorithm steps (Cont..)

- **Step 1:**
  - Start with itemsets containing just a **single item (Individual items).**
- **Step 2:**
  - Determine the support for itemsets.
  - **Keep** the itemsets that **meet your minimum support threshold** and **remove** itemsets that **do not support minimum support**.
- **Step 3:**
  - Using the itemsets you have kept from Step 1, **generate all the possible itemset combinations**.
- **Step 4:**
  - **Repeat** steps 1 & 2 until there are **no more new itemsets**.

Apriori algorithm - Pseudo code (Cont..)

C_k: Candidate itemset of size k

L_k: Frequent itemset of size k

L_1= {frequent items};

for (k = 1; L_k != ∅; k++) **do begin**

C_{k+1} = candidates generated from L_k;

for each transaction **t** in database do

Increment the count of all candidates in C_{k+1}

That are contained in **t**

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

Apriori algorithm - Example

Minimum Support = 2

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D →

C_1

ItemSet	Min. Sup
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

→

L_1

ItemSet	Min. Sup
{1}	2
{2}	3
{3}	3
{5}	3

C_2

Itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

← Scan D

C_2

Itemset	Min. Sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

←

L_2

ItemSet	Min. Sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Apriori algorithm - Example

Minimum Support = 2

L_2

ItemSet	Min. Sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Scan D

C_3

ItemSet	Min. Sup
{1 2 3}	1
{1 2 5}	1
{1 3 5}	1
{2 3 5}	2

L_3

Items	Sup
{2 3 5}	2

Rules generation

Association Rule	Support	Confidence	Confidence (%)
2 ^ 3 →5	2	2/2 = 1	100 %
3 ^ 5 →2	2	2/2 = 1	100 %
2 ^ 5 →3	2	2/3 = 0.66	66%
2 = 3 ^ 5	2	2/3 = 0.66	66%
3 = 2 ^ 5	2	2/3 = 0.66	66%
5 = 2 ^ 3	2	2/3 = 0.66	66%

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Activity:

Use Apriori algorithm and generate the rules with 100% confidence.
Minimum Support = 2

TID	ITEMS
1	A B C D
2	B C E
3	A B C E
4	B E F
5	A B F D
6	B C D E F