Nile University

CSCI467 – Data Mining

Lab 04

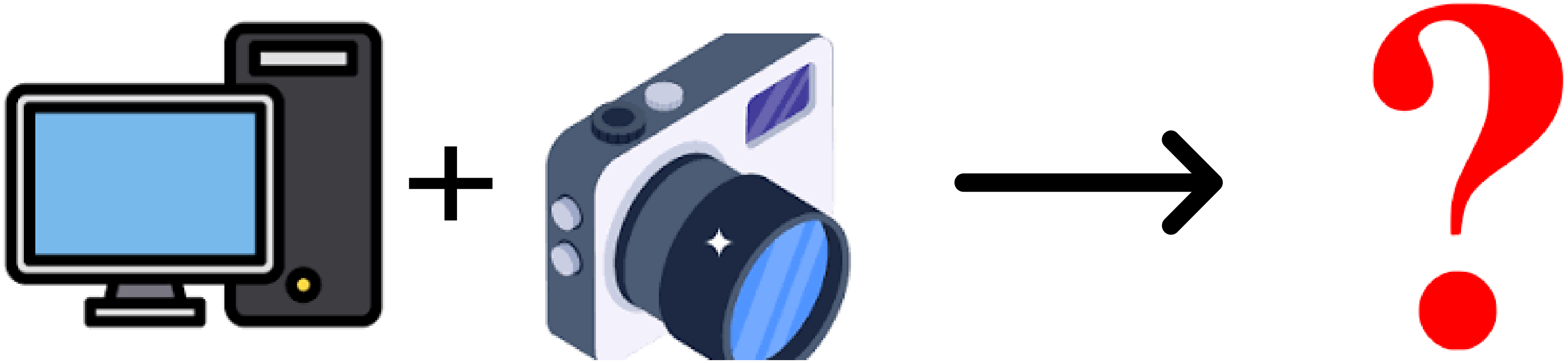
Table of Contents

Association Rules Mining

Introduction

Imagine that you are a sales manager, and you are talking to a customer who recently bought a **PC** and a **digital camera** from the store.

What should you **recommend** to her next?



Introduction

Information about which products are **frequently purchased** by your customers **following** their purchases of a **PC and a digital camera** in sequence would be very helpful in making your recommendation.

Frequent patterns and **association rules** are the knowledge that you want to mine in such a scenario.

Frequent patterns are patterns that appear frequently in a dataset.

For example, a set of items, such as **milk** and **bread**, that appear frequently together in a transaction data set is a **frequent itemset.**

Introduction

The discovery of **interesting relationships** among huge amounts of business **transaction** records can help in many **business decision-making** processes such as:

- Catalog design.
- Products recommendation.
- Cross-marketing.
- Customer shopping behavior analysis.
- Develop marketing strategies.
- Filling out missing data.

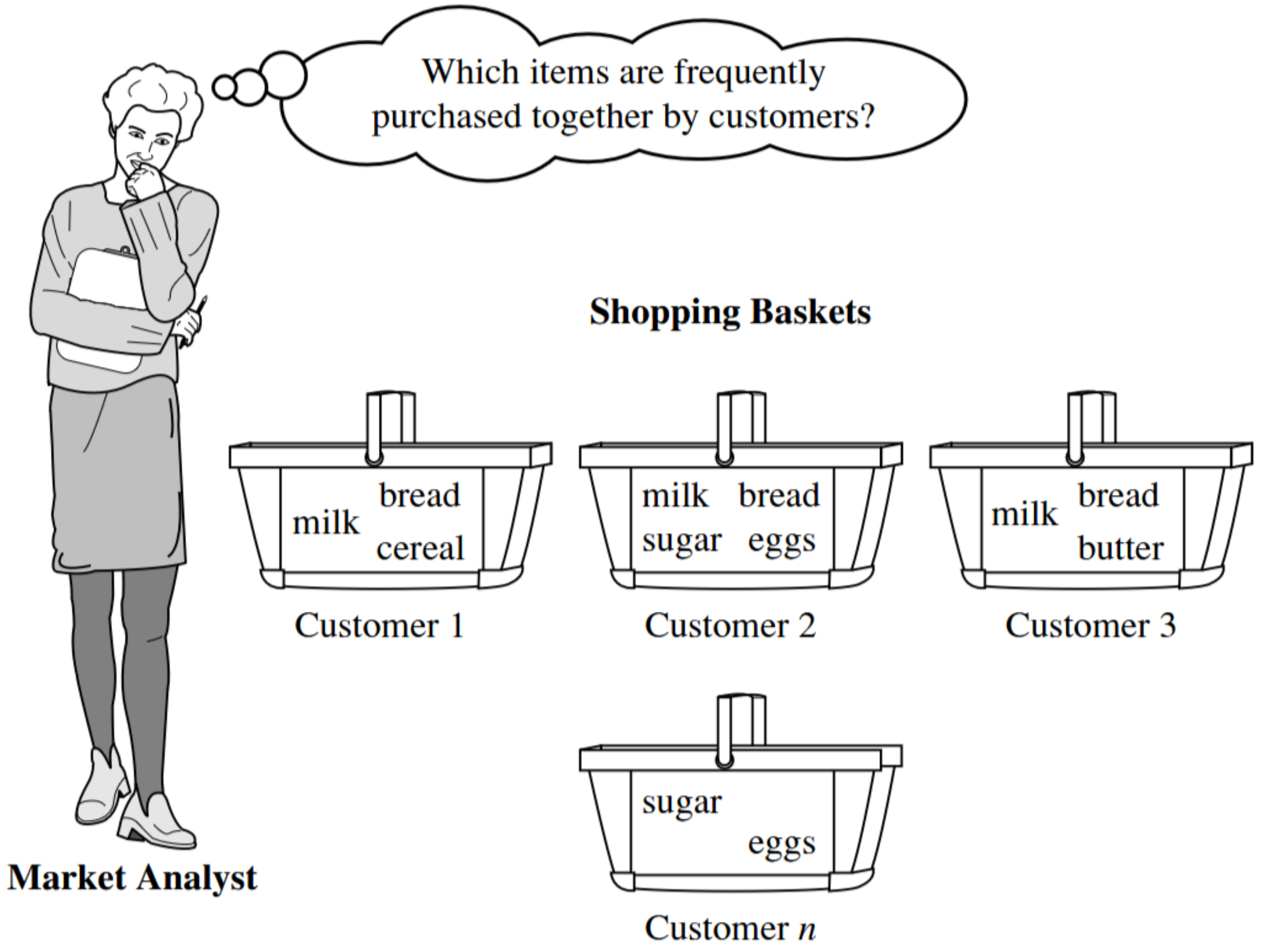
For example, the association rule **{Bread --> Milk}** has 2 parts:

- Antecedent {**Bread**}.
- Consequent {**Milk**}.

For a **high confidence** rule: **if** Bread exists, **then** Milk will exist.

Market Baset Analysis: A Motivating Example

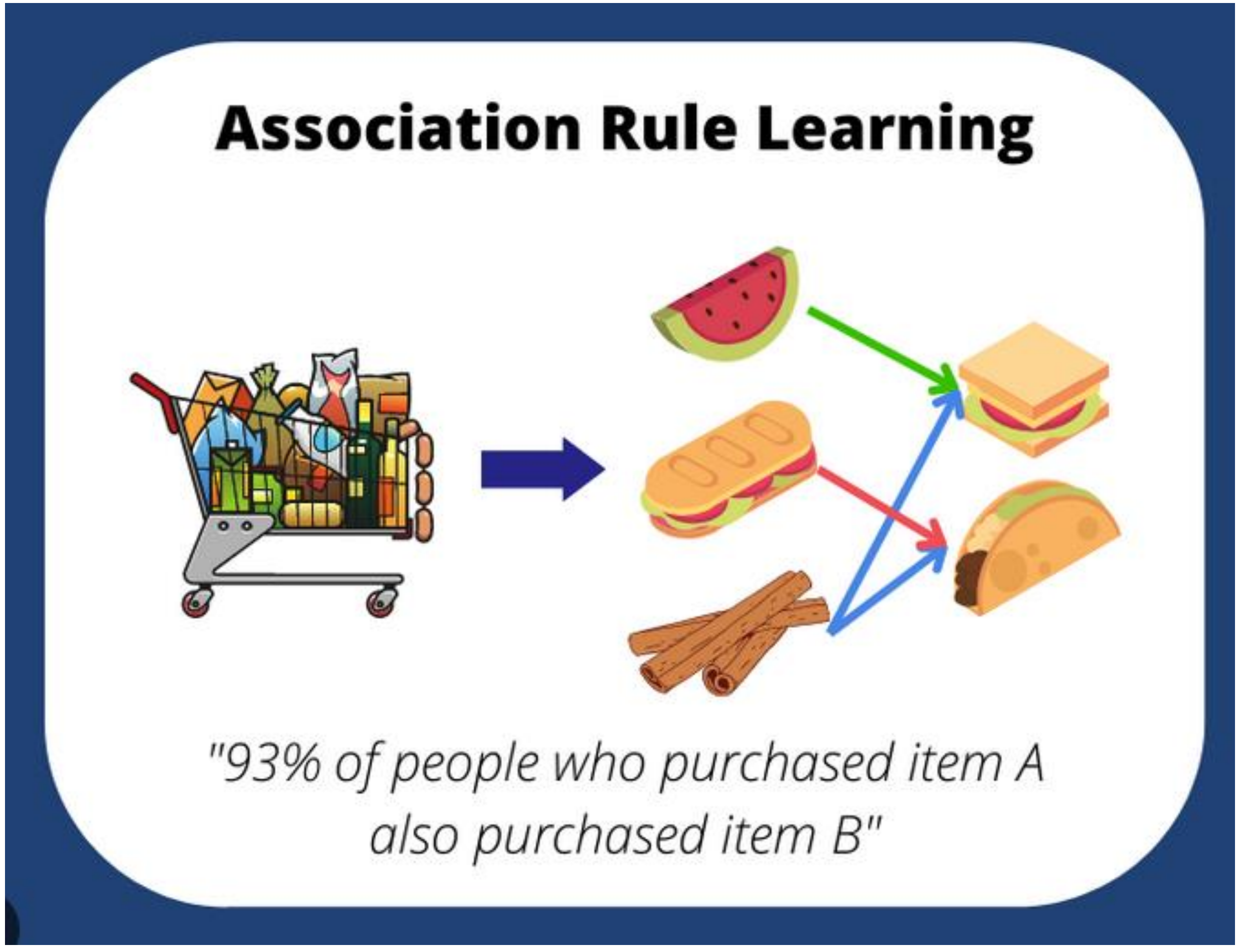
A typical example of **frequent itemset mining** is **market basket analysis.**



Introduction

Algorithms to mine frequent patterns:

1. Apriori algorithm.
2. FP Growth algorithm.
3. ECLAT algorithm.



How to measure an association rules?

$$\text{Support}(X) = \frac{\text{Transactions containing } X}{\text{Total transactions}}$$	How frequently an itemset appears in the dataset.
$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$	For each time X appears, how many times does Y appear in the same tuple? **Or** The likelihood that Y is purchased given that X is purchased.
$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$	Measures the strength of a rule. **>1:**products are likely purchased together. **<1:** products are not likely purchased together. **=1:** no association (bought together by chance).

Example 1

Given 5 transactions:

Transaction	Items Bought
1	Milk, Bread, Butter
2	Milk, Bread
3	Bread, Butter
4	Milk
5	Bread, Butter

<u>(1) Support:</u>

Support of (`Milk`): $\frac{3}{5} = 0.6$

Support of (`Milk, Bread`): $\frac{2}{5} = 0.4$

Example 1

Given 5 transactions:

Transaction	Items Bought
1	Milk, Bread, Butter
2	Milk, Bread
3	Bread, Butter
4	Milk
5	Bread, Butter

(2) Confidence: $\text{Confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$

Support {Milk, Bread} = $\frac{2}{5} = 0.4$

Support {Milk} = $\frac{3}{5} = 0.6$

Confidence (Milk --> Bread) = 0.4 / 0.6 = **0.67 (67%)**

This means that **67%** of customers **who bought Milk also bought Bread.**

Example 1

Given 5 transactions:

Transaction	Items Bought
1	Milk, Bread, Butter
2	Milk, Bread
3	Bread, Butter
4	Milk
5	Bread, Butter

<u>**(3) Lift:**</u> $\text{Lift}(X \Rightarrow Y) = \dfrac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$

Confidence {Milk, Bread} = 0.67

Support {Bread} = $\frac{4}{5} = 0.8$

Lift (Milk --> Bread) = $\dfrac{0.67}{0.8} = 0.84$

Milk and Bread are **not likely** bought together.

Introduction

Apriori Algorithm:

- Any subset of **frequent** itemset; must be **frequent**.
- Ex: if {x,y,z} is frequent, then {y,z} also {x} is frequent.
- If any subset of itemset S is infrequent, then there is no chance for S to be frequent, hence you do **not** have to mine S.
- Applied in two steps:
  - Candidate generation (self join).
  - Pruning (filter items by the minimum support).

Example 2

A database has 5 transactions. Let min sup = 60% and min conf = 80%. Use Apiriori algorithm to **find** the frequent item-sets.

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I ,E}

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I ,E}

Example 2

At K = 1

Step 01

1. Candidates generation:		2. Pruning:
M	3	M
O	3	O
N	2	
K	5	K
E	4	E
Y	3	Y
D	1	
A	1	
C	2	
U	1	
I	1	

At K = 2

Step 02

1. Candidates generation:		2. Pruning
MO	1	
MK	3	MK
ME	2	
MY	2	
OK	3	OK
OE	3	OE
OY	2	
KE	4	KE
KY	3	KY
EY	2	

At K = 3

Step 03

1. Candidates generation:		2. Pruning:
MKO	1	
MKE	2	
MKY	2	
OKE	3	OKE
OKY	2	
OEY	2	
KEY	2	

Example 2

C: Candidates (all item sets).
L: Frequent item set.

C1 = {M,O,N,K,E,Y,D,A,C,U,I}
L1 = {E, K, M, O, Y}

C2 = {EK, EM, EO, EY, KM, KO, KY, MO, MY, OY}
L2 = {EK, EO, KM, KO, KY}

C3 = {EKO}
L3 = {EKO}

C4 = Ø
L4 = Ø

The complete set of frequent itemsets (all in pruning steps):
{M, O, E, K, Y, EK, EO, KM, KO, KY, EKO}

The most frequent itemset is (at highest K): {EKO}

Disadvantage:
We generate lots of useless combinations (candidates) that are pruned later. To overcome this limitation, use **FP Growth** algorithm.

Example 3

Find the frequent item-sets and **generate** association rules on this. Assume that minimum support threshold (s = 33.33%) and minimum confident threshold (c = 60%)

Transaction ID	Items
T1	Hot Dogs, Buns, Ketchup
T2	Hot Dogs, Buns
T3	Hot Dogs, Coke, Chips
T4	Chips, Coke
T5	Chips, Ketchup
T6	Hot Dogs, Coke, Chips

Example 3

(1) Frequent item sets:

C at K = 1

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

L at K = 1

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

C at K = 2

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Ketchup	1
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Buns, Ketchup	1
Buns, Coke	0
Buns, Chips	0
Ketchup, Coke	0
Ketchup, Chips	1
Coke, Chips	3

L at K = 2

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Coke, Chips	3

C at K = 3

Item set	Sup-count
Hot Dogs, Buns, Coke	0
Hot Dogs, Buns, Chips	0
Hot Dogs, Coke, Chips	2

L at K = 3

Item set	Sup-count
Hot Dogs, Coke, Chips	2

$$\text{minimum support count} = \frac{33.33}{100} \times 6 = 2$$

The most frequent itemset (I) = {Hot Dogs, Coke, Chips}

Example 3

(2) Association rules:

- [Hot Dogs^Coke]=>[Chips]

confidence = sup(Hot Dogs^Coke^Chips)/sup(Hot Dogs^Coke) = 2/2*100=**100%** **Selected**

- [Hot Dogs^Chips]=>[Coke]

confidence = sup(Hot Dogs^Coke^Chips)/sup(Hot Dogs^Chips) = 2/2*100=**100%** **Selected**

- [Coke^Chips]=>[Hot Dogs]

confidence = sup(Hot Dogs^Coke^Chips)/sup(Coke^Chips) = 2/3*100=**66.67%** **Selected**

- [Hot Dogs]=>[Coke^Chips]

confidence = sup(Hot Dogs^Coke^Chips)/sup(Hot Dogs) = 2/4*100=**50%** **Rejected**

Example 3

(2) Association rules:

- [Coke]=>[Hot Dogs^Chips]

confidence = sup(Hot Dogs^Coke^Chips)/sup(Coke) = 2/3*100=**66.67%** **Selected**

- [Chips]=>[Hot Dogs^Coke]

confidence = sup(Hot Dogs^Coke^Chips)/sup(Chips) = 2/4*100**=50%** **Rejected**

There are **four** strong results (minimum confidence greater than 60%).

Extra Example

Given the following transactions with missing values, use the Apriori algorithm to fill out these missing values:
Butter in ID2 and Eggs in ID4.

ID	Items
ID1	Milk, Bread, Eggs
ID2	Milk, Bread, **? (Butter)**, Eggs
ID3	Milk, Butter, Eggs
ID4	Bread, Butter, **? (Eggs)**
ID5	Milk, Bread, Butter, Eggs

Extra Example

ID	Items
ID1	Milk, Bread, Eggs
ID2	Milk, Bread, **? (Butter)**, Eggs
ID3	Milk, Butter, Eggs
ID4	Bread, Butter, **? (Eggs)**
ID5	Milk, Bread, Butter, Eggs

Step 1:
Convert the dataset into a transactional format.

ID	Milk	Bread	Butter	Eggs
1	1	1	0	1
2	1	1	?	1
3	1	0	1	1
4	0	1	1	?
5	1	1	1	1

Extra Example

ID	Milk	Bread	Butter	Eggs
1	1	1	0	1
2	1	1	?	1
3	1	0	1	1
4	0	1	1	?
5	1	1	1	1

Step 2:

Apply Apriori Algorithm to get association rules with high confidence:

- `{Milk, Bread} → {Butter}` (Confidence = 85%)
- `{Bread, Butter} → {Eggs}` (Confidence = 90%)

Step 3:

Fill out missing values:

- **ID**: Since `{Milk, Bread} → {Butter}` is **85% confident**, we can assume `Butter = 1`.
- **ID4**: Since `{Bread, Butter} → {Eggs}` is **90% confident**, we can assume `Eggs = 1`.

Step 4:

Update the table.

ID	Milk	Bread	Butter	Eggs
1	1	1	0	1
2	1	1	**1**	1
3	1	0	1	1
4	0	1	1	**1**
5	1	1	1	1

Task

Task

Find the frequent item-sets and **generate** association rules. Assume that:

- Minimum support threshold (s = 50%)
- Minimum confident threshold (c = 70%)

Transaction	Items appearing in the transaction
T1	{pasta, lemon, bread, orange}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{pasta, lemon, orange, cake}

Thank you!