Data Mining & Informatics Lecture #5

Amin Noroozi
University of Wolverhampton
✉ a.noroozifakhabi@wlv.ac.uk
https://www.linkedin.com/in/amin-n-148350218/

Reminder

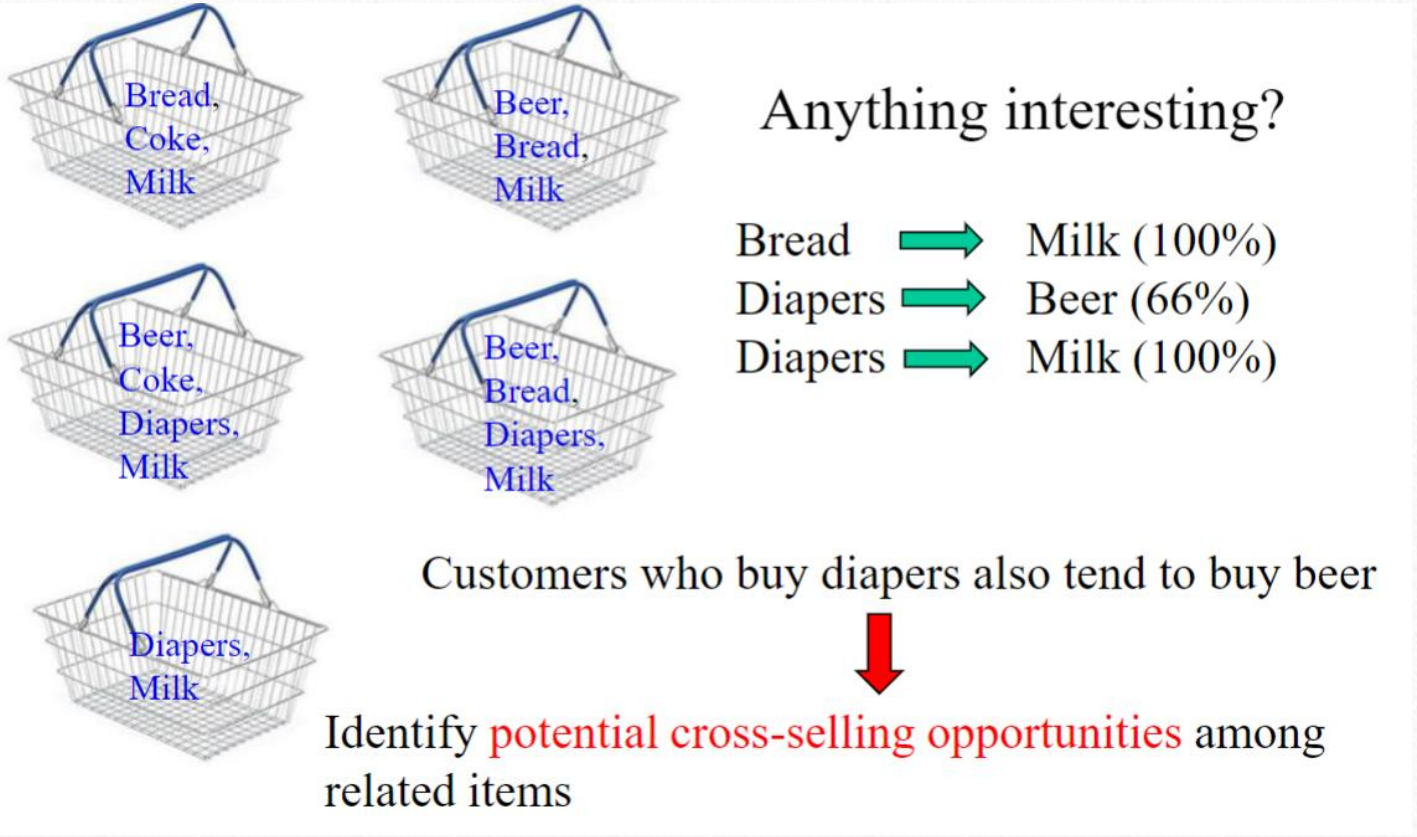
- **Extra lessons**

- Extra lessons will be on Tuesday 5-7 pm

Association Rule Mining

Association Rule Mining

- **Example**



Association Rule Mining

- The goal of association rule mining is to identify relationships between items in a dataset that occur frequently together

- This can increase sales. For example, it is likely that if a customer buys **Milk** and **bread** he/she also buys **Butter**. So the association rule is **('milk','bread')=>'butter'**. So the seller can suggest the customer buy butter if he/she buys Milk and Bread.

Association Rule Mining

- **Rule**: A rule in association rule mining is shown in one of the following forms:

A => B

It is highly likely that a customer who buys item A also buys item B

A, B => C (or A∩B =>C)

It is highly likely that a customer who buys item A and B also buys item C

A=> B, C (or A => B ∩ C)

It is highly likely that a customer who buys item A also buys items B and C

NOTE: You can generalise the above rules using more than two items, for example A, B,C => D

Association Rule Mining

- **K- Item set**: An item set containing K items

- **Support of an item or an item set**: Support of item X or item set X is calculated using the following formula:

$$\text{Support}(X) = \frac{Number\ of\ transactions\ containing\ X}{Total\ number\ of\ transactions}$$

- **Confidence**: Confidence of rule A=>B is calculated as follows:

$$\text{Support}(A=>B) = \frac{Number\ of\ transactions\ containing\ A\ and\ B}{number\ of\ transactions\ containing\ A}$$

Association Rule Mining

NOTE: The confidence value shows how low likely it is for a customer who buys A to also buy B. A confidence level of 60% per cent means in 60% of cases a customer who buys A also buys B.

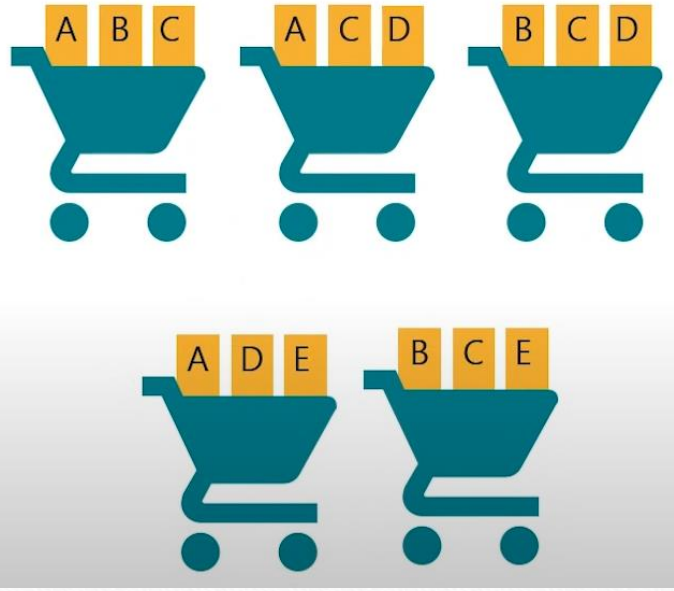
Frequent item set: An item set whose support is bigger than a threshold called 'minimum support'.

NOTE: The minimum support value is a hyperparameter set by the user

Strong association rule: An association rule whose confidence level is bigger than a threshold called the minimum confidence level

NOTE: The minimum confidence level is a hyperparameter set by the user

Association Rule Mining



Transaction at a Local Market

T1	A	B	C
T2	A	C	D
T3	B	C	D
T4	A	D	E
T5	B	C	E

Association Rule Mining

Rule	Support	Confidence
A=>D	2/5	2/3
C=>A	2/5	2/4
A=>C	2/5	2/3
B, C=>A	1/5	1/3

Association Rule Mining

How can we mine association rules? → Apriori algorithms

Apriori algorithm

Apriori algorithm

Apriori property:
All subsets of a frequent itemset must be frequent

- In apriori algorithm, we find all frequent k-item sets starting at k=1. Then we increment k until no resulting k-item set is frequent. If an item set is frequent but any of its subsets is not frequent, the item set will be removed (pruning). At this point, we create all possible association rules and calculate the confidence level for each rule. Any rule whose confidence level is bigger than the minimum confidence level will be selected as a strong association rule.

Apriori algorithm

Example: Given the following item set, mine association rules with the minimum support count of 2 and minimum confidence level of 60%

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Apriori algorithm

Step 1: (I) Set k=1 and create a table containing support counts of 1- item sets

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

C1

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

Apriori algorithm

Step 1: (II) Compare item sets support count with the minimum support count (here min_support=2). Remove item sets whose support counts are less than the minimum support count. This gives us table F1

C1

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

F1

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

Apriori algorithm

Step 2: Set k=2. Generate 2-item sets C2 by joining F1 with itself. The condition of joining is that the two joining items in F1 should have (K-2) elements in common (here k-2=0). Drop item sets with a support less than the minimum support or item sets whose subsets are not frequent. This gives us table F2.

F1

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

Only Items present in F1

C2

Itemset	Support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Apriori algorithm

Step 3: Set k=3. Generate 3-item sets C3 by joining F2 with itself. The condition of joining is that the two joining items in F1 should have (K-2) elements in common (here k-2=1). Drop item sets with a support less than the minimum support or item sets whose subsets are not frequent. This gives us table F3.

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

C3

Itemset	In F2?
{1,2,3}, {1,2}, {1,3}, {2,3}	NO
{1,2,5}, {1,2}, {1,5}, {2,5}	NO
{1,3,5},{1,5}, {1,3}, {3,5}	YES
{2,3,5}, {2,3}, {2,5}, {3,5}	YES

Apriori algorithm

Step 3: Set k=3. Generate 3-item sets C3 by joining F2 with itself. The condition of joining is that the two joining items in F1 should have (K-2) elements in common (here k-2=1). Drop item sets with a support less than the minimum support or item sets whose subsets are not frequent. This gives us table F3.

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

F3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

Apriori algorithm

Step 4: Repeat Step 3 for k=4. At this point, no frequent item set is generated. When this happens, we no longer generate item sets and start to mine association rules using the last frequent item set generated.

F2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

F3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

C3

Itemset	Support
{1,2,3,5}	1

Apriori algorithm

Step 5: Generate association rules. Remove any association rule whose confidence level is below the minimum confidence level.

F3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

For I = {1,3,5}, subsets are {1,3}, {1,5}, {3,5}, {1}, {3}, {5}

For I = {2,3,5}, subsets are {2,3}, {2,5}, {3,5}, {2}, {3}, {5}

- For every subsets S of I, output the rule:

S → (I-S) (S recommends I-S)

if ***support(I)/support(S) >= min_conf value***

Apriori algorithm

Applying Rules to Item set F3

1. **{1,3,5}**

   ✓ Rule 1: **{1,3}** → **({1,3,5} - {1,3})** means 1 & 3 → 5
   Confidence = support(1,3,5)/support(1,3) = 2/3 = 66.66% > 60%
   Rule 1 is selected

   ✓ Rule 2: **{1,5}** → **({1,3,5} - {1,5})** means 1 & 5 → 3
   Confidence = support(1,3,5)/support(1,5) = 2/2 = 100% > 60%
   Rule 2 is selected

   ✓ Rule 3: **{3,5}** → **({1,3,5} - {3,5})** means 3 & 5 → 1
   Confidence = support(1,3,5)/support(3,5) = 2/3 = 66.66% > 60%
   Rule 3 is selected

Apriori algorithm

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

Apriori algorithm

Applying Rules to Item set F3

1. **{1,3,5}**

   - ✓ Rule 4: **{1}** → **({1,3,5} - {1})** means 1 → 3 & 5
     Confidence = support(1,3,5)/support(1) = 2/3 = 66.66% > 60%
     Rule 4 is selected

   - ✓ Rule 5: **{3}** → **({1,3,5} - {3})** means 3 → 1 & 5
     Confidence = support(1,3,5)/support(3) = 2/4 = 50% <60%
     Rule 5 is rejected

   - ✓ Rule 6: **{5}** → **({1,3,5} - {5})** means 5 → 1 & 3
     Confidence = support(1,3,5)/support(3) = 2/4 = 50% < 60%
     Rule 6 is rejected

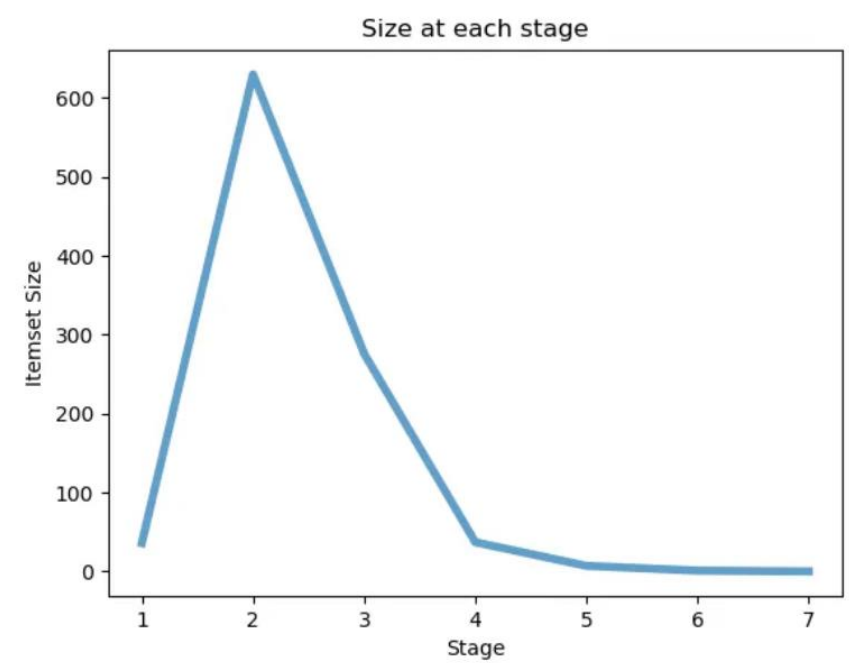
Apriori algorithm

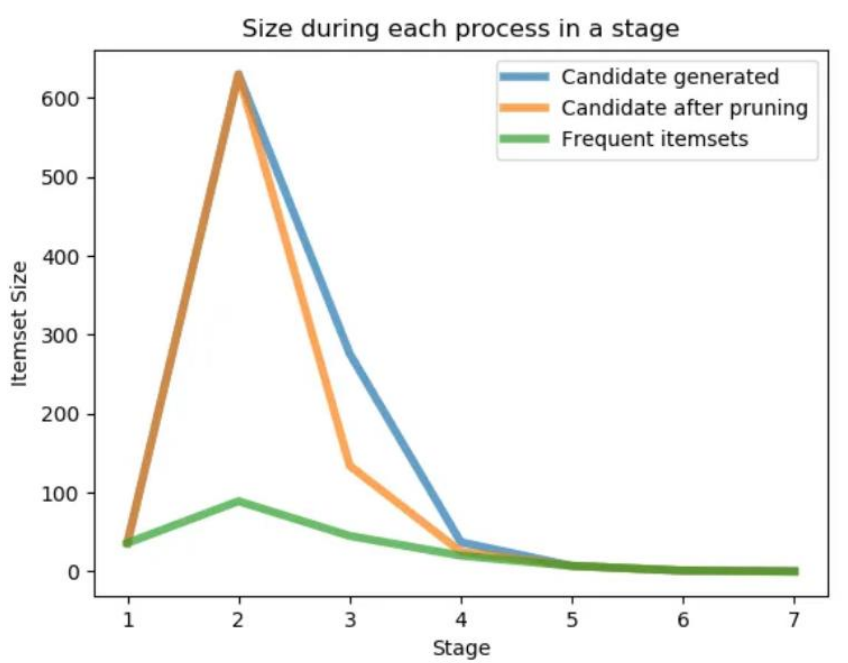
Disadvantages of the Apriori algorithm:

- The size of itemset from candidate generation could be extremely large
- Time consuming

Apriori algorithm

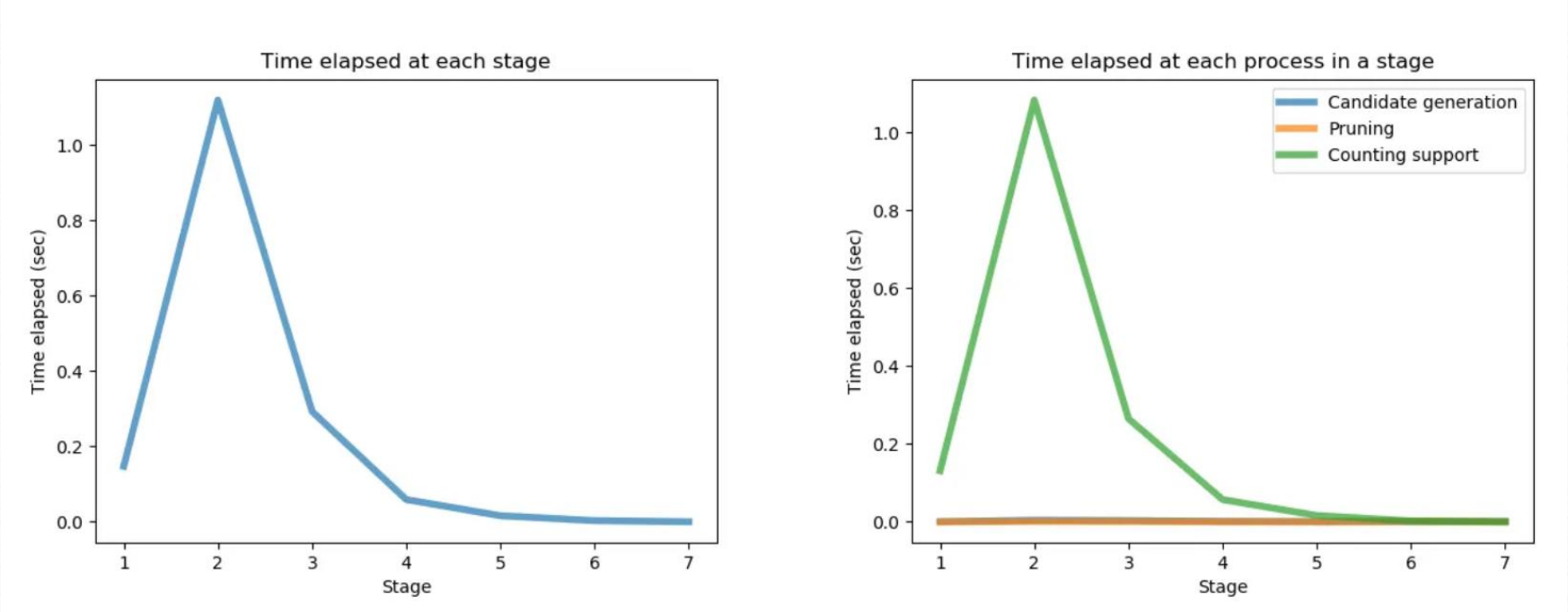
Disadvantages of the Apriori algorithm:





Apriori algorithm

Disadvantages of the Apriori algorithm:



More efficient algorithms have been introduced -> FP Growth algorithm