Tomáš Horváth

INTRODUCTION TO DATA SCIENCE

Lecture 3

Frequent Pattern Mining

Data Science and Engineering Department
Faculty of Informatics
ELTE University

Before we start

Clustering

- assigns objects with similar attributes to certain groups
- **object-attribute data**
  - rows (objects) = vectors of attributes
  - columns (attributes) = characteristics with well-defined domains
  - cells = concrete values of attributes for objects

Pattern mining

- Can we find such combinations of attributes that are common to many objects?
- Are there any significant (or confident) associations between these combinations?
- **transactional data**
  - rows (transactions) = sets of items
  - columns (items) = certain (not only physical) concepts
  - cells = connection of items and transactions

Transactional data

TID	Arabic	Indian	Mediterranean	Oriental	Fast Food
Andrew	✓		✓		
Bernhard		✓		✓	
Carolina		✓	✓		
Dennis	✓		✓	✓	
Eve				✓	✓
Fred	✓	✓			
Gwyneth		✓	✓	✓	
Hayden			✓	✓	✓
Irene		✓	✓	✓	
James		✓	✓	✓	

- *Is this a good format for such data?*
  - *If not, why?*
  - *If yes, when?*

Transactional data

TID	Items
Andrew	Indian, Mediterranean
Bernhard	Indian, Oriental, Fast Food
Carolina	Indian, Mediterranean, Oriental
Dennis	Arabic, Mediterranean
Eve	Oriental
Fred	Indian, Mediterranean, Oriental
Gwyneth	Arabic, Mediterranean
Hayden	Indian, Oriental, Fast Food
Irene	Indian, Mediterranean, Oriental
James	Arabic, Mediterranean

Let's define things. . .

- items $I = \{i_1, i_2, \ldots, i_m\}$
- transactional data $D = \{T \mid T \in \mathcal{P}(I),\ T \neq \emptyset\}$
- itemset $S \in \mathcal{P}(I)$
- support $sup_D : \mathcal{P}(I) \to \mathbb{R}$

$$sup_D(S) = \frac{\sum\limits_{T \in D} \delta(S \subseteq T)}{|D|}$$

  - $\delta(x) = 1$ if x is true, otherwise $\delta(x) = 0$

- itemset S is **frequent** if $sup_D(S) \geq \sigma$
  - minimum support threshold σ

Naïve algorithm

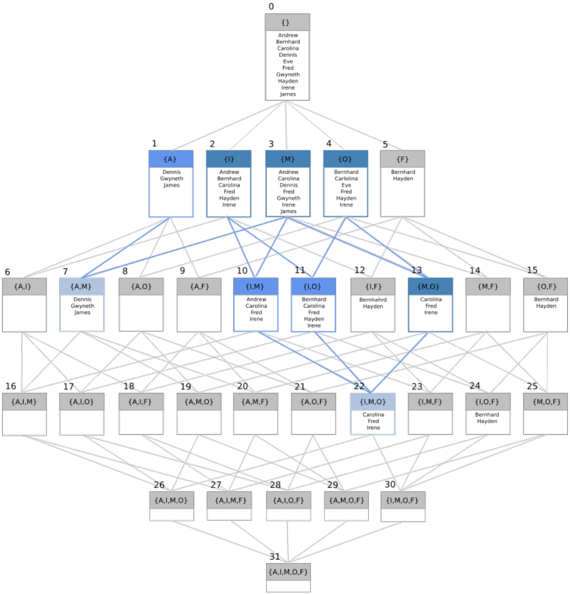
Can you propose it?

- *What would be its complexity?*

- if generating 1 itemset and count its support takes 1 ms

$\lvert I \rvert$	possible itemsets	Expected Runtime
5	31	31 milliseconds
10	1 023	>1 seconds
20	1 048 576	>17 minutes
30	1 073 741 823	>12 days
40	$> 10^{12}$	>34 years
50	$> 10^{15}$	>35 millennia

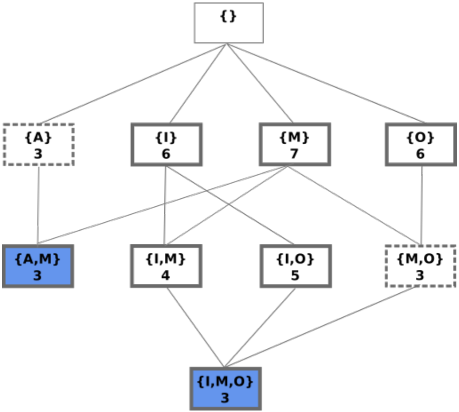
Itemset lattice



Closed and Maximal Frequent Itemsets

a frequent itemset is called

- **maximal** if none of its superset is frequent
- **closed** if none of its superset has the same support



Monotonicity

$$\forall S, Z \in \mathcal{P}(I) \ : \quad S \subseteq Z \ \Rightarrow \ sup_D(S) \geq sup_D(Z)$$

- each subset of a frequent itemset is frequent, too
- none of the supersets of a non-frequent itemset is frequent

Apriori

procedure $\textsc{Apriori}(D, I, \sigma)$

  $k \leftarrow 1$

  $F_k \leftarrow \{\ \{i\} \in I \mid sup_D(\{i\}) \geq \sigma\ \}$ ▷ frequent **1-itemsets**

  while $F_k \neq \emptyset$ **do**

    $C_{k+1} \leftarrow \text{GenerateCandidates}(F_k, k+1)$

    $F_{k+1} \leftarrow \{X \in C_{k+1} \mid sup_D(X) \geq \sigma\}$

    $k \leftarrow k + 1$

  return $\bigcup_k F_k$

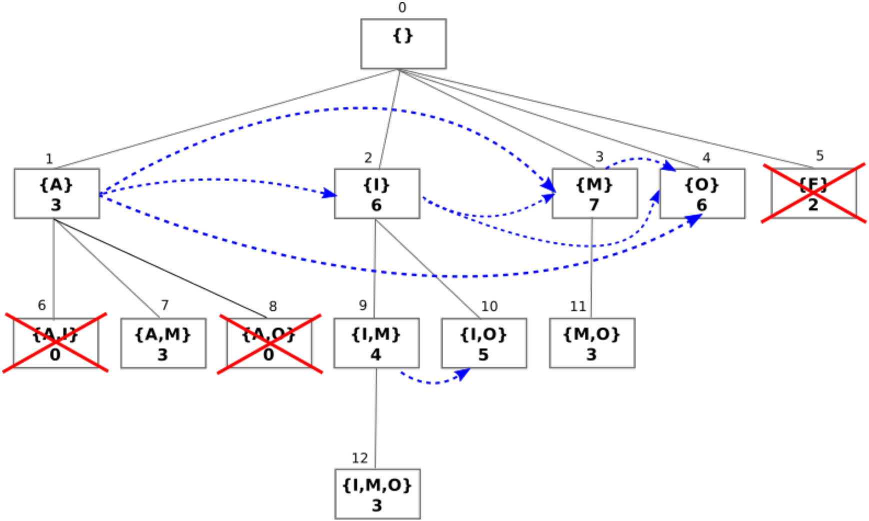
procedure $\textsc{GenerateCandidates}(F, k)$

  $C \leftarrow \{X \cup Y \mid X, Y \in F,\ |X \cup Y| = k\}$

  $C \leftarrow \{X \in C \mid (\forall Y \subset X)\ |Y| = k - 1 \Rightarrow Y \in F\}$

  return C

Enumeration Tree



Good to know

Apriori

- basically, all frequent itemset mining methods can be considered as variations of Apriori
    - various strategies to generate and explore the space of candidate itemsets defined by the enumeration tree
- *What is the bottleneck?*
    - **counting the support**

How to speed up counting the support?

- if the database fits into the memory
    - without passing all the transactions and computing if the candidate itemset is a subset of the transaction...
- if the database does not fit into the memory

Eclat

similar to Apriori

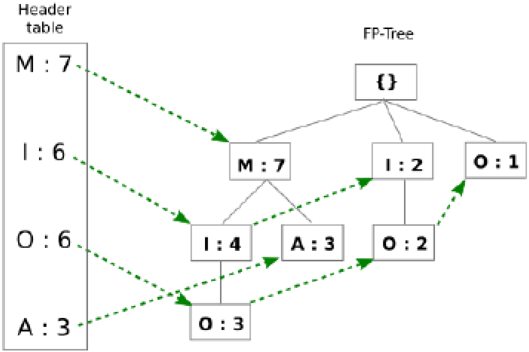
- **vertical data format** to count the support of an itemset
  - **intersection of TID-sets** assigned to items

Item	TID-set	Cardinality
Arabic	{Dennis, Gwyneth, James}	3
Indian	{Andrew, Bernhard, Carolina, Fred, Hayden, Irene}	6
Mediterranean	{Andrew, Carolina, Dennis, Fred, Gwyneth, Irene, James}	7
Oriental	{Bernhard, Carolina, Eve, Fred, Hayden, Irene}	6
Fast Food	{Bernhard, Hayden}	2

FP-Tree

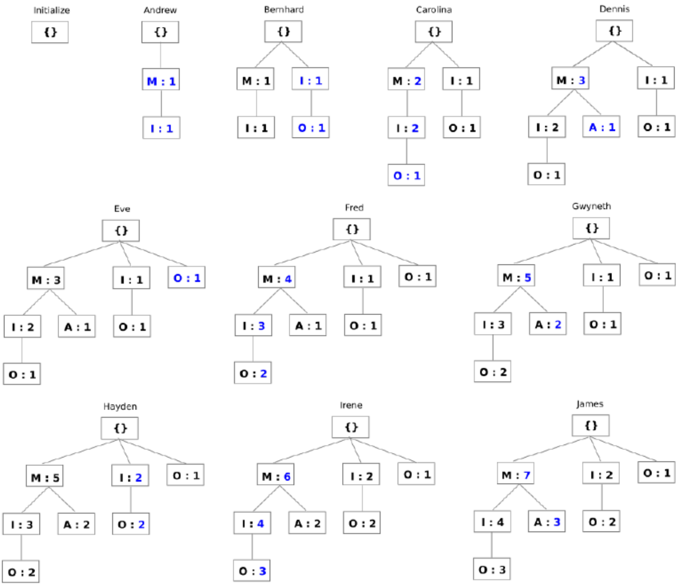
compact representation of the transactional data in a tree structure

- **fast** support count and itemset generation



- only **two passes of data** required
  1. all frequent items and their support are found
  2. items in each transaction are processed in a decreasing order according to their support

Building the FP-Tree



FP-Growth

Observation

- the set of all frequent itemsets can be divided into non-overlapping subsets of itemsets
  - containing item A
  - containing item O but not containing item A
  - having item I but not having items A and O and
  - an itemset having only the item M
  - *Why are they considered in decreasing order?*

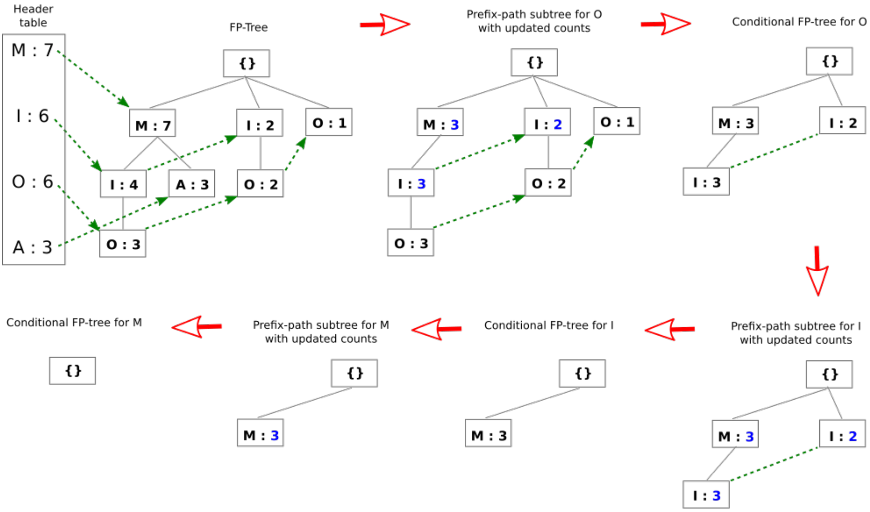
A recursive procedure using

- prefix-path subtrees
- consitional FP-Trees

FP-Growth

procedure FP-GROWTH(FPT, σ, IS) ▷ Current Itemset Suffix
  if FPT is a single path or empty **then**
    for all combination C of nodes **do**
      report all patterns $C \cup P$
  else
    for all $i \in FPT$ **do**
      generate pattern $P_i = \{i\} \cup P$
      report P_i as frequent
      use pointer chasing to extract conditional prefix paths for item i
      construct conditional FP-Tree FPT_i from conditional prefix paths after removing infrequent items
      if FPT_i is not empty **then**
        FP-Growth(FPT_i,σ, P_i)

FP-Growth


Header table
M : 7
I : 6
O : 6
A : 3
FP-Tree
{}
M : 7
I : 2
O : 1
I : 4
A : 3
O : 2
O : 3
Prefix-path subtree for O with updated counts
{}
M : 3
I : 2
O : 1
I : 3
O : 2
O : 3
Conditional FP-tree for O
{}
M : 3
I : 2
I : 3
Prefix-path subtree for I with updated counts
{}
M : 3
I : 2
I : 3
Conditional FP-tree for I
{}
M : 3
Prefix-path subtree for M with updated counts
{}
M : 3
Conditional FP-tree for M
{}

Associations

Implications

- **"if-then"** relations
- if $S_1 \subset T \in D$ implies $S_2 \subset T$ then S_1 and S_2 are **associated** in T

Association Rule

$$A \Rightarrow C$$

- $A \in \mathcal{P}(I)$ is the **antecedent**
- $C \in \mathcal{P}(I)$ is the **consequent**
- $A \cap C = \emptyset$

The quality of a rule

Given D, how do we measure the quality of $A \Rightarrow C$?

- **Support**

$$sup_D(A \Rightarrow C) = sup_D(A \cup C)$$

  - *Is it a good measure?*
    - $sup_D(\{\text{A}\} \Rightarrow \{\text{M}\}) = sup_D(\{\text{M}\} \Rightarrow \{\text{A}\})$
  - a kind of a **quantitative measure**

- **Confidence**

$$conf_D(A \Rightarrow C) = \frac{sup_D(A \cup C)}{sup_D(A)}$$

  - measures the reliability of the rule
    - $1 = conf_D(\{\text{A}\} \Rightarrow \{\text{M}\}) \neq conf_D(\{\text{M}\} \Rightarrow \{\text{A}\}) = 0.43$
  - a kind of a **qualitative measure**

Mining Association Rules

Given D, I, σ and θ, where

- σ is the minimum support threshold
- θ is the minimum confidence threshold

the goal is to find all association rules $A \Rightarrow C$ such that

- $sup_D(A \Rightarrow C) \geq \sigma$
- $conf_D(A \Rightarrow C) \geq \theta$

How much association rules can we generate from I?

- $3^{|\mathcal{I}|} - 2^{|\mathcal{I}|+1} + 1$
  - number of different splits into A and C
  - minus rules with empty antecedents or consequents
    - twice the number of itemsets
  - minus a rule with empty antecedent and consequent (empty rule)
  - $= 3^{|\mathcal{I}|} - 2(2^{|\mathcal{I}|} - 1) - 1$

Naïve algorithm and some speed up...

1. Mine frequent itemsets meeting the σ threshold
   - *Why are we utilizing frequent itemsets?*
2. Generate association rules meeting the θ threshold
   - $2^{|I|} - 2$ association rules for each itemset $|I|$

Confidence-based pruning

$$conf_D(X \Rightarrow Y - X) < \theta \;\Rightarrow\; conf_D(X' \Rightarrow Y - X') < \theta$$

- where $X' \subset X$
- $sup_D(X') \geq sup_D(X) \;\Rightarrow\; \frac{sup_D(Y)}{sup_D(X)} \geq \frac{sup_D(Y)}{sup_D(X')}$

Y={I,M,O}, X={I,O}, $\theta = 0.75$

$conf_D(X \Rightarrow Y - X) = conf_D(\{\text{I,O}\} \Rightarrow \{\text{M}\}) = 3/5 = 0.6$

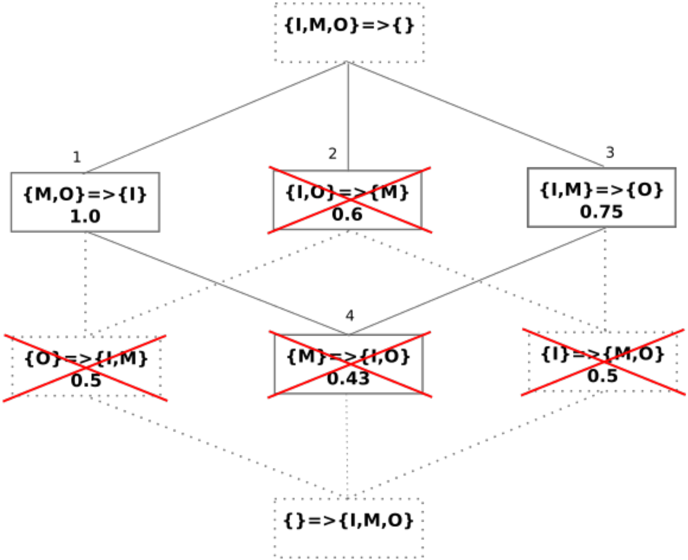
$X' = \{\text{I}\}:\; conf_D(X' \Rightarrow Y - X') = conf_D(\{\text{I}\} \Rightarrow \{\text{M,O}\}) = 3/6 = 0.5$

$X' = \{\text{O}\}:\; conf_D(X' \Rightarrow Y - X') = conf_D(\{\text{O}\} \Rightarrow \{\text{M,I}\}) = 3/6 = 0.5$

Generating Association Rules

INPUT Z, θ $\triangleright$ Z – frequent itemset
for all item i in Z **do**
    Construct a rule $Z - \{i\} \Rightarrow \{i\}$
    if $confidence(Z - \{i\} \Rightarrow \{i\}) \geq min\_conf$ **then**
        output $Z - \{i\} \Rightarrow \{i\}$
        add $\{i\}$ to the set C_1
Set $k = 2$
repeat
    for all V $(|V| = k)$ generated by joining $A, B \in C_{k-1}$ **do**
        Construct a rule $Z - V \Rightarrow V$
        if $confidence(Z - V \Rightarrow V) \geq min\_conf$ **then**
            output $Z - V \Rightarrow V$
            add V to the set C_k
    $k = k + 1$
until $k < |Z| - 1$

Association rules lattice



Behind support and confidence

Cross-support pattern

- containing low-support items together with high-support items
  - can be interesting
  - but also spurious
    - the items it contains are weakly correlated in the transactions

Support ratio

$$sup\_ratio_D(P) = \frac{min\{sup_D(i_1), sup_D(i_2), \ldots, sup_D(i_k)\}}{max\{sup_D(i_1), sup_D(i_2), \ldots, sup_D(i_k)\}}$$

- $i_1, i_2, \ldots, i_k \in P$

Behind support and confidence

Contingency table

frequency counts		Y: Present	Y: Absent	Total
X	Present	12	4	16
	Absent	68	16	84
	Total	80	20	100

- $sup_D(X \Rightarrow Y) = 0.12$
- $conf_D(X \Rightarrow Y) = 0.12/0.16 = 0.75$

How about mutual influence between X and Y?

- $sup_D(Y) = 0.8$, regardless if X is present
- thus, the occurence of X negatively influences the occurence of Y

Behind support and confidence

Lift

- high confidence and good support does not necessarily imply cause and effect between X and Y

$$lift_D(X \Rightarrow Y) = \frac{conf_D(X \Rightarrow Y)}{sup_D(Y)}$$

- $lift_D(X \Rightarrow Y) > 1$
  - positive correlation between the X and Y
  - i.e. the occurrence of X has a positive effect on the occurrence of Y
- $lift_D(X \Rightarrow Y) < 1$
  - negative correlation between the X and Y
  - i.e. the occurrence of X has a negative effect on the occurrence of Y
- $lift_D(X \Rightarrow Y)$ is near 1
  - no correlation between the X and Y
  - i.e. the occurrence of X has almost no effect on the occurrence of Y

Behind support and confidence

Simpson's paradox

- certain correlations between pairs of itemsets (antecedents and consequents of rules) appearing in different groups of data may disappear or be reversed when these groups are combined

Group A		Y Yes	Y No	Total
X	Yes	20	5	25
	No	105	150	255
	Total	125	155	280

Group B		Y Yes	Y No	Total
X	Yes	100	150	250
	No	25	245	270
	Total	125	395	520

Combined Groups A and B		Y Yes	Y No	Total
X	Yes	120	155	275
	No	130	395	525
	Total	250	550	800

- A: $conf_D(X \Rightarrow Y) = 0.8$, $lift_D(X \Rightarrow Y) = 1.79$
- B: $conf_D(Y \Rightarrow X) = 0.8$, $lift_D(Y \Rightarrow X) = 1.66$
- A and B: $conf_D(X \Rightarrow Y) = 0.44$, $conf_D(Y \Rightarrow X) = 0.48$

Sequential Patterns

Transaction ID	Consecutive events (itemsets) recorded in time
1	$\langle \{a,b\}, \{a,b,c\}, \{a,c,d,e\}, \{b,f\} \rangle$
2	$\langle \{a\}, \{a,b,f\}, \{a,c,e\} \rangle$
3	$\langle \{a\}, \{c\}, \{b,e,f\}, \{a,d,e\}, \{e,f\} \rangle$
4	$\langle \{e,d\}, \{c,f\}, \{a,c,f\}, \{a,b,d,e,f\} \rangle$
5	$\langle \{b,c\}, \{a,e,f\} \rangle$

- $s_1 = \langle X_1, X_2, \ldots, X_n \rangle$ and $s_2 = \langle Y_1, Y_2, \ldots, Y_m \rangle$
  - $n \leq m$
- s_1 is a subsequence of s_2 if there exists $1 \leq i_1 < i_2 < \cdots < i_n \leq m$ such that $X_1 \subseteq Y_{i_1}, X_2 \subseteq Y_{i_2}, \ldots, X_n \subseteq Y_{i_n}$

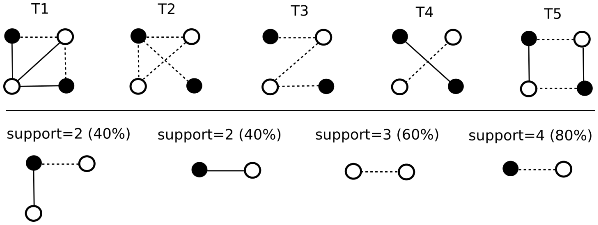
support is counted as in case of itemsets

- $sup_D(\langle \{a\}, \{f\} \rangle) = 4/5 = 0.8$

similarly, **closed** and **maximal** frequent sequences

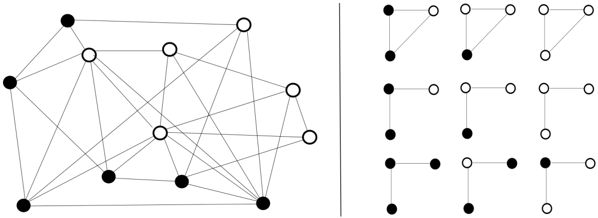
Graph patterns

transaction-based (for $\sigma = 0.4$)



graph-based (for $\sigma = 2$)

- support count has to consider overlapping patterns




That's all Folks!
Thanks for your attention

References

- Charu C. Aggarwal and Jiawei Han (2014). *Frequent Pattern Mining*. Springer Publishing Company, Incorporated.
- Pang-Ning Tan, Michael Steinbach, and Vipin Kumar (2005). *Introduction to Data Mining*, (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Frequent Itemset Mining Implementations Repository. `http://fimi.ua.ac.be/`
- EasyMiner – pattern mining in the browser. `http://www.easyminer.eu/` `https://github.com/KIZI/EasyMiner`

Homework

1. Choose a dataset and three implementation from the Frequent Itemset Mining Implementations Repository
   - Apriori, Eclat and FP-Growth
2. find frequent, closed and maximal itemsets from the chosen data
   - for various minimum support thresholds σ (e.g. 0.2, 0.4, 0.6 and 0.8)
3. find association rules in the data
   - for various combinations of minimum support thresholds σ and minimum confidence thresholds θ

Questions?

tomas.horvath@inf.elte.hu