Data Mining and Knowledge Discovery

Petra Kralj Novak

January 13, 2020

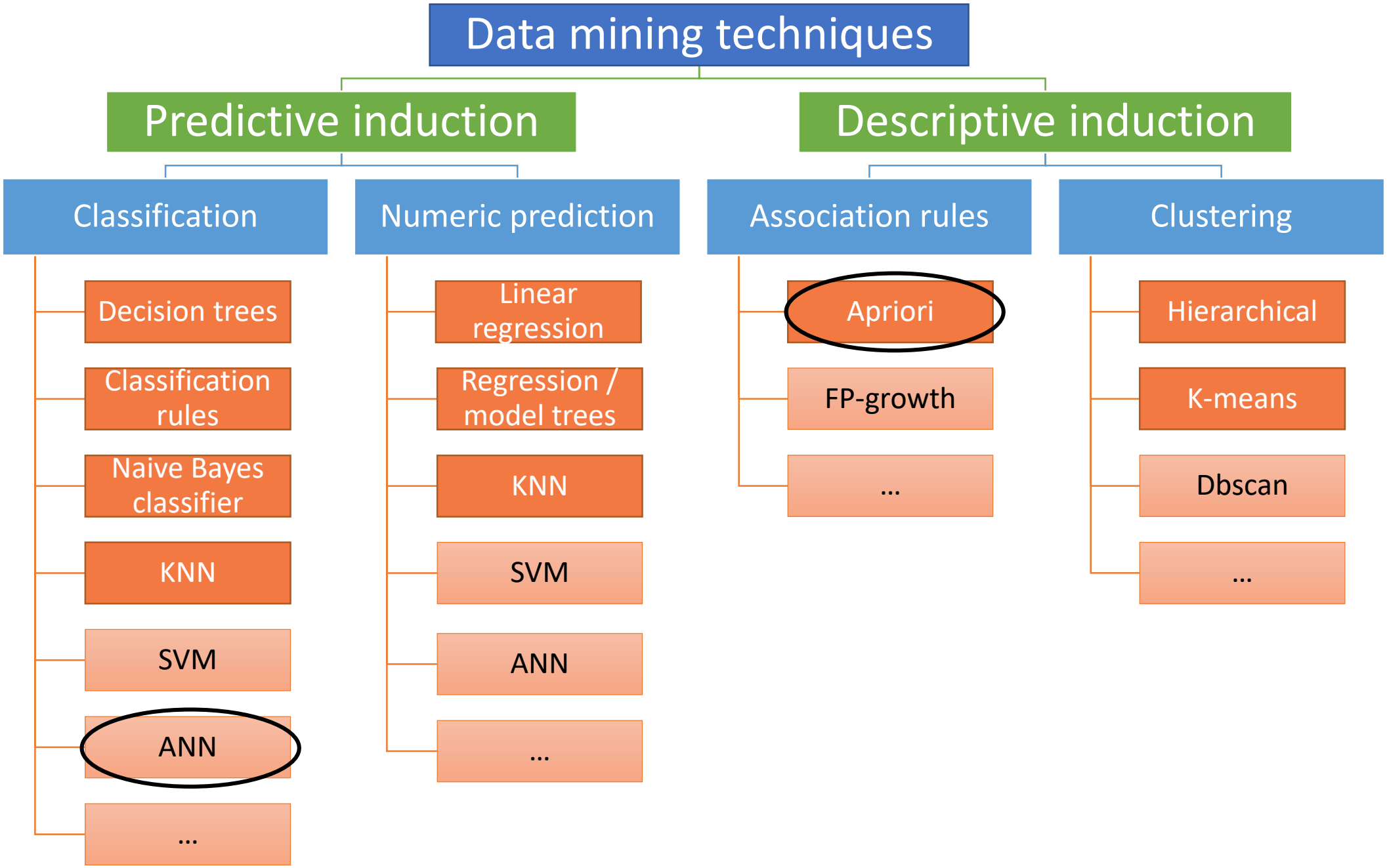
http://kt.ijs.si/petra_kralj/dmtm2.html

In previous episodes …

- 23-Oct-19
  - **Data**, data types
  - Interactive **visualization** (Orange)
  - **Classification** with decision trees (root, leaves, rules, entropy, info gain, TDIDT, ID3)
- 6-Nov-19
  - Classification: train – test (evaluate) - apply
  - **Decision tree** example (on blackboard)
  - Decision tree language bias (Orange workflow)
  - Homework:
    - InfoGain questions
    - Orange workflow
    - Reading “Classification and regression by randomForest” by Liaw & Wiener, 2002
- 25-Nov-19
  - **Evaluation**:
    - Methods: train-test, leave-one-out, randomized sampling,…
    - Metrics: accuracy, confusion matrix, precision, recall, F1,…
  - Homework: XOR, questions, precision and recall

… continued …

- 2-Dec-19
  - Evaluation: **ROC**
  - **Naïve Bayes** classifier
  - Probability estimation: relative frequency, Laplace estimate
  - **Numeric prediction (**linear regression, regression tree, model tree, KNN) and **evaluation** (MSE, MAE, RMSE)
- 16-Dec-19
  - Clustering
  - K-means
  - Silhouette coefficient
  - Agglomerative clustering, dendrogram
  - DB-scan
  - Similarity, distance


Data mining techniques
Predictive induction
Descriptive induction
Classification
Numeric prediction
Association rules
Clustering
Decision trees
Classification rules
Naive Bayes classifier
KNN
SVM
ANN
…
Linear regression
Regression / model trees
KNN
SVM
ANN
…
Apriori
FP-growth
…
Hierarchical
K-means
Dbscan
…

Or harried dads rewarding themselves with impulse buys

Association rules

Association rules – Market basket analysis

- What do customers buy together?
- Which items imply the purchase of other items?

* Terminology from market basket analysis (transactions, items, itemsets, …)

- Determine associations between groups of items bought by customers.
- No predefined target variable(s).
- Find interesting, useful patterns and relationships.
- Data mining, business intelligence.

Confidence and support

- The dataset consists of **n** transactions
- We have an association rule A→ B

The **support** of an itemset A is defined as the fraction of the transactions in the database T = {T1 . . . Tn} that contain A as a subset.

$$supp(A) = \frac{|A|}{n}$$

$$supp(A \rightarrow B) = \frac{|A \wedge B|}{n}$$

The **confidence** of the rule A→ B is the conditional probability of A and B occurring in a transaction, given that the transaction contains A.

$$conf(A \rightarrow B) = \frac{|A \wedge B|}{|A|} = P(B|A)$$

Exercise

tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

$$supp(A) = \frac{|A|}{n}$$

$$supp(A \rightarrow B) = \frac{|A \wedge B|}{n}$$

$$conf(A \rightarrow B) = \frac{|A \wedge B|}{|A|} = P(B|A)$$

Supp ({Bread}) =

Supp ({Milk, Yogurt}) =

Conf ({Milk} → {Yogurt}) =

Conf ({Yogurt} → {Milk}) =

Charu C. Aggarwal : Data Mining: The Textbook (2015, pg. 95)

Association rules

- Rules **A → B**, where A and B are conjunctions of items
- Task: Find **all** association rules that satisfy the minimum support and minimum confidence constraints

- **Support**:

$$supp(A \rightarrow B) = \frac{|A \wedge B|}{n}$$

- **Confidence**:

$$conf(A \rightarrow B) = \frac{|A \wedge B|}{|A|} = P(B|A)$$

Hard problem

- In practice
  - Millions of transactions
  - Many (thousands) of items
- Too many possible combinations
  - 1000 items for sale → 2^{1000} - 1 candidate market baskets
- Solution
  - ***Apriori*** *algorithm*

Frequent itemsets: intuition

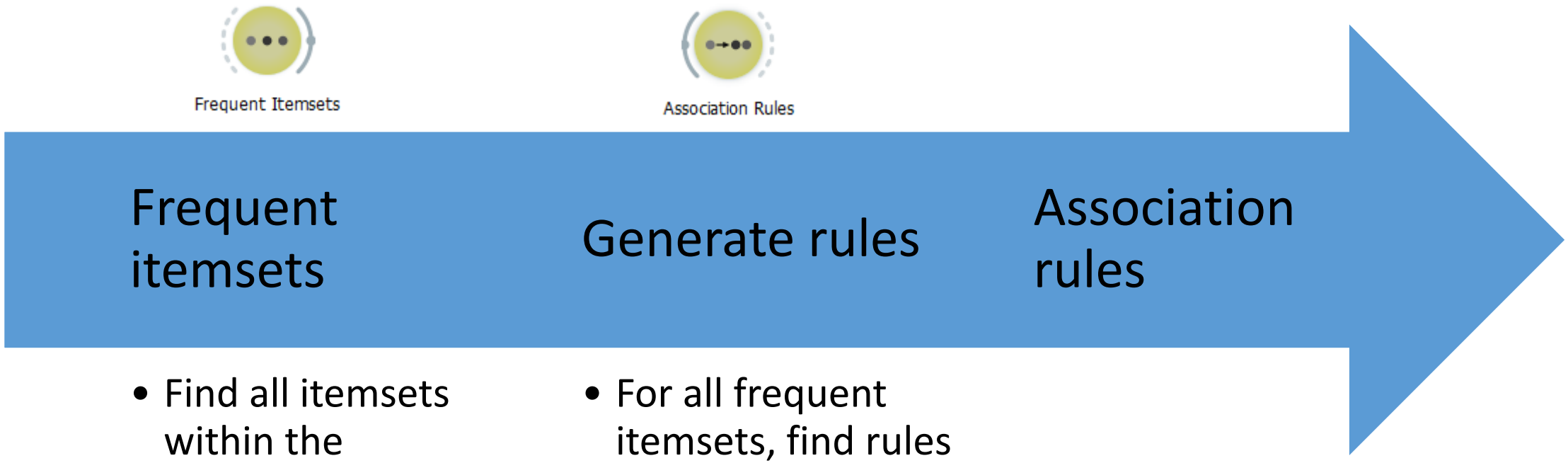
- We have ***n*** transactions containing (at least) {gloves, scarf and hat}
- What can we say about the number of transaction containing {gloves and scarf}?

*At least **n**.*

- The **anti-monotone property of support:** if we drop out an item from an itemset, support value of new itemset generated will either be the same or will increase.

$$\forall A, B: \ A \subseteq B \Rightarrow supp(A) \geq supp(B)$$

Apriori



- Find all itemsets within the *minSupport* constraint
- For all frequent itemsets, find rules which satisfy the *minConfidence* constraint

*Frequent itemsets = large itemsets, sometimes also frequent patterns

Apriori

Create L_1 = set of supported itemsets of cardinality one
Set k to 2
while $(L_{k-1} \neq \emptyset)$ {
    Create C_k from L_{k-1}
    Prune all the itemsets in C_k that are not
        supported, to create L_k
    Increase k by 1
}
The set of all supported itemsets is $L_1 \cup L_2 \cup \cdots \cup L_k$

(Generates C_k from L_{k-1})
Join Step
Compare each member of L_{k-1}, say A, with every other member, say B, in turn. If the first $k-2$ items in A and B (i.e. all but the rightmost elements of the two itemsets) are identical, place set $A \cup B$ into C_k.
Prune Step
For each member c of C_k in turn {
Examine all subsets of c with $k-1$ elements
Delete c from C_k if any of the subsets is not a member of L_{k-1}
}

Apriori: Frequent itemset mining

Create L_1 = set of supported itemsets of cardinality one
Set k to 2
while $(L_{k-1} \neq \emptyset)$ {
    Create C_k from L_{k-1}
    Prune all the itemsets in C_k that are not
        supported, to create L_k
    Increase k by 1
}
The set of all supported itemsets is $L_1 \cup L_2 \cup \cdots \cup L_k$

- The items in the sets should be ordered (alphabetically, …)

Apriori: constructing the next level from the previous one

- Since items in the sets are ordered (alphabetically, …)
- Join Step:
  - Merge sets that have all the elements the same except for the rightmost one
- Prune Step:
  - Remove the set if any of its subsets are not on the previous level

(Generates C_k from L_{k-1})

Join Step

Compare each member of L_{k-1}, say A, with every other member, say B, in turn. If the first $k-2$ items in A and B (i.e. all but the rightmost elements of the two itemsets) are identical, place set $A \cup B$ into C_k.

Prune Step

For each member c of C_k in turn {
Examine all subsets of c with $k-1$ elements
Delete c from C_k if any of the subsets is not a member of L_{k-1}
}

Rules from frequent itesmets

- Generate rules with a certain confidence
- All the counts we need are in the lattice (no database scanning)
- Confidence of rules generated from the same itemset has an anti-monotone property
- No need to check all the rules, since

Conf ({A,B} → {C}) >= Conf ({A} → {B,C})
Conf({A,B,C} → {D}) ≥ Conf({A,B} → {C,D}) ≥ Conf({A} → {B,C,D})

*In general, confidence does not have an anti-monotone property: Conf(ABC → D) can be larger or smaller than Conf(AB → D)

Charu C. Aggarwal : Data Mining: The Textbook (2015), pg. 98

Exercise: Association rules

Generate frequent itemsets with support at least 2/6 and confidence at least 75%.

Items: **A**=apple, **B**=banana, **C**=coca-cola, **D**=doughnut

- Client 1 bought: A, B, C, D
- Client 2 bought: B, C
- Client 3 bought: B, D
- Client 4 bought: A, C
- Client 5 bought: A, B, D
- Client 6 bought: A, B, C

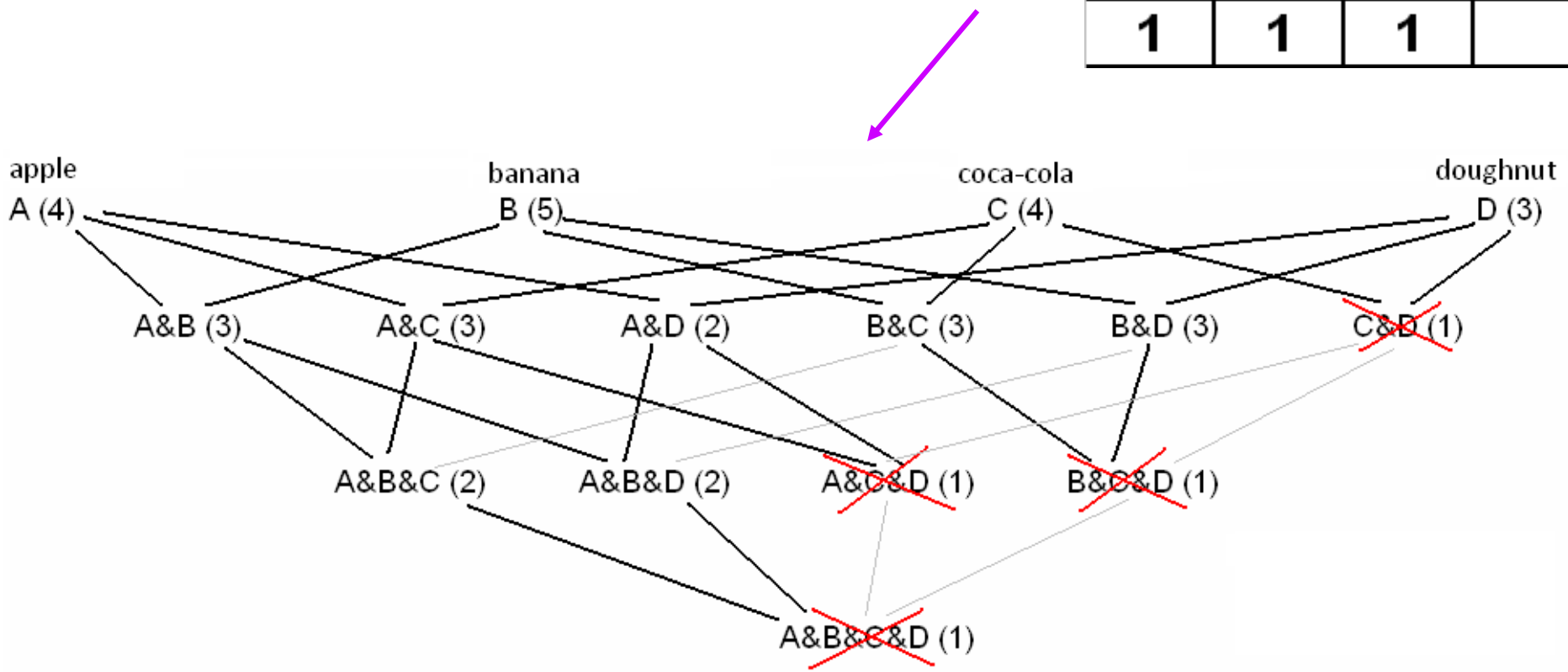
Exercise: Frequent itemsets

To ease the counting, we transcribe into a binary representation.

A	B	C	D
1	1	1	1
	1	1	
	1		1
1		1	
1	1		1
1	1	1	

Frequent itemsets (= large itemsets)

A	B	C	D
1	1	1	1
	1	1	
	1		1
1		1	
1	1		1
1	1	1	



Association rules …

Itemset (count)	Rule	Support	Confidence	Over threshold
AB (3)	A → B	3/6	3/4 = 75%	✔
	B → A	3/6	3/5 = 60%	
AC (3)	A → C	3/6	3/4 = 75%	✔
	C → A	3/6	3/4 = 75%	✔
AD (2)	A → D	2/6	2/4 = 50%	
	D → A	2/6	2/3 = 67%	
BC (3)	B → C	3/6	3/5 = 60%	
	C → B	3/6	3/4 = 75%	✔
BD (3)	B → D	3/6	3/5 = 60%	
	D → B	3/6	3/3 = 100%	✔

… association rules

ABC (2)	AB → C	2/6	2/3 = 67%	
	AC → B	2/6	2/3 = 67%	
	BC → A	2/6	2/3 = 67%	
	A → BC	We do not generate these rules because transferring members of a supported itemset from the left-hand side of a rule to the right-hand side cannot increase the value of rule confidence.		
	B → AC			
	C → AB			
ABD (2)	AB → D	2/6	2/3 = 67%	
	AD → B	2/6	2/2 = 100%	✔
	BD → A	2/6	2/3 = 67%	
	A → BD	2/6	2/4 = 50%	
	B → AD	We do not generate this rule.		
	D → AB	2/6	2/3 = 67%	

Lift

- The lift of rule **L → R** measures how many more times the items in L and R occur together in transactions than would be expected if the itemsets L and R were statistically independent.

$$\text{lift}(L \to R) = \frac{\text{support}(L \cup R)}{\text{support}(L) \times \text{support}(R)}$$

$$\text{lift}(L \to R) = \text{lift}(R \to L)$$

Leverage

- The leverage of rule **L → R** is the difference between the support for L ∪ R (i.e. the items in L and R occurring together in the database) and the support that would be expected if L and R were independent.

$$\text{leverage}(L \rightarrow R) = \text{support}(L \cup R) - \text{support}(L) \times \text{support}(R)$$

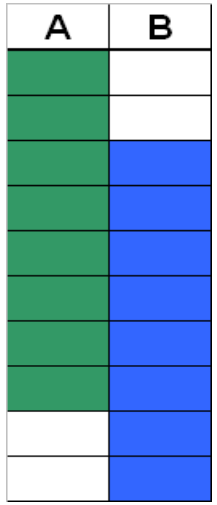
Literature

- Max Bramer: Principles of data mining (2007)
  - 13. Association Rule Mining II

- What is the "true story" about using data mining to identify a relation between sales of beer and diapers? http://www.dssresources.com/newsletters/66.php

Homework

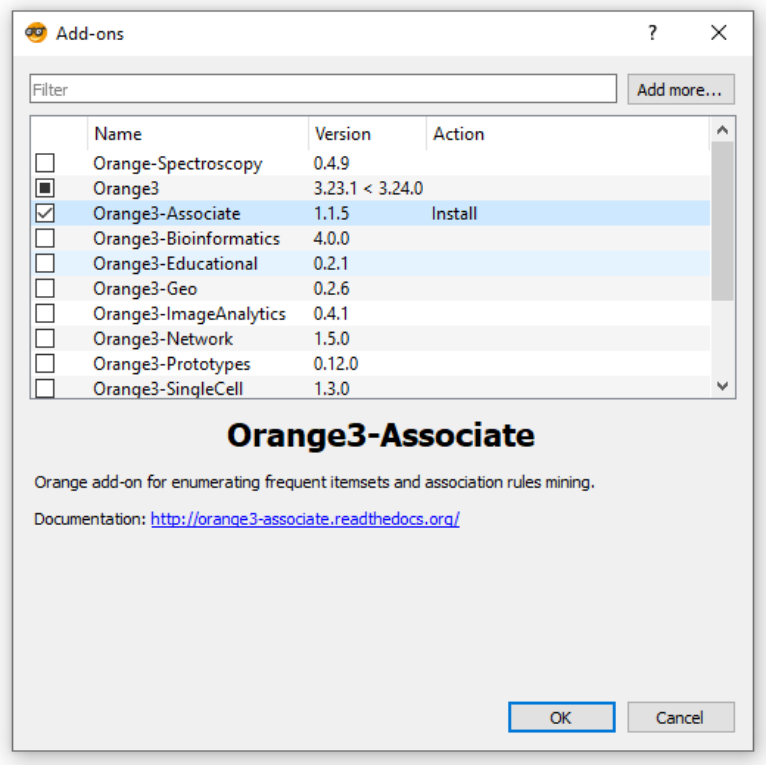
1. Transformation of an attribute-value dataset to a transaction dataset.
2. What would be the association rules for a dataset with two items A and B, each of them with support 80% and appearing in the same transactions as rarely as possible?
   a. minSupport = 50%, min conf = 70%
   b. minSupport = 20%, min conf = 70%
3. What if we had 4 items: A, ¬A, B, ¬ B

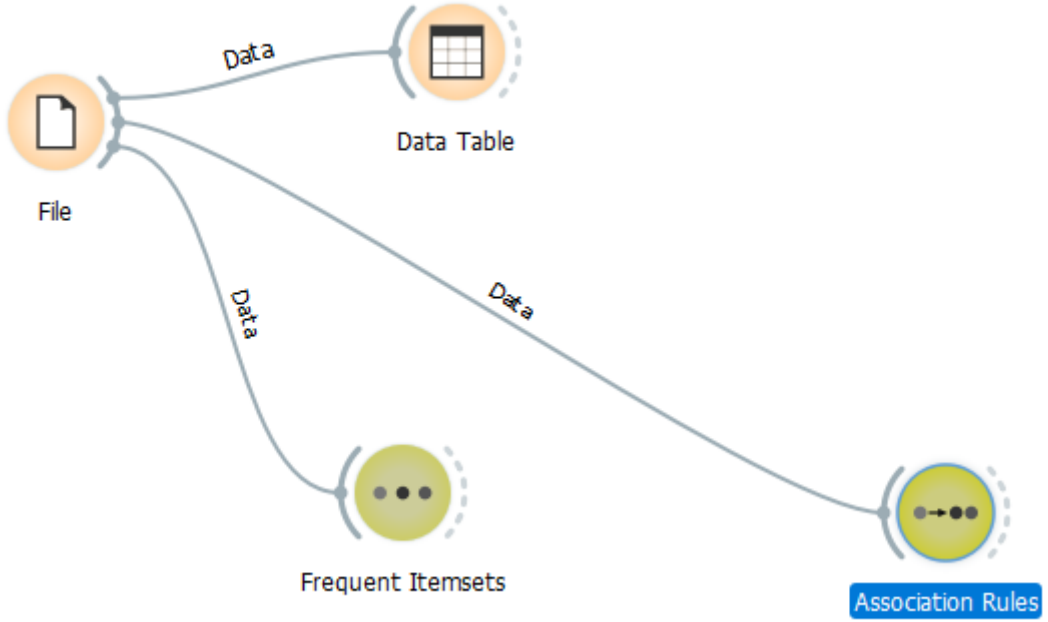


4. Compare decision trees and association rules regarding handling an attribute like “PersonID”. What about attributes that have many values (eg. Month of year)

Association rules: Orange workflow

1. Install Add-on Orange3-Associate





* Start with a small minSupport and we increase it gradually (to avoid running out of memory)

Association rules quality measures in Orange

Supp	Conf	Covr	Strg	Lift	Levr	Antecedent		Consequent
0.050	0.178	0.283	0.618	1.017	0.001	Fresh Vegetables	→	Fresh Fruit
0.050	0.287	0.175	1.619	1.017	0.001	Fresh Fruit	→	Fresh Vegetables

- **support, confidence, lift**, **leverage**
- **coverage**: how often antecedent items are found in the data set (support of antecedent/data)
- **strength**: (support of consequent/support of antecedent)

Lab exercise

Datasets

- https://biolab.si/core/foodmart.basket
- https://github.com/digizeph/data_mining/blob/master/data/FoodMart.csv
- http://file.biolab.si/datasets/voting.tab

1. Compare the two datasets (files)
2. Generate frequent itemsets and association rules for both datasets. What is the difference?
3. Frequent itemsets and association rules for „Voting.tab“