UNIVERSITÄT MANNHEIM



Heiko Paulheim

Outline

- 1. What is Association Analysis?
- 2. Frequent Itemset Generation
- 3. Rule Generation
- 4. Interestingness Measures
- 5. Handling Continuous and Categorical Attributes

Association Analysis

- First algorithms developed in the early 90s at IBM by Agrawal & Srikant
- Motivation
 - Availability of barcode cash registers







11/25/20 Heiko Paulheim

Association Analysis

- initially used for Market Basket Analysis
 - to find how items purchased by customers are related
- later extended to more complex data structures
 - sequential patterns (see Data Mining II)
 - subgraph patterns
- and other application domains
 - life science
 - social science
 - web usage mining

Simple Approaches

- To find out if two items x and y are bought together, we can compute their correlation
- E.g., Pearson's correlation coefficient:

$$\frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2} \sqrt{\sum (y_i - \overline{y})^2}}$$

- Numerical coding:
 - 1: item was bought
 - 0: item was not bought
- \overline{x} : average of x (i.e., how often x was bought)



Correlation Analysis in RapidMiner

| Correlation Matrix (Correlation Matrix) | | | | | | | | | | |
|---|--------|--------|--------|--------|--------|-------------|--------|--------|--------|--------|
| Table View O Pairwise Table O Plot View Annotations | | | | | | | | | | |
| Attributes ThinkPad X Asus EeePC HP Laserjet 2 GB DDR3 8 GB DDR3 Lenovo Tab Netbook-Sc HP CE50 T LT Laser M LT Minimaus | | | | | | LT Minimaus | | | | |
| ThinkPad X2 | 1 | -1 | 0.356 | -0.816 | 0.612 | 0.583 | -0.667 | 0.356 | 0.167 | -0.408 |
| Asus EeePC | -1 | 1 | -0.356 | 0.816 | -0.612 | -0.583 | 0.667 | -0.356 | -0.167 | 0.408 |
| HP Laserjet | 0.356 | -0.356 | 1 | -0.218 | -0.327 | 0.356 | -0.535 | 1 | -0.089 | -0.655 |
| 2 GB DDR3 | -0.816 | 0.816 | -0.218 | 1 | -0.500 | -0.816 | 0.816 | -0.218 | 0 | 0.200 |
| 8 GB DDR3 | 0.612 | -0.612 | -0.327 | -0.500 | 1 | 0.102 | -0.408 | -0.327 | 0.102 | 0 |
| Lenovo Tabl | 0.583 | -0.583 | 0.356 | -0.816 | 0.102 | 1 | -0.667 | 0.356 | -0.250 | 0 |
| Netbook-Sch | -0.667 | 0.667 | -0.535 | 0.816 | -0.408 | -0.667 | 1 | -0.535 | 0.167 | 0.408 |
| HP CE50 To | 0.356 | -0.356 | 1 | -0.218 | -0.327 | 0.356 | -0.535 | 1 | -0.089 | -0.655 |
| LT Laser Ma | 0.167 | -0.167 | -0.089 | 0 | 0.102 | -0.250 | 0.167 | -0.089 | 1 | -0.408 |
| LT Minimaus | -0.408 | 0.408 | -0.655 | 0.200 | 0 | 0 | 0.408 | -0.655 | -0.408 | 1 |

Correlation Analysis in Python

• e.g., using Pandas:

```
import seaborn as sns
corr = dataframe.corr()
sns.heatmap(corr)
```



11/25/20 Heiko Paulheim

Association Analysis

 Given a set of transactions, find 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, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

Examples of Association Rules

{Diaper} \rightarrow {Beer}, {Milk, Bread} \rightarrow {Eggs,Coke}, {Beer, Bread} \rightarrow {Milk},

> → denotes co-occurence, not causality!

Correlation vs. Causality



http://xkcd.com/552/

Definition: Frequent Itemset

- Itemset
 - A collection of one or more items
 - Example: {Milk, Bread, Diaper}
 - k-itemset
 - An itemset that contains k items
- Support (s)
 - Frequency of occurrence of an itemset
 - e.g. s({Milk, Bread, Diaper}) = 2/5
- Frequent Itemset
 - An itemset w/ support ≥ a minimum support threshold (minsup)

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

Definition: Association Rule

- Association Rule
 - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
- Interpretation: when X occurs, Y occurs with a certain probability

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

- More formally, it's a *conditional probability*
 - P(Y|X) given X, what is the probability of Y?
- Known as confidence (c)
 - e.g., for {Bread, Milk} \rightarrow {Diaper}, the probability is 2/3

Definition: Evaluation Metrics

- Given the rule {Milk, Diaper} \rightarrow {Beer}
- Support:
 - Fraction of total transactions which contain both X and Y

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\mathsf{T}|} = \frac{2}{5} = 0.4$$

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

- Confidence:
 - Fraction of transactions containing X which also contain Y

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
 - support \geq minsup threshold
 - confidence ≥ minconf threshold
- minsup and minconf are provided by the user
- Brute-force approach:
 - List all possible association rules
 - Compute the support and confidence for each rule
 - Remove rules that fail the minsup and minconf thresholds
 - → Computationally prohibitive due to large number of candidates!

Mining Association Rules

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

Examples of Rules:

- {Milk, Diaper} \rightarrow {Beer} (s=0.4, c=0.67)
- {Milk, Beer} \rightarrow {Diaper} (s=0.4, c=1.0)
- {Diaper, Beer} \rightarrow {Milk} (s=0.4, c=0.67)
- {Beer} \rightarrow {Milk, Diaper} (s=0.4, c=0.67)
- {Diaper} \rightarrow {Milk, Beer} (s=0.4, c=0.5)
- {Milk} \rightarrow {Diaper, Beer} (s=0.4, c=0.5)

Observations

- All the above rules are partitions of the same itemset, i.e. {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support

$$s(X \rightarrow Y) := \frac{|X \cup Y|}{|T|}$$

- but can have different confidence
- \rightarrow we may decouple the support and confidence requirements

Apriori Algorithm: Basic Idea

- Two-step approach
- First: Frequent Itemset Generation
 - Generate all itemsets whose support \geq minsup
- Second: Rule Generation
 - Generate high confidence rules from each frequent itemset
 - where each rule is a binary partitioning of a frequent itemset
- However: Frequent itemset generation is still computationally expensive....

Frequent Itemset Generation



11/25/20 Heiko Paulheim

Brute-force Approach

- Amazon sells 12M different products (as of 2020)
- 21200000=8.871×103612359
 - That's a 3.6M digit number
- Today's fastest computer
 - 200 Petaflops, i.e., 2x1017 floating point operations per second
- Even if an itemset could be checked with one single floating point operation
 - this would take 1.4×10³⁶¹²³³⁵ years
- Comparision: age of the universe is 1.4×10¹⁰ years!





Brute-force Approach

- Each itemset in the lattice is a candidate frequent itemset
- Count the support of each candidate by scanning the database
- Match each transaction against every candidate



- Complexity ~ $O(NMw) \rightarrow Expensive since M = 2^d$
- A smarter algorithm is required

Anti-Monotonicity of Support

- What happens when an itemset gets larger?
- s({Bread}) = 0.8
 - s({Bread,Milk}) = 0.6
 - s({Bread,Milk,Diaper}) = 0.4

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

- s({Milk}) = 0.8
 - s({Milk,Diaper}) = 0.6
 - s({Milk,Diaper,Beer}) = 0.4
- There is a pattern here!

Anti-Monotonicity of Support

- There is a pattern here!
 - It is called *anti-monitonicity* of support
- If X is a subset of Y
 s(Y) is at most as large as s(X)

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

$$\forall X, Y : (X \subseteq Y) \Longrightarrow s(X) \ge s(Y)$$

- Consequence for frequent itemset search (aka Apriori principle):
 - If Y is frequent, X also has to be frequent
 - i.e.: all subsets of frequent itemsets are frequent

Illustrating the Apriori Principle



The Apriori Algorithm

- 1. Start at k=1
- 2. Generate frequent itemsets of length k=1
- 3. Repeat until no new frequent itemsets are identified
 - 1. Generate length (k+1) candidate itemsets from length k frequent itemsets; increase k
 - 2. Prune candidate itemsets that cannot be frequent because they contain subsets of length k that are infrequent (Apriori Principle)
 - 3. Count the support of each remaining candidate by scanning the DB
 - 4. Eliminate candidates that are infrequent, leaving only those that are frequent

Illustrating the Apriori Principle



From Frequent Itemsets to Rules

- Given a frequent itemset F, find all non-empty subsets f ⊆ F such that f → F \ f satisfies the minimum confidence requirement
- Example Frequent Itemset:
 - F= {Milk,Diaper,Beer}
- Example Rule:
 - f = {Milk,Diaper}
 - {Milk,Diaper} \rightarrow {Beer}

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

Challenge: Combinatorial Explosion

- Given a 4-itemset {A,B,C,D}, we can generate ${A} \rightarrow {B,C,D}, {B} \rightarrow {A,C,D}, {C} \rightarrow {A,B,D}, {D} \rightarrow {A,B,C},$ ${A,B} \rightarrow {C,D}, {A,C} \rightarrow {B,D}, {A,D} \rightarrow {B,C},$ ${B,C} \rightarrow {A,D}, {B,D} \rightarrow {A,C}, {C,D} \rightarrow {A,B},$ ${A,B,C} \rightarrow {D}, {A,B,D} \rightarrow {C}, {A,C,D} \rightarrow {B}, {B,C,D} \rightarrow {A}$
- i.e., a total of 14 rules for just one itemset!
- General number for a k-itemset: 2^k-2
 - it's not 2^k since we ignore $\emptyset \to {\dots}$ and ${\dots} \to \emptyset$

Challenge: Combinatorial Explosion

- Wanted: another pruning trick like Apriori principle
- However

 $\{Milk, Diaper\} \rightarrow \{Beer\} c=0.67$ $\{Milk\} \rightarrow \{Beer\} c=0.5$ $\{Diaper\} \rightarrow \{Beer\} c=0.8$

• It's obviously not as straight forward

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

Challenge: Combinatorial Explosion

- Wanted: another pruning trick like Apriori principle
- Let's look at it another way
 - {Milk,Diaper,Beer} $\rightarrow Ø$ c=1.0
 - {Milk,Diaper} \rightarrow {Beer} c=0.67
 - {Milk} \rightarrow {Diaper,Beer} c=0.5
 - {Diaper} \rightarrow {Milk,Beer} c=0.5
 - {Milk,Beer} \rightarrow {Diaper} c=1.0
 - {Milk} \rightarrow {Diaper,Beer} c=0.5
 - {Beer} \rightarrow {Milk,Diaper} c=0.67

| TID | Items |
|-----|---------------------------|
| 1 | Bread, Milk |
| 2 | Bread, Diaper, Beer, Eggs |
| 3 | Milk, Diaper, Beer, Coke |
| 4 | Bread, Milk, Diaper, Beer |
| 5 | Bread, Milk, Diaper, Coke |

• **Observation:** moving elements in the rule from left to right never increases confidence!

Rule Generation

- Confidence is anti-monotone w.r.t. number of items on the RHS of the rule
 - i.e., "moving elements from left to right" cannot increase confidence
 - reason:

$$c(AB \rightarrow C) := \frac{s(ABC)}{s(AB)} \quad c(A \rightarrow BC) := \frac{s(ABC)}{s(A)}$$

- Due to anti-monotone property of support, we know
 - $S(AB) \leq S(A)$
- Hence
 - $c(AB \rightarrow C) \ge C(A \rightarrow BC)$

Rule Generation for Apriori Algorithm



Rule Generation for Apriori Algorithm

- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
- join(CD=>AB, BD=>AC)
 - would produce the candidate rule D => ABC
- Prune rule D=>ABC
 - if its subset AD=>BC does not have high confidence



• All the required information for confidence computation has already been recorded during itemset generation. \rightarrow No need to see the data anymore! $c(X \rightarrow Y) := \frac{s(X \cup Y)}{s(X)}$

Complexity of Apriori Algorithm

- Expensive part is scanning the database
 - i.e., when counting the support of frequent itemsets
- The database is scanned once per pass of frequent itemset generation
 - one pass to count frequencies of 1-itemsets
 - one pass to count frequencies of 2-itemsets
 - etc.
- i.e., for frequent itemsets of size $\leq k$,
 - k passes over the database are required

FP-growth Algorithm

- An alternative method for finding frequent itemsets
 - usually faster than Apriori
 - requires at most two passes over the database
- Use a compressed representation of the database using an FP-tree
- Once an FP-tree has been constructed, it uses a recursive divideand-conquer approach to mine the frequent itemsets



After reading TID=3:





11/25/20 Heiko Paulheim

- Properties of the FP-Tree
 - a very compact representation
 - fits in memory
 - even for larger transaction databases
 - more transactions of the same kind do not increase the tree size
 - can be optimized
 - sorting most frequent items first
 - good compression for many similar transactions
 - up-front pruning of infrequent itemsets

From the FP Tree to Patterns

• Naively:



From the FP Tree to Paths

- Enumeration:
 - A:7, AB:5, AC:3, AD:1, BC:3, CD:1, ABC:3, ABCD:1, ...
- However, we can do better
 - Single path tree: enumerate all subsets
 - Multi path tree: Build FP-Tree of subtrees recursively
 - For that recursion, we use the links
 - e.g., build FP-Tree for all itemsets ending in E
- Details
 - See literature

FP-Growth (Summary)

- Scans the database only twice:
 - first scan counts all 1-itemsets
 - for ordering by most frequent (more compact tree)
 - and for removing itemsets below minsup
 - second scan for constructing the FP-tree
 - recursive constructions only work on compact representation, not the actual database
- Finding patterns from the tree
 - algorithm recursively decomposes the tree into smaller subtrees
 - details: see books

Frequent Itemset Generation in Rapidminer

| 🌔 📴 Parameters 😹 🖨 🔟 | | | | | |
|------------------------|-------------------------------|--------|--|--|--|
| að 🗸 🛊? 🔟 🔯 🌛 🗸 | 🍒 😼 🖻 🦻 🕏 🗸 | | | | |
| | 🎘 FP- | Growth | | | |
| FP-Growth | 🔲 find min number of itemsets | | | | |
| exa exa res fre res | positive value | | | | |
| • | min support | 0.3 | | | |
| | max items | -1 | | | |
| | must contain | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Frequent Itemset Generation in Rapidminer

| No. of Sets: 22 | Size | Support | Item 1 | Item 2 | Item 3 | Item 4 |
|--------------------|------|---------|-------------|-------------|-------------|-------------|
| Total Max. Size: 4 | 1 | 0.600 | Asus EeePC | | | |
| | 1 | 0.500 | LT Minimaus | | | |
| | 1 | 0.500 | 2 GB DDR3 | | | |
| Min. Size: 1 | 1 | 0.400 | ThinkPad X2 | | | |
| Max. Size: 4 | 1 | 0.400 | Netbook-Sch | | | |
| Contains Item: | 1 | 0.400 | Lenovo Tabl | | | |
| | 1 | 0.400 | LT Laser Ma | | | |
| | 1 | 0.300 | HP Laserjet | | | |
| Undate View | 1 | 0.300 | HP CE50 To | | | |
| | 2 | 0.400 | Asus EeePC | LT Minimaus | | |
| | 2 | 0.500 | Asus EeePC | 2 GB DDR3 | | |
| | 2 | 0.400 | Asus EeePC | Netbook-Sch | | |
| | 2 | 0.300 | LT Minimaus | 2 GB DDR3 | | |
| | 2 | 0.300 | LT Minimaus | Netbook-Sch | | |
| | 2 | 0.400 | 2 GB DDR3 | Netbook-Sch | | |
| | 2 | 0.300 | ThinkPad X2 | Lenovo Tabl | | |
| | 2 | 0.300 | HP Laserjet | HP CE50 To | | |
| | 3 | 0.300 | Asus EeePC | LT Minimaus | 2 GB DDR3 | |
| | 3 | 0.300 | Asus EeePC | LT Minimaus | Netbook-Sch | |
| | 3 | 0.400 | Asus EeePC | 2 GB DDR3 | Netbook-Sch | |
| | 3 | 0.300 | LT Minimaus | 2 GB DDR3 | Netbook-Sch | |
| | 4 | 0.300 | Asus EeePC | LT Minimaus | 2 GB DDR3 | Netbook-Sch |

Creating Association Rules in Rapidminer

| | 🖉 Parameters 💥 🖨 🖻 | | |
|--|------------------------------------|--------------|--|
| að 🗕 🗊 🔟 🌛 🗕 | 🍒 😼 💀 🦻 🕵 🕶 | | |
| | Create Association Rules | | |
| | criterion | confidence 💌 | |
| | min confidence | 0.1 | |
| FP-Growth Create Association Rules exa exa fre ite rul ite | \Lambda 2 hidden expert parameters | | |

Exploring Association Rules in Rapidminer

🛒 Result Overview 🛛 🕅 🕅 AssociationRules (Create Association Rules) 🛛 🕅 💿 Table View 🔘 Graph View 🔘 Text View 🔘 Annotations Premises Conclusion Support Confidence Show rules matching No. marital-status = Never-married class = <=50K 0.310 0.957278 all of these conclusions: 266 age = range1 [-∞ - 31.500] class = <=50K 0.330 0.938 236 sex = Female class = <=50K 0.308 0.917 native-country = United-States workclass = Private race = White 157 class = <=50K 0.510 0.775 class = <=50K native-country = United-States, worl class = <=50K 0.751 154 0.440 sex = Male race = White, workclass = Private 0.749 153 class = <=50K 0.418 workclass = Private 0.736 native-country = United-States 150 class = <=50K 0.646 marital-status = Married-civ-spouse native-country = United-States, race class = <=50K 149 0.732 0.376 relationship = Husband 148 race = White class = <=50K 0.614 0.721 age = range1 [-∞ - 31.500] 0.715 147 native-country = United-States, race class = <=50K 0.556 sex = Female sex = Male, workclass = Private 0.699 146 class = <=50K 0.302 age = range3 [44.500 - ∞] marital-status = Never-married Min. Criterion: confidence Min. Criterion Value:

Frequent Itemset Mining in Python

- Various packages exist
 - In the exercise, we'll use the Orange3 package



itemsets = dict(fp_growth.frequent_itemsets(X, .2))
rules = association_rules(itemsets, .8)

11/25/20 Heiko Paulheim

Interestingness Measures

- Association rule algorithms tend to produce too many rules
 - many of them are uninteresting or redundant
 - Redundant if {A,B,C} → {D} and {A,B} → {D} have same support & confidence
- Interestingness measures can be used to prune or rank the derived rules
- In the original formulation of association rules, support & confidence are the only interest measures used
- Later, various other measures have been proposed
 - See Tan/Steinbach/Kumar, Chapter 6.7
 - We will have a look at two: Correlation & Lift

Drawback of Confidence

| | Coffee | Coffee | |
|-----|--------|--------|-----|
| Теа | 15 | 5 | 20 |
| Tea | 75 | 5 | 80 |
| | 90 | 10 | 100 |

Association Rule: Tea \rightarrow Coffee

• Confidence= s(Tea \cap Coffee)/s(Tea) = 15/20 = 0.75

Correlation

- We discover a high confidence rule for tea \rightarrow coffee
 - 75% of all people who drink tea also drink coffee
 - Hypothesis: people who drink tea are likely to drink coffee
 - Implicitly: more likely than people not drinking tea
- Cross check:
 - What is the confidence of not(tea) \rightarrow coffee?
 - Even higher: ~94% of people who **don't** drink tea do drink coffee
- We have two rules here
 - One is learned on all people who drink tea
 - The other is learned on all people who don't trink tea
 - Only together, they cover the whole dataset

Correlation

- Correlation takes into account all data at once
- In our scenario: corr(tea,coffee) = -0.25
 - i.e., the correlation is negative
 - Interpretation: people who drink tea are **less** likely to drink coffee

| | Coffee | Coffee | |
|-----|--------|--------|-----|
| Теа | 15 | 5 | 20 |
| Tea | 75 | 5 | 80 |
| | 90 | 10 | 100 |



- We discover a high confidence rule for tea \rightarrow coffee
 - 75% of all people who drink tea also drink coffee
 - Hypothesis: people who drink tea are likely to drink coffee
 - Implicitly: more likely than **all** people
- Test: Compare the confidence of the two rules
 - Rule: Tea \rightarrow coffee
 - Default rule: all \rightarrow coffee
- $c(tea \rightarrow coffee) = s(tea \cap coffee)/s(tea)$
- c(all → coffee) = s(all ∩ coffee)/s(all) = s(coffee) / 1



- Test: Compare the confidence of the two rules
 - Rule: tea \rightarrow coffee
 - Default rule: all \rightarrow coffee
- We accept a rule iff its confidence is higher than the default rule
 - c(tea \rightarrow coffee) = s(tea \cap coffee)/s(tea)
 - $c(all \rightarrow coffee) = s(all \cap coffee)/s(all) = s(coffee) / 1$

 $\begin{array}{l} \mathsf{c}(\mathsf{tea} \to \mathsf{coffee}) > \mathsf{c}(\mathsf{all} \to \mathsf{coffee}) \\ \leftrightarrow \mathsf{c}(\mathsf{tea} \to \mathsf{coffee}) / \mathsf{c}(\mathsf{all} \to \mathsf{coffee}) > 1 & Lift(X \Rightarrow Y) = \frac{s(X \cap Y)}{s(X) \times S(Y)} \\ \leftrightarrow \mathsf{s}(\mathsf{tea} \cap \mathsf{coffee}) / (\mathsf{s}(\mathsf{tea}) * \mathsf{s}(\mathsf{coffee})) > 1 & \end{array}$

Lift

• The *lift* of an association rule $X \rightarrow Y$ is defined as:

$$Lift(X \rightarrow Y) = \frac{s(X \cap Y)}{s(X) \times S(Y)}$$

- Interpretation:
 - if lift > 1, then X and Y are positively associated
 - if lift < 1, then X are Y are negatively associated
 - if lift = 1, then X and Y are independent.

Example: Lift

| | Coffee | Coffee | |
|-----|--------|--------|-----|
| Теа | 15 | 5 | 20 |
| Tea | 75 | 5 | 80 |
| | 90 | 10 | 100 |

Association Rule: Tea \rightarrow Coffee

s(Tea \cap Coffee) = 0.15

s(Tea) = 0.2, s(Coffee) = 0.9

 \Rightarrow Lift = 0.15/(0.2*0.9)= 0.8333 (< 1, therefore is negatively associated)

11/25/20 Heiko Paulheim

Combination of Confidence and Lift/Correlation

- So why not try to find rules with high lift/correlation directly?
- By design, lift and correlation are symmetric
 - i.e., lift(tea \rightarrow coffee) = lift(coffee \rightarrow tea)
- Confidence is *asymmetric*
 - c(coffee \rightarrow tea) is only 15/90 = 0.167

| | Coffee | Coffee | |
|-----|--------|--------|-----|
| Теа | 15 | 5 | 20 |
| Tea | 75 | 5 | 80 |
| | 90 | 10 | 100 |

Interestingness Measures

- There are lots of measures proposed in the literature
- Some measures are good for certain applications, but not for others
- Details: see literature (e.g., Tan et al.)

| # | Measure | Formula |
|----|---|---|
| 1 | ϕ -coefficient | $\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$ |
| 2 | Goodman-Kruskal's (λ) | $\frac{\sum_{j \max_{k} P(A_{j}, B_{k}) + \sum_{k \max_{j} P(A_{j}, B_{k}) - \max_{k} P(A_{j}, B_{k}) - \max_{k} P(A_{j}, B_{k}) - \max_{k} P(A_{j}) - \max_{k} P(B_{k})}{2 - \max_{j} P(A_{j}) - \max_{k} P(B_{k})}$ |
| 3 | ${\rm Odds\ ratio}\ (\alpha)$ | $\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$ |
| 4 | Yule's Q | $\frac{P(A,B)P(\overline{AB}) - P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B}) + P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha - 1}{\alpha + 1}$ |
| 5 | Yule's Y | $\frac{\sqrt{P(A,B)P(\overline{AB})} - \sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},B)} + \sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha} - 1}{\sqrt{\alpha} + 1}$ |
| 6 | Kappa (κ) | $\frac{\sqrt{P(A,B)P(AB)} + \sqrt{P(A,B)P(A,B)}}{\frac{P(A,B) + \sqrt{P(A,B)P(A)P(B)}}{1 - P(A)P(B) - P(\overline{A})P(\overline{B})}}$ |
| 7 | Mutual Information (M) | $\frac{\sum_{i}\sum_{j} P(A_{i},B_{j}) \log \frac{\nabla(A_{i},Y_{j})}{P(A_{i})P(B_{j})}}{\min(-\sum_{i} P(A_{i}) \log P(A_{i}), -\sum_{j} P(B_{j}) \log P(B_{j}))}$ |
| 8 | J-Measure (J) | $\max \Big(P(A,B) \log(\frac{P(B A)}{P(B)}) + P(A\overline{B}) \log(\frac{P(B A)}{P(\overline{B})}),$ |
| | | $P(A,B)\log(rac{P(A B)}{P(A)}) + P(\overline{A}B)\log(rac{P(\overline{A} B)}{P(\overline{A})}))$ |
| 9 | Gini index (G) | $\left \max \left(P(A) [P(B A)^2 + P(\overline{B} A)^2] + P(\overline{A}) [P(B \overline{A})^2 + P(\overline{B} \overline{A})^2] \right \right $ |
| | | $(-P(B)^2 - P(\overline{B})^2,$ |
| | | $P(B)[P(A B)^{2} + P(\overline{A} B)^{2}] + P(\overline{B})[P(A \overline{B})^{2} + P(\overline{A} \overline{B})^{2}]$ |
| | | $-P(A)^2 - P(\overline{A})^2$ |
| 10 | Support (s) | P(A,B) |
| 11 | Confidence (c) | $\max(P(B A), P(A B))$ |
| 12 | Laplace (L) | $\max\left(\frac{NP(A,B)+1}{NP(A)+2},\frac{NP(A,B)+1}{NP(B)+2}\right)$ |
| 13 | Conviction (V) | $\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})},\frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$ |
| 14 | Interest (I) | $\frac{P(A,B)}{P(A)P(B)}$ |
| 15 | $\cos (IS)$ | $\frac{P(A,B)}{\sqrt{P(A)P(B)}}$ |
| 16 | $\operatorname{Piatetsky-Shapiro's}\left(PS ight)$ | P(A,B) - P(A)P(B) |
| 17 | Certainty factor (F) | $\max\left(rac{P(B A)-P(B)}{1-P(B)},rac{P(A B)-P(A)}{1-P(A)} ight)$ |
| 18 | Added Value (AV) | $\max(P(B A)-P(B),P(A B)-P(A))$ |
| 19 | Collective strength (S) | $\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$ |
| 20 | Jaccard (ζ) | $\frac{P(A,B)}{P(A) \perp P(B) - P(A,B)}$ |
| 21 | Klosgen (K) | $\sqrt{P(A,B)} \max(P(B A) - P(B), P(A B) - P(A))$ |

Handling Continuous and Categorical Attributes

 How to apply association analysis formulation to other types of variables?

| Session Id | Country | Session Length (sec) | Number of Web Pages viewed | Gender | Browser Type | Buy |
|---------------|-----------|----------------------------|----------------------------------|--------|-----------------|-----|
| 1 | USA | 982 | 8 | Male | IE | No |
| 2 | China | 811 | 10 | Female | Netscape | No |
| 3 | USA | 2125 | 45 | Female | Mozilla | Yes |
| 4 | Germany | 596 | 4 | Male | IE | Yes |
| 5 | Australia | 123 | 9 | Male | Mozilla | No |
| | | | | | | |

• Example of Association Rule:

{Number of Pages \in [5,10) \land (Browser=Mozilla)} \rightarrow {Buy = No}

Handling Categorical Attributes

- Transform categorical attribute into asymmetric binary variables
- Introduce a new "item" for each distinct attribute-value pair
 - Example: replace Browser Type attribute with
 - Browser Type = Internet Explorer
 - Browser Type = Mozilla

| | Nominal to | Binominal | |
|----|------------|-----------|--|
| -(| exa | exa) | |
| | R | ori 🕽 | |
| | ~ | pre D | |
| | • | | |

Handling Categorical Attributes

- Introduce a new "item" for each distinct attribute-value pair
 - Example: replace Browser Type attribute with
 - Browser Type = Internet Explorer
 - Browser Type = Mozilla
- This method is also known as one-hot-encoding
 - We create n new variables, only one of which is 1 ("hot") at a time

```
from sklearn.preprocessing import OneHotEncoder
enc = OneHotEncoder()
enc.fit transform(data)
```

Handling Categorical Attributes

- Potential Issues
 - Many attribute values
 - Many of the attribute values may have very low support
 - Potential solution: Aggregate the low-support attribute values

 bin for "other"
 - Highly skewed attribute values
 - Example: 95% of the visitors have Buy = No
 - Most of the items will be associated with (Buy=No) item
 - Potential solution: drop the highly frequent items

Handling Continuous Attributes

- Transform continuous attribute into binary variables using discretization
 - Equal-width binning
 - Equal-frequency binning



- Issue: Size of the intervals affects support & confidence $\{\text{Refund} = \text{No}, (51,253 \leq \text{Income} \leq 51,254)\} \rightarrow \{\text{Cheat} = \text{No}\}$ $\{\text{Refund} = \text{No}, (60\text{K} \leq \text{Income} \leq 80\text{K})\} \rightarrow \{\text{Cheat} = \text{No}\}$ $\{\text{Refund} = \text{No}, (0\text{K} \leq \text{Income} \leq 1\text{B})\} \rightarrow \{\text{Cheat} = \text{No}\}$
 - Too small intervals: not enough support
 - Too large intervals: not enough confidence

Effect of Support Distribution

Many real data sets have a skewed support distribution



11/25/20 Heiko Paulheim

Effect of Support Distribution

- How to set the appropriate *minsup* threshold?
 - If *minsup* is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
 - If *minsup* is set too low, it is computationally expensive and the number of itemsets is very large
- Using a single minimum support threshold may not be effective

Multiple Minimum Support

- How to apply multiple minimum supports?
 - MS(i): minimum support for item i
 - e.g.: MS(Milk)=5%, MS(Coke) = 3%, MS(Broccoli)=0.1%, MS(Salmon)=0.5%
 - MS({Milk, Broccoli}) = min (MS(Milk), MS(Broccoli))= 0.1%
- Challenge: Support is no longer anti-monotone
 - Suppose: Support(Milk, Coke) = 1.5% and Support(Milk, Coke, Broccoli) = 0.5
 - \rightarrow {Milk,Coke} is infrequent but {Milk,Coke,Broccoli} is frequent
 - Requires variations of Apriori algorithm
 - Details: see literature

Association Rules with Temporal Components

- Good example:
 - Star Wars ep. 7, Star Wars ep. 8 → Star wars ep. 9



- Bad example:
 - mobile phone \rightarrow charger vs. charger \rightarrow mobile phone
 - are indistinguishable by frequent pattern mining
 - both will be used for recommendation
 - customers will select a charger after a mobile phone
 - but not the other way around!
 - however, Amazon does not respect sequences...
- See: Data Mining 2 for sequential pattern mining

Wrap-up

- Association Analysis:
 - discovering patterns in data
 - patterns are described by rules
- Apriori & FP-Growth algorithm:
 - Finds rules with minimum support (i.e., number of transactions)
 - and minimum confidence (i.e., strength of the implication)
- You'll play around with it in the upcoming exercise...

What's Next?

- Data Mining 2 (next FSS)
- Machine Learning / Hot Topics in Machine Learning (HWS / FSS), Prof. Gemulla
- Relational Learning (HWS), Dr. Meilicke
- Information Retrieval and Web Search (FSS), Prof. Glavaš
- Text Analytics (HWS), Prof. Ponzetto & Prof. Glavaš
- Web Mining (FSS), Prof. Ponzetto
- Image Processing (HWS) and Higher-Level Computer Vision (FSS), Prof. Keuper
- Network Analysis (HWS), Prof. Stuckenschmidt
- Process Mining & Analytics (HWS), Prof. van der Aa & Prof. Rehse

Questions?

