Association Rule Mining

Yanchang Zhao

http://www.RDataMining.com

Machine Learning 102 Workshop
S P Jain School of Global Management, Mumbai, India

30 May - 5 June 2016

Outline

Association Rules

- To discover association rules showing itemsets that occur together frequently [Agrawal et al., 1993].
- Widely used to analyze retail basket or transaction data.
- An association rule is of the form $A \Rightarrow B$, where A and B are items or attribute-value pairs.
- The rule means that those database tuples having the items in the left hand of the rule are also likely to having those items in the right hand.
- Examples of association rules:
  - *bread* $\Rightarrow$ *butter*
  - *computer* $\Rightarrow$ *software*
  - *age in [20,29] & income in [60K,100K]* $\Rightarrow$ *buying up-to-date mobile handsets*

Association Rules

Association rules are rules presenting association or correlation between itemsets.

$$
\begin{aligned}
\text{support}(A \Rightarrow B) &= P(A \cup B) \\
\text{confidence}(A \Rightarrow B) &= P(B|A) \\
&= \frac{P(A \cup B)}{P(A)} \\
\text{lift}(A \Rightarrow B) &= \frac{\text{confidence}(A \Rightarrow B)}{P(B)} \\
&= \frac{P(A \cup B)}{P(A)P(B)}
\end{aligned}
$$

where $P(A)$ is the percentage (or probability) of cases containing A.

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support $=$

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06
- confidence =

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06
- confidence = support / P(R) = 0.06/0.08 = 0.75

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06
- confidence = support / P(R) = 0.06/0.08 = 0.75
- lift =

An Example

- Assume there are 100 students.
- 10 out of them know data mining techniques, 8 know R language and 6 know both of them.
- *knows R* $\Rightarrow$ *knows data mining*
- support = P(R & data mining) = 6/100 = 0.06
- confidence = support / P(R) = 0.06/0.08 = 0.75
- lift = confidence / P(data mining) = 0.75/0.10 = 7.5

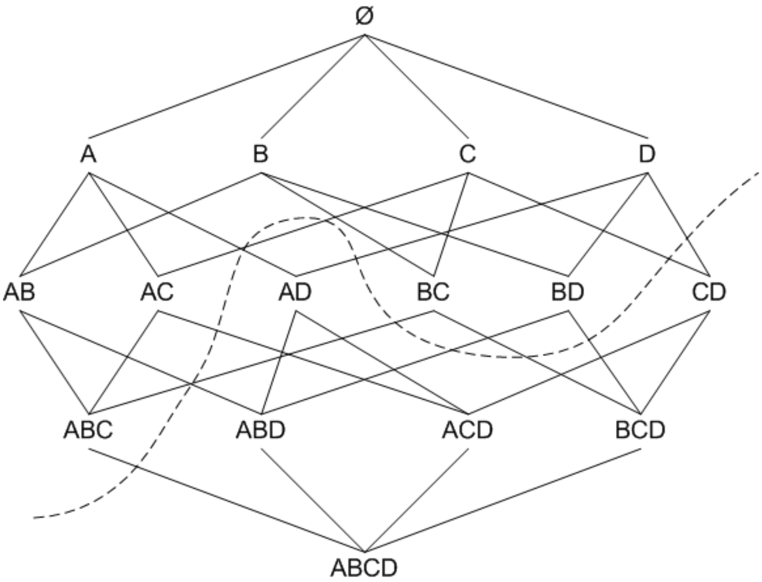
Association Rule Mining

- Association Rule Mining is normally composed of two steps:
  - Finding all frequent itemsets whose supports are no less than a minimum support threshold;
  - From above frequent itemsets, generating association rules with confidence above a minimum confidence threshold.
- The second step is straightforward, but the first one, frequent itemset generateion, is computing intensive.
- The number of possible itemsets is $2^n - 1$, where n is the number of unique items.
- Well-known algorithms: Apriori, ECLAT, FP-Growth

Downward-Closure Property

- Downward-closure property of support, a.k.a. anti-monotonicity
- For a frequent itemset, all its subsets are also frequent. if {A,B} is frequent, then both {A} and {B} are frequent.
- For an infrequent itemset, all its super-sets are infrequent. if {A} is infrequent, then {A,B}, {A,C} and {A,B,C} are infrequent.
- useful to prune candidate itemsets

Itemset Lattice



Outline

Apriori

- Apriori [Agrawal and Srikant, 1994]: a well-known algorithm for association rule mining
- A level-wise, breadth-first algorithm
- Counts transactions to find frequent itemsets
- Generates candidate itemsets by exploiting downward closure property of support

Apriori Process

1. Find all frequent 1-itemsets L_1
2. Join step: generate candidate k-itemsets by joining L_{k-1} with itself
3. Prune step: prune candidate k-itemsets using downward-closure property
4. Scan the dataset to count frequency of candidate k-itemsets and select frequent k-itemsets L_k
5. Repeat above process, until no more frequent itemsets can be found.

Dataset

$\mathbf{D}$	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

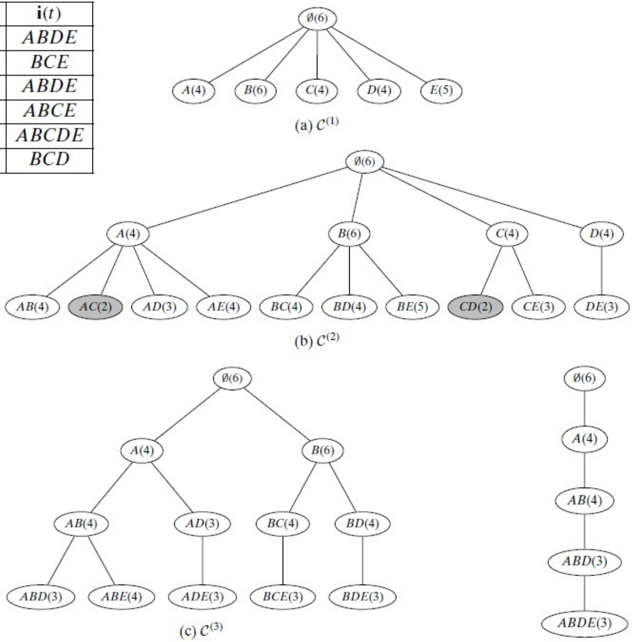
(b) Transaction database

x	A	B	C	D	E
$\mathbf{t}(x)$	1	1	2	1	1
	3	2	4	3	2
	4	3	5	5	3
	5	4	6	6	4
		5			5
		6			

(c) Vertical database

From [Zaki and Meira, 2014]

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD



(a) $\mathcal{C}^{(1)}$

(b) $\mathcal{C}^{(2)}$

(c) $\mathcal{C}^{(3)}$

(d) $\mathcal{C}^{(4)}$

From [Zaki and Meira, 2014]

Inefficiency of Apriori

- Apriori may generate a huge number of candidate itemsets.
- It scans data repeatedly, as it needs one full scan of data to find each L_k.

Outline

Improving Efficiency of Apriori

- Hashing itemset counts
- Transaction reduction
- Partitioning
- Sampling
- Dynamic itemset counting

Hashing Itemset Counts

- When scanning each transaction to generate frequent 1-itemset, generate all 2-itemsets for each transaction, hash them into different buckets of a hash table and increase the corresponding bucket counts.
- A 2-itemsets whose corresponding bucket count is below minimum support cannot be frequent and can be removed from the candidate set.
- It may significantly reduce the number of candidate itemsets to examine, especially for 2-itemsets.

Transaction Reduction

- A transaction not containing any frequent k-itemsets can not conatin any frequent (k+1)-itemsets.
- Such a transaction can be removed when generating j-itemsets, where $j > k$.
- The data to scan becomes smaller and smaller as k increases.

Partitioning

- Idea: If dividing a dataset into multiple partitions, any itemset that is potentially frequent in the dataset must be a frequent itemset in at least one partition.
- Split data into multiple partitions, so that each partition can fit into main memory.
- Find local frequent itemsets from each partition
- The collection of all local frequent itemsets forms the global candidate itemsets.
- Scan the whole data, count frequency of candidates and find global frequent itemsets.
- Needs only two scans of data

Sampling

- Idea: Trade off accuracy against efficiency by sampling
- Draw a random sample of data, such that the sample can fit in main memory.
- Find frequent itemsets in the sample data, using a lower support threshold than minimum support.
- Use the rest of data to compute the actual frequencies of each itemsets.

Dynamic itemset counting

- Idea: add candidate itemsets at different points during a scan
- Partition data into blocks marked by start points and new candidate itemsets can be added at any start point.
- Different from Apriori, which adds new candidate itemsets only prior to each complete data scan.
- It estimates the support of all itemsets that have been counted so far, adding new candidate itemsets if all of their subsets are estimated to be frequent.
- Needs fewer data scans than Apriori.

Outline

FP-growth

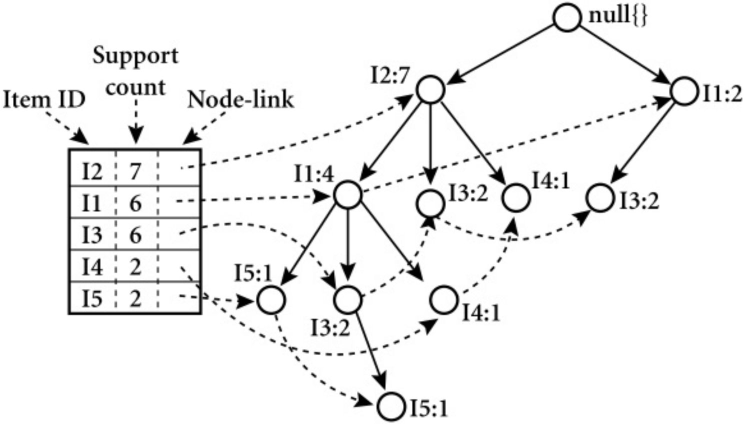
- FP-growth: frequent-pattern growth, which mines frequent itemsets without candidate generation [Han et al., 2004]
- Compresses the input database creating an FP-tree instance to represent frequent items.
- Divides the compressed database into a set of conditional databases, each one associated with one frequent pattern.
- Each such database is mined separately.
- It reduces search costs by looking for short patterns recursively and then concatenating them in long frequent patterns.*

* https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm

FP-tree

- The frequent-pattern tree (FP-tree) is a compact structure that stores quantitative information about frequent patterns in a dataset. It has two components:
  - A root labeled as "null" with a set of item-prefix subtrees as children
  - A frequent-item header table
- Each node has three attributes:
  - Item name
  - Count: number of transactions represented by the path from root to the node
  - Node link: links to the next node having the same item name
- Each entry in the frequent-item header table also has three attributes:
  - Item name
  - Head of node link: point to the first node in the FP-tree having the same item name
  - Count: frequency of the item

FP-tree



From [Han, 2005]

The FP-growth Algorithm

- In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to a header table.
- In the second pass, it builds the FP-tree structure by inserting instances.
- Items in each instance have to be sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly.
- Items in each instance that do not meet minimum coverage threshold are discarded.
- If many instances share most frequent items, FP-tree provides high compression close to tree root.

The FP-growth Algorithm

- Recursive processing of this compressed version of main dataset grows large item sets directly, instead of generating candidate items and testing them against the entire database.
- Growth starts from the bottom of the header table (having longest branches), by finding all instances matching given condition.
- New tree is created, with counts projected from the original tree corresponding to the set of instances that are conditional on the attribute, with each node getting sum of its children counts.
- Recursive growth ends when no individual items conditional on the attribute meet minimum support threshold, and processing continues on the remaining header items of the original FP-tree.
- Once the recursive process has completed, all large item sets with minimum coverage have been found, and association rule creation begins.

Dataset

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

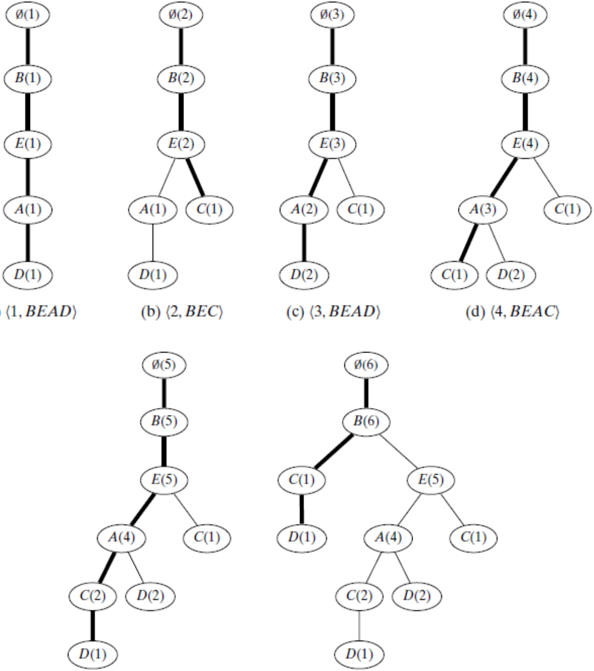
(b) Transaction database

x	A	B	C	D	E
$\mathbf{t}(x)$	1 3 4 5	1 2 3 4 5 6	2 4 5 6	1 3 5 6	1 2 3 4 5

(c) Vertical database

From [Zaki and Meira, 2014]

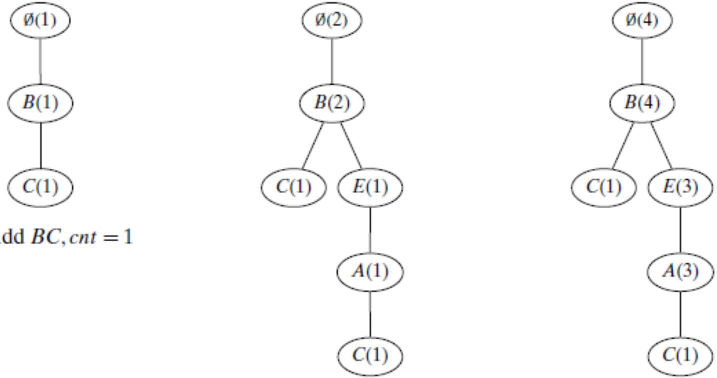
t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD



(a) $\langle 1, BEAD \rangle$ (b) $\langle 2, BEC \rangle$ (c) $\langle 3, BEAD \rangle$ (d) $\langle 4, BEAC \rangle$

(e) $\langle 5, BEACD \rangle$ (f) $\langle 6, BCD \rangle$

Projected FP-Tree for D



(a) Add $BC, cnt = 1$

(b) Add $BEAC, cnt = 1$

(c) Add $BEA, cnt = 2$

From [Zaki and Meira, 2014]

FP-Growth for Large Datasets

- In large datasets, its not possible to hold the FP-tree in the main memory.
- Partition the dataset into a set of smaller datasets (called projected datasets).
- Construct an FP-tree from each of these smaller datasets.

Outline

ECLAT

- ECLAT: equivalence class transformation [Zaki et al., 1997]
- A depth-first search algorithm using set intersection
- Idea: use tid set intersecion to compute the support of a candidate itemset, avoiding the generation of subsets that does not exist in the prefix tree.
- $t(AB) = t(A) \cap t(B)$
- $support(AB) = |t(AB)|$
- Eclat intersects the tidsets only if the frequent itemsets share a common prefix.
- It traverses the prefix search tree in a DFS-like manner, processing a group of itemsets that have the same prefix, also called a prefix equivalence class.

ECLAT

- It works recursively.
- The initial call uses all single items with their tid-sets.
- In each recursive call, it verifies each itemset tid-set pair $(X, t(X))$with all the other pairs to generate new candidates. If the new candidate is frequent, it is added to the set P_x.
- Recursively, it finds all frequent itemsets in the X branch.

The ECLAT Algorithm

// Initial Call: $\mathcal{F} \leftarrow \emptyset, P \leftarrow \left\{\langle i, \mathbf{t}(i)\rangle \mid i \in \mathcal{I}, |\mathbf{t}(i)| \geq minsup\right\}$

Eclat ($P, minsup, \mathcal{F}$):

foreach $\langle X_a, \mathbf{t}(X_a)\rangle \in P$ **do**

  $\mathcal{F} \leftarrow \mathcal{F} \cup \left\{(X_a, sup(X_a))\right\}$

  $P_a \leftarrow \emptyset$

  foreach $\langle X_b, \mathbf{t}(X_b)\rangle \in P$, *with* $X_b > X_a$ **do**

    $X_{ab} = X_a \cup X_b$

    $\mathbf{t}(X_{ab}) = \mathbf{t}(X_a) \cap \mathbf{t}(X_b)$

    if $sup(X_{ab}) \geq minsup$ **then**

      $P_a \leftarrow P_a \cup \left\{\langle X_{ab}, \mathbf{t}(X_{ab})\rangle\right\}$

  if $P_a \neq \emptyset$ **then** Eclat ($P_a, minsup, \mathcal{F}$)

From [Zaki and Meira, 2014]

Dataset

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

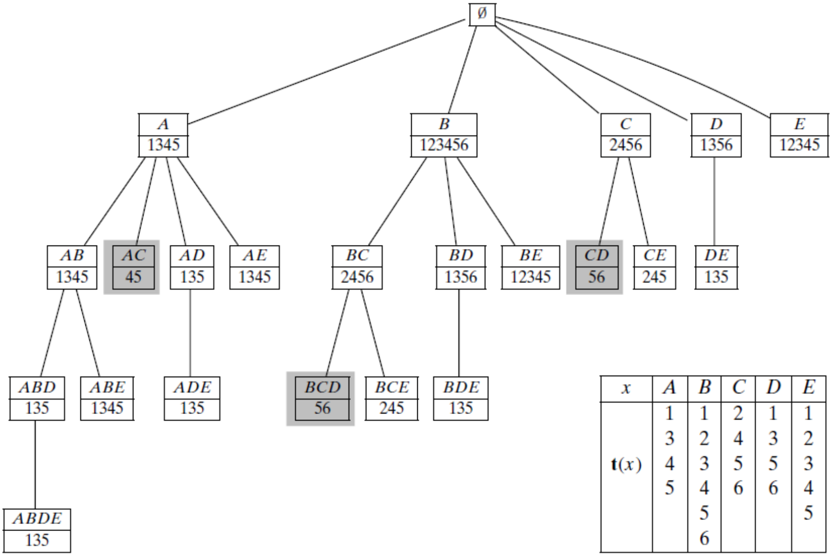
(b) Transaction database

x	A	B	C	D	E
$\mathbf{t}(x)$	1 3 4 5	1 2 3 4 5 6	2 4 5 6	1 3 5 6	1 2 3 4 5

(c) Vertical database

From [Zaki and Meira, 2014]

ECLAT



x	A	B	C	D	E
$\mathbf{t}(x)$	1	1	2	1	1
	3	2	4	3	2
	4	3	5	5	3
	5	4	6	6	4
		5			5
		6			

From [Zaki and Meira, 2014]

Outline

Interestingness Measures

- Which rules or patterns are the most interesting ones? One way is to rank the discovered rules or patterns with interestingness measures.
- The measures of rule interestingness fall into two categories, subjective and objective [Freitas, 1998, Silberschatz and Tuzhilin, 1996].
- Objective measures, such as *lift*, *odds ratio* and *conviction*, are often data-driven and give the interestingness in terms of statistics or information theory.
- Subjective (user-driven) measures, e.g., *unexpectedness* and *actionability*, focus on finding interesting patterns by matching against a given set of user beliefs.

Objective Interestingness Measures

- Support and confidence are the most widely used objective measures to select interesting rules.
- Many other objective measures introduced by Tan et al. [Tan et al., 2002], such as ϕ*-coefficient*, *odds ratio*, *kappa*, *mutual information*, *J-measure*, *Gini index*, *laplace*, *conviction*, *interest* and *cosine*.
- Their study shows that different measures have different intrinsic properties and there is no measure that is better than others in all application domains.
- In addition, any-confidence, all-confidence and bond, are designed by Omiecinski [Omiecinski, 2003].
- *Utility* is used by Chan et al. [Chan et al., 2003] to find top-k objective-directed rules.
- *Unexpected Confidence Interestingness* and *Isolated Interestingness* are designed by Dong and Li [Dong and Li, 1998] by considering its unexpectedness in terms of other association rules in its neighbourhood.

Subjective Interestingness Measures

- *Unexpectedness* and *actionability* are two main categories of subjective measures [Silberschatz and Tuzhilin, 1995].
- A pattern is unexpected if it is new to a user or contradicts the user's experience or domain knowledge.
- A pattern is actionable if the user can do something with it to his/her advantage [Silberschatz and Tuzhilin, 1995, Liu et al., 2003].
- Liu and Hsu [Liu and Hsu, 1996] proposed to rank learned rules by matching against expected patterns provided by the user.
- Ras and Wieczorkowska [Ras and Wieczorkowska, 2000] designed action-rules which show "what actions should be taken to improve the profitability of customers". The attributes are grouped into "hard attributes" which cannot be changed and "soft attributes" which are possible to change with reasonable costs. The status of customers can be moved from one to another by changing the values of soft ones.

Interestingness Measures

#	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)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j) \log \frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i) \log P(A_i), -\sum_j P(B_j) \log P(B_j))}$
8	J-Measure (J)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)}) + P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)}) + P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2 + P(\overline{B}\|A)^2] + P(\overline{A})[P(B\|\overline{A})^2 + P(\overline{B}\|\overline{A})^2]$ $-P(B)^2 - P(\overline{B})^2,$ $P(B)[P(A\|B)^2 + P(\overline{A}\|B)^2] + P(\overline{B})[P(A\|\overline{B})^2 + P(\overline{A}\|\overline{B})^2]$ $-P(A)^2 - P(\overline{A})^2\Big)$

From [Tan et al., 2002]

Interestingness Measures

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	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B) - P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)}\right)$
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)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A) - P(B), P(A\|B) - P(A))$

From [Tan et al., 2002]

Outline

Associative Classification

- Associative classification: build a classifier with association rules
- Class association rules
- Three phases for building associative classifiers [Antonie et al., 2009]:
  1. mining the training data for association rules and keeping only those that can classify instances,
  2. pruning the mined rules to weed out irrelevant or noisy rules,
  3. selecting and combining the rules to classify unknown items.
- Other algorithms for associative classification: [Baralis and Chiusano, 2004], [Liu et al., 1998], [Zaiane and Antonie, 2005], [Chiusano and Garza, 2009]

Outline

Applications - I

- Market basket analysis
  - Identifying associations between items in shopping baskets, i.e., which items are frequently purchsed together
  - Can be used by retailers to understand customer shopping habits, do selective marketing and plan shelf space
- Churn analysis and selective marketing
  - Discovering demographic characteristics and behaviours of customers who are likely/unlikely to switch to other telcos
  - Identifying customer groups who are likely to purchase a new service or product
- Credit card risk analysis
  - Finding characteristics of customers who are likely to default on credit card or mortgage
  - Can be used by banks to reduce risks when assessing new credit card or mortgage applications

Applications - II

- Stock market analysis
  - Finding relationships between individual stocks, or between stocks and economic factors
  - Can help stock traders select interesting stocks and improve trading strategies
- Medical diagnosis
  - Identifying relationships between symptoms, test results and illness
  - Can be used for assisting doctors on illness diagnosis or even on treatment

Outline

Further Readings

- Association Rule Learning
  https://en.wikipedia.org/wiki/Association_rule_learning
- Data Mining Algorithms In R: Apriori
  https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_Apriori_Algorithm
- Data Mining Algorithms In R: ECLAT
  https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_Eclat_Algorithm
- Data Mining Algorithms In R: FP-Growth
  https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm
- FP-Growth Implementation by Christian Borgelt
  http://www.borgelt.net/fpgrowth.html
- Frequent Itemset Mining Implementations Repository
  http://fimi.ua.ac.be/data/

Further Readings

- More than 20 interestingness measures, such as chi-square, conviction, gini and leverage

  Tan, P.-N., Kumar, V., and Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In Proc. of KDD '02, pages 32-41, New York, NY, USA. ACM Press.
- More reviews on interestingness measures: [Silberschatz and Tuzhilin, 1996], [Tan et al., 2002], [Omiecinski, 2003] and [Wu et al., 2007]
- Post mining of association rules, such as selecting interesting association rules, visualization of association rules and using association rules for classification [Zhao et al., 2009]

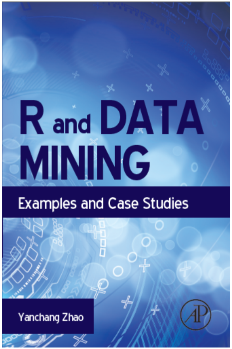
  Yanchang Zhao, et al. (Eds.). "Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction", ISBN 978-1-60566-404-0, May 2009. Information Science Reference.
- Package *arulesSequences*: mining sequential patterns

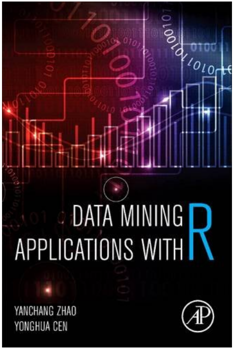
  `http://cran.r-project.org/web/packages/arulesSequences/`

Online Resources

- Chapter 9 - Association Rules, in book
  R and Data Mining: Examples and Case Studies
  `http://www.rdatamining.com/docs/RDataMining-book.pdf`
- RDataMining Reference Card
  `http://www.rdatamining.com/docs/RDataMining-reference-card.pdf`
- Free online courses and documents
  `http://www.rdatamining.com/resources/`
- RDataMining Group on LinkedIn (20,000+ members)
  `http://group.rdatamining.com`
- Twitter (2,500+ followers)
  @RDataMining

The End





Thanks!

Email: yanchang(at)rdatamining.com

References I

Agrawal, R., Imielinski, T., and Swami, A. (1993).
Mining association rules between sets of items in large databases.
In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington D.C. USA.

Agrawal, R. and Srikant, R. (1994).
Fast algorithms for mining association rules in large databases.
In *Proc. of the 20th International Conference on Very Large Data Bases*, pages 487–499, Santiago, Chile.

Antonie, M.-L., Chodos, D., and Zaiane, O. (2009).
Variations on associative classifiers and classification results analyses.
In Zhao, Y., Zhang, C., and Cao, L., editors, *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction*, pages 150–172. Information Science Reference.

Baralis, E. and Chiusano, S. (2004).
Essential classification rule sets.
ACM Trans. Database Syst., 29(4):635–674.

Chan, R., Yang, Q., and Shen, Y.-D. (2003).
Mining high utility itemsets.
In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 19–26.

Chiusano, S. and Garza, P. (2009).
Selection of high quality rules in associative classification.
In Zhao, Y., Zhang, C., and Cao, L., editors, *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction*, pages 173–198. Information Science Reference.

References II

Dong, G. and Li, J. (1998).
Interestingness of discovered association rules in terms of neighborhood-based unexpectedness.
In *PAKDD '98: Proceedings of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining*, pages 72–86, London, UK. Springer-Verlag.

Freitas, A. A. (1998).
On objective measures of rule surprisingness.
In *PKDD '98: Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 1–9, London, UK. Springer-Verlag.

Han, J. (2005).
Data Mining: Concepts and Techniques.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Han, J., Pei, J., Yin, Y., and Mao, R. (2004).
Mining frequent patterns without candidate generation.
Data Mining and Knowledge Discovery, 8:53–87.

Liu, B. and Hsu, W. (1996).
Post-analysis of learned rules.
In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 828–834, Portland, Oregon, USA.

Liu, B., Hsu, W., and Ma, Y. (1998).
Integrating classification and association rule mining.
In *KDD'98: Proc. of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 80–86. AAAI Press.

Liu, B., Ma, Y., and Yu, P. S. (2003).
Discovering business intelligence information by comparing company web sites.
In Zhong, N., Liu, J., and Yao, Y. Y., editors, *Web Intelligence*, pages 105–127. Springer-Verlag.

References III

Omiecinski, E. R. (2003).
Alternative interest measures for mining associations in databases.
IEEE Transactions on Knowledge and Data Engineering, 15(1):57–69.

Ras, Z. W. and Wieczorkowska, A. (2000).
Action-rules: How to increase profit of a company.
In *PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 587–592, London, UK. Springer-Verlag.

Silberschatz, A. and Tuzhilin, A. (1995).
On subjective measures of interestingness in knowledge discovery.
In *Knowledge Discovery and Data Mining*, pages 275–281.

Silberschatz, A. and Tuzhilin, A. (1996).
What makes patterns interesting in knowledge discovery systems.
IEEE Transactions on Knowledge and Data Engineering, 8(6):970–974.

Tan, P.-N., Kumar, V., and Srivastava, J. (2002).
Selecting the right interestingness measure for association patterns.
In *KDD '02: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–41, New York, NY, USA. ACM Press.

Wu, T., Chen, Y., and Han, J. (2007).
Association mining in large databases: A re-examination of its measures.
In *PKDD'07: Proc. of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007*, pages 621–628.

Zaiane, O. R. and Antonie, M.-L. (2005).
On pruning and tuning rules for associative classifiers.
In *Proceedings of the 9th International Conference on Knowledge-Based Intelligence Information & Engineering Systems (KES'05)*, Melbourne, Australia.

References IV

Zaki, M. and Meira, W. (2014).
Data Mining and Analysis: Fundamental Concepts and Algorithms.
Cambridge University Press.

Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997).
New algorithms for fast discovery of association rules.
Technical Report 651, Computer Science Department, University of Rochester, Rochester, NY 14627.

Zhao, Y., Zhang, C., and Cao, L., editors (2009).
Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction, ISBN 978-1-60566-404-0.

Information Science Reference, Hershey, PA.