CEFET/RJ

Pattern Mining

Eduardo Ogasawara
eduardo.ogasawara@cefet-rj.br
https://eic.cefet-rj.br/~eogasawara

What Is Frequent Pattern Analysis?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc) that occurs frequently in a data set
- First proposed by Agrawal (1993) in the context of frequent itemsets and association rule mining
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?
    - Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis

[1] R. Agrawal, T. Imieliński, and A. Swami, 1993, Mining Association Rules Between Sets of Items in Large Databases, *ACM SIGMOD Record*, v. 22, n. 2, p. 207–216.
[2] J. Han, H. Cheng, D. Xin, and X. Yan, 2007, Frequent pattern mining: Current status and future directions, *Data Mining and Knowledge Discovery*, v. 15, n. 1, p. 55–86.

Basic Concepts: Frequent Patterns

- Itemset
  - A set of one or more items
- k-itemset
  - $\mathrm{X} = \{x_1, \ldots, x_k\}$
- (absolute) support count of X:
  - occurrences of an itemset X
- (relative) support of X:
  - the fraction of transactions that contains X
  - the probability that a transaction contains X
  - $\mathrm{Sup(X)} = \mathrm{P(X)}$
- An itemset X is frequent if $\mathrm{Sup(X)} \geq minsup$

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Let $minsup$ = 50%

Freq. Pat.:

$Beer$:3, $Nuts$:3, $Diaper$:4, $Eggs$:3, $\{Beer, Diaper\}$:3

$$\mathrm{Sup}(\mathrm{X} \cup Y) = \mathrm{P}(\mathrm{X} \cap Y)^*$$

$$Sup(Diaper) = P(diaper) = \frac{4}{5}$$

$$Sup(Beer) = P(beer) = \frac{3}{5}$$

$$Sup(Beer \cup Diaper) = P(diaper \cap beer) = \frac{3}{5}$$

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

[2] L. Baroni, R. Salles, S. Salles, G. Guedes, F. Porto, E. Bezerra, C. Barcellos, M. Pedroso, and E. Ogasawara, 2020, An analysis of malaria in the Brazilian Legal Amazon using divergent association rules, Journal of Biomedical Informatics, v. 108

(*) Using Statistics notation

Basic Concepts: Association Rules

- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, s, probability that a transaction contains both X and Y
  - $Sup(X \rightarrow Y) = Sup(X \cup Y) = P(X \cap Y)$*
  - confidence, c, conditional probability that a transaction having X also contains Y
  - $Conf(X \rightarrow Y) = P(Y|X) = \frac{P(X \cap Y)}{P(X)} = \frac{\mathrm{Sup}(X \cup Y)}{\mathrm{Sup}(X)}$

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Let $minsup$ = 50%, $minconf$ = 50%

Freq. Pat.:

$Beer$:3, $Nuts$:3, $Diaper$:4, $Eggs$:3, $\{Beer, Diaper\}$:3

Association rules:

$Beer \rightarrow Diaper$ (60%, 100%)

$Diaper \rightarrow Beer$ (60%, 75%)

[1] J. Han, H. Cheng, D. Xin, and X. Yan, 2007, Frequent pattern mining: Current status and future directions, Data Mining and Knowledge Discovery, v. 15, n. 1, p. 55–86.
[2] J.M. Luna, P. Fournier-Viger, and S. Ventura, 2019, Frequent itemset mining: A 25 years review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 9, n. 6
(*) Using statistics notation

Closed Patterns and Max-Patterns

- A long pattern contains a combinatorial number of sub-patterns
  - E.g., $\{a_1, \ldots, a_{100}\}$ contains $\binom{100}{1} + \binom{100}{2} + \ldots + \binom{100}{100} =$
    $2^{100} - 1 = 1.27 \times 10^{30}$ sub-patterns
- Ways to reduce search space
  - Mine closed patterns and max-patterns instead
  - An itemset X is closed if X is frequent and there exists no super-pattern $Y \sqsupset X$, with the same support as X
  - An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \sqsupset X$
  - Closed pattern is a lossless compression of freq. patterns
    - Reducing the number of patterns and rules

[1] J. Wang, J. Han, and J. Pei, 2003, CLOSET+: Searching for the best strategies for mining frequent closed itemsets, In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 236–245
[2] D. Xin, J. Han, X. Yan, and H. Cheng, 2005, Mining compressed frequent-pattern sets, In: *VLDB 2005 - Proceedings of 31st International Conference on Very Large Data Bases*, p. 709–720

Scalable Frequent Itemset Mining Methods

- Apriori: A Candidate Generation-and-Test Approach
- FPGrowth: A Frequent Pattern-Growth Approach
- ECLAT: Frequent Pattern Mining with Vertical Data Format
- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent

If {beer, diaper, nuts} is frequent, so is {beer, diaper}

i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}

[1] J. Han, H. Cheng, D. Xin, and X. Yan, 2007, Frequent pattern mining: Current status and future directions, Data Mining and Knowledge Discovery, v. 15, n. 1, p. 55–86.
[2] J.M. Luna, P. Fournier-Viger, and S. Ventura, 2019, Frequent itemset mining: A 25 years review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 9, n. 6

Apriori: A Candidate Generation & Test Approach

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested

1. Initially, scan DB once to get frequent 1-itemset
2. Generate length (k+1) candidate itemsets from length k frequent itemsets
3. Test the candidates against DB
4. Terminate when no frequent or candidate set can be generated

[1] R. Agrawal, T. Imieliński, and A. Swami, 1993, Mining Association Rules Between Sets of Items in Large Databases, *ACM SIGMOD Record*, v. 22, n. 2, p. 207–216.
[2] R. Agrawal and R. Srikant, 1994, Fast Algorithms for Mining Association Rules in Large Databases, In: *Proceedings of the 20th International Conference on Very Large Data Bases*, p. 487–499

The Apriori Algorithm - An Example

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

$minsup = 2$

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
~~{D}~~	~~1~~
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
~~{A, B}~~	~~1~~
{A, C}	2
~~{A, E}~~	~~1~~
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

[1] R. Agrawal, T. Imieliński, and A. Swami, 1993, Mining Association Rules Between Sets of Items in Large Databases, *ACM SIGMOD Record*, v. 22, n. 2, p. 207–216.

The Apriori Algorithm (Pseudo-Code)

$\mathrm{C_k}$: Candidate itemset of size k
$\mathrm{L_k}$: frequent itemset of size k

$\mathrm{L_1}$ ← {frequent items}
k ← 1
while ($\mathrm{L_k} \neq \varnothing$) do begin
    $\mathrm{C_{k+1}}$ ← candidates generated from $\mathrm{L_k}$
    for each transaction t in database do
        increment the count of candidates in $\mathrm{C_{k+1}}$ that are in t
    $\mathrm{L_{k+1}}$ ← candidates in $\mathrm{C_{k+1}}$with $\mathrm{minsupport}$
    $\mathrm{k \leftarrow k + 1}$
end
return $\cup_k \mathrm{L_k}$

[1] R. Agrawal, T. Imieliński, and A. Swami, 1993, Mining Association Rules Between Sets of Items in Large Databases, *ACM SIGMOD Record*, v. 22, n. 2, p. 207–216.

Implementation of Apriori

- How to generate candidates?
  - Step 1: self-joining L_k
  - Step 2: pruning

Example of candidate-generation

L_3={abc, abd, acd, ace, bcd}

Self-joining: $L_3 \bowtie L_3$

$abcd$ from abc and abd

$acde$ from acd and ace

Pruning:

$acde$ is removed because ade is not in L_3

$C_4 = \{abcd\}$

[1] J. Han, H. Cheng, D. Xin, and X. Yan, 2007, Frequent pattern mining: Current status and future directions, Data Mining and Knowledge Discovery, v. 15, n. 1, p. 55–86.
[2] J.M. Luna, P. Fournier-Viger, and S. Ventura, 2019, Frequent itemset mining: A 25 years review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 9, n. 6

Candidate Generation: An SQL Implementation

Suppose the items in $\mathbf{L_{k+1}}$ are listed in an order

Step 1: self-joining L_k

insert into $\mathbf{C_{k+1}}$

select $\mathbf{p.item_1}$, $\mathbf{p.item_2}$, …, $\mathbf{p.item_k}$, $\mathbf{q.item_k}$

from $\mathbf{L_{k-1}}$ p, $\mathbf{L_{k-1}}$ q

where $\mathbf{p.item_1} = \mathbf{q.item_1}$ and … and $\mathbf{p.item_{k-1}} = \mathbf{q.item_{k-1}}$
and $\mathbf{p.item_k} < \mathbf{q.item_k}$

Step 2: pruning

forall itemsets c in C_{k+1} do

- forall (k)-subsets s of C do
  - if (s is not in L_k) then delete c from C_{k+1}

Use object-relational extensions like UDFs, BLOBs, and Table functions for efficient implementation

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Basic Limitations of Apriori Method

- Limitations
  - Multiple scans of transaction database
  - Huge number of candidates
  - High workload of support counting for candidates
- Bottlenecks of the Apriori approach
  - Breadth-first (i.e., level-wise) search
  - Candidate generation and test
    - Often generates a huge number of candidates
- Improvements
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Partition: Scan Database Only Twice

- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
    - Preferably fitting into main memory
  - Scan 2: consolidate global frequent patterns

DB_1 + DB_2 + . . . + DB_k = DB

$\sup_1(i) < \sigma DB_1$ $\quad$ $\sup_2(i) < \sigma DB_2$ $\quad$ $sup_k(i) < \sigma DB_k$ $\quad$ $\sup(i) < \sigma DB$

[1] R. Agrawal and J.C. Shafer, 1996, Parallel mining of association rules, *IEEE Transactions on Knowledge and Data Engineering*, v. 8, n. 6, p. 962–969.
[2] S. Biswas, N. Biswas, and K.C. Mondal, 2018, Parallel apriori based distributed association rule mining: A comprehensive survey, In: *Proceedings - 2018 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2018*, p. 202–207
[3] Y. Djenouri, D. Djenouri, J.C.-W. Lin, and A. Belhadi, 2018, Frequent itemset mining in big data with effective single scan algorithms, *IEEE Access*, v. 6, p. 68013–68026.

Sampling for Frequent Patterns

- Select a sample of original database, mine frequent patterns within sample using Apriori
- Scan database once to verify frequent itemsets found in sample, only borders of closure of frequent patterns are checked
  - Example: check $abcd$ instead of ab, ac, …, etc.
- Scan database again to find missed frequent patterns

[1] R. Agrawal and J.C. Shafer, 1996, Parallel mining of association rules, *IEEE Transactions on Knowledge and Data Engineering*, v. 8, n. 6, p. 962–969.
[2] S. Biswas, N. Biswas, and K.C. Mondal, 2018, Parallel apriori based distributed association rule mining: A comprehensive survey, In: *Proceedings - 2018 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2018*, p. 202–207
[3] Y. Djenouri, D. Djenouri, J.C.-W. Lin, and A. Belhadi, 2018, Frequent itemset mining in big data with effective single scan algorithms, *IEEE Access*, v. 6, p. 68013–68026.

Pattern-Growth Approach: Mining Frequent Patterns Without Candidate Generation

- The FPGrowth Approach
  - Depth-first search
  - Avoid explicit candidate generation
- Major philosophy:
  - Grow long patterns from short ones using local frequent items only

$\mathbf{abc}$ is a frequent pattern Get all transactions having $\mathbf{abc}$, i.e., project DB on $\mathbf{abc}$: DB$\vert\mathbf{abc}$ $\mathbf{d}$ is a local frequent item in DB$\vert\mathbf{abc}$ $\rightarrow$ $\mathbf{abcd}$ is a frequent pattern

[1] J. Han, J. Pei, and Y. Yin, 2000, Mining frequent patterns without candidate generation, *SIGMOD Record*, v. 29, n. 2, p. 1–12.

Construct FP-tree from a Transaction Database

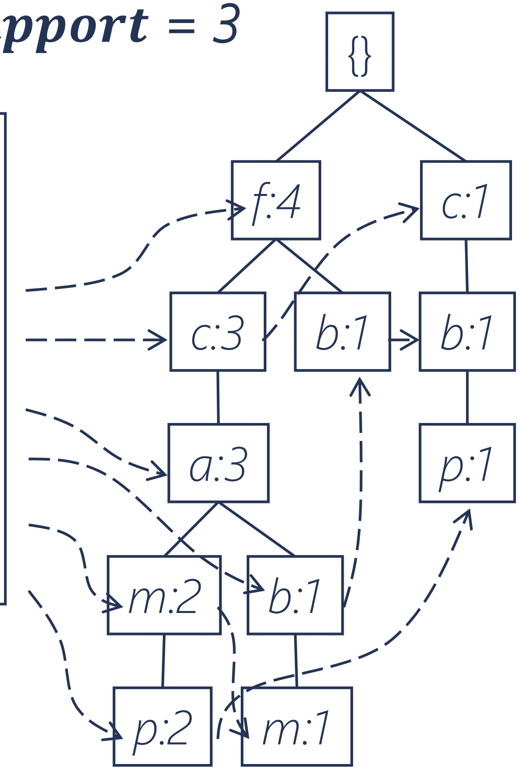
1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, f-list
3. Scan DB again, construct FP-tree

TID	*Items bought*	*(ordered) frequent items*
100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}
200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}
300	{*b, f, h, j, o, w*}	{*f, b*}
400	{*b, c, k, s, p*}	{*c, b, p*}
500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}

$\boldsymbol{minsupport} = 3$

Header Table

Item	*frequency*
f	4
c	4
a	3
b	3
m	3
p	3



F-list = $\boldsymbol{f - c - a - b - m - p}$

[1] J. Han, J. Pei, and Y. Yin, 2000, Mining frequent patterns without candidate generation, *SIGMOD Record*, v. 29, n. 2, p. 1–12.

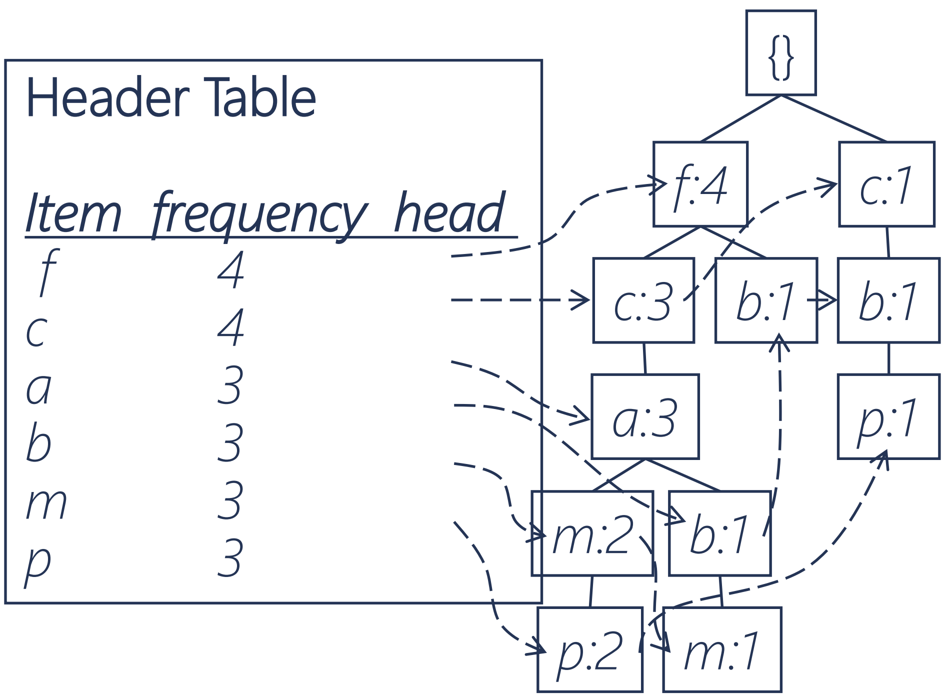
Partition Patterns and Databases

- Frequent patterns can be partitioned into subsets according to f-list
  - F-list = $f - c - a - b - m - p$
    - Patterns containing p
    - Patterns having m but no p
    - …
    - Patterns having c but no a, b, m, p
    - Pattern having f but no c, a, b, m, p
- Completeness and non-redundancy

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Find Patterns Having P From P-conditional Database

- Starting at the frequent item header table in the FP-tree
- Traverse the FP-tree by following the link of each frequent item p
- Sum all transformed prefix paths of item p to form p's conditional pattern base



F-list = $f - c - a - b - m - p$

Conditional pattern bases

item	*cond. pattern base*
p	*fcam:2, cb:1*
m	*fca:2, fcab:1*
b	*fca:1, f:1, c:1*
a	*fc:3*
c	*f:3*

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

From Conditional Pattern-bases to Conditional FP-trees

- For each pattern-base
  - Accumulate the count for each item in the base
  - Construct the FP-tree for the frequent items of the pattern base

m-conditional pattern base: *fca:2, fcab:1*

m-conditional FP-tree

{}
|
f:3
|
c:3
|
a:3
|
~~*b:1*~~

→

All frequent patterns relate to *m*
m,
fm, cm, am,
fcm, fam, cam,
fcam

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Benefits of the FP-tree Structure

- Completeness
  - Preserve complete information for frequent pattern mining
  - Never break a long pattern of any transaction
- Compactness
  - Reduce irrelevant info—infrequent items are gone
  - Items in frequency descending order: the more frequently occurring, the more likely to be shared
  - Never be larger than the original database (not count node-links and the count field)

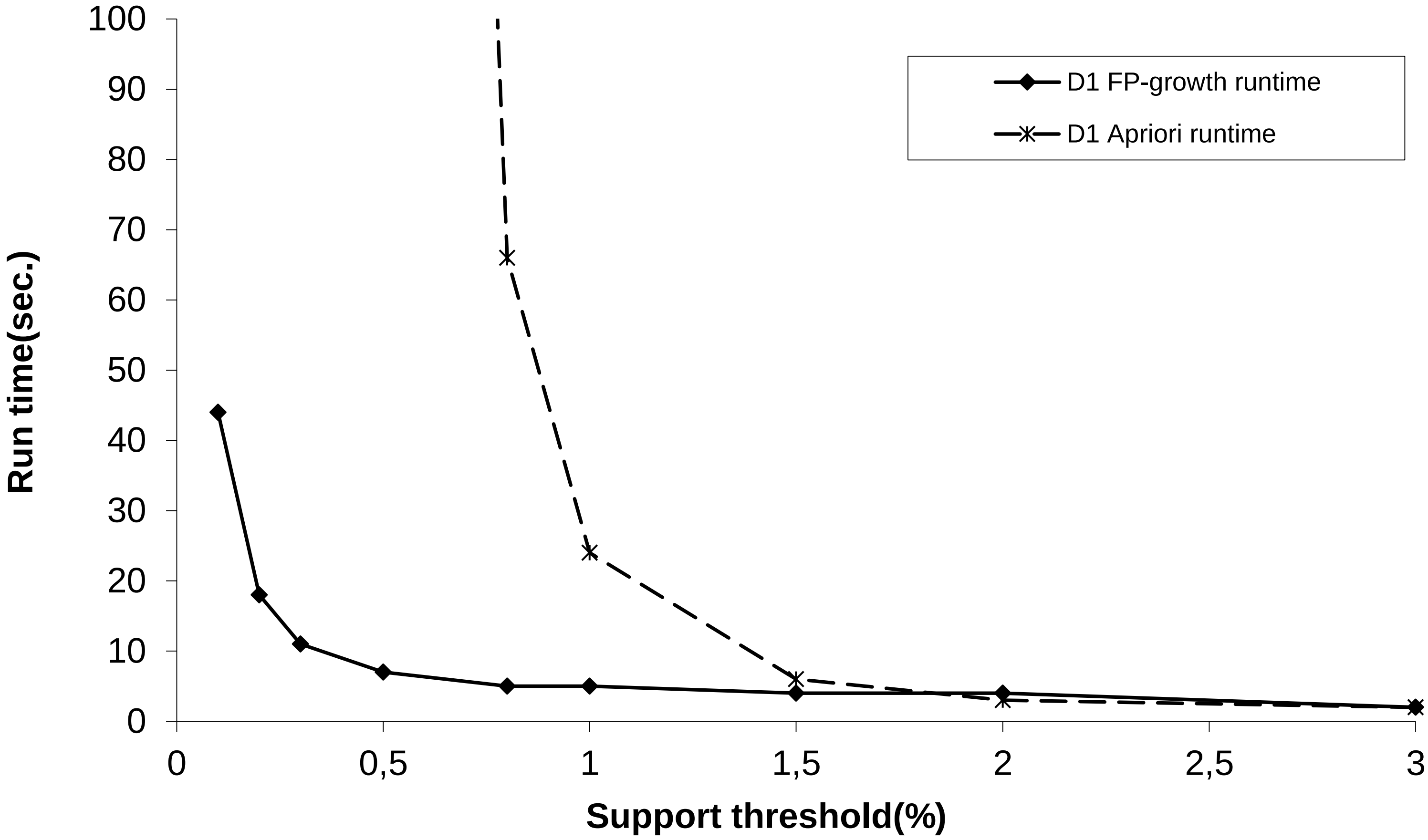
[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

The Frequent Pattern Growth Mining Method

- Idea: Frequent pattern growth
  - Recursively grow frequent patterns by pattern and database partition
- Method
  - For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
  - Repeat the process on each newly created conditional FP-tree
  - Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

FP-Growth vs. Apriori: Scalability With the Support Threshold



[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Advantages of the Pattern Growth Approach

- Divide-and-conquer:
  - Decompose both the mining task and DB according to the frequent patterns obtained so far
  - Lead to focused search of smaller databases
- Other factors
  - No candidate generation, no candidate test
  - Compressed database: FP-tree structure
  - No repeated scan of entire database

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

ECLAT: Mining by Exploring Vertical Data Format

- Vertical format: t($\{A, B\}$) = $\{T_{11}, T_{25}$, …}
  - tid-list: list of trans.-ids containing an itemset
- Deriving frequent patterns based on vertical intersections
  - t(X) = t(Y): X and Y always happen together
  - t(X) $\subset$ t(Y): transaction having X always has Y
  - Using diffset to accelerate mining
  - Only keep track of differences of tids
  - t(X) = $\{T_1, T_2, T_3\}$, t(XY) = $\{T_1, T_3\}$
  - Diffset (XY, X) = $\{T_2\}$

[1] M.J. Zaki, 2000, Scalable algorithms for association mining, *IEEE Transactions on Knowledge and Data Engineering*, v. 12, n. 3, p. 372–390.

Reading papers

Data Min Knowl Disc (2007) 15:55–86
DOI 10.1007/s10618-006-0059-1

Frequent pattern mining: current status and future directions

Jiawei Han · Hong Cheng · Dong Xin ·
Xifeng Yan

Received: 22 June 2006 / Accepted: 8 November 2006 / Published online: 27 January 2007
Springer Science+Business Media, LLC 2007

Abstract Frequent pattern mining has been a focused theme in data mining research for over a decade. Abundant literature has been dedicated to this research and tremendous progress has been made, ranging from efficient and scalable algorithms for frequent itemset mining in transaction databases to numerous research frontiers, such as sequential pattern mining, structured pattern mining, correlation mining, associative classification, and frequent pattern-based clustering, as well as their broad applications. In this article, we provide a brief overview of the current status of frequent pattern mining and discuss a few promising research directions. We believe that frequent pattern mining research has substantially broadened the scope of data analysis and will have deep impact on data mining methodologies and applications in the long run. However, there are still some challenging research issues that need to be solved before frequent pattern mining can claim a cornerstone approach in data mining applications.

Keywords Frequent pattern mining · Association rules · Data mining research · Applications

Responsible editor: Geoff Webb.

The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678/06-42771 and NSF BDI-05-15813. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

J. Han (✉) · H. Cheng · D. Xin · X. Yan
Department of Computer Science
University of Illinois, 1304 West Springfield Ave.
Urbana, IL 61801, USA
e-mail: hanj@cs.uiuc.edu

Springer

Han, J., Cheng, H., Xin, D., Yan, X., (2007), "Frequent pattern mining: Current status and future directions", Data Mining and Knowledge Discovery, v. 15, n. 1, p. 55–86.
Luna, J. M., Fournier-Viger, P., Ventura, S., (2019), "Frequent itemset mining: A 25 years review", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 9, n. 6

Analysis of rules

- Strong rules are not necessarily interesting
  - High support and confidence
- Look at the dataset
  - drink coffee → drink milk [40%, 66.7%]
    - is misleading. The overall % of customers drink milk is 75% > 66.7%.
  - drink coffee → ¬ drink milk [20%, 33.3%]
    - is more accurate, although with lower support and confidence

	Coffee	¬ Coffee	Sum (row)
Milk	2000	1750	3750
¬ Milk	1000	250	1250
Sum(col.)	3000	2000	5000

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Interestingness Measures: Lift between itemsets

- Lift
  - Measure of dependent or correlated events: lift
  - >= 1, correlated
  - < 1, non-correlated
- $lift = \frac{P(A \cap B)}{P(A)P(B)}$

	Coffee	¬ Coffee	Sum (row)
Milk	2000	1750	3750
¬ Milk	1000	250	1250
Sum(col.)	3000	2000	5000

$$\text{Coffee} \rightarrow \text{Milk}$$
$$lift(\text{Coffee}, \text{Milk}) = \frac{\frac{2000}{5000}}{\frac{3000}{5000} \cdot \frac{3750}{5000}} = 0.89$$

$$\text{Coffee} \rightarrow \neg\text{Milk}$$
$$lift(\text{Coffee}, \neg\text{Milk}) = \frac{\frac{1000}{5000}}{\frac{3000}{5000} \cdot \frac{1250}{5000}} = 1.33$$

(*) lift is computed using statistics notation

Interestingness Measures: χ^2 between variables

- $\chi^2 = \sum \frac{(observed_{ij} - expected_{ij})^2}{expected_{ij}}$
  - $expected_{ij} = \frac{observed_{ij}}{\sum observed_i} \times \frac{observed_{ij}}{\sum observed_j} \times \sum observed_{ij}$
  - p-value > 5% independent
  - p-value ≤ 5% (not independent)

Contingency Table with the Expected Values

	Coffee	¬ Coffee	Sum (row)
Milk	2000 (1777)	1750 (2041)	3750
¬ Milk	1000 (1333)	250 (125)	1250
Sum(col.)	3000	2000	5000

Expected value for Coffee and Milk = $\frac{2000}{3000} \cdot \frac{2000}{3750} \cdot 5000 = 1777$

$\chi^2 = 276$

p-value $\leq 2.2 \times 10^{-16}$ (are not independent)

Geng, L., Hamilton, H. J., (2006), "Interestingness measures for data mining: A survey", ACM Computing Surveys, v. 38, n. 3, p. 3.
McGarry, K., (2005), "A survey of interestingness measures for knowledge discovery", Knowledge Engineering Review, v. 20, n. 1, p. 39–61.

Null-transaction problem
Milk (m) and Coffee (c)

- Considering the analysis of two itemsets of interests: Milk and Coffee
  - In D_1 and D_2, mc are positively associated as they are higher than $\overline{m}c$ and $m\overline{c}$
  - There is a slightly higher chance of mc rather than $\overline{m}c$ or $m\overline{c}$
  - However, Lift(Milk, Coffee) and χ^2 provides contradictory signs due to the difference between $\overline{mc}$. It is due to null-transactions.
  - A null-transaction is a transaction that does not contain any of the itemsets being examined

	Milk Coffee	¬Milk Coffee	Milk ¬Coffee	¬Milk ¬Coffee	Lift(Milk, Coffee)	χ^2
D_1	10000	1000	1000	100000	9.25	90547*
D_2	10000	1000	1000	100	1	0

lift and χ^2 generate dramatically different measure values for D_1 and D_2 due to their sensitivity to $\overline{mc}$

Geng, L., Hamilton, H. J., (2006), "Interestingness measures for data mining: A survey", ACM Computing Surveys, v. 38, n. 3, p. 3.
McGarry, K., (2005), "A survey of interestingness measures for knowledge discovery", Knowledge Engineering Review, v. 20, n. 1, p. 39–61.

Null-invariant metrics: $kulc$ and IR

- $kulc(A, B) = \frac{\sup(A \cup B)}{2}\left(\frac{1}{\sup(A)} + \frac{1}{\sup(B)}\right)$
- $IR(A, B) = \frac{|\sup(A) - sup(B)|}{\sup(A) + \sup(B) - \sup(A \cup B)}$

$lift(a, b)$ varies from 0 to ∞ (<1: unrelated; ≥1: related)
$\chi^2(A, B)$ varies from 0 to ∞ (0 is independent; p-value $\leq$ 5% is related)
$kulc(a, b)$ varies from 0 to 1 (0 is negative related; 0.5 is neutral; 1 is positive related)
$IR(a, b)$ varies from 0 to 1 (0 is balanced; 1 is imbalanced)

Geng, L., Hamilton, H. J., (2006), "Interestingness measures for data mining: A survey", ACM Computing Surveys, v. 38, n. 3, p. 3.
McGarry, K., (2005), "A survey of interestingness measures for knowledge discovery", Knowledge Engineering Review, v. 20, n. 1, p. 39–61.

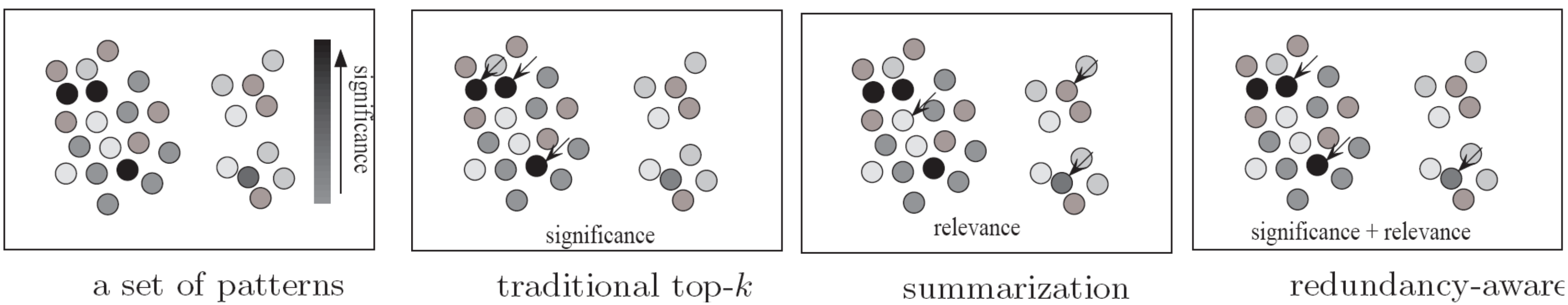
Comparison of Interestingness Measures for Milk (m) and Coffee (c)

- positively associated in D_1 and D_2
  - For D_1 and D_2, m and c are positively associated because mc (10000) is considerably greater than $\overline{m}c$ (1000) and $m\overline{c}$ (1000)
    - For people who bought milk (m = 10000 + 1000 = 11000), it is very likely that they also bought coffee (mc/m 10/11 = 91%), and vice versa
  - lift and χ^2 generate dramatically different measure values for D_1 and D_2 due to their sensitivity to $\overline{mc}$
- negatively associated in D_3
- neutral in D_4 and D_5
  - Kulk and IR together present a clear picture for datasets D_4 and D_5:
  - D_4 is balanced & neutral; D_5 is imbalanced & neutral

	Milk Coffee	¬Milk Coffee	Milk ¬Coffee	¬Milk ¬Coffee	Lift(Milk, Coffee)	χ^2	kulc(Milk, Coffee)	IR(Milk, Coffee)
D_1	10000	1000	1000	100000	9.25	90547*	0.91	0
D_2	10000	1000	1000	100	1	0	0.91	0
D_3	100	1000	1000	100000	8.43	662*	0.09	0
D_4	1000	1000	1000	100000	25.75	24715*	0.5	0
D_5	1000	100	10000	100000	9.18	8163*	0.5	0.89

Redundancy-Award Top-k Patterns

- Why redundancy-aware top-k patterns?
  - Desired patterns: high significance & low redundancy
  - Propose the MMS (Maximal Marginal Significance) for measuring the combined significance of a pattern set
  - Extracting Redundancy-Aware Top-K Patterns



[1] C.C. Aggarwal and J. Han, 2014, *Frequent Pattern Mining*. Springer.

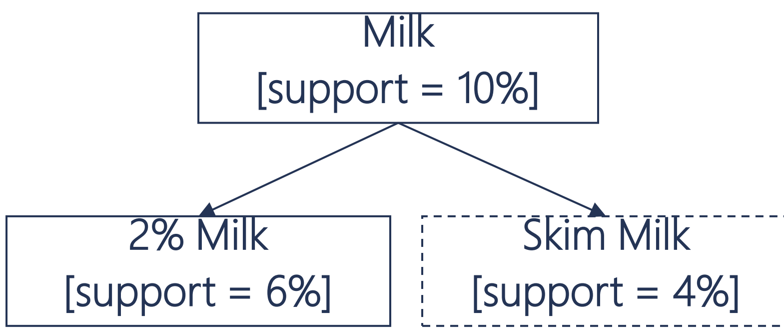
Mining Multiple-Level Association Rules

- Items often form hierarchies
- Flexible support settings
  - Items at the lower level are expected to have lower support
- Exploration of shared multi-level mining

uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%



reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 3%

[1] C.C. Aggarwal and J. Han, 2014, *Frequent Pattern Mining*. Springer.

Multi-level Association: Flexible Support and Redundancy filtering

- Flexible min-support thresholds: Some items are more valuable but less frequent
  - Use non-uniform, group-based min-support
  - E.g., {diamond, watch, camera}: 0.05%; {bread, milk}: 5%; …
- Redundancy Filtering: Some rules may be redundant due to "ancestor" relationships between items
  - milk → wheat bread [support = 8%, confidence = 70%]
    - light milk (2%) → wheat bread [2%, 72%]
  - The first rule is an ancestor of the second rule
- A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor

[1] C.C. Aggarwal and J. Han, 2014, *Frequent Pattern Mining*. Springer.

Mining Multi-Dimensional Association

- Single-dimensional rules:
    - buys(X, “milk”) → buys(X, “bread”)
- Multi-dimensional rules: ≥ 2 dimensions or predicates
  - Inter-dimension assoc. rules (no repeated predicates)
    - age(X,”19-25”) ∧ occupation(X,“student”) → buys(X, “coke”)
  - hybrid-dimension assoc. rules (repeated predicates)
    - age(X,”19-25”) ∧ buys(X, “popcorn”) → buys(X, “coke”)
- Categorical Attributes:
  - finite number of possible values, no ordering among values
- Quantitative Attributes:
  - Numeric, implicit ordering among values
  - discretization, clustering, and gradient approaches

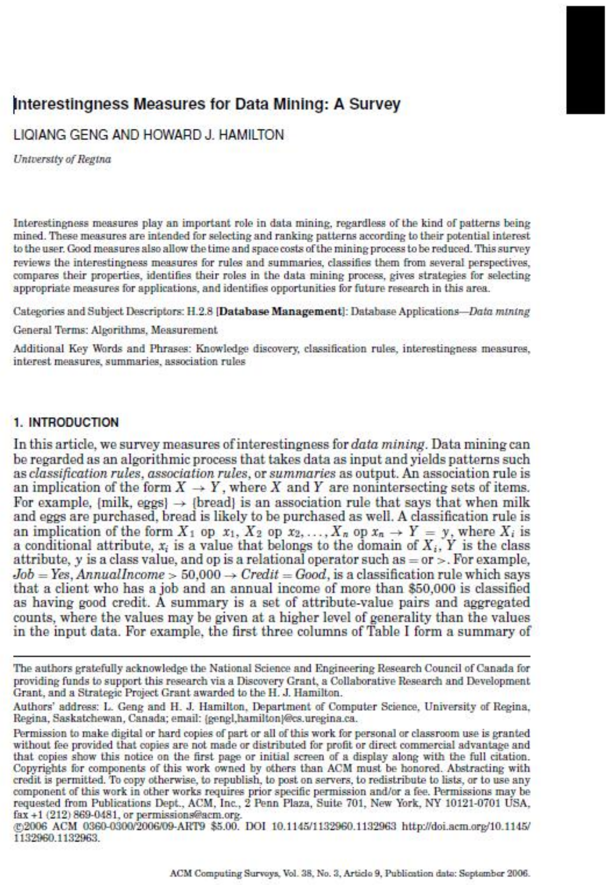
[1] C.C. Aggarwal and J. Han, 2014, *Frequent Pattern Mining*. Springer.

Negative and Rare Patterns

- Rare patterns: Very low support but interesting
  - E.g., buying Rolex watches
  - Mining: Setting individual-based or special group-based support threshold for valuable items
- Negative patterns
  - Since it is unlikely that one buys Ford Expedition (an SUV car) and Toyota Prius (a hybrid car) together, Ford Expedition and Toyota Prius are likely negatively correlated patterns
- Negatively correlated patterns that are infrequent tend to be more interesting than those that are frequent

Geng, L., Hamilton, H. J., (2006), "Interestingness measures for data mining: A survey", ACM Computing Surveys, v. 38, n. 3, p. 3.
McGarry, K., (2005), "A survey of interestingness measures for knowledge discovery", Knowledge Engineering Review, v. 20, n. 1, p. 39–61.

Reading papers



Interestingness Measures for Data Mining: A Survey

LIQIANG GENG AND HOWARD J. HAMILTON

University of Regina

Interestingness measures play an important role in data mining, regardless of the kind of patterns being mined. These measures are intended for selecting and ranking patterns according to their potential interest to the user. Good measures also allow the time and space costs of the mining process to be reduced. This survey reviews the interestingness measures for rules and summaries, classifies them from several perspectives, compares their properties, identifies their roles in the data mining process, gives strategies for selecting appropriate measures for applications, and identifies opportunities for future research in this area.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms, Measurement

Additional Key Words and Phrases: Knowledge discovery, classification rules, interestingness measures, interest measures, summaries, association rules

1. INTRODUCTION

In this article, we survey measures of interestingness for *data mining*. Data mining can be regarded as an algorithmic process that takes data as input and yields patterns such as *classification rules*, *association rules*, or *summaries* as output. An association rule is an implication of the form $X \rightarrow Y$, where X and Y are nonintersecting sets of items. For example, {milk, eggs} $\rightarrow$ {bread} is an association rule that says that when milk and eggs are purchased, bread is likely to be purchased as well. A classification rule is an implication of the form X_1 op x_1, X_2 op $x_2, \ldots, X_n$ op $x_n \rightarrow Y = y$, where X_i is a conditional attribute, x_i is a value that belongs to the domain of X_i, Y is the class attribute, y is a class value, and op is a relational operator such as = or >. For example, $Job = Yes, AnnualIncome > 50{,}000 \rightarrow Credit = Good$, is a classification rule which says that a client who has a job and an annual income of more than $50,000 is classified as having good credit. A summary is a set of attribute-value pairs and aggregated counts, where the values may be given at a higher level of generality than the values in the input data. For example, the first three columns of Table I form a summary of

The authors gratefully acknowledge the National Science and Engineering Research Council of Canada for providing funds to support this research via a Discovery Grant, a Collaborative Research and Development Grant, and a Strategic Project Grant awarded to the H. J. Hamilton.

Authors' address: L. Geng and H. J. Hamilton, Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada; email: {gengl,hamilton}@cs.uregina.ca.

Geng, L., Hamilton, H. J., (2006), "Interestingness measures for data mining: A survey", ACM Computing Surveys, v. 38, n. 3, p. 3.
McGarry, K., (2005), "A survey of interestingness measures for knowledge discovery", Knowledge Engineering Review, v. 20, n. 1, p. 39–61.

What Is Sequential Pattern Mining?

- Given a set of sequences, find the complete set of frequent subsequences
- Sequential pattern mining: find the complete set of patterns, satisfying the minimum support (frequency) threshold

A *sequence*: < (ef) (ab) (df) c b >

A sequence database

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

- An element may contain a set of items
- Items within an element are unordered and we list them alphabetically

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

Given *support threshold min_sup* = 2, <(ab)c> is a *sequential pattern*

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Sequential Pattern Mining Algorithms

- Concept introduction and an initial Apriori-like algorithm
  - Agrawal & Srikant: Mining sequential patterns
- Requirement: efficient, scalable, complete, minimal database scans, and be able to incorporate various kinds of user-specific constraints
- Representative algorithms
  - GSP (Generalized Sequential Patterns)
  - Vertical format-based mining: SPADE
  - Pattern-growth methods: PrefixSpan
- Constraint-based sequential pattern mining
- Mining closed sequential patterns: CloSpan

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

The Apriori Property of Sequential Patterns

- A basic property: Apriori
  - If a sequence S is not frequent
  - Then none of the super-sequences of S is frequent
  - E.g, <hb> is infrequent → so do <hab> and <(ah)b>

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Given *support threshold min_sup* =2

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

GSP—Generalized Sequential Pattern Mining

- GSP (Generalized Sequential Pattern) mining algorithm
- Outline of the method
  - Initially, every item in DB is a candidate of length: 1
  - for each level (i.e., sequences of length: k) do
    - scan database to collect support count for each candidate sequence
    - generate candidate length: (k+1) sequences from length: k frequent sequences using Apriori
  - repeat until no frequent sequence or no candidate can be found
- Major strength: Candidate pruning by Apriori

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

GSP Algorithm – Candidate Generation

1. **Join Phase.** We generate candidate sequences by joining L_{k-1} with L_{k-1}. A sequence s_1 joins with s_2 if the subsequence obtained by dropping the first item of s_1 is the same as the subsequence obtained by dropping the last item of s_2. The candidate sequence generated by joining s_1 with s_2 is the sequence s_1 extended with the last item in s_2. The added item becomes a separate element if it was a separate element in s_2, and part of the last element of s_1 otherwise. When joining L_1 with L_1, we need to add the item in s_2 both as part of an itemset and as a separate element, since both $\langle$(x) (y)$\rangle$ and $\langle$(x y)$\rangle$ give the same sequence $\langle$(y)$\rangle$ upon deleting the first item. (Observe that s_1 and s_2 are contiguous subsequences of the new candidate sequence.)

Frequent 3-Sequences	Candidate 4-Sequences after join	Candidate 4-Sequences after pruning
$\langle$(1, 2) (3)$\rangle$	$\langle$(1, 2) (3, 4)$\rangle$	$\langle$(1, 2) (3, 4)$\rangle$
$\langle$(1, 2) (4)$\rangle$	$\langle$(1, 2) (3) (5)$\rangle$	
$\langle$(1) (3, 4)$\rangle$		
$\langle$(1, 3) (5)$\rangle$		
$\langle$(2) (3, 4)$\rangle$		
$\langle$(2) (3) (5)$\rangle$		

Example Figure 3 shows L_3, and C_4 after the join and prune phases. In the join phase, the sequence $\langle$(1, 2) (3)$\rangle$ joins with $\langle$(2) (3, 4)$\rangle$ to generate $\langle$(1, 2) (3, 4)$\rangle$ and with $\langle$(2) (3) (5)$\rangle$ to generate $\langle$(1, 2) (3) (5)$\rangle$. The remaining sequences do not join with any sequence in L_3. For instance, $\langle$(1, 2) (4)$\rangle$ does not join with any sequence since there is no sequence of the form $\langle$(2) (4 x)$\rangle$ or $\langle$(2) (4) (x)$\rangle$. In the prune phase, $\langle$(1, 2) (3) (5)$\rangle$ is dropped since its contiguous subsequence $\langle$(1) (3) (5)$\rangle$ is not in L_3.

[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Finding Length-1 Sequential Patterns

- Examine GSP using an example
- Initial candidates: all singleton sequences
  - <a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>
- Scan database once, count support for candidates

min_sup =2

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Cand	Sup
<a>	3
<b>	5
<c>	4
<d>	3
<e>	3
<f>	2
~~<g>~~	1
~~<h>~~	1

Mooney, C. H., Roddick, J. F., (2013), "Sequential pattern mining - Approaches and algorithms", ACM Computing Surveys, v. 45, n. 2

GSP: Generating Length-2 Candidates

Isolated items

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

51 length-2 Candidates

Merged items

	<a>	<b>	<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

Without Apriori property, 8*8+8*7/2=92 candidates

Apriori prunes 44.57% candidates

Mooney, C. H., Roddick, J. F., (2013), "Sequential pattern mining - Approaches and algorithms", ACM Computing Surveys, v. 45, n. 2

Reading papers

Sequential Pattern Mining – Approaches and Algorithms

CARL H. MOONEY and JOHN F. RODDICK, Flinders University

Sequences of events, items, or tokens occurring in an ordered metric space appear often in data and the requirement to detect and analyze frequent subsequences is a common problem. Sequential Pattern Mining arose as a subfield of data mining to focus on this field. This article surveys the approaches and algorithms proposed to date.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Data mining

General Terms: Algorithms, Design

Additional Key Words and Phrases: Sequential pattern mining

ACM Reference Format:
Mooney, C. H. and Roddick, J. F. 2013. Sequential pattern mining – Approaches and algorithms. ACM Comput. Surv. 45, 2, Article 19 (February 2013), 39 pages.
DOI = 10.1145/2431211.2431218. http://doi.acm.org/10.1145/2431211.2431218.

1. INTRODUCTION

1.1. Background and Previous Research

Sequences are common, occurring in any metric space that facilitates either total or partial ordering. Events in time, codons, or nucleotides in an amino acid, Web site traversal, computer networks, and characters in a text string are examples of where the existence of sequences may be significant and where the detection of frequent (totally or partially ordered) subsequences might be useful. Sequential pattern mining has arisen as a technology to discover such subsequences.

The sequential pattern mining problem was first addressed by Agrawal and Srikant [1995] and was defined as follows.

> "Given a database of sequences, where each sequence consists of a list of transactions ordered by transaction time and each transaction is a set of items, sequential pattern mining is to discover all sequential patterns with a user-specified minimum support, where the support of a pattern is the number of data sequences that contain the pattern."

Since then there has been a growing number of researchers in the field, evidenced by the volume of papers produced, and the problem definition has been reformulated in a number of ways. For example, Garofalakis et al. [1999] described it as follows.

> "Given a set of data sequences, the problem is to discover subsequences that are frequent, that is, the percentage of data sequences containing them exceeds a user-specified minimum support"

19

Authors' addresses: C. H. Mooney (corresponding author), J. F. Roddick, School of Computer Science, Engineering and Mathematics, Flinders University, P.O. Box 2100, Adelaide 5001, South Australia; email: Carl.mooney@flinders.edu.au.

DOI 10.1145/2431211.2431218. http://doi.acm.org/10.1145/2431211.2431218.

Mooney, C. H., Roddick, J. F., (2013), "Sequential pattern mining - Approaches and algorithms", ACM Computing Surveys, v. 45, n. 2

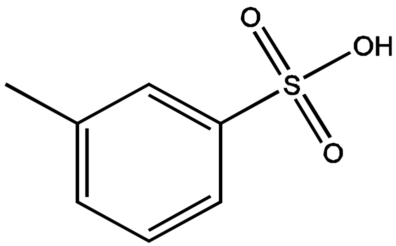
Graph Pattern Mining

- Frequent subgraphs
  - A (sub)graph is frequent if its support (occurrence frequency) in a given dataset is no less than a minimum support threshold
- Applications of graph pattern mining
  - Mining biochemical structures
  - Program control flow analysis
  - Mining XML structures or Web communities
  - Building blocks for graph classification, clustering, compression, comparison, and correlation analysis

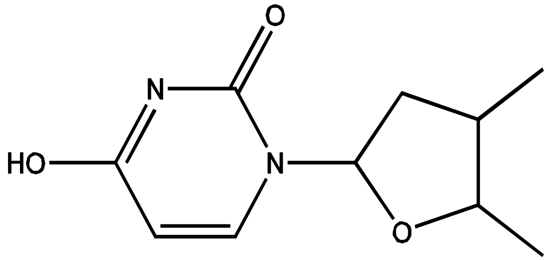
[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

Example: Frequent Subgraphs

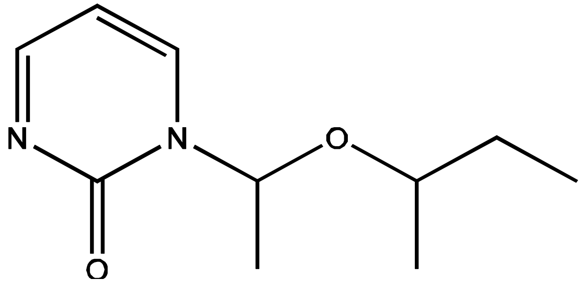
Graph Dataset



(A)



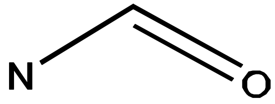
(B)



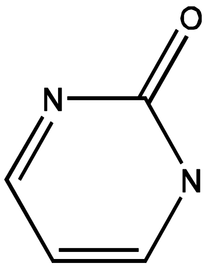
(C)

Frequent Patterns (Min Support Is 2)

(1)

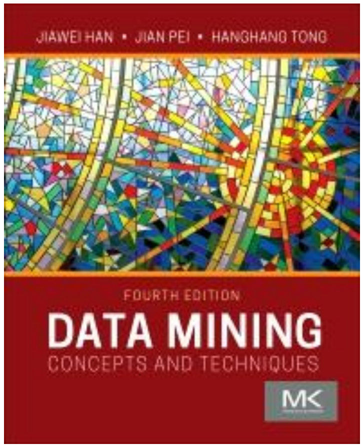


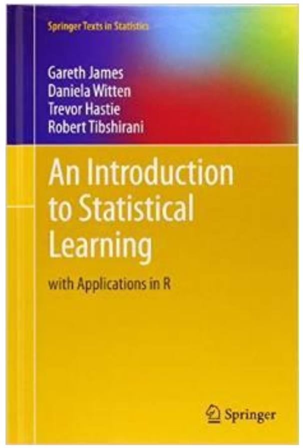
(2)

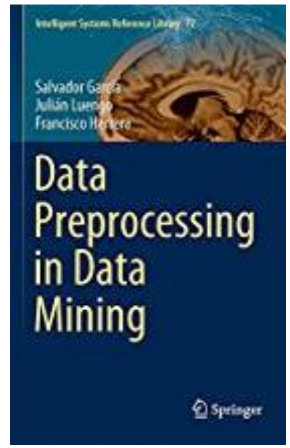


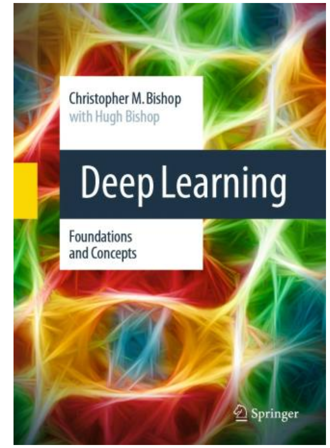
[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022..

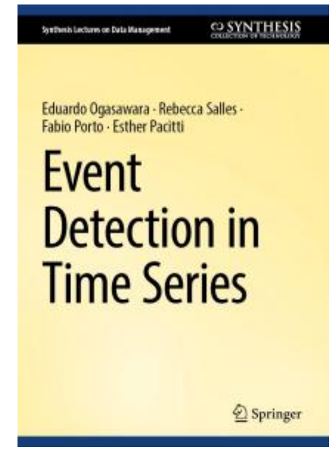
Main References











[1] J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022.

[2] G. M. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: With Applications in R. Springer Nature, 2021.

[3] S. Garcia, J. Luengo, and F. Herrera, Data Preprocessing in Data Mining. Springer, 2014.

[4] C. M. Bishop and H. Bishop, Deep Learning: Foundations and Concepts. Springer Nature, 2023.

[5] E. Ogasawara, R. Salles, F. Porto, and E. Pacitti, Event Detection in Time Series, 1st ed. in Synthesis Lectures on Data Management. Cham: Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-75941-3.

Slides and videos at: https://eic.cefet-rj.br/~eogasawara/data-mining/