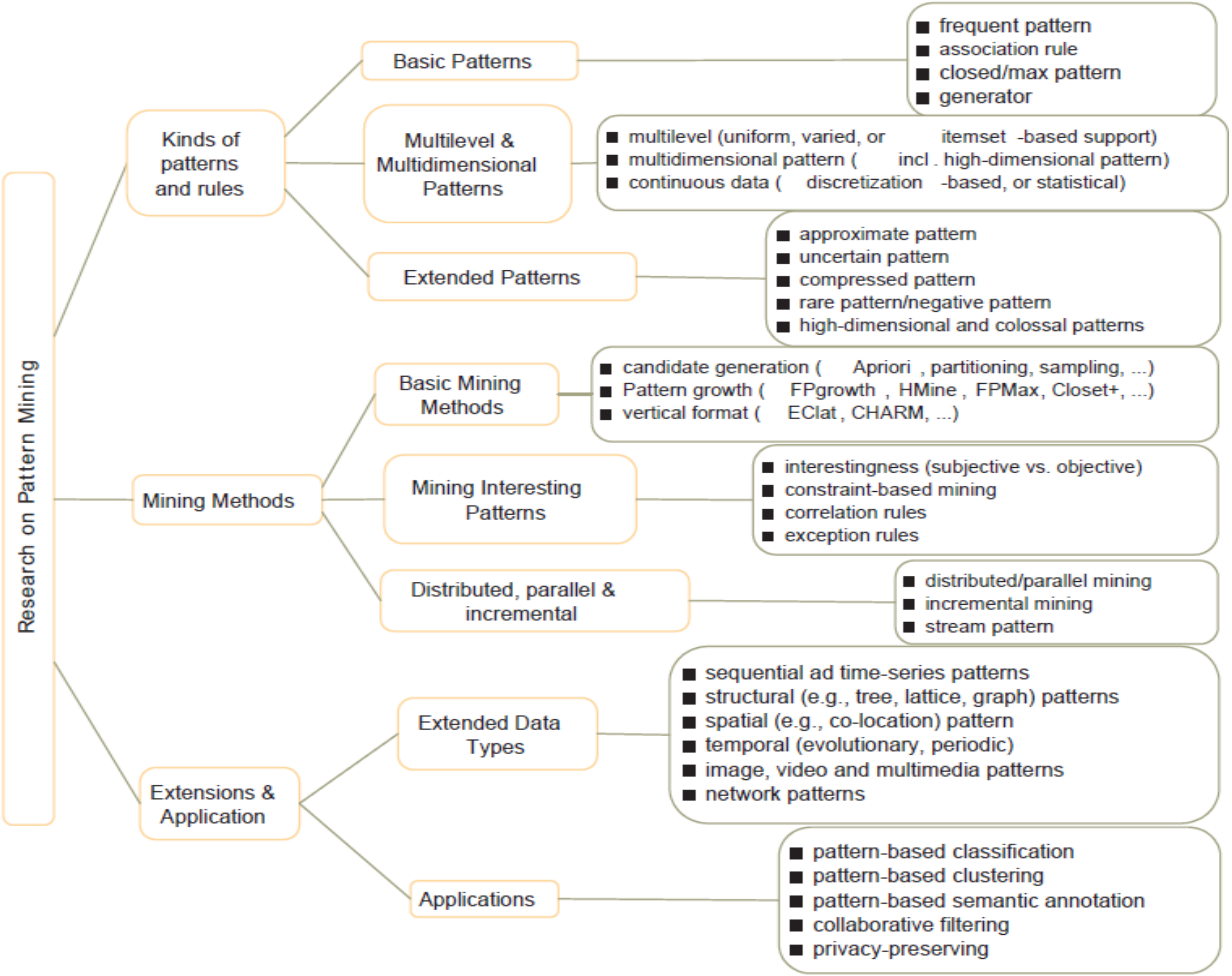
Chapter 7. Advanced Frequent Pattern Mining: Diverse Patterns

Meng Jiang

CSE 40647/60647 Data Science Fall 2017

Introduction to Data Mining

Research on Pattern Mining: A Road Map



Advanced Frequent Pattern Mining

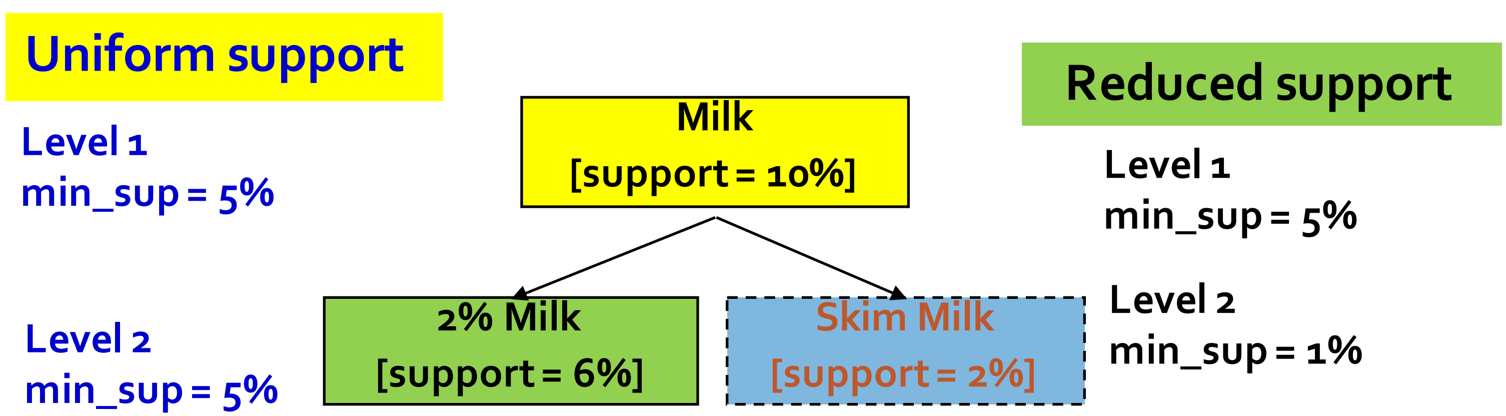
- **Mining Diverse Patterns**
- Constraint-Based Frequent Pattern Mining
- Sequential Pattern Mining
- Graph Pattern Mining

Mining Diverse Patterns

- Mining Multiple-Level Associations
- Mining Multi-Dimensional Associations
- Mining Quantitative Associations
- Mining Negative Correlations

Mining Multiple-Level Frequent Patterns

- Items often form hierarchies
  - Ex.: Dairyland 2% milk; Wonder wheat bread
- How to set min-support thresholds?
  - Uniform min-support across multiple levels (reasonable?)
  - Level-reduced min-support: Items at the lower level are expected to have lower support



Redundancy Filtering at Mining Multi-Level Associations

- Multi-level association mining may generate many redundant rules
- Redundancy filtering: Some rules may be redundant due to "ancestor" relationships between items

  (Suppose the 2% milk sold is about ¼ of milk sold in gallons)

  - milk $\Rightarrow$ wheat bread [support = 8%, confidence = 70%] (1)
  - 2% milk $\Rightarrow$ wheat bread [support = 2%, confidence = 72%] (2)
- A rule is *redundant* if its support is close to the "expected" value, according to its "ancestor" rule, and it has a similar confidence as its "ancestor"
  - Rule (1) is an ancestor of rule (2), which one to prune?

Customized Min-Supports for Different Kinds of Items

- We have used the same min-support threshold for all the items or item sets to be mined in each association mining
- In reality, some items (e.g., diamond, watch, …) are valuable but less frequent
- It is necessary to have customized min-support settings for different kinds of items
- One Method: Use group-based "individualized" min-support
  - E.g., {diamond, watch}: 0.05%; {bread, milk}: 5%; …

Mining Multi-Dimensional Associations

- Single-dimensional rules (e.g., items are all in "product" dimension)
  - buys(X, "milk") ⇒ buys(X, "bread")
- Multi-dimensional rules (i.e., items in ≥ 2 dimensions or predicates)
  - Inter-dimension association rules (*no repeated predicates*)
    - age(X, "18-25") ∧ occupation(X, "student") ⇒ buys(X, "coke")
  - Hybrid-dimension association rules (*repeated predicates*)
    - age(X, "18-25") ∧ buys(X, "popcorn") ⇒ buys(X, "coke")
- Attributes can be categorical or numerical
  - Categorical Attributes (e.g., *profession*, *product*: no ordering among values): Data cube for inter-dimension association
  - Quantitative Attributes: Numeric, implicit ordering among values—discretization, clustering, and gradient approaches

Mining Quantitative Associations

- Mining quantitative associations
  - Ex.: Gender = female ⇒ Wage: mean=$7/hr (overall mean = $9)
  - LHS: a subset of the population
  - RHS: an ***extraordinary*** behavior of this subset
- Rule condition can be categorical or numerical
  - Ex.: (Gender = female) ^ (South = yes) ⇒ mean wage = $6.3/hr
  - Ex.: Education in [14-18] (yrs) ⇒ mean wage = $11.64/hr
- Data cube technology?

Rare Patterns vs. Negative Patterns

- Rare patterns
  - Very low support but interesting (e.g., buying Rolex watches)
  - How to mine them? Setting individualized, group-based min-support thresholds for different groups of items
- Negative patterns
  - Negatively correlated: Unlikely to happen together
  - Ex.: Since it is unlikely that the same customer buys both a Ford Expedition (an SUV car) and a Ford Fusion (a hybrid car), buying a Ford Expedition and buying a Ford Fusion are likely negatively correlated patterns
  - How to define negative patterns?

Defining Negative Correlated Patterns

- A support-based definition
  - If itemsets A and B are both frequent but rarely occur together, i.e., sup(A U B) << sup (A) × sup(B)
  - Then A and B are negatively correlated

Does this remind you the definition of *lift*?

- Is this a good definition for large transaction datasets?
- Ex.: Suppose a store sold two needle packages A and B 100 times each, but only one transaction contained both A and B
  - When there are in total 200 transactions, we have
    - s(A U B) = 0.005, s(A) × s(B) = 0.25, s(A U B) << s(A) × s(B)
  - But when there are 10^5 transactions, we have
    - s(A U B) = $1/10^5$, s(A) × s(B) = $1/10^3 \times 1/10^3$, s(A U B) > s(A) × s(B)
  - What is the problem? — Null transactions: The support-based definition is not null-invariant!

Defining Negative Correlation: Need Null-Invariance in Definition

- A good definition on negative correlation should take care of the null-invariance problem
  - Whether two itemsets A and B are negatively correlated should not be influenced by the number of null-transactions
- A Kulczynski measure-based definition
  - If itemsets A and B are frequent but $(P(A|B) + P(B|A))/2 < \epsilon$, where ϵ is a negative pattern threshold, then A and B are negatively correlated
- For the same needle package problem:
  - No matter there are in total 200 or 10^5 transactions
  - If $\epsilon = 0.01$, we have $(P(A|B) + P(B|A))/2 = (0.01 + 0.01)/2 < \epsilon$

Advanced Frequent Pattern Mining

- Mining Diverse Patterns
- **Constraint-Based Frequent Pattern Mining**
- Sequential Pattern Mining
- Graph Pattern Mining

Why Constraint-Based Mining?

- Finding all the patterns in a dataset autonomously? — unrealistic!
  - Too many patterns but not necessarily user-interested!
- Pattern mining should be an interactive process
  - User directs what to be mined using a data mining query language (or a graphical user interface)
- Constraint-based mining
  - User flexibility: provides constraints on what to be mined
  - Optimization: explores such constraints for efficient mining
    - Constraint-based mining: Constraint-pushing, similar to push selection first in DB query processing

Meta-Rule Guided Mining

- A meta-rule can contain partially instantiated predicates & constants
  - $P_1(X, Y) \wedge P_2(X, W) \Rightarrow$ buys(X, "iPad")
- The resulting mined rule can be
  - age(X, "15-25") ^ profession(X, "student") ⇒ buys(X, "iPad")
- In general, (meta) rules can be in the form of
  - $P_1 \wedge P_2 \wedge \ldots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_r$
- Method to find meta-rules
  - Find frequent (l + r) predicates (based on *min-support*)
  - Push constants deeply when possible into the mining process
  - Also, push min_sup, min_conf, and other measures as early as possible (measures acting as constraints)

Different Kinds of Constraints Lead to Different Pruning Strategies

- Constraints can be categorized as
  - **Pattern space** pruning constraints vs. **data space** pruning constraints
- Pattern space pruning constraints
- Data space pruning constraints

Pattern Space Pruning with **Pattern Anti-Monotonicity**

- Constraint c is anti-monotone
  - If an itemset S violates constraint c, so does any of its superset
  - That is, mining on itemset S can be terminated
- Ex. 1: c1: sum(S.price) ≤ v is anti-monotone
- Ex. 2: c2: range(S.profit) ≤ 15 is anti-monotone
  - Itemset ab violates c2 (range(ab) = 40)
  - So does every superset of ab
- Ex. 3. c3: sum(S.Price) ≥ v is not anti-monotone
- Ex. 4. Is c4: support(S) ≥ σ anti-monotone?
  - Yes! Apriori pruning is essentially pruning with an anti-monotonic constraint!

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

price(item) > 0

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	5

Pattern Monotonicity and Its Roles

- A constraint c is monotone: if an itemset S satisfies the constraint c, so does any of its superset
  - That is, we do not need to check c in subsequent mining
- Ex. 1: c1: sum(S.Price) ≥ v is monotone
- Ex. 2: c2: min(S.Price) ≤ v is monotone
- Ex. 3: c3: range(S.profit) ≥ 15 is monotone
  - Itemset ab satisfies c3
  - So does every superset of ab

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

price(item) > 0

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	5

Data Space Pruning with **Data Anti-Monotonicity**

- A constraint c is *data anti-monotone*: In the mining process, if a data entry (transaction) *t* cannot satisfy constraint *c*, *t* cannot satisfy any pattern *p* under *c*
  - Data space pruning*:* Data entry *t* can be pruned
- Ex. 1: c_1: *sum(S.Profit)* ≥ *v* is data anti-monotone
  - Let constraint c_1 be: sum{S.Profit} ≥ 25
    - T_{30}: {b, c, d, f, g} can be removed since none of their combinations can make an S whose sum of the profit is ≥ 25
- Ex. 2: c_2: *min(S.Price)* ≤ *v* is data anti-monotone
    - Consider *v* = 5 but every item in transaction T_{50} has a price higher than 10

TID	Transaction
10	a, b, c, d, f, h
20	b, c, d, f, g, h
30	b, c, d, f, g
40	a, c, e, f, g

min_sup = 2

price(item) > 10

Item	Profit
a	40
b	0
c	-20
d	-15
e	-30
f	-10
g	20
h	5

Different Kinds of Constraints Lead to Different Pruning Strategies

- Constraints can be categorized as
  - Pattern space pruning constraints vs. data space pruning constraints
- Pattern space pruning constraints
  - **Anti-monotonic:** If constraint c is violated, its further mining can be terminated (=no superset)
  - **Monotonic:** If c is satisfied, no need to check c again (=all supersets)
  - Succinct: If c can be enforced by directly manipulating the data
  - Convertible: c can be converted to monotonic or anti-monotonic if items can be properly ordered in processing
- Data space pruning constraints
  - **Data anti-monotonic:** If a transaction t does not satisfy c, then t can be pruned to reduce data processing effort (=no that transaction)
  - Data succinct: Data space can be pruned at the initial pattern mining process

References: Mining Diverse Patterns

- R. Srikant and R. Agrawal, “Mining generalized association rules”, VLDB'95
- Y. Aumann and Y. Lindell, “A Statistical Theory for Quantitative Association Rules”, KDD'99
- K. Wang, Y. He, J. Han, “Pushing Support Constraints Into Association Rules Mining”, IEEE Trans. Knowledge and Data Eng. 15(3): 642-658, 2003
- D. Xin, J. Han, X. Yan and H. Cheng, "On Compressing Frequent Patterns", Knowledge and Data Engineering, 60(1): 5-29, 2007
- D. Xin, H. Cheng, X. Yan, and J. Han, "Extracting Redundancy-Aware Top-K Patterns", KDD'06
- J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent Pattern Mining: Current Status and Future Directions", Data Mining and Knowledge Discovery, 15(1): 55-86, 2007
- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, “Mining Colossal Frequent Patterns by Core Pattern Fusion”, ICDE'07

References: Constraint-Based Frequent Pattern Mining

- R. Srikant, Q. Vu, and R. Agrawal, “Mining association rules with item constraints”, KDD'97
- R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang, “Exploratory mining and pruning optimizations of constrained association rules”, SIGMOD’98
- G. Grahne, L. Lakshmanan, and X. Wang, “Efficient mining of constrained correlated sets”, ICDE'00
- J. Pei, J. Han, and L. V. S. Lakshmanan, “Mining Frequent Itemsets with Convertible Constraints”, ICDE'01
- J. Pei, J. Han, and W. Wang, “Mining Sequential Patterns with Constraints in Large Databases”, CIKM'02
- F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, “ExAnte: Anticipated Data Reduction in Constrained Pattern Mining”, PKDD'03
- F. Zhu, X. Yan, J. Han, and P. S. Yu, “gPrune: A Constraint Pushing Framework for Graph Pattern Mining”, PAKDD'07