1. Basics and Algorithms
2. Itemset Frequent Pattern & *Apriori* Principle
3. FP-Growth, FP-Tree
4. Handling Categorical Attributes
5. Sequential, Subgraph, and Infrequent Patterns

1. Basics and Algorithms

- **Given**: a database of transactions, where each transaction is a list of items
- **Find**: all rules that associate the presence of one set of items with that of another set of items
- Example *98% of people who purchase tires and auto accessories also get automotive services done*

Terminologies used in association analysis

- **Itemset**
  - A collection of one or more items. **Example**: {Milk, Bread, Diaper}
  - An itemset that contains k items is called k-itemset.
- **Support**: The support of an association pattern is the percentage of task-relevant data transaction for which the pattern is true.
  Support (A): Number of tuples containing A / Total number of tuples
  Support (A = > B): Number of tuples containing A and B / Total number of tuples
  - If minsup is set too high, we could miss item sets involving interesting rare items (e.g., expensive products)
  - If minsup is set too low, it is computationally expensive and the number of item sets is very large.
- **Confidence**: Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern.
  Confidence **(A = > B):** Number of tuples containing A and B / Total count of A
- **Association Rule:** An implication expression of the form X => Y, where X and Y are item sets.
  Example: {Milk, Diaper} => {Beer} i.e. If customer buy Milk, Diaper then they can also buy Beer
- **Closed Itemset:** An itemset is closed if none of its immediate supersets has same support as of the itmeset.
- **Lift**
  - Lift is a measure of the performance of a targeting model (association rule) at predicting or classifying cases as having an enhanced response with respect to the population as a whole, measured against a random choice targeting model.
  - Lift can be found by dividing the confidence by the unconditional probability of the consequent, or by dividing the support by the probability of the antecedent times the probability of the consequent.
    Lift = P(Y | X) /P(Y)
  - If some rule had a **lift of 1**, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events.
  - If the **lift is > 1**, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.
- **Frequent Itemset Generation Strategies**
  - **Reduce the number of candidates (M):** For complete search, **M=2d**. Use pruning techniques to reduce M.
  - **Reduce the number of transactions (N):** Reduce size of N as the size of itemset increases.
  - **Reduce the number of comparisons (NM):** Use efficient data structures to store the candidates or transactions. No need to match every candidate against every transaction.

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**

{Diaper} → {Beer},

{Milk, Bread} → {Diaper, Coke},

{Beer, Bread} → {Milk},

Fig. Association Rule in Market-Basket transactions

Association rules

- Given a set of transactions **D**, find rules that will predict the occurrence of an item (or a set of items) based on the occurrences of other items in the transaction
  Example: **Market-Basket transactions**

What Is Association Rule Mining?

- Finding frequent patterns called associations, among sets of items or objects in *transaction databases, relational databases, and other information repositories.*
- Association Rules (in data mining) are **if - then statements** that help uncover relationships between seemingly unrelated data in a relational database or other information repository.
  E.g. "*If a customer buys a dozen eggs, he is 80% likely to also purchase milk."*
- An association rule has two parts,
  i. an **antecedent (if):** is an item found in the data
  ii. a **consequent (then):** is an item that is found in combination with the antecedent.
- Association rules are created by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships.
- **Support** is an indication of how frequently the items appear in the database. **Confidence** indicates the number of times the if/then statements have been found to be true.
- In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in *shopping basket data analysis, product clustering, catalog design and store layout.*

- **Programmers** use association rules to build programs capable of machine learning. Machine learning is a type of artificial intelligence (AI) that seeks to build programs with the ability to become more efficient without being explicitly programmed.

Approaches for association rules mining

Brute- Force Approach

- List all possible association rules.
- Compute the support and confidence for each rule.
- Prune rules that fail to minimum support and minimum confidence level.
- *This approach is computationally very expensive.

Aspects of Association Rule Mining

How do we generate rules fast?

- Performance measured in
  - Number of database scans
  - Number of itemsets that must be counted
- Which are the interesting rules?

Applications:

- Basket data analysis: study of items that are purchased or grouped together in a single transaction or multiple, sequential transactions
- Cross-marketing, Catalog design, Loss-leader analysis, Clustering, Classification, Bioinformatics, Medical diagnosis, Web mining, and Scientific data analysis etc.
- Analysis of Earth science data: association pattern may reveal interesting connections among the ocean, land, and atmospheric processes.
- For a financial services company
  - Analysis of credit and debit card purchases.
  - Analysis of cheque payments made.
  - Analysis of services/products taken *e.g. a customer who has taken executive credit card is also likely to take personal loan.*
- For a telecommunication
  - Analysis of telephone calling patterns.
  - Analysis of value-add services taken together.

Approaches of Finding Association Rule

1. Frequent Itemset Generation: Generate all itemsets whose support ≥ minsup
2. Rule Generation: Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset.

2. Frequent Itemset Pattern & *Apriori* Principle

An even simpler concept: Frequent Itemset - Given a set of transactions D, find combination of items that occur frequently.

Example: 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

{Diaper} → {Beer},
{Milk, Bread} → {Diaper, Coke},
{Beer, Bread} → {Milk},

Fig. Market-Basket transactions

- Support count (σ) - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread, Diaper}) = 2
- Support - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- Frequent Itemset - An itemset whose support is greater than or equal to a ***minsup*** (minimum support) threshold
- Maximal Frequent Itemset: An itemset is maximal if none of its immediate supersets is frequent.

Why do we want to find frequent itemsets?

- Find all combinations of items that occur together
- They might be interesting (e.g., in placement of items in a store ☺)
- Frequent itemsets are only positive combinations (we do not report combinations that do not occur frequently together)
- Frequent itemsets aims at providing a summary for the data

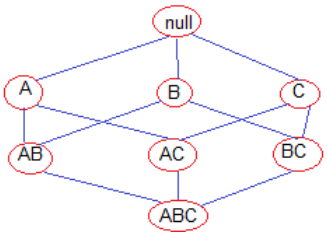


Fig. A

How many itemsets are there in Fig. A?

Ans: Given n items, No. of possible itemsets = $\mathbf{2^n}$. So, n = 3; itemsets = $2^3 = 8$

- No. of possible Association rules = $\mathbf{3^n - 2^{n+1} + 1} = 3^3 - 2^4 + 1 = 27 - 8 + 1 = 20$

Apriori principle:

– If an itemset is frequent, then all of its subsets must also be frequent

Apriori approach is two step approach

i. Frequent item generation and
ii. Rules generation

Apriori Principle:

- Supersets of non-frequent item are also non-frequent. Or, If an itemset is frequent, then all of its subset also be frequent.
- Apriori algorithm is an influential algorithm for mining frequent itemset.
- It use a level-wise search, k-itemsets are used to explore k+1 itemsets.
- At first, the set of frequent itemset is found and used to generate to frequent itemset at next level and so on.

Apriori Algorithm:

- Read the transaction database and get support for each itemset, compare the support with minimum support to generate frequent itemset at level 1.
- Use join to generate a set of candidate k-itmesets at next level.
- Generate frequent ietmsets at next level using minimum support.
- Repeat step 2 and 3 until no frequent itme sets can be generated.
- Generate rules form frequent itemsets from level 2 onwards using minimum confidence.

  **This approach has faster than Brute-Force approach but still has higher computational complexity.

[Example is given next document…]

3. FP-Growth, FP-Tree

FP-Growth (Frequent Pattern Growth)

- Does not generate candidates
- Typically just need to scan database twice

- Mining frequent itmesets without candidate generation.
- It is a divide and conquers strategy.
- It compresses the database representing frequent items into a frequent –pattern tree (FP-Tree), which retains the itemset association information.
- Divides the compressed database into a set of conditional databases, each associated with one frequent item or pattern fragment and then mines each such database separately.
- FP-Growth method transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix.
- It uses least frequent items as suffix .
- Advantage: Reduce search cost, has good selectivity, faster than apriori.
- Disadvantage: When the database is large, it is sometimes unrealistic to construct a man memory based FP-tree.

FP-Tree algorithm

- Create root node of tree, labeled with null.
- Scan the transactional database.
- The items in each transaction are processed in sorted order (Descending) and branch is created for each transaction.

FP-Tree algorithm

- Start from each frequent length pattern as an initial suffix pattern.
- Construct conditional pattern base. (Pattern base is a sub database which consists of the set of prefix paths in the FP-tree co-occurring with suffix pattern.
- Construct its FP-tree and perform mining recursively on such a tree

Why Is FP-Growth the Winner?

- Divide-and-conquer:
  - decompose both the mining task and DB according to the frequent patterns obtained so far
  - leads 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
  - basic ops—counting local freq items and building sub FP-tree, no pattern search and matching

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)

[…Example of FP-Tree in document…]

4. Handling Categorical Attributes

Categorical data

- Categorical data is a statistical data type consisting of categorical variables e.g. Gender= Male, Female or Temp.=Low, Mediaum, High, used for observed data whose value is one of a fixed number of nominal categories.
- More specifically, categorical data may derive from either or both of observations made of qualitative data e.g. Skin Colour: Black, Red, where the observations are summarized as counts or cross tabulations, or of quantitative data.
- Observations might be directly observed counts of events happening or they might count of values that occur within given intervals.
- Often, purely categorical data are summarized in the form of a contingency table.
- However, particularly when considering data analysis, it is common to use the term "categorical data" to apply to data sets that, while containing some categorical variables, may also contain non-categorical variables.

Potential Issues

- What if attribute has many possible values**:** Example: attribute country has more than 200 possible values. Many of the attribute values may have very low support.
  Potential solution: Aggregate the low-support attributes values.
- What if distribution of attribute values is highly skewed: 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 Categorical Attributes

- Transform categorical attribute into asymmetric binary variables **e.g. medical test (positive vs. negative)**. i.e If the outcomes of a binary variable are not equally important.
- Introduce a new "item" for each distinct attribute - value pair

5. Sequential, Subgraph, and Infrequent Patterns

Sequential Pattern

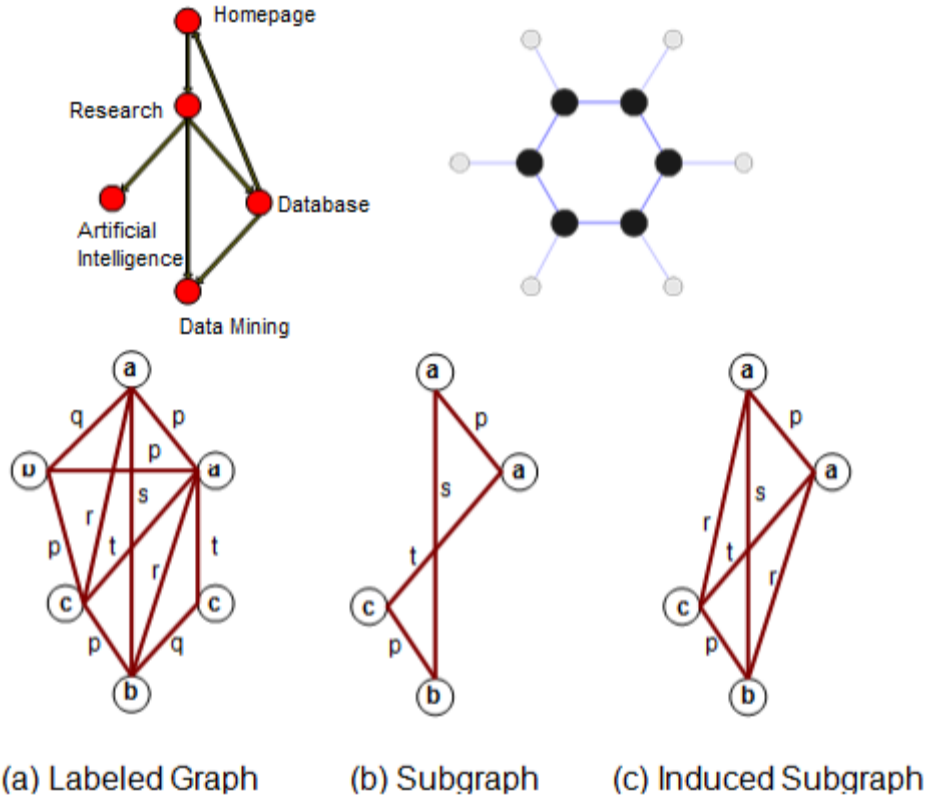
- Mining of frequently occurring ordered events or subsequences as patterns.
  Eg: **web sequence** < {Homepage} {Electronics} {Digital Cameras} {Canon Digital Camera} {Shopping Cart} {Order Confirmation} {Return to Shopping} >
  , **book checked out in library** <{Fellowship of the Ring} {The Two Towers} {Return of the King}> etc.
- Used mostly in marketing, customer analysis, prediction modeling.
- A sequence is an ordered list of elements(transactions)-set of items/events where an item can occur at most in an event of a sequence but can occur multiple times in different events of a sequence.
- Given a set of sequences, where each sequence consists of a list of events or elements and each event consists of set of items, given a minimum support threshold, sequential pattern mining finds all frequent subsequences.
- *Sequence with minimum support is called frequent sequence or sequential pattern.*
- A sequential pattern with length 'l' is called an l-pattern sequential pattern.
- Sequential pattern is computationally challenging because such mining may generate combinationally explosive number of intermediate subsequences.
- For efficient and scalable sequential pattern mining two common **approaches** are:
  - Mining the full set of sequential patterns
  - Mining only the set of closed sequential pattern
- A sequence database is a set of tuples with sequence_ID and sequences.

Sequence_ID	Sequence
1	{(a, (a,b,c), (a,c), (b,c)}
2	{(a,b,c), (a,d),e,(d,e)}
3	{(c,d), (a,d,e),e}
4	{ (e,f,),d,(a,b,c),f]

Sub-graph Patterns

- It finds characteristics sub-graphs within the network.
- It is a form of graph search.
- Given a labeled graph data set, D = {G1, G2, ….,Gn}, *a frequent graph has minimum support not less than minimum threshold support*.
- **Frequent sub-graph pattern** can be discovered by generating frequent substructures candidate and hence check the frequency of each candidate.
- Apriori methos and frequent –growth are tow common basic methods for finding frequent sub-graph
- Extend association rule mining to finding frequent subgraphs

- Useful for Web Mining, computational chemistry, bioinformatics, spatial data sets, etc
  Eg:: Chemical Structure, Geographical Nodes



Challenges

- Node may contain duplicate labels.
- How to define support and confidence?
- Additional constraints imposed by pattern structure
  - Support and confidence are not the only constraints
  - Assumption: frequent subgraphs must be connected

***Apriori-like approach:**

- Use frequent k-subgraphs to generate frequent (k+1) subgraphs

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, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining
- Frequent-pattern mining finds a set of patterns that occur frequently in a data set, where a pattern can be a set of items (called an itemset), a subsequence, or a substructure.
- A pattern is considered frequent if its count satisfies a minimum support. Scalable methods for mining frequent patterns have been extensively studied for static data sets.
- Challenges in mining data streams:
  - Many existing frequent-pattern mining algorithms require the system to scan the whole data set more than once, but this is unrealistic for infinite data streams.
  - A frequent itemset can become infrequent as well. The number of infrequent itemsets is exponential and so it is impossible to keep track of all of them.
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

Why Is Freq. Pattern Mining Important?

- Discloses an intrinsic and important property of data sets
- Forms the foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: associative classification
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing: iceberg cube and cube-gradient
  - Semantic data compression: fascicles
  - Broad applications

Basic Concepts: Frequent Patterns and Association Rules

- Itemset X = $\{x_1, \ldots, x_k\}$
- Find all the rules *X* → *Y* with minimum support and confidence
  - support, *s*, probability that a transaction contains X ∪ Y
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let sup_{min} *= 50%,* $conf_{min}$ *= 50%*

Freq. Pat.: {A:3, B:3, D:4, E:3, AD:3}

Association rules:

A → *D* (60%, 100%)

D → *A* (60%, 75%)

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F