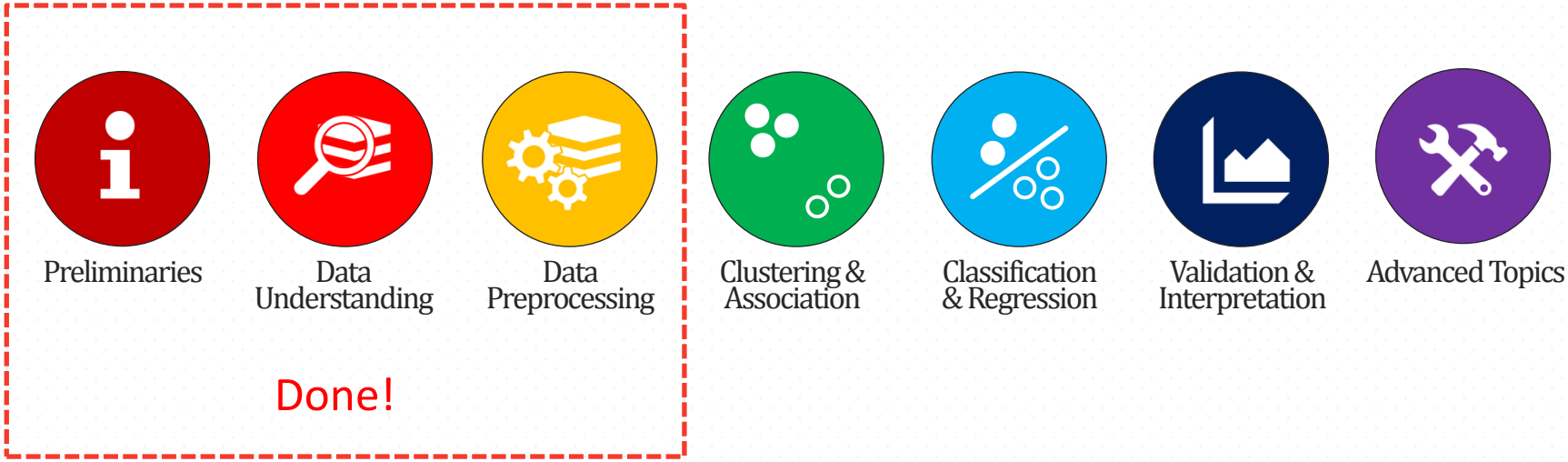
Course Topics



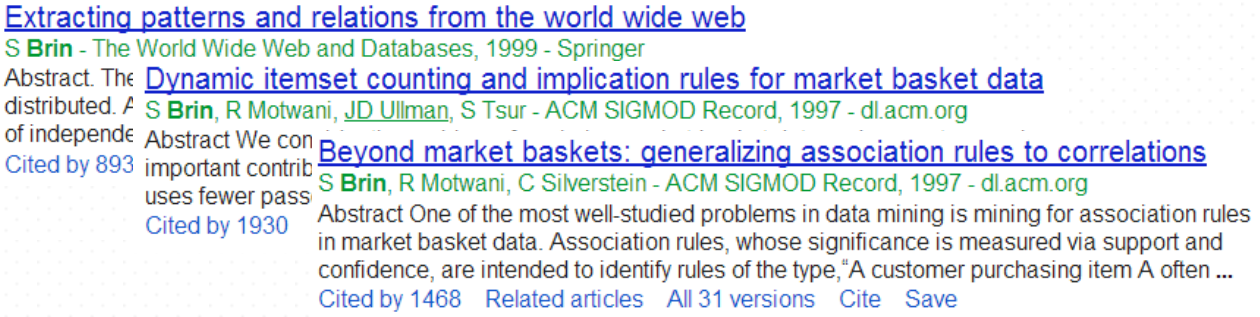
Clustering & Association

Association Rules

The process of discovering interesting relations between variables in large datasets.

Association Rules

- Introduced by Agrawal, Imielinski and Swami in 1993
- Vastly studied over the years, with countless improvements proposed by many researchers
- Before dropping out of Stanford's graduate program, Sergey Brin (Google's co-founder) published several papers on Association Rules



Association Rules

1. Market basket analysis
2. The Apriori algorithm
3. FP-Growth

Market Basket Analysis

TID	Beer	Milk	Bread	Diapers	Coke
T1	0	1	1	0	0
T2	1	0	0	0	0
T3	1	0	0	1	0
T4	0	1	1	0	0
T5	0	0	1	0	1

Market Basket Analysis

- Example:
  - 5% of transactions contain both these items
  - 30% of the transactions containing beer also contain diapers

- Beer ⇒ Diapers(0.05, 0.30)
  - 5% – **Support** of the rule
  - 30% – **Confidence** of the rule

Market Basket Analysis Applications

- Retail: *what sells with what*
- Marketing : *population segments, recommendations, etc.*
- Finance: *investment portfolios, "basket of stocks"*
- Biology: *genetics, microarrays, gene expressions*

Market Basket Analysis - Definitions

- $I = \{i_1, i_2, \dots, i_n\}$: a set of literals, called **items**
- **Transaction** T: a set of items such that $T \subseteq I$
- **Dataset** D: a set of transactions
- A transaction T contains X, a set of items in I if $X \subseteq T$
- An **association rule** is an implication of the form $X \Longrightarrow Y$, where $X, Y \subset I$
- The rule $X \Longrightarrow Y$ has **support** s in transaction set D if s% of transactions in D contain $X \cup Y$
- The rule $X \Longrightarrow Y$ holds in transaction set D with **confidence** c if c% of transaction in D that contain X also contain Y

Market Basket Analysis – Definitions (cont.)

- **Itemset**:
  - A collection of one or more items
    - Example: $\{Milk, Bread, Diaper\}$
  - **k-itemset**
    - An itemset containing k items
- Support count ($\boldsymbol{\sigma}$)
  - Frequency of an itemset
    - $\sigma(\{Milk, Bread, Diaper\}) = 2$
- Support ($\boldsymbol{s}$)
  - Fraction of transactions containing an itemset
  - $s(\{Milk, Bread, Diaper\}) = \frac{2}{5}$
- **Frequent Itemset**
  - One whose support is greater than or equal to a *minsup threshold*

TID	Items
T1	Bread, milk
T2	Bread, diaper, beer, eggs
T3	Milk, diaper, beer, coke
T4	Bread, milk, diaper, beer
T5	Bread, milk, diaper, coke

Association rules

- Ex.: $\{Milk, Diaper\} \Rightarrow \{Beer\}$

- Rule Evaluation Metrics
  - Support:

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

  - Confidence:

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

TID	Items
T1	Bread, milk
T2	Bread, diaper, beer, eggs
T3	Milk, diaper, beer, coke
T4	Bread, milk, diaper, beer
T5	Bread, milk, diaper, coke

Association rules

- Why use *support* and *confidence*?
  - Rules with low support may occur simply by chance
  - A low support rule is not interesting from a business perspective
  - Confidence measures the reliability of the inference made by a rule.
    - For $X \Rightarrow Y$, the higher the confidence, the more likely it is for Y to be present in transactions containing X

Association rules – Caution!

- Association rules results should be interpreted with caution
  - They do not imply causality, which requires extra knowledge of your data
  - Instead, they simply imply a strong co-occurrence relationship between items

Example

- Itemset $I = \{i_1, i_2, \dots, i_n\}$
- Find all rules $X \Rightarrow Y$ such that:
  - $min\_support = 50\%$
  - $min\_conf = 50\%$

TID	Items
T1	A, B, C
T2	A, C
T3	A, D
T4	B, E, F

$\boldsymbol{A \Rightarrow D}$ $(25\%, 33.3\%)$ ✘

$\boldsymbol{A \Rightarrow B}$ $(25\%, 33.3\%)$ ✘

$\boldsymbol{A \Rightarrow C}$ $(50\%, 66.7\%)$ ✔

$\boldsymbol{C \Rightarrow A}$ $(50\%, 100\%)$ ✔

Strong Association Rules

- Most times, we will be interested in finding strong association rules
  - $sup(R) \geq minsup$ and $conf(R) \geq minconf$
- The problem of finding weak association rules can also have interesting applications
  - "What two items are almost never purchased together?"

The Association Rule Mining Problem

Definition. Given a set of transactions T, find all the rules having support $\geq minsup$ and confidence $\geq minconf$, where $minsup$ and $minconf$ are the corresponding support and confidence thresholds.

The Association Rule Mining Problem

- The brute-force approach
  - Compute the support and confidence for every possible rule
- Prohibitively expensive
  - Given an itemset with n items, there exist $\boldsymbol{R = 3^n + 2^{n+1} + 1}$ rules
- Suppose we had 6 items:

$$R = 3^6 + 2^{6+1} + 1 = 602$$

Mining Association Rules

- Two-step approach:

1. Frequent Itemset Generation
   - Generate all item sets whose support $\geq minsup$
     - Note that the support of a rule $X \Rightarrow Y$ depends only on the support of $X \cup Y$
     - All rules below have the same support:

$$\{Beer, Diapers\} \Rightarrow \{Milk\}$$
$$\{Diapers, Milk\} \Rightarrow \{Beer\}$$
$$\{Milk\} \Rightarrow \{Beer, Diapers\}$$
$$\{Beer\} \Rightarrow \{Milk, Diapers\}$$
$$\ldots$$

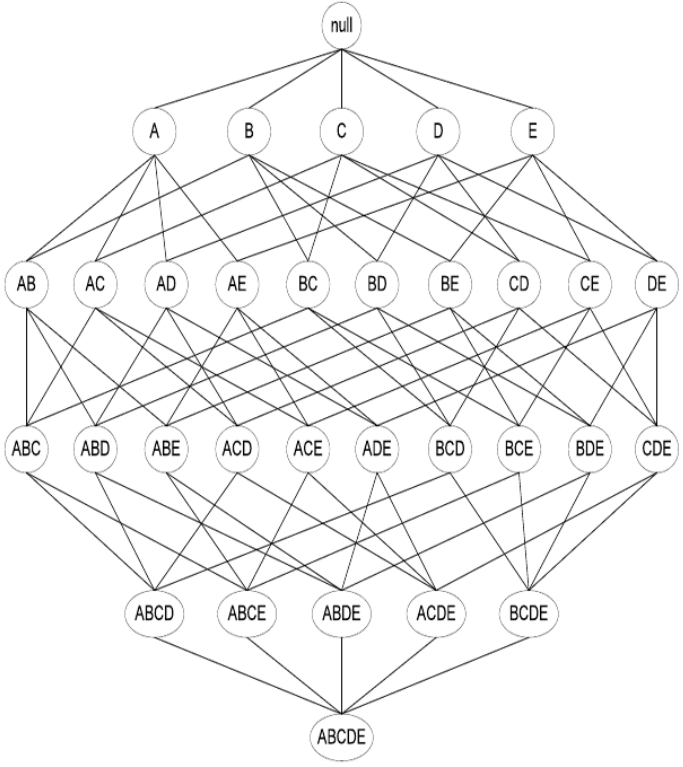
Mining Association Rules

- Two-step approach:

  1. Frequent Itemset Generation
     - Generate all item sets whose support $\geq$ *minsup*
  2. Rule Generation
     - Generate high confidence (strong) rules from each frequent itemset

- The computational complexity of (1) is higher.

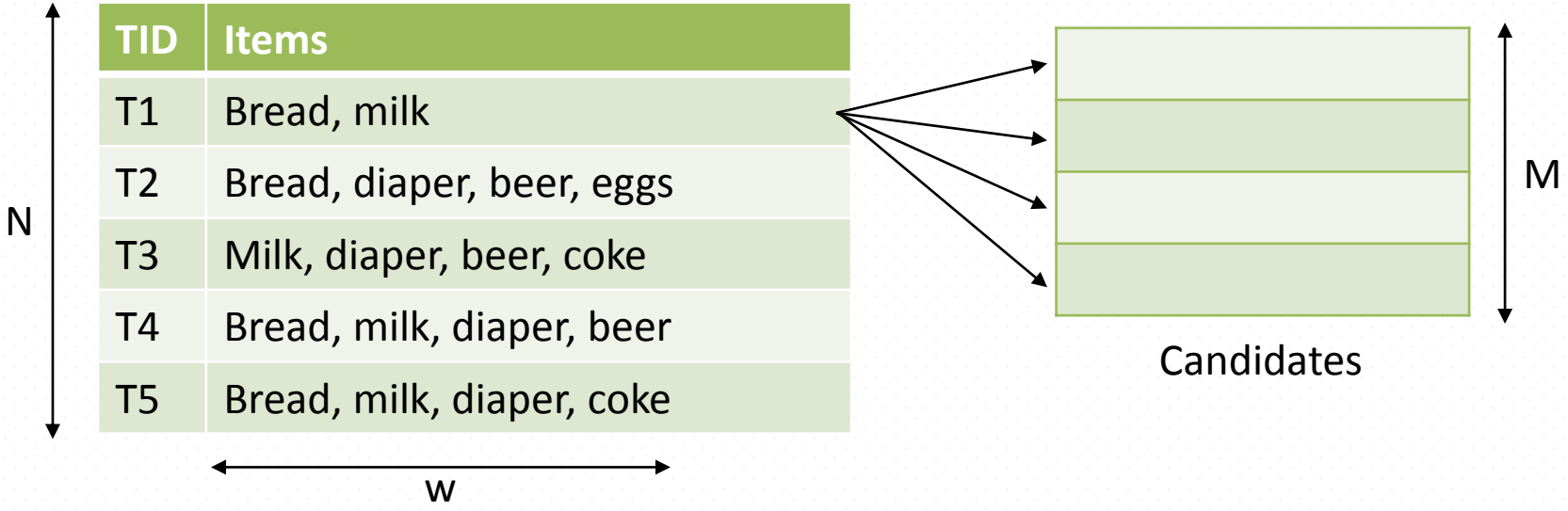
Frequent Itemset Generation



Itemset lattice

- For a set with n items:
  - 2^n-1 possible itemsets
- Each of these is called a *candidate* frequent itemset

Frequent Itemset Generation



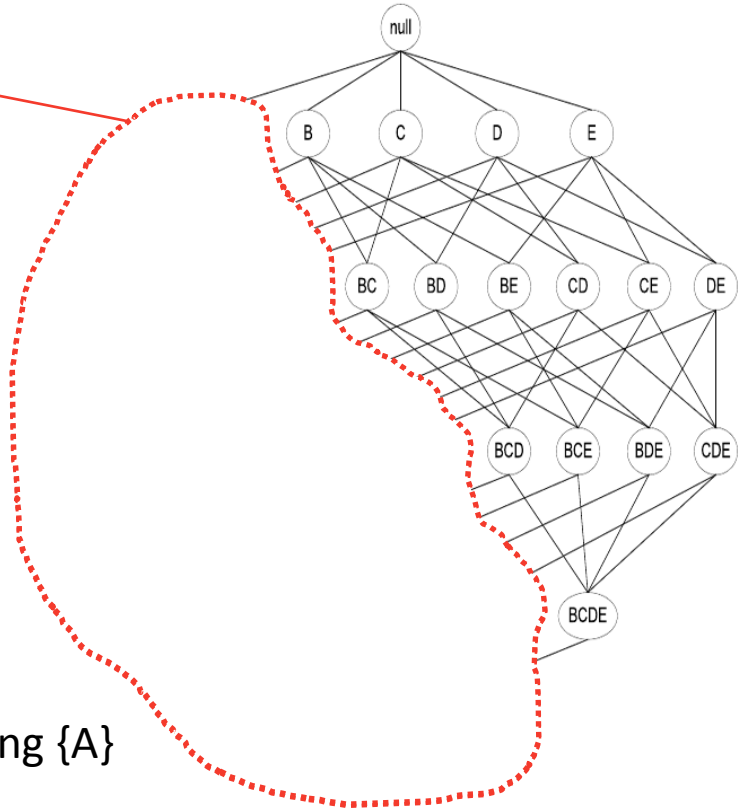
- Compare each candidate against every transaction
- Complexity: $O(NMw)$

Frequent Itemset Generation

- Several ways to reduce the computational complexity:
  - **Reduce the number of candidate itemsets** (M) by pruning the itemset lattice → **Apriori Algorithm**
  - **Reduce the number of comparisons** by using advanced data structures to store the candidate itemsets or to compress the dataset → **FP Growth**

Apriori Algorithm

If {A} is infrequent



Prune all itemsets containing {A}