BIG DATA ANALYTICS

Frequent Itemsets

Association Rule Discovery

Market-basket model:

- **Goal**: Identify items that are bought together
- **Approach**: Process the sales data to find dependencies among items
- Bread & milk: little interest
- Hot dogs & mustard $\Longrightarrow$ clever marketing:
  - A sale on hot dogs
  - Raise the price of mustard

Analysis of true market baskets

- Unexpected: diapers & beer
- A baby at home $\implies$ you are unlikely to be drinking at a bar
- Same marketing as for hot dogs and mustard

The market-basket model

- *Items* and *baskets*
    - Each basket $=$ a subset of items
    - The number of items in a basket is small $\ll$ the total number of items
    - The number of baskets is very large $\gg$ main memory
- Want to discover association rules:
    - People who bought $\{Diaper, Milk\}$ tend to buy $\{Beer\}$

Applications

- **Items** = words
- **baskets** = documents (e.g., Web pages, blogs, tweets)
- A basket (document) contains items (words) that are present in the document
    - Ignore the stop words
- **Find pairs of words that represent a joint concept**
    - For example: {Biden, White house}

Applications: plagiarism

- **Items** = documents
- **baskets** = sentences
- An item (document) is "in" a basket (sentence) if the sentence is in the document
- Pairs of items that appear together in several baskets
- even one or two sentences in common $\Longrightarrow$ plagiarism.

Application: Biomarkers

- Items = biomarkers and diseases
- Basket = the set of data about a patient
- A frequent itemset that consists of one disease and one or more biomarkers $\implies$ a test for the disease.

Frequent Itemsets

Goal: Find sets of items that appear together "frequently" in baskets

- $s =$ the support threshold
- $I =$ a set of items
- the support for $I =$ the number of baskets for which I is a subset
  - Often expressed as a fraction of the total number of baskets
- **I is frequent if its support is $\geq s$**

Toy example

1. {Cat, and, dog, bites }
2. { Yahoo, news, claims, a, cat, mated, with, a, dog, and, produced, viable,offspring }
3. { Cat, killer, likely, is, a, big, dog }
4. { Professional, free, advice, on, dog, training, puppy, training }
5. { Cat, and, kitten, training, and, behavior }
6. { Dog, &, Cat, provides, dog, training, in, Eugene, Oregon }
7. { “Dog, and, cat”, is, a, slang, term, used, by, police, officers, for, a, male–female, relationship }
8. { Shop, for, your, show, dog, grooming, and, pet, supplies }

Singleton

- "Dog": support is 7
- "Cat": support is 6
- "And": support is 5
- ...
- Threshold at $s = 3$:

Five frequent singleton itemsets: {dog}, {cat}, {and}, {a}, and {training}.

Doubletons

- A doubleton cannot be frequent unless both items in the set are frequent by themselves
- There are five frequent doubletons if $s = 3$: {dog, a}, {dog, and}, {dog, cat}, {cat, a} {cat, and}
- A single frequent triple: {dog, cat, a}

Association Rules

- **Association Rules**: If - then rules about the contents of baskets
- $I = \{a, b, c\} \rightarrow j$ means: "if a basket contains all of a, b, c then it is likely to contain j"
- **Goal: find significant/interesting rules**
- **Confidence** of an association rule is the probability of j given $I = \{a, b, c\}$:

$$conf(I \rightarrow j) = \frac{support(I \cup j)}{support(I)}$$

Interesting Association Rules

- Not all high-confidence rules are interesting
  - The rule $X \to milk$ may have high confidence for many itemsets X as milk is purchased very often
  - $\implies$ the confidence will be high
- **Lift** of an association rule $I \to j$:

$$Lift(I \to j) = \frac{support(I \cup j) \times support(E)}{support(I) \times support(j)}$$

  where E is the set of all items.
- Interesting rules: $lift > 1$ or $lift < 1$

Example: Confidence and Lift

$$S_1 = \{Bread, Coke, Milk\} \qquad S_2 = \{milk, pepsi, juice\}$$
$$S_3 = \{bread, milk\} \qquad S_4 = \{Coke, juice\}$$
$$S_5 = \{milk, pepsi, bread\} \qquad S_6 = \{milk, coke, bread, juice\}$$
$$S_7 = \{coke, bread, juice\} \qquad S_8 = \{bread, coke\}$$

- Association rule: $\{milk, bread\} \rightarrow coke$
  - Confidence$= 2/4 = 0.5$
  - Lift$= 0.5/(5/8) = 0.8$
  - Rule is not very interesting

Finding Association Rules

- Problem: Find all association rules with support $\geq s$ and confidence $\geq c$
  - Support of an association rule is the support of $\{I\}$
- Hard part: Finding the frequent itemsets!
- If $I \rightarrow j$ has high support and confidence, then both I and $\{I, j\}$ will be "frequent"!

$$conf(I \rightarrow j) = \frac{support(I \cup j)}{support(I)}$$

Mining Association Rules

- **Step 1:** Find all frequent itemsets I
- **Step 2:** Rule generation:
  - For every j in I generate a rule $I \setminus j \rightarrow j$
    - Since I is frequent, $I \setminus j$ at least as frequent
    - compute the rule confidence
  - Output the rules above the confidence threshold

Example

$$S_1 = \{Bread, Coke, Milk\} \qquad S_2 = \{milk, pepsi, juice\}$$
$$S_3 = \{bread, milk\} \qquad S_4 = \{Coke, juice\}$$
$$S_5 = \{milk, pepsi, bread\} \qquad S_6 = \{milk, coke, bread, juice\}$$
$$S_7 = \{coke, bread, juice\} \qquad S_8 = \{bread, coke\}$$

- Support threshold $s = 3$, confidence $c = 0.75$
- Frequent itemsets: $\{milk, bread\}$, $\{coke, bread\}$, $\{juice, coke\}$
- Generate rules:

$$\mathbf{milk \rightarrow bread : c = 4/5 > 0.75} \qquad \mathbf{coke \rightarrow bread : c = 4/5 > 0.}$$
$$bread \rightarrow milk : c = 4/6 < 0.75 \qquad bread \rightarrow coke : c = 4/6 < 0.$$
$$\mathbf{juice \rightarrow coke : c = 3/4 = 0.75} \qquad coke \rightarrow juice : c = 3/5 < 0.$$

Reducing the number of outputs

- Rules must be acted upon
- Reduce the number of rules:
  - adjust the support threshold

Outline

Representation of Market-Basket Data

- Items = integers
- Data is kept in flat files
  - Stored basket-by-basket
  - Baskets are small but we have many baskets and many items
  - E.g. Items are positive integers, and boundaries between baskets are -1 or

$$\{123, 6, 5067\}, \{12, 67, 50, 796\}, \ldots$$

Main Memory Bottleneck

- Main-memory is the critical resource:
  - Reading baskets, we need to count occurrences of pairs of items
  - The number of different things we can count is limited by main memory
  - Example:
    - we have n items and we need count all pairs
    - $\binom{n}{2} \sim n^2/2$ pairs of integers
    - Integers take 4 bytes $\implies 2n^2$ bytes
    - machine has 2 gigabytes$= 2^{31}$ bytes $\implies n < 33,000$.

Naive approach

- Read file once, counting in main memory the occurrences of each pair:
  - From each basket generate its pairs
  - Counting in the main memory the occurences of each pair?

Counting pairs in memory

- Two approaches:
  - Approach 1: Count all pairs using a matrix
  - Approach 2: Keep a table of triples $[i, j, c]$ ="the count of the pair of items (i, j) is c
    - If integers and item ids are 4 bytes, we need approximately 12 bytes for pairs with count > 0
  - Approach 1 only requires 4 bytes per pair
  - Approach 2 uses 12 bytes per pair (but only for pairs with count > 0)

Problem: we have so many items that the pairs do not fit into the memory.

Can we do better?

Outline

Computation Model

- Association-rule algorithms read the data in passes
- The cost = **number of passes** an algorithm makes over the data

A-Priori Algorithm (1)

- A two-pass approach
- Limits the need for main memory
- Key idea: **monotonicity**
  - If a set of items I appears at least s times, so does every subset J of I
- **If item i does not appear in s baskets, then no pair including i can appear in s baskets.**

A-Priori Algorithm (2)

- **Pass 1**: Read baskets and count in main memory the occurrences of each individual item
  - Requires memory proportional to #items
- Items that appear $\geq s$ times: the frequent items
- **Pass 2**: Read baskets again and count in main memory only those pairs where both elements are frequent
  - Requires memory proportional to square of frequent items
  - $+$ a list of the frequent items

Frequent triples, etc

- For each k, we construct two sets of k-tuples :
  - C_k =candidate k-tuples = those that might be frequent sets (support $> s$) based on information from the pass for k–1
  - L_k =the set of truly frequent k-tuples
  - E.g., C_3 = the set of triples, any two of which is a pair in L_2 .

A-priori for all frequent itemsets

- One pass for each k (itemset size)
- Needs room in main memory to count each candidate k–tuple
- For typical market-basket data and reasonable support (e.g., 1%), $k = 2$ requires the most memory
- Many possible extensions:
  - Association rules with intervals:
    - For example: Men over 65 have 2 cars
  - Association rules when items are in a taxonomy
    - Bread, Butter $\rightarrow$ Fruit, Jam
    - BakedGoods, MilkProduct $\rightarrow$ PreservedGoods

References

- **J. Leskovec, A. Rajaraman and J. D. Ullman** *Mining of Massive Datasets* **(2014), Chapter 6**
- R. Agrawal, T. Imielinski, and A. Swami, "Mining associations between sets of items in massive databases," Proc. ACM SIGMOD Intl. Conf. on Management of Data, pp. 207–216, 1993.
- R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," Intl. Conf. on Very Large Databases, pp. 487–499, 1994.