Lecture 13 - Market Basket Analysis

Summary

Modern supermarkets and shops capture large amounts of information about what their customers buy in a transaction. **Market Basket Analysis**, or more generally **Association Rule Mining** has become a popular approach to mining these transactions in order to identify common trends in terms of products that are often brought together. Once these association rues have been identified they can be used in many ways, although often with the goal of increasing or optimising profit.

Key points

Association rules and their applications

- Association Rules are "if-then" implications
    - i.e. X=>Y :: X implies Y :: if X then Y
- Market basket analysis attempts to find the association rules from a database of transactions
    - Each transaction records what items were brought by a single customer at a single point in time
    - The association rules will tell us which items are commonly brought together
    - Why do we want to do this?
        - *Identify* who customers are (in terms of broad demographics)
        - *Understand* why they make certain purchases
        - *Gain insight* about products:
            - Fast and slow movers
            - Products which are purchased together
            - Products which might benefit from promotion
        - *Take action* to increase profit/sales/turnover/etc:
            - Optimise store layouts
            - Which products to put on specials, promote, coupons…
        - *Find out* what customers **do not** purchase
        - *Determine* the key drivers of purchases
    - Other application areas:
        - Telecommunication
            - each customer is a transaction containing the set of phone calls
        - Credit Cards/ Banking Services
            - each card/account is a transaction containing the set of customer's payments
        - Medical Treatments
            - each patient is represented as a transaction containing the ordered set of diseases
        - Basketball-Game Analysis
            - each game is represented as a transaction containing the ordered set of ball passes

Mining Association Rules

- Formal Definitions
  - I={i_1, i_2, ..., i_n}: a set of all the items
  - Transaction T: a set of items such that T ⊆ I
  - Transaction Database D: a set of transactions
  - A transaction T ⊆ I contains a set X ⊆ I of some items, if X ⊆ T
  - An **Association Rule**: is an implication of the form X ⇒ Y, where X, Y ⊆ I
  - A set of items is referred as an itemset.
    - An itemset that contains k items is a k-itemset.
  - The **support** s of an itemset X is the percentage of transactions in the transaction database D that contain X.
  - The **support** of the rule X ⇒ Y in the transaction database D is the support of the items set X ∪ Y in D.
  - The **confidence** of the rule X ⇒ Y in the transaction database D is the ratio of the number of transactions in D that contain X ∪ Y to the number of transactions that contain X in D.
  - Example:
    - If a supermarket database has 100,000 point-of-sale transactions, out of which 2,000 include both items A and B and 800 of these include item C, the association rule "If A and B are purchased then C is purchased on the same trip" has a support of 800/100000 = 0.8% and a confidence of 800/2000 = 40%.
  - One way to think of support is that it is the probability that a randomly selected transaction from the database will contain all items in the antecedent and the consequent, whereas the confidence is the conditional probability that a randomly selected transaction will include all the items in the consequent given that the transaction includes all the items in the antecedent.
- Association Rule Problem
  - Find the association rules in a database of transactions
  - We're not interested in all association rules
    - only those with high support and high confidence
  - Can break the problem into two parts
    1. Find all sets of items that have minimum support (**frequent itemsets**)
    2. Use the frequent itemsets to generate the desired rules
  - The **Apriori algorithm** is an efficient way to do this
    - Makes use of the *frequent itemset property*:
      - **Any subset of a frequent itemset is frequent.**
    - and its contrapositive:
      - **If an itemset is not frequent, none of its supersets are frequent.**
- The **Apriori algorithm**:

```
Let $L_k$ be the set of frequent itemsets of size k (with min support)
Let $C_k$ be the set of candidate itemset of size k (potentially frequent itemsets)

$L_1$ = {frequent items};
for (k=1; $L_k$ != ∅; k++) do
   $C_{k+1}$ = candidates generated from $L_k$;
   for each transaction t in database do:
      increment the count of all candidates in $C_{k+1}$ that are contained in t;
   $L_{k+1}$ = candidates in $C_{k+1}$ with min_support;
return $\cup_k L_k$;
```

- Candidates can be generated with:

```
Input: $L_{i-1}$: set of frequent itemsets of size i-1
Output: $C_i$: set of candidate itemsets of size i
$C_i$ = empty set;
for each itemset J in $L_{i-1}$ do
   for each itemset K in $L_{i-1}$ s.t. K≠J do
      if i-2 of the elements in J and K are equal then
         if all subsets of {K ∪ J} are in $L_{i-1}$ then
            Ci = Ci ∪ {K ∪ J}
return Ci;
```

- Rules can be discovered with:

```
for each frequent itemset I do
   for each subset C of I do
      if (support(I) / support(I - C) >= minconf) then
         output the rule (I - C) ⇒ C,
         with confidence = support(I) / support (I - C)
         and support = support(I)
```

- Apriori algorithm considerations:
  - Advantages:
    - Uses large itemset property
    - Easily parallelised
    - Easy to implement
  - Disadvantages:
    - Assumes transaction database is memory resident
    - Requires many database scans

Improved rule measures

- **Confidence** measure has problems with independent itemsets that will lead to high confidences when it doesn't really make sense
- **Lift** indicates the departure from independence of X and Y
  - But is symmetric & doesn't consider direction of implications
- **Conviction** indicates the departure from independence of X and Y taking into account the implication direction.

Further applications

- Find **linked concepts** in text documents
  - Transactions = documents
  - Items = words in those documents
- Detecting **plagiarism** in text documents
  - Transactions = sentences
  - Items = documents containing those sentences
- Finding web pages about a **common topic**
  - Transactions = web pages
  - Items = pages linked to from those pages
- Finding **mirrors or similar pages**
  - Transactions = web pages

    - Items = pages linking to that page

Further Reading

- Chapter 6 of “Introduction to Data Mining” by Tan et al is very detailed: https://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf (https://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf)
- Wikipedia’s page on mining association rules is good: https://en.wikipedia.org/wiki/Association_rule_learning (https://en.wikipedia.org/wiki/Association_rule_learning)