Lecture notes

Mining Association Rules

The goal of association rule finding is to extract correlation relationships in the large datasets of items. Ideally, these relationships will be causative. Many businesses are interested in extracting interesting relationships in order to raise profit. Scientists, on the other hand, are interested in discovering previously unknown patterns in their field of research.

An illustrative example of association rule mining is so-called *market basket analysis*. This process is based on transactional data, which are huge amounts of records of individual purchases. For example, department stores such as Wal-Mart keep track of all transactions in a given period of time. One transaction could be \( t_1 = \{ \text{italian bread, 1\% milk, energizer batteries} \} \). It can contain an arbitrary (technically, it is limited to what’s in the store) number of items. All items purchased belong to a set of items \( \Omega \). In the computer science jargon, such a set is typically called a *universe*. Another notational detail is that transaction \( t_1 \) can also be called an *itemset*, because it contains a set of purchased items.

What could be a rule and what kind of rules are we looking for?

Let’s analyze a hypothetical *AllElectronics* retailer. It is likely that this business contains items such as *batteries*, *computer*, *mp3 player*, etc. An association rule could then be

\[
\text{computer} \Rightarrow \text{financial\_management\_software}
\]

which means that a purchase of a computer implies a purchase of financial management software. Naturally, this rule may not hold for all customers and every single purchase. Thus, we are going to associate two numbers with every such rule. These two numbers are called *support* and *confidence*. They can be considered as measures the interestingness of the rules.
Notation and Basic Concepts

Let $\Omega = \{i_1, i_2, \ldots, i_m\}$ be a universe of items. Also, let $T = \{t_1, t_2, \ldots, t_n\}$ be a set of all transactions collected over a given period of time. To simplify a problem, we will assume that every item $i$ can be purchased only once in any given transaction $t$. Thus, $t \subseteq \Omega$ (“$t$ is a subset of omega”). In reality, each transaction $t$ is assigned a number, for example a transaction id (TID).

Let now $A$ be a set of items (or an itemset). A transaction $t$ is said to contain $A$ if and only if $A \subseteq t$. Now, mathematically, an association rule will be an implication of the form

$$A \Rightarrow B$$

where both $A$ and $B$ are subsets of $\Omega$ and $A \cap B = \emptyset$ (“the intersection of sets A and B is an empty set”).

What is support?

Support (sometimes called frequency) is simply a probability that a randomly chosen transaction $t$ contains both itemsets $A$ and $B$. Mathematically,

$$\text{support}(A \Rightarrow B) = P(A \subset t \land B \subset t) = \frac{\# \text{ of transactions containing both } A \text{ and } B}{\text{total } \# \text{ of transactions}}$$

we will use a simplified notation that

$$\text{support}(A \Rightarrow B) = P(A \land B)$$

What is confidence?

Confidence (sometimes called accuracy) is simply a probability that an itemset $B$ is purchased in a randomly chosen transaction $t$ given that the itemset $A$ is purchased. Mathematically,

$$\text{confidence}(A \Rightarrow B) = P(B \subset t \mid A \subset t) = \frac{\# \text{ of transactions containing both } A \text{ and } B}{\text{total } \# \text{ of transactions containing } A}$$

we will use a simplified notation that

$$\text{confidence}(A \Rightarrow B) = P(B \mid A)$$

Thus, a valid rule could be

$$\text{computer} \Rightarrow \text{financial_management_software}$$

$$[\text{support} = 2\%, \text{confidence} = 60\%]$$

Recall, our goal is to find interesting rules. Hence, we would like to detect those with high support and high confidence. Typically, we will select appropriate thresholds for both measures and then look for all subsets that fulfill given support and confidence criteria.
**Definitions:** a set of $k$ items is called a *$k$-itemset*. For example, \{*bread*, *skim milk*, *pringles*\} is a 3-itemset. The occurrence frequency of an itemset (or simple the count) is the number of transactions that contain the itemset. For example, the database $T$ may contain 1542 transactions that contain itemset $A$. An itemset whose count (or probability) is greater than some pre-specified threshold is called a frequent itemset. A set of all frequent $k$-itemsets will be denoted as $L_k$.

**How are we going to find interesting rules from the database $T$?** It will be a two-step process:

1. **Find all frequent itemsets** (each of these itemsets will occur at least as frequently as pre-specified by the minimum support threshold)
2. **Generate strong association rules from the frequent itemsets** (these rules will satisfy both minimum support threshold and minimum confidence threshold)

**Classification of Association Rules**

Based on the types of values handled in the rule we can have Boolean or quantitative association rules. A quantitative association rule would look like

$$age(X, \text{“30..39”}) \land income(X, \text{“42K..48K”}) \Rightarrow buys(X, \text{plasma TV})$$

where $X$ is an arbitrary customer.

Based on the levels of abstractions involved in the rule set we can have

$$age(X, \text{“30..39”}) \Rightarrow buys(X, \text{plasma TV})$$

or

$$age(X, \text{“30..39”}) \Rightarrow buys(X, \text{TV})$$

more general

**Apriori Algorithm**

In the following transaction database $D$ find all frequent itemsets. The minimum support count is 2.

<table>
<thead>
<tr>
<th>TID</th>
<th>List of item IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>11, 12, 15</td>
</tr>
<tr>
<td>T200</td>
<td>12, 14</td>
</tr>
<tr>
<td>T300</td>
<td>12, 13</td>
</tr>
<tr>
<td>T400</td>
<td>11, 12, 14</td>
</tr>
<tr>
<td>T500</td>
<td>11, 13</td>
</tr>
<tr>
<td>T600</td>
<td>12, 13</td>
</tr>
<tr>
<td>T700</td>
<td>11, 13</td>
</tr>
<tr>
<td>T800</td>
<td>11, 12, 13, 15</td>
</tr>
<tr>
<td>T900</td>
<td>11, 12, 13</td>
</tr>
</tbody>
</table>
The steps of the algorithm are shown below:

Apriori algorithm employs a level-wise search for frequent itemsets. In particular, frequent $k$-itemsets are used to find frequent $(k+1)$-itemsets. This is all based on the following property:

**all nonempty subsets of a frequent itemset must also be frequent**

This means that, in order to find $L_{k+1}$, we should only be looking at $L_k$. There are two steps in this process, the **join** step and the **prune** step.

1. **The join step.** To find $L_k$, a set of candidate $k$-itemsets is generated by joining $L_{k-1}$ with itself. This set of candidates is denoted $C_k$. Let $l_1$ and $l_2$ be itemsets in $L_{k-1}$. Then, $l_1[j]$ refers to the $j$-th element in the itemset $l_1$ (Apriori assumes that the elements are sorted in some lexicographical order). The join operation $\text{join}(L_{k-1}, L_{k-1})$ is performed only when $l_1[1] = l_2[1] \wedge l_1[2] = l_2[2] \wedge \ldots l_1[k-2] = l_2[k-2] \wedge l_1[k-1] \neq l_2[k-1]$ (one should make sure no duplicate candidates are created; to ensure this $<$ is used instead of $\neq$).

2. **The prune step.** $C_k$ is a superset of $L_k$, because the members created in the join step may not be frequent. We can scan the database $D$ to determine count of each itemset in $C_k$. However, this would be costly since $C_k$ can be huge. Thus, here we use the property that any subset of a frequent itemset must also be frequent. Thus, if any subset of an itemset from $C_k$ is not present in $L_{k-1}$ it should be pruned without a database scan.
How can we generate association rules from frequent itemsets?

Once we find all frequent itemsets, we can easily calculate confidence for any rule. We simply use

\[ \text{confidence}(A \Rightarrow B) = P(B \mid A) = \frac{\text{count}(A \text{ and } B)}{\text{count}(A)} \]

Thus, for every frequent itemset, we generate all nonempty proper subsets. Then, we simply run each subset (and its complement) through the formula above. Those rules that create confidence above the pre-specified threshold are outputted as association rules. A proper subset is a subset that is strictly smaller than the original set.

In the example below an itemset \( l = \{I_1, I_2, I_5\} \) is frequent. All nonempty proper subsets are \( \{I_1, I_2\} \), \( \{I_1, I_5\} \), \( \{I_2, I_5\} \), \( \{I_1\} \), \( \{I_2\} \), \( \{I_5\} \). The confidence of all the candidate association rules are now

- \( I_1 \land I_2 \Rightarrow I_5 \) confidence = 2/4 = 50%
- \( I_1 \land I_5 \Rightarrow I_2 \) confidence = 2/2 = 100%
- \( I_2 \land I_5 \Rightarrow I_1 \) confidence = 2/2 = 100%
- \( I_1 \Rightarrow I_2 \land I_5 \) confidence = 2/6 = 33%
- \( I_2 \Rightarrow I_1 \land I_5 \) confidence = 2/7 = 29%
- \( I_5 \Rightarrow I_1 \land I_2 \) confidence = 2/2 = 100%

If a minimum confidence rule was 75%, only the second, third and sixth rule would be considered strong and thus outputted.

What could we mine for association rules and what is our benefit?

- transaction data (to organize store layout, to create catalogs, to design discount patterns etc)
- web log data (to track behavior of users, to improve add positioning, etc.)
- bioinformatics data (to find functional/structural patterns for a set of proteins etc.)

-----

Source: Data mining: concepts and techniques by Jiawei Han and Micheline Kamber, Academic Press, 2001.
Principles of data mining by David Hand, Heikki Mannila and Padhraic Smyth, MIT Press, 2001