Mining Infrequent Patterns

Johan Bjarnle (JohbJ551)
Peter Zhu (Petzh912)

Linköping University, 2009
TNM033 Data Mining
1 Introduction

Often when considering data mining, the focus is on frequent patterns. Although the majority of the most interesting patterns will lie within the frequent ones, there are important patterns that will be ignored with this approach. These are called infrequent patterns.

Take for example the sale of VHS:s and DVD:s. There will be low occurrences of people buying both of them. In terms of data mining, the itemset \{VHS, DVD\} will be infrequent and therefore ignored. However, people that buys DVD:s does not tend to buy VHS:s and vice versa. These items will be competing and an interesting pattern is found.

Data mining comes with a lot of concepts and terms. To be able to follow and understand the concepts and definitions in this report, one should be familiar with two basic terms, support and confidence. Support is defined as how frequent an itemset is among all the records:

\[
\text{Support: } s = \frac{\sigma(X)}{|R|}
\]

where \(\sigma(X)\) is the number of occurrences of itemset \(X\) and \(|R|\) is the total number of records.

Confidence is defined as the probability function \(P(X|Y)\) which measures how often the itemset \(X\) appears in records that contains itemset \(Y\).

\[
\text{Confidence: } c = \frac{\sigma(X)}{\sigma(Y)}
\]

where \(\sigma(X)\) and \(\sigma(Y)\) are defined as the number of occurrences of the itemsets \(X\) respective \(Y\).
Infrequent patterns are described as itemsets or rules that don’t pass the \textit{minsup} criteria. The \textit{minsup} threshold is manually set and separates frequent- and infrequent patterns. If the \textit{minsup} is set too high then itemsets involving interesting rare items could be missed. For example, in a shopping mall, itemsets containing expensive colognes might be ignored, causing eventual interesting patterns to be missed.

\textbf{DEFINITION 1:} An infrequent pattern is an itemset or a rule whose support is less than the \textit{minsup} threshold.

Usually most of the infrequent patterns are not interesting, and they also account for a large number of itemsets. Good algorithms to filter out interesting patterns which at the same time are computationally efficient are required. The definitions in this report were extracted from \textit{Introduction to Data Mining} [1].

2 Techniques

2.1 Negative Patterns

Given an itemset \(I = \{i_1, i_2, ..., i_k\}\), \(\bar{i}_k\) denotes the absence of item \(i_k\) in the itemset. For example, \(\text{\textbar{milk}}\) corresponds to the absence of \textit{milk}. With the negative item defined, a \textit{negative itemset} can be described as an itemset with both positive and negative items, passing the \textit{minsup} threshold.

\textbf{DEFINITION 2:} A negative itemset \(X\) contains positive items \(A\) and negative items \(\overline{A}\), where \(|\overline{A}| \geq 1\) and \(s(X) \geq \text{minsup}\).

From these negative itemsets, negative association rules can be defined.

\textbf{DEFINITION 3:} A negative association rules has to have following three properties; the rules is extracted from a negative itemset, the support of the rule is greater than or equal to \textit{minsup}, and the confidence of the rule is greater than or equal to \textit{minconf}.

Such rules will be denoted as \textit{coffee} \(\rightarrow\) \textit{tea}, which means that people who buys coffee tends to not buy tea. Negative itemsets and negative association rules will be referred to as negative patterns in this report.

2.2 Negative Correlated Patterns

The basic idea of a negative correlated pattern is that the actual support should be less than the expected support, as stated in the definitions below:

\textbf{DEFINITION 4:} An itemset \(X\) is negatively correlated if

\[s(X) < \prod_{j=1}^{k} s(x_j) = s(x_1) \times s(x_2) \times \cdots \times s(x_k),\]

where \(s(x_j)\) is the support of an item \(x_j\).
The right hand side, the expected support, represents an estimate of the probability that all the items in $X$ are statistically independent. The smaller the support is compared to the expected support, the more negatively correlated the pattern is.

**Definition 5:** An association rule $X \rightarrow Y$ is negatively correlated if

$$s(X \cup Y) < s(X)s(Y),$$

where $X$ and $Y$ are disjoint itemsets, for example $X \cup Y = \emptyset$. This definition, however, only provides a partial condition for negative correlation between items in $X$ and $Y$. The full condition is stated below:

$$s(X \cup Y) < \prod_i s(x_i) \prod_j s(y_j),$$

where $x_i \in X$ and $y_j \in Y$. The negatively correlated itemsets and association rules will be referred to as negatively correlated patterns.

These patterns are closely related. Infrequent patterns and negatively correlated patterns refer to itemsets that only contain positive items. This is shown in the figure below.

![Figure 1. Comparisons between infrequent patterns, negative patterns, and negative correlated patterns.](image_url)
2.3 Mining Negative Patterns

This technique utilizes a binary table, where every itemset is translated into binary values. If one considers the earlier described negative patterns, the records can be binarized by augmenting it with negative items. The tables below demonstrate how this works.

<table>
<thead>
<tr>
<th>RecordID</th>
<th>Items</th>
<th>RecordID</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{A,B}</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>{B,C}</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>{A}</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. Original records table.

<table>
<thead>
<tr>
<th>RecordID</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2. The records translated to binary values.

This can then be combined with an algorithm such as Apriori [2], to derive all the negative itemsets. This technique is only usable when the itemsets covers few items, because of the following reasons:

1. The number of items is doubled, since every item also has a corresponding negative value. When exploring an itemset, a larger set than the usual $2^d$ has to be evaluated, where the $d$ stands for number of items in the original data set.
2. Pruning based on support is no longer effective, since either $x$ or $\bar{x}$ will have a value greater than or equal 50%.
3. The width of each record will increase when adding the corresponding negative part for each item. If the total number of items is $d$, then a record originally is much smaller than $d$. Consider a food market, then each record is almost guaranteed to not include all items. However, using this technique, the width of each record will be $d$.

Many of the existing algorithms can’t handle these kinds of datasets due to the large data size, and will eventually break down.

To decrease the computational cost; the support computation for a negative itemset can be done by calculating the support for the positive parts only.

**EQUATION 1:** $s(X \cup Y) = s(X) + \sum_{i=1}^{n} \sum_{Z \in Y \setminus |Z| = i} \{-1\}^i \times s(X \cup Z)$

2.4 Mining Negative Correlated Patterns

The second technique is based on the expected support. An infrequent pattern is considered to be interesting if the actual support is less than the expected support. Two approaches to calculate the expected support will be discussed. The first is based upon a concept hierarchy and the second uses a neighborhood-based approach.

2.4.1 Concept Hierarchy

The usual support calculation (see support) is not a sufficient measure to filter away uninteresting infrequent patterns. Consider an itemset e.g. \{TV, coke\}, whose items are frequent. The itemset is considered to be infrequent due to low support and maybe negatively correlated, but it is uninteresting
due to the fact that the items belong to different categories. For an expert this may seem to be obvious, however for a computer to ignore these kinds of infrequent patterns, a good measure has to be used.

**Figure 2.** Concept hierarchy

This approach is based on the assumption that products within same product family have similar types of interaction with other items. For example, if \{Cookies, Pork\} (see figure 2) is a frequent itemset then their children will have similar relations. The expected support for \{Oatmeal, Bacon\} can then be calculated with the following equation:

**Example 1:**
\[
\varepsilon(s(Oatmeal, Bacon)) = s(Cookies, Pork) \times \frac{s(Oatmeal)}{s(Cookies)} \times \frac{s(Bacon)}{s(Pork)}
\]

If the actual support for \{Oatmeal, Bacon\} is less than the expected support calculated above then we have found an interesting infrequent pattern.

The algorithm for this case where we only consider direct children of the frequent itemset (in example 1 \{Cookies, Pork\}) can be generalized into the following:

**Equation 2:**
\[
\varepsilon(s(p, q, \ldots, t)) = s(p \cup q \cup \ldots \cup t) \times \frac{s(p) \times s(q) \times \ldots \times s(t)}{s(p) \times s(q) \times \ldots \times s(t)}
\]

where \{\hat{p}, \hat{q}, \ldots, \hat{t}\} is a frequent itemset and \(\hat{p}\) denotes the parent of \(p\).

In the second case we can consider one of the items in the frequent itemset and one child from the other item, for example \{Cookies, Bacon\}. The general equation for this:

**Equation 3:**
\[
\varepsilon(s(p, q, r, \ldots, t)) = s(p \cup q \cup \hat{r} \cup \ldots \cup \hat{t}) \times \frac{s(r) \times \ldots \times s(t)}{s(p) \times \ldots \times s(t)}
\]
The third case considers one of the items in the frequent itemset and one sibling from the other item, for example \{Cookies, Chicken\}. The general equation for this:

\[
\text{EQUATION 4: } \epsilon(s(p, q, r, ..., t)) = s(p \cup q \cup r' \cup ... \cup t') \times \frac{s(r) \times ... \times s(t)}{s(r') \times ... \times s(t')}
\]

where \(t'\) denotes the sibling of \(t\).

In our example we only take into account that the frequent itemset contains two items to make it simpler to understand, however this is not the case when mining data. These three cases are the only interesting when mining for negative itemsets, for example we do not consider the itemset where we only have the siblings.

2.4.2 Indirect Association

In the previous section we described how to calculate the expected support using concept hierarchy. This technique will instead use indirect association to get the expected support. Indirect association means looking through a set of items in common, the mediator set.

\[\text{DEFINITION 6: A pair of items } a, b \text{ is indirectly associated through a mediator set } Y \text{ if the following conditions hold:}\]

1. \(s(\{a, b\}) < t_s\) (Itempair support condition).
2. \(\exists Y \neq 0\) such that:
   
   I. \(s(\{a\} \cup Y) \geq t_f\) and \(s(\{b\} \cup Y) \geq t_f\) (Mediator support condition).
   II. \(d((a), Y) \geq t_d, d((b), Y) \geq t_d\), where \(d(X, Z)\) is an objective measure of the association between \(X\) and \(Z\) (Mediator dependence condition).
3 Applications

Infrequent patterns can be used in many applications.

- In text mining, indirect associations can be used to find synonyms, antonym or words that are used in different contexts. For example, the word data might be indirectly associated with the word gold, using the mediator mining.

- In the market basket domain, indirect associations can be used to find competing items, such as desktop computers and laptops, which states that people whom buys desktop computers won’t buy laptops.

- Infrequent patterns can be used to detect errors. For example, if \(\{\text{Fire} = \text{Yes}\}\) is frequent, but \(\{\text{Fire} = \text{Yes}, \text{Alarm} = \text{On}\}\) is infrequent, then the alarm system probably is faulting.

When evaluating Weka [3], we could not find any signs of an infrequent pattern classifier.

4 Conclusion

The techniques discussed in this report show that we can find interesting infrequent patterns that wouldn’t be discovered with the usual algorithms. One may believe that these techniques are computationally heavy, but that’s not the case. Savasere et. al [4] ran the negative correlation algorithm technique on a SPARCStation 5 which was released almost fifteen years ago [5]. The time it took to complete the algorithm over 50,000 records with an average of ten items per itemset was within the interval 300 to 1000 seconds. On a modern computer the same operations would probably be done within seconds.

Noteworthy is that negative and negative correlated patterns do not have to be infrequent, see figure 1.

It is a shame that Weka does not contain any of the discussed algorithms for finding infrequent patterns. However Apriori is implemented and since the binary tables uses Apriori to find negative patterns, it should be possible to extend the current Apriori in Weka.

References


