

DATA MINING ASSIGNMENT 5

(Constrained Association Mining)

Done by

Debarati Das

1PI13CS052

9.2.2016

Overview

Association rule learning is a method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

This assignment involved the supermarket.arff file in which we were supposed to apply the FPGrowth algorithm with the minimum support of 0.3 and minimum lift of 1.2, with the rules containing "vegetables" items only in the consequence.

Answers to the Questions :

1. What do the measures “Leverage” and “Conviction” mean?

Leverage

Introduced by Piatetsky-Shapiro, G.

$$PS(X \Rightarrow Y) = \text{leverage}(X \Rightarrow Y) = \text{supp}(X \Rightarrow Y) - \text{supp}(X)\text{supp}(Y) = P(X \wedge Y) - P(X)P(Y)$$

Leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y were statistically dependent. The rationale in a sales setting is to find out how many more units (items X and Y together) are sold than expected from the independent sells. Note that *leverage* and lift measure similar things, except that leverage measures the difference between the probability of co-occurrence of L and R as the independent probabilities of each of L and R, (given rule $L \Rightarrow R$) i.e.,

$$\text{leverage} = \Pr(L, R) - \Pr(L) \cdot \Pr(R)$$

Conviction

Introduced by Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Turk.

$$\text{conviction}(X \Rightarrow Y) = \frac{1 - \text{supp}(Y)}{1 - \text{conf}(X \Rightarrow Y)} = \frac{P(X)P(\bar{Y})}{P(X \wedge \bar{Y})}$$

Conviction was developed as an alternative to confidence which was found to not capture direction of associations adequately. Conviction compares the probability that X appears without Y if they were dependent with the actual frequency of the appearance of X without Y. *Conviction* is similar to lift, but it measures the effect of the right-hand-side not being true. It also inverts the ratio. So, if rule is given by $(L \Rightarrow R)$ conviction is measured as:

$$\text{conviction} = \Pr(L) \cdot \Pr(\text{not } R) / \Pr(L, R).$$

2. How are they calculated in Weka? (Browse the source code for FPGrowth.java. Note that the Weka install includes a jar file which contains the sources)

Calculation for leverage :

$$\text{Lev}(A \Rightarrow B) = P(A \wedge B) - P(A)P(B)$$

/* As calculated in Weka :

```
double coverageForItemSet = (double)totalSupport / (double)totalTransactions;
```

```
double expectedCoverageIfIndependent =
```

```

        ((double)premiseSupport / (double)totalTransactions) *
        ((double)consequenceSupport / (double)totalTransactions);
    return coverageForItemSet - expectedCoverageIfIndependent;
*/

```

Calculation for conviction :

So, if rule is given by $(L \Rightarrow R)$ conviction is measured $= \frac{\Pr(L) \cdot \Pr(\text{not } R)}{\Pr(L, R)}$.

/* As calculated in Weka :

```

double num = (double)premiseSupport * (double)(totalTransactions - consequenceSupport) /
        (double)totalTransactions;
double denom = premiseSupport - totalSupport + 1;
return num / denom;
*/

```

3. Notice that Weka can print out a string representation of a rule (try it out). Suppose you wanted to change default way in which a rule is printed, which method in which class needs to be modified?

We need to modify the string returned by the `toString()` method. Another way of accessing the rules is by using the method, `getAssociationRules()` which returns a list of all the mined rules.