THE UNIVERSITY OF TEXAS
AT AUSTIN

CS363D STATISTICAL LEARNING AND DATA MINING

Homework 05

Edited by LaTeX

Department of Computer Science

STUDENT

Jimmy Lin
xl5224

INSTRUCTOR

Pradeep Ravikumar

TASSISTANT

Adarsh Prasad

RELEASE DATE

April. 21 2014

DUE DATE

April. 28 2014

TIME SPENT

7 hours

April 23, 2014

Contents

List of Figures

1 Short Answer Question

1.1 superset

(a) Can a superset of an infrequent itemset be frequent? Why or why not?

Of course not. This is because the support of superset is always lower than or equals to the support of that infrequent itemset itself. (the total number of transactions are fixed and the support count of superset is always lower than or equals to the support count of that infrequent itemset itself.)

1.2 confidence

(b) Let x denote an item that occurs in every transaction of a dataset. What can you say about the confidence of a rule of the form $y \to x$, where y is some item in the dataset that appears in at least one transaction?

According to the given condition, we can see the support count

$$\sigma(y,x) = \sigma(y) \tag{1}$$

$$\sigma(x) = |T| \tag{2}$$

The confidence related to that rule $y \to x$ is

$$c(y \to x) = \frac{\sigma(y,x)}{\sigma(y)} = 1 \tag{3}$$

Hence, the confidence of $y \to x$ is 1.

1.3 threshold

(c) Let Y denote a frequent itemset. We are interested in generating rules from Y that have a confidence of at least c. Let $X \to Y - X$ be a rule that does not satisfy the confidence threshold. Let $X \subseteq X$. Can $X \to Y - X$ satisfy the confidence threshold? Give reasons.

According to the known condition,

$$c(X \to Y - X) = \frac{\sigma(Y)}{\sigma(X)} < minconf \tag{4}$$

Since the X's is the subset of X, we have

$$\sigma(X') \geq \sigma(X) \tag{5}$$

We get the relationship between

$$c(X \to Y - X) = \frac{\sigma(Y)}{\sigma(X')} \leq \frac{\sigma(Y)}{\sigma(X)} < minconf \tag{6}$$

Hence, it can be concluded that

$$c(X \to Y - X) \text{ does not satisfy the confidence threshold.} \tag{7}$$

2 Apriori algorithm

2.1 Lattice

(a) Draw an itemset lattice representing the dataset given in Table 1.

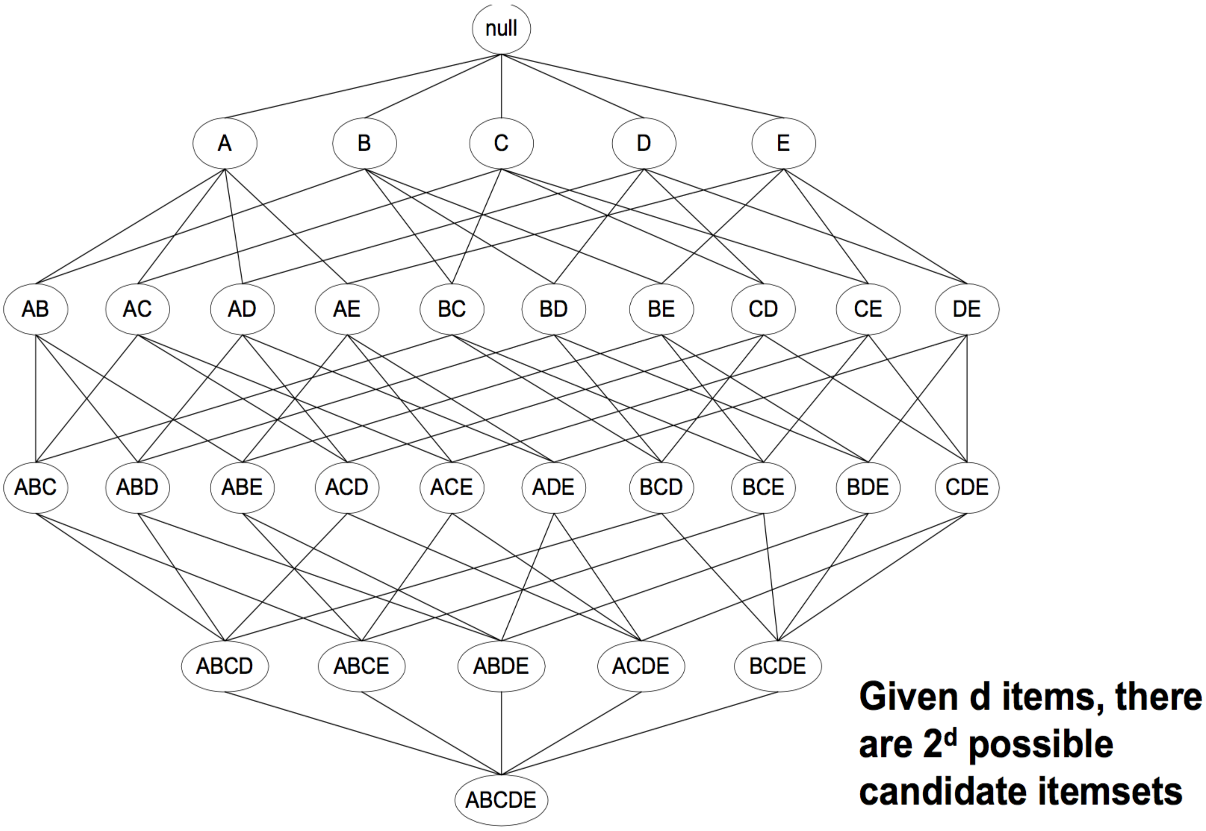

Figure 1: Itemset Lattice

2.2 Frequent Itemsets

(b) What is the percentage of frequent itemsets (with respect to all itemsets in the lattice)?

The total number of nodes in the lattice is $|T| = 2^5 = 32$.

The number of frequent itemsets is 16.

The percentage of frequent itemsets is $\frac{16}{32} = \frac{1}{2}$.

2.3 Candidate Itemsets

(c) What is the percentage of candidate itemsets that are found to be infrequent after support counting?

The total number of nodes in the lattice is $|T| = 2^5 = 32$.

Candidate itemsets that found to be infrequent after support counting are ac, ce, abd, abe, bcd, abde.

The percentage of infrequent itemsets is $\frac{6}{32} = \frac{3}{16}$

A Codes to count frequency

```
############################################################
##    FILENAME:   count.py
##    VERSION:    1.0
##    SINCE:      2014-04-23
##    AUTHOR:
##        Jimmy Lin (xl5224) - JimmyLin@utexas.edu
##    DESCRIPTION:
##        For CS363D data mining homework 05
##
############################################################
##    Edited by MacVim
##    Documentation auto-generated by Snippet
############################################################

import itertools

def query(data, toquery, thredshould=3):
    count = 0
    for datum in data:
        if set(toquery).issubset(datum):
            count += 1
    frequent = count >= thredshould
    global total
    if frequent: total += 1
    print toquery, count, frequent

if __name__ == ’__main__’:
    global total
    total = 0
    Items = [’a’, ’b’, ’c’, ’d’, ’e’]
    nTransactions = 10
    data = [set([]) for x in range(nTransactions)]
    data[0].update([’a’, ’b’, ’d’, ’e’])
    data[1].update([’b’, ’c’, ’d’])
    data[2].update([’a’, ’b’, ’d’, ’e’])
    data[3].update([’a’, ’c’, ’d’, ’e’])
    data[4].update([’b’, ’c’, ’d’, ’e’])
    data[5].update([’b’, ’d’, ’e’])
    data[6].update([’c’, ’d’])
    data[7].update([’a’, ’b’, ’c’])
    data[8].update([’a’, ’d’, ’e’])
    data[9].update([’b’, ’d’])

    for i in range(len(Items)):
        for x in list(itertools.combinations(Items, i)):
            query(data, x)
    print total
```