**MSC-BDT5002/MSC-IT 5210 Knowledge Discovery and Data Mining, Fall 2017**

Assignment 1

Student No. 20470873

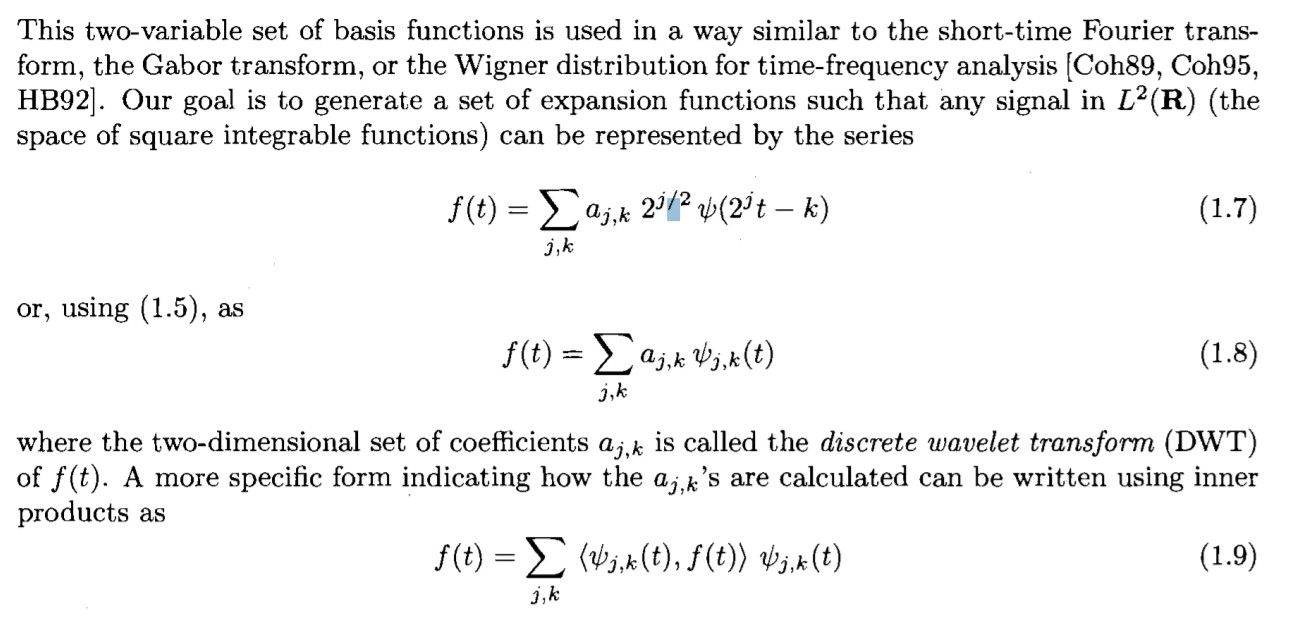
Li Jiawei

# 2.Data Preprocessing

## 2.1 Wavelet Transform

Q1:

The definition of discrete wavelet transform is below:



In my opinion, descrete wavelet transform(DWT) is a technology about linear sinal processing skill. We can build a data vector X with our data, using DWT to tansform it into another X’ with different value. We can keep these two vectors with the same length. In this way, we can create another vector contains the zip data of the remote vector. The most simple way is above, the coefficient of f(t) means the value of 2-D data vector, we call X’ above.

Q2:

1.From the specific S=[1,4,2,3,-2,-1,2,1], we divide it into two part equally, the result is:

A=[1,4,2,3]

B=[-2,-1,2,1]

2.Next,we make some pairs using the number has the same index in these two pairs like(A[i],B[i])(i=0,1,2,3),and we calculate the value of (A[i]+B[i])\*h and put it into index i, also calculate the value of (A[i]-B[i])\*h and put it in to index (i+4). In this question, h is 1/sqrt(2).and we can make another S’ like below:

**S’=[-0.707, 2.121, 0.828, 2.828, 2.121, 1.414, 0, 1.414]**

3.Next,we calculate the S’’ from S’ using the way like step 2. But we just using the front part of S’. And keep the back part stable. We can calculate the S’’ like below:

**S’’=[1.500, 3.500, -2.500, -0.500, 2.121, 1.414, 0, 1.414]**

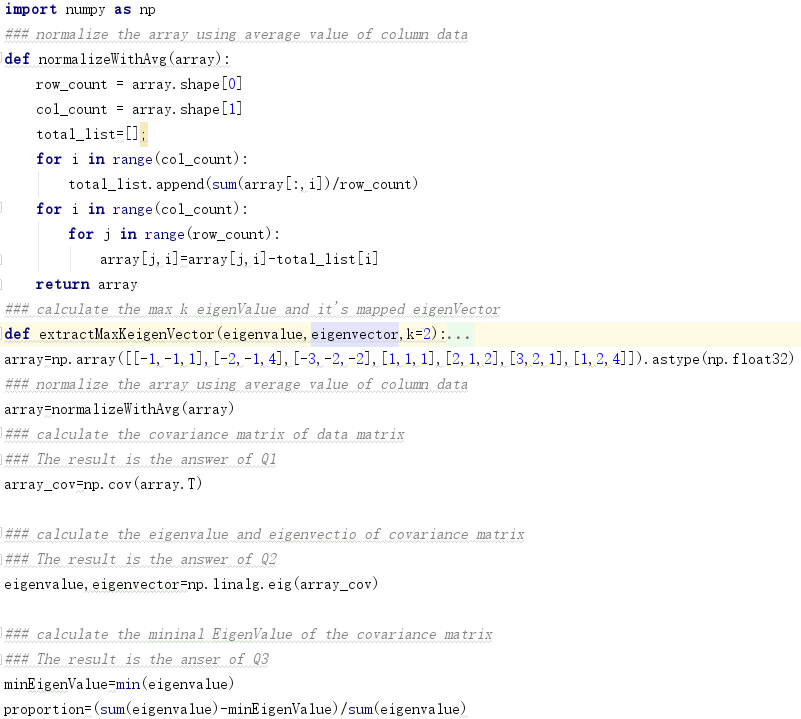
4.Finally,we calculate the S’’’ from S’’ using the way like step 3. But we just using the front quarter part of S’’. And keep another parts stabe. We can calculate the S’’’ like below:

**S’’’=[3.536, -1.414, -2.500, -0.500, 2.121, 1.414, 0, 1.414]**

S’’’ is the final result of Q2.

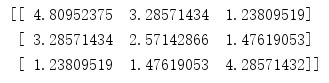
## 2.2 Principal Components Analysis

The code of PCA with python is below:



The output is below:

Q1:

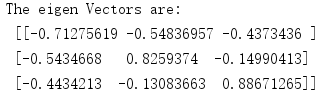


Q2:

EigenValue:



EigenVector:

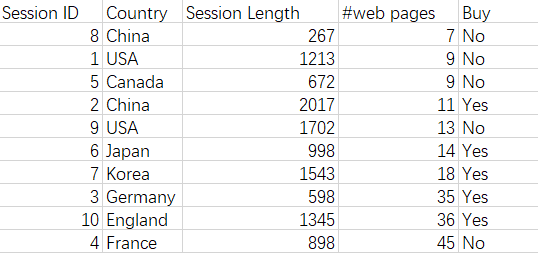


Q3:

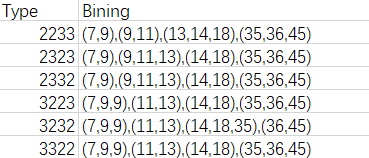


# 3.Pattern Discovey

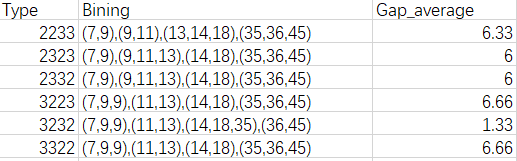
First, we user equal-depth binning to discretize Number of web pages, Now we have 10 rows, we sort the data by the web-pages column like below:



Next we use equal-depth binning to discretize the web-page column with the number of bins 4. That means we should have bin types like ‘3223’ or ‘2323’. Actually, we have C42=6 types of these bins. The binning method is below:

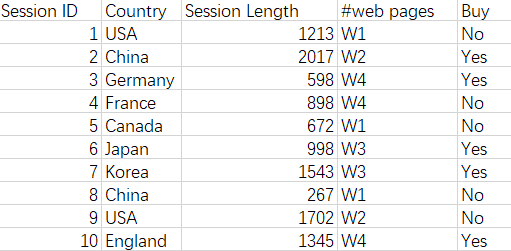


We can calculate the gap of each bin in order to choose one way of binning. For example, we calculate gap between the max value in front bin and the min value in following bin, then calculate the average value of them. We can have the following table:



In the table we hope we can get bigger gap between bins, so we choose the type 3223 with 6.66 gap score.

Next we encode the 4 bins with w1,w2,w3 and w4, then we can achieve a table with bining by web pages colum like below:



Next we try to encode the Session Length column using equal-width binning method. We first calculate the maximum gap of the Session Length values like below：

Maximum gap=Max(Session\_Length)-Min(Session\_Length)

=2017-267=1740

Next we calculate the interval of bins using the following method:

BinInterval=Maximum gap/binNum=1740/3=580

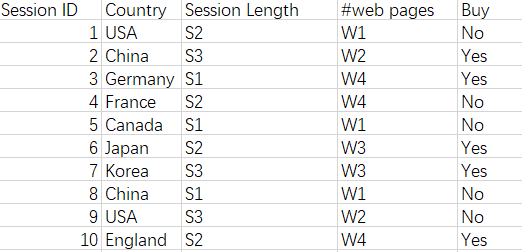
That means we build the BinInterval like the following format:

S1-The number in [267+0,267+580）=[267,847)

S2-The number in [267+580,267+1160)=[847,1427)

S3-The number in [267+1160,267+1740)=[1427,2017]

So we sort the table by Session Length column like below:



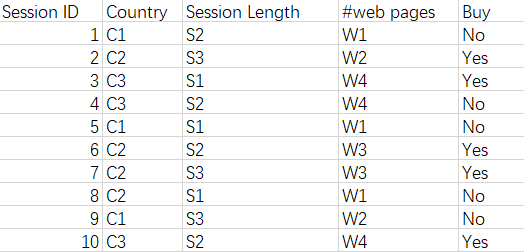
Finally, we handle the column ‘County’, follow the next format:

(USA, Canada) -> NA ->C1

(China,Japan,Korea) -> ASIA ->C2

(France,Germany,England) ->EU ->C3

So we achieve the final table like below:



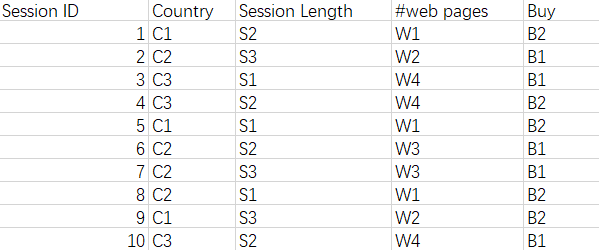
Q1:

Also, we put the column ‘Buy’ using discrete value witu the format below:

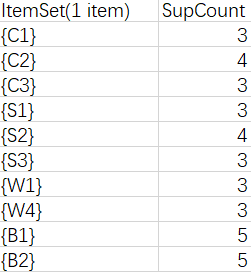
Yes-> B1

NO-> B2

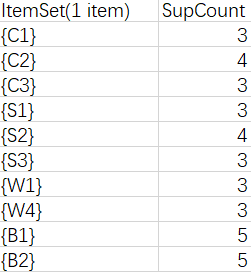
Now we get the table below:



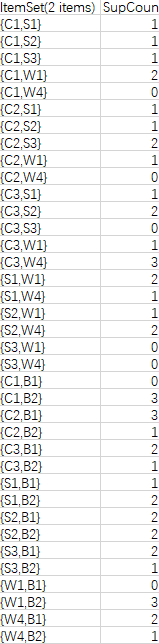
First, we list the all 1-itemset like below:



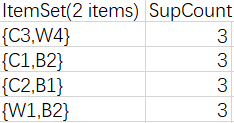
Because the value of min\_sup=3, so we remove the itemset which its supcount less than 3, that means we will remove the item {W2} and {W3}, like below:



Next, we build our 2-itemsets using the 1-itemsets. We connect all the 1-itemsets and calculate their supcount like below:



As the method behind, we remove the items with supcount value less than 3, keep just 4 values like below:

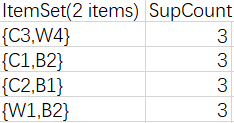


We connect the itemsets behind to build 3-itemsets like below, and list theirs supcount value:



But its value of supcount is less than 3, so we abandon this 3-itemsets.

Finally, we get all the frequent itemsets like below:



We consider the frequent pattern and build the following rules:

{C3} -> {W4}, confidence = 3/3=100%

{W4} -> {C3}, confidence = 3/3=100%

{C1} -> {B2}, confidence = 3/3=100%

{B2} -> {C1}, confidence = 3/5=60%

{C2} -> {B1}, confidence = 3/4=75%

{B1} -> {C2}, confidence = 3/5=60%

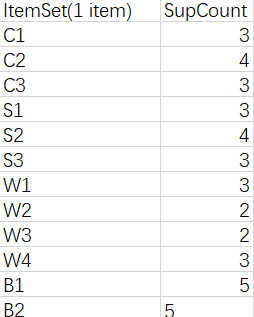
{W1} -> {B2}, confidence = 3/3=100%

{B2} -> {W1}, confidence = 3/5=60%

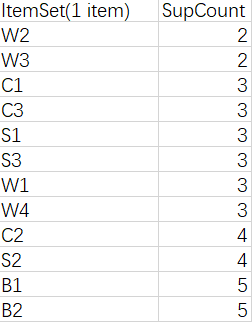
Because all the rules are all bigger than 0.3, so they a all frequent pattern of the data table.

Q2:

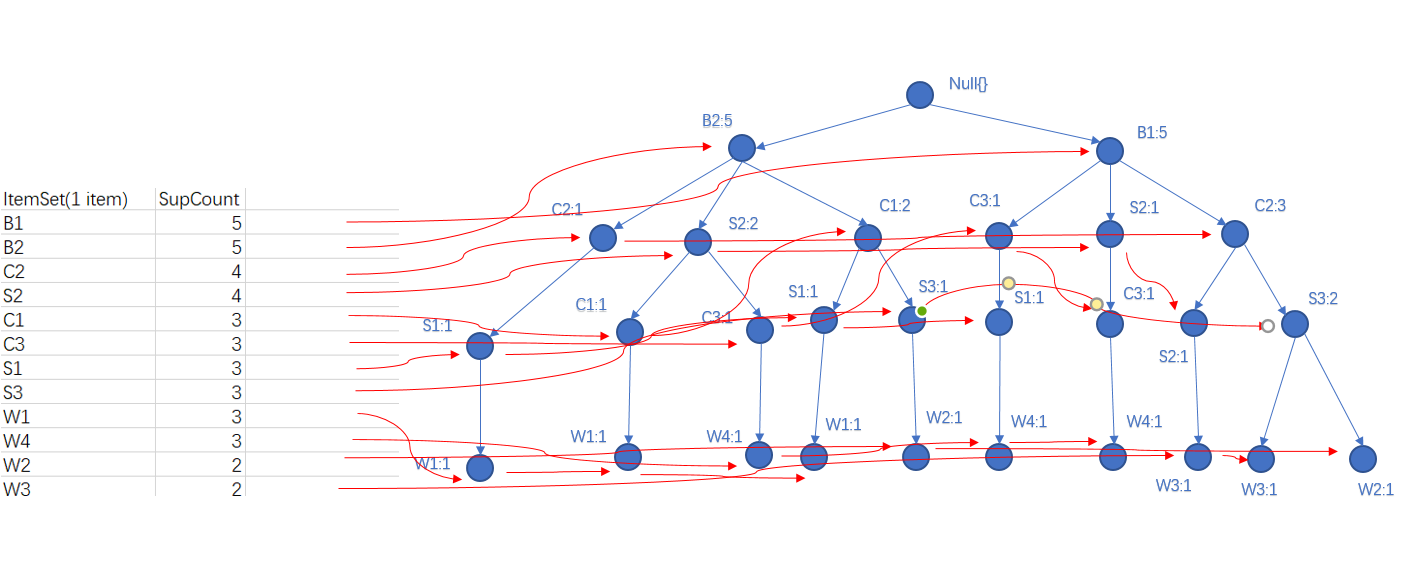
First, we list the all 1-itemset like below:



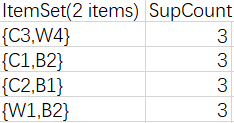
We sort the supCount value like below:



We build the FP tree like below:

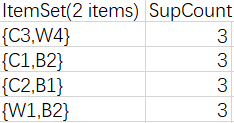


We use the FP method to build our frequent itemsets as below:

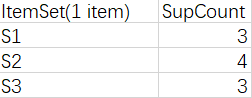


Q3:

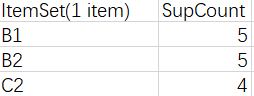
First we search the space of 2-itemsets, and we find all the 2-itemsets are all max frequent patterns:



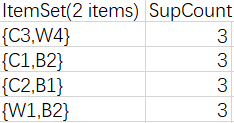
Then we search the space of 1-itemsets, and we find the 1-itemsets are max frequent patterns as below:



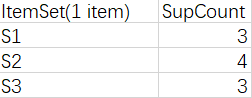
We can search the closed frequent patterns from max frequent patterns as below:



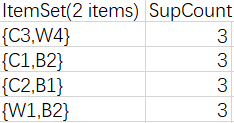
So finally, our max frequent patterns are:



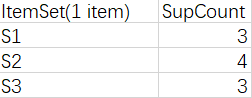
And



And our closed frequent patterns are:



And



And

