Master of Technology in Enterprise Business Analytics
(Web Analytics Elective)

Web Usage Mining Assignment

Dr. Barry Shepherd
Institute of Systems Science
National University of Singapore
Email: barryshepherd@nus.edu.sg

Assignment Details

- 20 marks
- Teams of 4 to 6
- Pick one from a given selection of datasets
- Tools = SPSS Modeler, R, other …. its your choice
- Methods = association and sequence mining
- Goals (depending on the dataset, more than one may apply)
  - Find associations between pages or items
  - Find frequent sequences of pages or items
  - Make recommendations to users – recommend a page or an item
- Validation of Findings
  - Present test/validation results for all of your findings. Typically this will involve testing your findings against a held-back (test) dataset.

Microsoft.com – Vroots Data Set

- The data was created by sampling and processing the www.microsoft.com logs. The data records the use of www.microsoft.com by 38,000 anonymous, randomly-selected users. For each user, the data lists all the areas of the web site (Vroots) that the user visited in a one week timeframe.
- *Attribute records:*
  - E.g.: A, 1277, 1, "NetShow for PowerPoint", "/stream"

attributeID for the website area (the vroot) — ignore — The vroot title — The URL relative to "http://www.microsoft.com"

- *Case and Vote records:*
  - For each user, there is a case line followed by zero or more vote lines, e.g.

```
C,"10164",10164
V,1123,1
V,1009,1
V,1052,1
```

'C' marks this as a case line,
'10164' is the case ID number of a user,
'V' marks the vote lines for this case,
'1123', 1009', 1052' are the attributes ID's of Vroots that a user visited.
'1' may be ignored.

Goal = Find and test page recommendations using association and sequence finding

PKDD 2005 Challenge Dataset

- This data comes from a Czech company running several internet shops.
- The log data covers the traffic on the web server of about three weeks - about 3 mil. records (each record is a single page view). Each log file contains the information collected during one hour, hence over 500+ files (big data).
- *Structure of the log file*
  - shopID; unix-time; IPaddress, sessionID; visited page details; referring URL

Generated when first entering a page of a web shop (user will get new ID when moving to another shop) - valid for a single session only (user will get new ID for new session)

The visited page details include:
- The internet shop (anonymised)
- Product category (63 categories, mostly electronics)
- Product brand (197 brands)
- The type of page (e.g. shopping cart, product detail, online advice etc)

Goals
- Find & test associations between product categories and brands viewed in the same session. Can also look for associations between category/brand with page type = "shopping cart", this might detect top-selling category/brands
- The data is big*, but will be much smaller after brand and category extraction. e.g. extracted records would look like: *sessionID, product category, product brand, page type*
- Can further reduce size by using sub-sampling, e.g. a subset of hours

NASA Website Data

- Two month's worth of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida. Two separate files:
  - First log was July 1, 1995 through July 31, 1995, a total of 31 days.
  - Second log was August 1, 1995 through August 31, 1995, a total of 7 days.
- The logs are an ASCII file with one line per request, with the following columns:
  - host making the request. A hostname when possible, otherwise the Internet address if the name could not be looked up.
  - timestamp in the format "DAY MON DD HH:MM:SS YYYY". The timezone is -0400.
  - request given in quotes.
  - HTTP reply code.
  - bytes in the reply.
  - Measurement
- From 1/Aug/1995:14:52:01 until 3/Aug/1995:04:36:13 there are no records, as the server was shut down due to Hurricane Erin

Goal = Find and test associations and sequences (similar to vroots)

Note: You will need to sessionise the data yourself either using:
- host (try treating all visits as one)
- host + datetime (apply the 30 min rule to get actual visits)

http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

RecSys Challenge 2015

- Click events performed by users on an e-retail website
- File1 ~ non-purchase clicks on items (products).
  - Format = Session ID, Timestamp, Item ID, Category
    - *Session ID* – the id of the session. In one session there are one or many clicks.
    - *Timestamp* – the time when the click occurred. Format of YYYY-MM-DDThh:mm:ss.SSSZ
    - *Item ID* – the unique identifier of the item that has been clicked (an integer)
    - *Category* – the context of the click.
      - » "S" indicates a special offer
      - » "0" indicates a missing value
      - » a number between 1 to 12 indicates a real category identifier
      - » any other number indicates a brand.

    E.g. if an item was clicked in the context of a promotion or special offer then the value will be "S".
    If the context was a brand (e.g. BOSCH) then the value will be an 8-10 digits number.
    If the item was clicked under regular category (e.g. sport) then the value is a number from 1 to 12.
- File2 ~ the buying events:
  - Format = Session ID, Timestamp, Item ID, Price, Quantity

Goal = find and test associations and sequences that predict if a user will buy something (and what they will buy)

http://recsys.yoochoose.net/challenge.html
https://www.kaggle.com/chadgostopp/recsys-challenge-2015

RetailRocket

- Data (3 files) from a real-world ecommerce website. It is raw data, i.e. without any content transformations, however, all values are hashed due to confidential issues.
- **Behaviour data**: events like clicks, add to carts, transactions were collected over a period of 4.5 months. There are three types of events: "view", "addtocart" or "transaction"
  - Format = timestamp, visitorID, eventtype, itemID (time is unixtime)
  - E.g. 1439694000000, 1, view, 100
- **Item properties:** since the property of an item can vary in time (e.g., price changes over time), every row in the file has corresponding timestamp
  - Format = timestamp, itemid, property, value
  - E.g. 1439694000000, 1, 100, 1000
- **Category tree:** Every row specifies a child categoryId and the corresponding parent. E.g.:
  - Line "100,200" means that categoryid=1 has parent with categoryid=200
  - Line "300," means that categoryid hasn't parent in the tree

Goal = Try to predict properties of items in "addtocart" events by using data from "view" events for any visitor.

https://www.kaggle.com/retailrocket/ecommerce-dataset/home

Assessment and Report Guidelines

- Hand in your report + your R code (and SPSS streams) by Nov 6th
  - Upload files to IVLE – ensure the names of all team members are in the report.
    Make sure your files have a name unique to you (remember all teams upload to same IVLE directory)
- The report should contain the following:
  - **Executive Summary**
    - 1 page at most – describe the problem you are solving and summarise your results
  - **Model Build & Test Process**
    - Details of any data cleaning and preprocessing performed
    - What tool and algorithms did you use? What problems (if any) did you face?
    - What settings did you use , e.g. state if you reduced or raised min. rule confidence and support before model build
    - Details of how you split the data into training and test sets and how you performed testing
  - **Associations and Sequences Found**
    - Show the top associations (and sequences) that you found
  - **Model Test Results**
    - Show model precision and recall - how many recommendations did your rules make on the test data and how many of these were correct?
    - Do you think your model performance good enough to deploy? Would it make sufficient recommendations to be useful?

Appendix

- Model Testing Hints.....

Association Rule Testing: General Concept

- We use a separate test set of users
- We apply the rules to each item in each test users basket to obtain a set of predictions
- Lets consider only simple rules to begin with (one antecedent only)
  - E.g. User1 basket = {A,B,C} rules : A=>B, A=>C, A=>D, B=>C, C => E
  - Predictions for A ~ B, C, D
    Predictions for B ~ C
    Predictions for C ~ E
- If the prediction is also in the basket we can say that the prediction is likely correct since the user has already bought, seen or liked the predicted item
  - Predictions for A ~ B, C, D (2 correct)
    Predictions for B ~ C (1 correct)
    Predictions for C ~ E (0 correct)

Testing Association Rules in R*

```
#build the rules as before
rules <- apriori(trainegs, parameter = list(supp=0.1, conf=0.1, minlen=2))

#read the test data
testegs = read.csv(file="simplebasket-test.csv");
colnames(testegs) <- c("basketID","items")  # set standard names

#execute rules against test data
rulesDF = as(rules,"data.frame")
testegs$preds = apply(testegs,1,function(X) makepreds(X["items"], rulesDF))

# extract unique predictions for each test user
userpreds = as.data.frame(aggregate(preds ~ basketID, data = testegs, paste, collapse=","))
userpreds$preds = apply(userpreds,1,function(X) uniqueitems(X["preds"]))

# extract unique items bought (or rated highly) for each test user
baskets = as.data.frame(aggregate(items ~ basketID, data = testegs, paste, collapse=","))
baskets$items = apply(baskets,1,function(X) uniqueitems(X["items"]))

#count how many unique predictions made are correct
correctpreds = sum(apply(userpreds,1,function(X) checkpreds(X["preds"],X["basketID"])))

# count total number of unique predictions made
totalpreds = sum(apply(userpreds,1,function(X) countpreds(X["preds"][[1]])))

precision = correctpreds*100/totalpreds
```

**Rattle does not enable testing of association rules*

Testing Association Rules in R contd.

```
#remove duplicate items from a basket (itemstrg)
uniqueitems <- function(itemstrg) {
  unique(as.list(strsplit(gsub(" ","",itemstrg),","))[[1]])
}

# execute ruleset using item as rule antecedent (handles single item antecedents only)
makepreds <- function(item, rulesDF) {
  antecedent = paste("{",item,"} =>",sep="")
  firingrules = rulesDF[grep(antecedent, rulesDF$rules,fixed=TRUE),1]
  gsub(" ","",toString(sub("\\}","",sub(".*=> \\{","",firingrules))))
}

# count how many predictions are in the basket of items already seen by that user
# Caution : refers to "baskets" as a global
checkpreds <- function(preds, baskID) {
  plist = preds[[1]]
  blist = baskets[baskets$basketID == baskID,"items"][[1]]
  cnt = 0
  for (p in plist) {
    if (p %in% blist) cnt = cnt+1
  }
  cnt
}

# count all predictions made
countpreds <- function(predlist) {
  len = length(predlist)
  if (len > 0 && (predlist[[1]] == "")) 0 # avoid counting an empty list
  else len
}
```

Testing Association Rules in R contd.

```
> rulesDF
          rules support confidence      lift
1    {E} => {D}     0.2  1.0000000 2.5000000
2    {D} => {E}     0.2  0.5000000 2.5000000
3    {C} => {B}     0.2  1.0000000 2.5000000
4    {B} => {C}     0.2  0.5000000 2.5000000
5    {C} => {A}     0.2  1.0000000 1.6666667
6    {A} => {C}     0.2  0.3333333 1.6666667
7    {D} => {A}     0.2  0.5000000 0.8333333
8    {A} => {D}     0.2  0.3333333 0.8333333
9    {B} => {A}     0.4  1.0000000 1.6666667
10   {A} => {B}     0.4  0.6666667 1.6666667
11 {B,C} => {A}     0.2  1.0000000 1.6666667
12 {A,C} => {B}     0.2  1.0000000 2.5000000
13 {A,B} => {C}     0.2  0.5000000 2.5000000
>
```

```
> testegDF
  basketID items preds
1        6     A C,D,B
2        6     C   B,A
3        6     D   E,A
4        7     B   C,A
5        7     C   B,A
6        8     G
7        8     H
>
> predDF
  basketID          preds
1        6 C, D, B, A, E
2        7       C, A, B
3        8
>
> basketDF
  basketID   items
1        6 A, C, D
2        7    B, C
3        8    G, H
>
```

```
> cat("precision=",precision, "corr=",correctpreds,"total=",totalpreds)
precision= 62.5 corr= 5 total= 8
>
```


Testing Association Rules in R contd.

- What about rules with multiple antecedents?
  - E.g. User1 basket = {A,B,C} rules : A, B=>C; A, B=>D; B, C=>A; A,B,C=>D
  - Predictions for A, B ~ C , D {1 correct}
    Predictions for B, C ~ A {1 correct}
    Predictions for A,B,C ~ D {0 correct}

- Code change
  - Derive all subsets of basket items
  - Match against rules
  - Proceed as before

Basket Subset
A
B
C
A,B
A,C
B,C
A,B,C

Running Spade in R

- Input data is a sequence of baskets, each basket is contained in a separate record.
  - Record format = Sequence-ID, event-ID , item-count, item-list (i.e. basket items)

```
> data("zaki")
> as(zaki,"data.frame")
   transactionID.sequenceID transactionID.eventID transactionID.SIZE     items
1                         1                    10                  2     {C,D}
2                         1                    15                  3   {A,B,C}
3                         1                    20                  3   {A,B,F}
4                         1                    25                  4 {A,C,D,F}
5                         2                    15                  3   {A,B,F}
6                         2                    20                  1       {E}
7                         3                    10                  3   {A,B,F}
8                         4                    10                  3   {D,G,H}
9                         4                    20                  2     {B,F}
10                        4                    25                  3   {A,G,H}
>
```

  - For vroots dataset,
    - sequenceID = userID
    - eventID ~ order of page view
    - Item lists contain one page only (assume two pages are not viewed at the same time)
    - Use read_baskets() to input the data file (type ?read_baskets() in R for documentation)

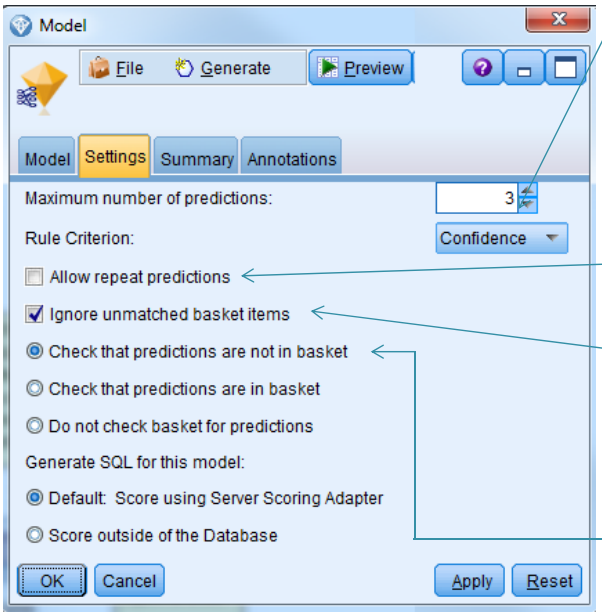
Testing cSpade Sequences

- First convert to rules using ruleInduction()
- Then execute and test in a similar manner to association rules

```
> s2 <- cspade(zaki, parameter = list(support = 0.4))
> as(s2,"data.frame")
          sequence support
1            <{A}>    1.00
2            <{B}>    1.00
3            <{D}>    0.50
4            <{F}>    1.00
5          <{A,F}>    0.75
6          <{B,F}>    1.00
7        <{D},{F}>    0.50
8      <{D},{B,F}>    0.50
9        <{A,B,F}>    0.75
10         <{A,B}>    0.75
11       <{D},{B}>    0.50
12       <{B},{A}>    0.50
13       <{D},{A}>    0.50
14       <{F},{A}>    0.50
15   <{D},{F},{A}>    0.50
16     <{B,F},{A}>    0.50
17 <{D},{B,F},{A}>    0.50
18   <{D},{B},{A}>    0.50
>
> r2 <- ruleInduction(s2, confidence = 0.5,control = list(verbose = TRUE))
> as(r2,"data.frame")
                    rule support confidence lift
1        <{D}> => <{F}>     0.5        1.0  1.0
2      <{D}> => <{B,F}>     0.5        1.0  1.0
3        <{D}> => <{B}>     0.5        1.0  1.0
4        <{B}> => <{A}>     0.5        0.5  0.5
5        <{D}> => <{A}>     0.5        1.0  1.0
6        <{F}> => <{A}>     0.5        0.5  0.5
7    <{D},{F}> => <{A}>     0.5        1.0  1.0
8      <{B,F}> => <{A}>     0.5        0.5  0.5
9  <{D},{B,F}> => <{A}>     0.5        1.0  1.0
10   <{D},{B}> => <{A}>     0.5        1.0  1.0
>
```

Executing SPSS Association Rules

Edit the nugget node to set the rule execution settings



Many association rules can execute for each test record. The default number of predictions to make is 3. Rules are executed in the order of confidence (most confident first)

Don't select this – don't recommend same item twice!
e.g. *bread & cheese => wine*
cheese & fruit => wine

(multiple consequents, e.g. bread & cheese => wine & pate are considered repeat predictions only if all consequents (wine & pate) have been predicted before)

Usually Select , but you decide!
e.g. if rule is: *tent & sleeping bag => gas stove*
and user basket is: *(tent, sleeping bag, kettle)*
then does having a kettle impact whether we should recommend the gas stove?

Select this – don't recommend what's already been seen or bought
exceptions ~ e.g. if basket contains all pages a user has ever seen on a news website then maybe don't select

Executing SPSS Association Rules

Many rules can fire for each test record => multiple predictions

- Three columns are generated for each rule executed:
  (1) the prediction (2) the prediction confidence and (3) the ID of the rule used.
- These new columns get auto-generated names (beginning with $A, $AC, $A-Rule-ID). The auto-name may also includes number of input variables used for model build

The input data fields | *Most confident prediction* | *Second most confident prediction* | *Third most confident prediction*

user	frontpage	news	tech	....	$A-17 fields-1	$AC-17 fields-1	$A-Rule_ID-1	$A-17 fields-2	$AC-17 fields-2	$A-Rule_ID-2	$A-17 fields-3	$AC-17 fields-3	$A-Rule_ID-3
1	1	0	0		news	0.235	83	sports	0.136	65	on-air	0.129	81
2	0	1	0		frontpage	0.420	82	local	0.154	75	on-air	0.145	79
3	0	1	1		frontpage	0.420	82	misc	0.222	43	on-air	0.176	73
4	0	0	0		$null$	$null$	$null$	$null$	$null$	$null$	$null$	$null$	$null$
5	1	0	0		news	0.235	83	sports	0.136	65	on-air	0.129	81
6	0	0	0		frontpage	0.188	80	misc	0.149	45	news	0.119	78
7	1	0	0		news	0.235	83	sports	0.136	65	on-air	0.129	81
8	0	0	0		frontpage	0.188	80	misc	0.149	45	news	0.119	78
9	0	0	0		frontpage	0.462	48	local	0.339	42	news	0.246	46
10	0	0	1		frontpage	0.510	15	news	0.346	14	misc	0.222	43
11	1	0	0		news	0.309	27	on-air	0.197	26	local	0.197	25
12	0	0	0		frontpage	0.387	64	news	0.210	62	tech	0.128	57
13	1	0	0		news	0.235	83	sports	0.136	65	on-air	0.129	81
14	0	0	0		news	0.129	19	frontpage	0.113	20	local	0.111	17
15	0	0	0		frontpage	0.188	80	misc	0.149	45	news	0.119	78

Viewing rule predictions using a Table node

Understanding SPSS Rule Execution Output

- If the test data is in transaction format then each transaction is considered in sequence (as if they occurred over time)

- E.g. assume the ruleset is:
  - Rule1: A => D , confidence=0.6
  - Rule2: A => E, confidence = 0.7
  - Rule3: A & D => F, confidence = 0.8
  - Rule4: A & C => D, confidence = 0.9

- Assume test user1 has the basket (A, B, C, F, E). In transaction format this is 5 test records:
  - User1, A
  - User1, B
  - User1, C
  - User1, F
  - User1, E

- When the first transaction is read then rules 1 & 2 can execute. The output hence looks like:

user	movie	pred1	cf1	id1	pred2	cf2	id2	pred3	cf3	id3
1	A	E	0.7	2	D	0.6	1	n/a	n/a	n/a

Understanding SPSS Rule Execution Output

- When the second transaction (B) is read no new rules execute, but existing predictions still hold for that user. Since one record is output for every test transaction the output now looks like this:

user	movie	pred1	cf1	id1	pred2	cf2	id2	pred3	cf3	id3
1	A	E	0.7	2	D	0.6	1	n/a	n/a	n/a
1	B	E	0.7	2	D	0.6	1	n/a	n/a	n/a

- When the third test transaction (C) is read then rule4 can fire, it has highest confidence so far hence appears as the left most prediction pushing the other two to the right

user	movie	pred1	cf1	id1	pred2	cf2	id2	pred3	cf3	id3
1	A	E	0.7	2	D	0.6	1	n/a	n/a	n/a
1	B	E	0.7	2	D	0.6	1	n/a	n/a	n/a
1	C	F	0.9	4	E	0.7	2	D	0.6	1

The rules:

R1: A =>D , cf=0.6
R2: A =>E, cf = 0.7
R3: A & F=>G, cf= 0.8
R4: A & C =>F, cf = 0.9

Understanding SPSS Rule Execution Output

- When transaction4 (F) is read then rule3 can now fire. The prediction for F is removed since we earlier checked the option "check that predictions are not in the basket". The highest confidence prediction is now from rule3

user	movie	pred1	cf1	id1	pred2	cf2	id2	pred3	cf3	id3
1	A	E	0.7	2	D	0.6	1	n/a	n/a	n/a
1	B	E	0.7	2	D	0.6	1	n/a	n/a	n/a
1	C	F	0.9	4	E	0.7	2	D	0.6	1
1	F	G	0.8	3	E	0.7	2	D	0.6	1

- When transaction5 (E) is read then the prediction for E is also removed

user	movie	pred1	cf1	id1	pred2	cf2	id2	pred3	cf3	id3
1	A	E	0.7	2	D	0.6	1	n/a	n/a	n/a
1	B	E	0.7	2	D	0.6	1	n/a	n/a	n/a
1	C	F	0.9	4	E	0.7	2	D	0.6	1
1	F	G	0.8	3	E	0.7	2	D	0.6	1
1	E	G	0.8	3	D	0.6	1	n/a	n/a	n/a

The rules:

R1: A =>D , cf=0.6
R2: A =>E, cf = 0.7
R3: A & F=>G, cf= 0.8
R4: A & C =>F, cf = 0.9