

Almost Famous

Cindy Lamm

21:25, Sunday 18th January, 2015

```
writeLevelMappingToFile <- function(df, variable, file) {  
  varIdx <- which(names(df)==variable)  
  tmp <- unclass(unique(df[,varIdx]))  
  mapping <- data.frame(numericLevel=tmp, characterLevel=levels(tmp))  
  write.csv(mapping, file=file, row.names=FALSE)  
}
```

```
getMapFileName <- function(variableName, type) {  
  paste("out/",type,"_level_map_",variableName,".csv", sep="")  
}
```

Load variable names and types:

```
nameTypeDataFile <- "resources/raw_variables.csv"  
variableNames <- read.csv(nameTypeDataFile, header=TRUE, stringsAsFactors=FALSE)  
variableNames  
  
##      name      type  
## 1  visit_id  factor  
## 2      uid  factor  
## 3  campaign  factor  
## 4    tstamp character  
## 5 experiments  factor  
## 6     action  factor  
## 7     query  factor  
  
factorIdx <- which(variableNames$type=="factor")  
factorNames <- variableNames$name[factorIdx]
```

Read the top 2000 lines of the web.log data:

```
webFile <- "../data/head2000.csv"  
webData <- read.csv(webFile, stringsAsFactors=FALSE, col.names=variableNames$name,  
                    colClasses=variableNames$type, na.strings=c("NA",""))  
webData$tstamp <- as.POSIXct(webData$tstamp)  
str(webData)  
  
## 'data.frame': 2000 obs. of 7 variables:  
## $ visit_id : Factor w/ 1719 levels "10040801398",...: 158 43 1712 223 433 477 37 69 176 590 ...
```

```
## $ uid      : Factor w/ 1719 levels "100007286","100049500",...: 1417 1513 31 1183 773 1222 1468 15...
## $ campaign : Factor w/ 10 levels "103","127","14",...: 10 8 1 9 7 8 3 1 10 7 ...
## $ tstamp   : POSIXct, format: "2014-09-15 00:00:01" "2014-09-15 00:00:02" ...
## $ experiments: Factor w/ 4 levels "[1 3]","[1 4]",...: 3 1 1 2 4 1 2 3 2 3 ...
## $ action    : Factor w/ 4 levels "adclick","landed",...: 2 2 2 2 2 2 2 2 2 ...
## $ query     : Factor w/ 5 levels "advanced analytics",...: 4 5 5 1 1 5 2 5 4 1 ...
```

Add variable with the total time spent per visit, `total_time_spent`, and `time_diff` indicating the seconds that passed inbetween the logged entries within a visit:

```
require(plyr)
webData <- ddply(webData, .(visit_id), mutate,
  total_time_spent=max(tstamp)-min(tstamp),
  time_diff=c(NA,diff(tstamp)))
viewExample(webData,"web")

##      visit_id      uid campaign      tstamp experiments action
## 850 4789307146 179647219      103 2014-09-15 00:07:53      [2 4] landed
## 851 4789307146 179647219      <NA> 2014-09-15 00:09:33      [2 4] signup
##               query total_time_spent time_diff
## 850 predictive modeling      1.666667 secs      NA
## 851      <NA>      1.666667 secs      1.666667
```

Look at a summary for the web data:

```
summary(webData)

##      visit_id      uid      campaign      tstamp
## 10424039824: 4 105521118: 4 103 :384 Min. :2014-09-15 00:00:01
## 19883673173: 4 106305837: 4 558 :373 1st Qu.:2014-09-15 00:07:03
## 38814711307: 4 110827261: 4 31 :264 Median :2014-09-15 00:13:23
## 43032154989: 4 116041288: 4 59 :260 Mean :2014-09-15 00:13:40
## 50850280949: 4 120457434: 4 127 :107 3rd Qu.:2014-09-15 00:20:35
## 58444220460: 4 145965967: 4 (Other):331 Max. :2014-09-15 00:27:14
## (Other) :1976 (Other) :1976 NA's :281
## experiments      action      query      total_time_spent
## [1 3]:469 adclick: 126 advanced analytics :524 Length:2000
## [1 4]:484 landed :1719 building predictive models:113 Class :difftime
## [2 3]:554 order : 51 data science :111 Mode :numeric
## [2 4]:493 signup : 104 data science training :214
##               predictive modeling :757
##               NA's :281
##
##      time_diff
## Min. : 1.000
## 1st Qu.: 2.483
## Median : 4.067
## Mean : 10.020
## 3rd Qu.: 9.000
## Max. :201.000
## NA's :1719
```

Look at a summary per visit for the web data:

```
webAggVisits <- aggregatePerVisit(webData)
summary(webAggVisits)
```

##	visit_id	nb_entries	uid	campaign	nb_experiments
##	10040801398:	1 Min. :1.000	100007286:	1 103 :384	[1 3]:409
##	10060610948:	1 1st Qu.:1.000	100049500:	1 558 :373	[1 4]:424
##	10109427525:	1 Median :1.000	100181847:	1 31 :264	[2 3]:460
##	10278786916:	1 Mean :1.163	100307194:	1 59 :260	[2 4]:426
##	10296243639:	1 3rd Qu.:1.000	100323489:	1 127 :107	
##	10342204026:	1 Max. :4.000	100340661:	1 94 :107	
##	(Other) :1713		(Other) :1713	(Other):224	

```
##
```

##	actions	queries	median_time_diff
##	landed :1491	advanced analytics :524	Min. : 1.000
##	landed,signup : 101	building predictive models:113	1st Qu.: 2.533
##	landed,order : 50	data science :111	Median : 4.075
##	landed,adclick : 40	data science training :214	Mean : 10.335
##	landed,adclick,adclick : 18	predictive modeling :757	3rd Qu.: 8.000
##	landed,adclick,adclick,adclick: 16		Max. :114.000
##	(Other) : 3		NA's :1491

```
viewAggExample(webAggVisits, "web", "visit")
```

##	visit_id	nb_entries	uid	campaign	nb_experiments	actions
##	1492	8786064200	2	17968217	103	[2 4] landed,order
##						queries median_time_diff
##	1492	predictive modeling				2.4

Look at a summary per uid (supposedly user) for the web data:

```
webAggUids <- aggregatePerUid(webData)
summary(webAggUids)
```

##	uid	nb_entries	visit_ids	campaign	nb_experiments
##	100007286:	1 Min. :1.000	10040801398:	1 103 :384	[1 3]:409
##	100049500:	1 1st Qu.:1.000	10060610948:	1 558 :373	[1 4]:424
##	100181847:	1 Median :1.000	10109427525:	1 31 :264	[2 3]:460
##	100307194:	1 Mean :1.163	10278786916:	1 59 :260	[2 4]:426
##	100323489:	1 3rd Qu.:1.000	10296243639:	1 127 :107	
##	100340661:	1 Max. :4.000	10342204026:	1 94 :107	
##	(Other) :1713		(Other) :1713	(Other):224	

```
##
```

##	actions	queries	median_time_diff
##	landed :1491	advanced analytics :524	Min. : 1.000
##	landed,signup : 101	building predictive models:113	1st Qu.: 2.533
##	landed,order : 50	data science :111	Median : 4.075
##	landed,adclick : 40	data science training :214	Mean : 10.335
##	landed,adclick,adclick : 18	predictive modeling :757	3rd Qu.: 8.000
##	landed,adclick,adclick,adclick: 16		Max. :114.000
##	(Other) : 3		NA's :1491

```
viewAggExample(webAggUids, "web", "uid")
```

##	uid	nb_entries	visit_ids	campaign	nb_experiments
----	-----	------------	-----------	----------	----------------

```
## 110 110827261          4 9909532485          31          [2 3]
##                  actions              queries median_time_diff
## 110 landed,adclick,adclick,adclick advanced analytics          9
```

Read spam data:

```
spamFile <- "../data/spam.csv"
spamData <- read.csv(spamFile, stringsAsFactors=FALSE, col.names=variableNames$name,
                     colClasses=variableNames$type, na.strings=c("NA",""))
spamData$tstamp <- as.POSIXct(spamData$tstamp)
str(spamData)

## 'data.frame': 4404 obs. of 7 variables:
## $ visit_id : Factor w/ 1482 levels "10199862810",...: 146 146 130 130 130 602 602 602 602 1409 ...
## $ uid      : Factor w/ 1060 levels "100191","100547",...: 1038 1038 238 238 238 9 9 9 9 320 ...
## $ campaign : Factor w/ 10 levels "103","127","14",...: 6 NA 6 NA NA 1 NA NA NA 1 ...
## $ tstamp   : POSIXct, format: "2014-09-15 00:06:27" "2014-09-15 00:06:33" ...
## $ experiments: Factor w/ 4 levels "[1 3]","[1 4]",...: 3 3 4 4 4 2 2 2 2 3 ...
## $ action    : Factor w/ 2 levels "adclick","landed": 2 1 2 1 1 2 1 1 1 2 ...
## $ query     : Factor w/ 5 levels "advanced analytics",...: 3 NA 3 NA NA 5 NA NA NA 5 ...
```

I again add a variable time_spent

Look at a summary of the spam data:

```
summary(spamData)

##      visit_id      uid      campaign      tstamp
## 1097758223 :    4   180718 :   14   103      : 339   Min.   :2014-09-15 00:06:27
## 1101067381 :    4   152118 :   12   558      : 303   1st Qu.:2014-09-18 22:06:23
## 11428883192:    4   23119  :   12   31       : 221   Median :2014-09-23 03:00:47
## 1191433828 :    4   8235   :   12   59       : 217   Mean    :2014-09-23 00:33:30
## 12119332951:    4   86179  :   12   127      : 106   3rd Qu.:2014-09-27 04:53:49
## 12160456931:    4   12204  :   11   (Other): 296   Max.    :2014-09-30 23:52:15
## (Other)    :4380   (Other):4331   NA's     :2922
## experiments      action      query      total_time_spent
## [1 3]:1135   adclick:2922   advanced analytics      : 438   Min.    : 1.00
## [1 4]:1153   landed :1482   building predictive models: 96   1st Qu.: 8.00
## [2 3]:1054      data science      : 102   Median :12.00
## [2 4]:1062      data science training : 204   Mean    :12.32
##                  predictive modeling : 642   3rd Qu.:17.00
##                  NA's                :2922   Max.    :29.00
##
##      time_diff
## Min.    : 1.000
## 1st Qu.: 3.000
## Median  : 6.000
## Mean    : 5.636
## 3rd Qu.: 8.000
## Max.    :10.000
## NA's    :1482
```

Look at a summary per visit for the spam data:

```

spamAggVisits <- aggregatePerVisit(spamData)
summary(spamAggVisits)

##          visit_id      nb_entries      uid      campaign      nb_experiments
## 10199862810: 1      Min.      :2.000  152118 : 4      103      :339  [1 3]:382
## 10219041924: 1      1st Qu.:2.000  176470 : 4      558      :303  [1 4]:384
## 10346637545: 1      Median :3.000  180718 : 4      31       :221  [2 3]:353
## 10427993218: 1      Mean   :2.972  62370  : 4      59       :217  [2 4]:363
## 10441154073: 1      3rd Qu.:4.000  86179  : 4      127      :106
## 10485842186: 1      Max.    :4.000  93067  : 4      94       : 98
## (Other)      :1476                    (Other):1458  (Other):198
##
##              actions              queries      median_time_diff
## landed,adclick              :509  advanced analytics      :438  Min.    : 1.000
## landed,adclick,adclick      :506  building predictive models: 96  1st Qu.: 4.000
## landed,adclick,adclick,adclick:467  data science              :102  Median : 6.000
##                                     data science training      :204  Mean   : 5.659
##                                     predictive modeling          :642  3rd Qu.: 7.500
##                                     Max.    :10.000
##
viewAggExample(spamAggVisits, "spam", "visit")

##          visit_id nb_entries      uid campaign nb_experiments      actions
## 754 54864188974      4 193279      59      [2 4] landed,adclick,adclick,adclick
##              queries median_time_diff
## 754 advanced analytics      3

```

Look at a summary per uid (supposedly user) for the spam data:

```

spamAggUids <- aggregatePerUid(spamData)
summary(spamAggUids)

##          uid      nb_entries      visit_ids      campaign
## 100191 : 1      Min.      : 2.000  10199862810      : 1      103      :180
## 100547 : 1      1st Qu.: 3.000  10219041924      : 1      558      :158
## 10060  : 1      Median : 4.000  10346637545,9973480327 : 1      31       :112
## 101345 : 1      Mean   : 4.155  10427993218      : 1      59       :112
## 101493 : 1      3rd Qu.: 5.000  10441154073,62074161015: 1      94       : 52
## 101645 : 1      Max.    :14.000  10485842186      : 1      127      : 49
## (Other):1054                    (Other)      :1054  (Other):397
##
## nb_experiments      actions
## [1 3]:265      landed,adclick              :245
## [1 4]:275      landed,adclick,adclick        :245
## [2 3]:256      landed,adclick,adclick,adclick :234
## [2 4]:264      landed,adclick,landed,adclick,adclick : 36
##                                     landed,adclick,adclick,adclick,landed,adclick: 35
##                                     landed,adclick,adclick,landed,adclick,adclick: 32
##                                     (Other)              :233
##
##              queries      median_time_diff
## predictive modeling      :373  Min.    : 1.000
## advanced analytics      :236  1st Qu.: 4.000
## data science training    :104  Median : 6.000

```

```
## data science : 51 Mean : 5.694
## building predictive models : 48 3rd Qu.: 7.500
## predictive modeling,advanced analytics: 44 Max. :10.000
## (Other) :204

viewAggExample(spamAggUids, "spam", "uid")

## uid nb_entries visit_ids campaign nb_experiments
## 539 188071 8 44678193401,45541719747,73898419674 31,558 [1 4]
## actions
## 539 landed,adclick,landed,adclick,landed,adclick,adclick,adclick
## queries median_time_diff
## 539 advanced analytics,predictive modeling 5
```

Write out a file which can be processed by Spark, meaning all factors as numeric values. Also unclass factors with digits as levels to have resulting variables on roughly the same scale:

```
numericSpamVisits <- data.frame(visit_id=spamAggVisits$visit_id,
                                nb_actions=spamAggVisits$nb_entries,
                                uid=unclass(spamAggVisits$uid),
                                campaign=unclass(spamAggVisits$campaign),
                                actions=unclass(spamAggVisits$actions),
                                queries=unclass(spamAggVisits$queries),
                                median_time_diff=spamAggVisits$median_time_diff)

head(numericSpamVisits)

## visit_id nb_actions uid campaign actions queries median_time_diff
## 1 10199862810 2 1053 1 1 5 8.0
## 2 10219041924 3 244 8 2 5 7.5
## 3 10346637545 2 745 10 1 4 7.0
## 4 10427993218 3 95 6 2 3 4.5
## 5 10441154073 3 324 8 2 5 4.0
## 6 10485842186 3 431 7 2 1 4.5

write.csv(numericSpamVisits, file="out/spam_visits_numeric.csv", row.names=FALSE)
```

Also write the level mapping in to files:

```
writeLevelMappingToFile(spamAggVisits, "uid", getMapFileName("uid","spam"))
writeLevelMappingToFile(spamAggVisits, "campaign", getMapFileName("campaign","spam"))
writeLevelMappingToFile(spamAggVisits, "actions", getMapFileName("actions","spam"))
writeLevelMappingToFile(spamAggVisits, "queries", getMapFileName("queries","spam"))
```