checkin data: global analysis

Ming Li

Wednesday, August 13, 2014

Based on the [summarization](checkin-analysis-summarize.html/) of [previous work](checkin-analysis.html/), we want to perform similar exploratory analysis in a global scale.

The assumption is: despite the sparsity of any single users, the gloabl dataset will express some relevant patterns that might be hidden from single users, because of **LLN**.

#### 1. chi-square for temporal and meteorological

Load the data.

## checkin data  
## nrows=529931 ~~ 19 weeks/133 days's data  
checkin.global = read.csv( paste0(basedir, "data\\allcheckins.csv"),   
 header=TRUE, sep=",", nrows=529931, na.strings = "none",  
 colClasses = c("numeric","numeric","factor","factor", "numeric","numeric",  
 "numeric","character","factor","factor")  
)  
checkin.global$datetime = strptime( strtrim(checkin.global$localtime,19), format="%Y-%m-%d %H:%M:%S")  
  
## weather data   
weather = read.csv( paste0(basedir, "data\\weather.csv"),   
 header=TRUE, sep=",", na.strings = c("-9999","Unknown"),  
 colClasses = c("numeric","numeric","numeric","character","numeric","factor",  
 "numeric","numeric","numeric","numeric","numeric","numeric",  
 "numeric","numeric")  
)  
## deal with logical data  
weather$fog=as.logical(weather$fog)  
weather$rain=as.logical(weather$rain)  
weather$snow=as.logical(weather$snow)  
weather$thunder=as.logical(weather$thunder)  
weather$tornado=as.logical(weather$tornado)  
  
## join checkin data with weather data based on timestamps   
checkin.global = joindfsbytime(checkin.global, weather)  
  
## deal with time   
checkin.global$hour = as.factor(format(checkin.global$datetime,"%H"))  
checkin.global$yearday = format(checkin.global$datetime,"%j")  
checkin.global$weekday = format(checkin.global$datetime,"%w")  
checkin.global$isweekend = as.factor(ifelse( ( checkin.global$weekday>5 ), "Saturday",   
 ifelse( ( checkin.global$weekday<1 ),"Sunday","Workday")))  
  
## add record for last checkin  
# checkin.global = copylastcheckinrec(checkin.global)  
checkin.global = checkin.global[complete.cases(checkin.global$conds),]

Again, chi-square test:

# venue\_cate v.s. hour of day  
cate.hour = xtabs(~hour+cate\_l1, data=checkin.global)  
chisq.test(cate.hour)

##   
## Pearson's Chi-squared test  
##   
## data: cate.hour  
## X-squared = 132062, df = 207, p-value < 2.2e-16

# venue\_cate v.s. weekend/workday  
cate.weekend = xtabs(~isweekend+cate\_l2, data=checkin.global)  
chisq.test(cate.weekend)

## Warning: Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: cate.weekend  
## X-squared = 31567, df = 756, p-value < 2.2e-16

# venue\_cate v.s. weather condition  
cate.conds = xtabs(~conds+cate\_l1, data=checkin.global)  
cate.conds = as.table(cate.conds[rowSums(cate.conds)>0,colSums(cate.conds)>0])  
chisq.test(cate.conds)

## Warning: Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: cate.conds  
## X-squared = 3216, df = 117, p-value < 2.2e-16

#### interactions between factors  
# hour v.s. weekend/workday  
hour.isweekend = xtabs(~hour+isweekend, data=checkin.global)  
chisq.test(hour.isweekend)

##   
## Pearson's Chi-squared test  
##   
## data: hour.isweekend  
## X-squared = 25317, df = 46, p-value < 2.2e-16

# hour v.s. weather condition  
hour.conds = xtabs(~hour+conds, data=checkin.global)  
hour.conds = as.table(hour.conds[rowSums(hour.conds)>0,colSums(hour.conds)>0])  
chisq.test(hour.conds)

## Warning: Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: hour.conds  
## X-squared = 37243, df = 299, p-value < 2.2e-16

The results shows that there are some interactions between these factors. The hour-weekend interaction can be explained like this: the number of checkins at a given hour of a given weekday cannot be estimated by the marginal checkins in each hour and marginal checkins in Saturday, sunday and workday; they have differnt patterns of checkins across the hour.

Well, this seems to make things even more complicated. However, we still have to move on.

category with weather:

# the probability of checkin a category under certain weather conditon  
cate.conds.p=t(apply(cate.conds, 1, function(x) x/sum(x) ) )  
  
#cate.conds.p.vec = as.vector(cate.conds.p)  
cate.conds.vec = as.vector(cate.conds)  
weather.factor = factor(rep(rownames(cate.conds),10))  
category.factor = factor(rep(colnames(cate.conds),each=14))  
  
#summary(aov(cate.conds.p.vec~weather.factor))  
summary(aov(cate.conds.vec~weather.factor+category.factor))

## Df Sum Sq Mean Sq F value Pr(>F)   
## weather.factor 13 6.64e+09 5.11e+08 14.1 < 2e-16 \*\*\*  
## category.factor 9 1.27e+09 1.41e+08 3.9 0.00024 \*\*\*  
## Residuals 117 4.24e+09 3.62e+07   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

correspondence analysis

ref:<http://www.statmethods.net/advstats/ca.html>

library(ca)  
prop.table(cate.conds, 1) # row percentages

## cate\_l1  
## conds Arts & Entertainment College & University Event  
## Clear 0.101861 0.017618 0.005038  
## Fog 0.046559 0.025304 0.002024  
## Haze 0.093115 0.026233 0.004727  
## Heavy Rain 0.149035 0.013514 0.005792  
## Heavy Snow 0.000000 0.000000 0.000000  
## Light Freezing Rain 0.037037 0.004630 0.000000  
## Light Rain 0.117519 0.016183 0.003367  
## Light Snow 0.105657 0.013955 0.009220  
## Mostly Cloudy 0.101194 0.017465 0.005972  
## Overcast 0.107677 0.018207 0.004662  
## Partly Cloudy 0.114640 0.018503 0.005910  
## Rain 0.130571 0.017235 0.002263  
## Scattered Clouds 0.098943 0.018828 0.007371  
## Snow 0.073097 0.025952 0.005623  
## cate\_l1  
## conds Food Nightlife Spot Outdoors & Recreation  
## Clear 0.296071 0.124188 0.070688  
## Fog 0.348178 0.061741 0.058704  
## Haze 0.259020 0.094690 0.052151  
## Heavy Rain 0.266795 0.191120 0.030888  
## Heavy Snow 0.083333 0.333333 0.083333  
## Light Freezing Rain 0.180556 0.023148 0.101852  
## Light Rain 0.290757 0.148202 0.046432  
## Light Snow 0.272116 0.104909 0.070272  
## Mostly Cloudy 0.302727 0.109475 0.069002  
## Overcast 0.288940 0.120485 0.058022  
## Partly Cloudy 0.303785 0.120656 0.069403  
## Rain 0.307625 0.156511 0.041086  
## Scattered Clouds 0.311318 0.095573 0.078851  
## Snow 0.231834 0.042388 0.097751  
## cate\_l1  
## conds Professional & Other Places Residence Shop & Service  
## Clear 0.094189 0.028155 0.155553  
## Fog 0.153846 0.027328 0.169028  
## Haze 0.147708 0.024579 0.169608  
## Heavy Rain 0.084170 0.036680 0.106950  
## Heavy Snow 0.333333 0.083333 0.083333  
## Light Freezing Rain 0.245370 0.060185 0.199074  
## Light Rain 0.095254 0.030303 0.146356  
## Light Snow 0.112136 0.038874 0.146524  
## Mostly Cloudy 0.101028 0.023318 0.167293  
## Overcast 0.105399 0.027174 0.156648  
## Partly Cloudy 0.089137 0.023181 0.157310  
## Rain 0.070334 0.033078 0.135097  
## Scattered Clouds 0.100375 0.021566 0.172318  
## Snow 0.198529 0.035035 0.147059  
## cate\_l1  
## conds Travel & Transport  
## Clear 0.106640  
## Fog 0.107287  
## Haze 0.128171  
## Heavy Rain 0.115058  
## Heavy Snow 0.000000  
## Light Freezing Rain 0.148148  
## Light Rain 0.105626  
## Light Snow 0.126339  
## Mostly Cloudy 0.102527  
## Overcast 0.112787  
## Partly Cloudy 0.097474  
## Rain 0.106198  
## Scattered Clouds 0.094857  
## Snow 0.142734

prop.table(cate.conds, 2) # column percentages

## cate\_l1  
## conds Arts & Entertainment College & University Event  
## Clear 4.715e-01 4.731e-01 4.772e-01  
## Fog 8.334e-04 2.628e-03 7.416e-04  
## Haze 2.141e-02 3.500e-02 2.225e-02  
## Heavy Rain 6.993e-03 3.679e-03 5.562e-03  
## Heavy Snow 0.000e+00 0.000e+00 0.000e+00  
## Light Freezing Rain 1.449e-04 1.051e-04 0.000e+00  
## Light Rain 3.921e-02 3.133e-02 2.299e-02  
## Light Snow 7.682e-03 5.887e-03 1.372e-02  
## Mostly Cloudy 7.705e-02 7.716e-02 9.307e-02  
## Overcast 2.569e-01 2.521e-01 2.277e-01  
## Partly Cloudy 5.905e-02 5.529e-02 6.229e-02  
## Rain 1.359e-02 1.041e-02 4.820e-03  
## Scattered Clouds 4.256e-02 4.699e-02 6.489e-02  
## Snow 3.062e-03 6.307e-03 4.820e-03  
## cate\_l1  
## conds Food Nightlife Spot Outdoors & Recreation  
## Clear 4.862e-01 4.972e-01 5.179e-01  
## Fog 2.211e-03 9.560e-04 1.663e-03  
## Haze 2.114e-02 1.884e-02 1.899e-02  
## Heavy Rain 4.442e-03 7.757e-03 2.294e-03  
## Heavy Snow 6.429e-06 6.269e-05 2.868e-05  
## Light Freezing Rain 2.507e-04 7.836e-05 6.310e-04  
## Light Rain 3.442e-02 4.277e-02 2.452e-02  
## Light Snow 7.020e-03 6.598e-03 8.088e-03  
## Mostly Cloudy 8.179e-02 7.210e-02 8.317e-02  
## Overcast 2.446e-01 2.487e-01 2.192e-01  
## Partly Cloudy 5.552e-02 5.375e-02 5.658e-02  
## Rain 1.136e-02 1.409e-02 6.768e-03  
## Scattered Clouds 4.751e-02 3.556e-02 5.369e-02  
## Snow 3.446e-03 1.536e-03 6.482e-03  
## cate\_l1  
## conds Professional & Other Places Residence Shop & Service  
## Clear 4.585e-01 5.010e-01 4.788e-01  
## Fog 2.897e-03 1.881e-03 2.012e-03  
## Haze 3.573e-02 2.173e-02 2.594e-02  
## Heavy Rain 4.154e-03 6.617e-03 3.337e-03  
## Heavy Snow 7.622e-05 6.966e-05 1.205e-05  
## Light Freezing Rain 1.010e-03 9.055e-04 5.180e-04  
## Light Rain 3.342e-02 3.887e-02 3.247e-02  
## Light Snow 8.575e-03 1.087e-02 7.084e-03  
## Mostly Cloudy 8.091e-02 6.826e-02 8.470e-02  
## Overcast 2.645e-01 2.493e-01 2.486e-01  
## Partly Cloudy 4.829e-02 4.590e-02 5.388e-02  
## Rain 7.699e-03 1.323e-02 9.349e-03  
## Scattered Clouds 4.541e-02 3.566e-02 4.929e-02  
## Snow 8.747e-03 5.642e-03 4.096e-03  
## cate\_l1  
## conds Travel & Transport  
## Clear 4.789e-01  
## Fog 1.863e-03  
## Haze 2.860e-02  
## Heavy Rain 5.239e-03  
## Heavy Snow 0.000e+00  
## Light Freezing Rain 5.625e-04  
## Light Rain 3.419e-02  
## Light Snow 8.913e-03  
## Mostly Cloudy 7.575e-02  
## Overcast 2.611e-01  
## Partly Cloudy 4.871e-02  
## Rain 1.072e-02  
## Scattered Clouds 3.959e-02  
## Snow 5.801e-03

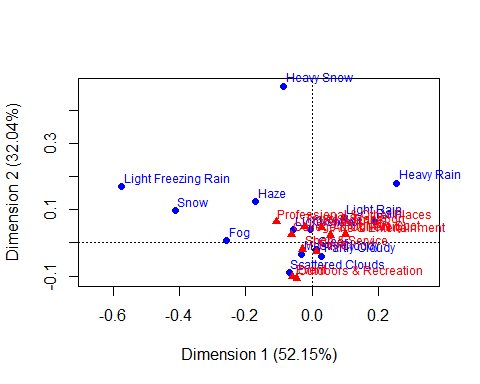
fit <- ca(cate.conds)  
print(fit) # basic results

##   
## Principal inertias (eigenvalues):  
## 1 2 3 4 5 6 7   
## Value 0.003175 0.00195 0.000545 0.000175 8.4e-05 6.3e-05 5.7e-05  
## Percentage 52.16% 32.04% 8.95% 2.87% 1.38% 1.03% 0.94%   
## 8 9   
## Value 2.7e-05 1.1e-05  
## Percentage 0.44% 0.18%   
##   
##   
## Rows:  
## Clear Fog Haze Heavy Rain Heavy Snow  
## Mass 0.483522 0.001870 0.024025 0.004902 0.000023  
## ChiDist 0.028982 0.327464 0.215717 0.323466 1.211348  
## Inertia 0.000406 0.000201 0.001118 0.000513 0.000033  
## Dim. 1 0.214951 -4.584354 -3.037591 4.503458 -1.534756  
## Dim. 2 -0.422001 0.128074 2.814400 4.071209 10.678418  
## Light Freezing Rain Light Rain Light Snow Mostly Cloudy Overcast  
## Mass 0.000409 0.03485 0.007595 0.079543 0.249277  
## ChiDist 0.694485 0.12563 0.137614 0.056361 0.042680  
## Inertia 0.000197 0.00055 0.000144 0.000253 0.000454  
## Dim. 1 -10.182340 1.71823 -0.989541 -0.584982 -0.092711  
## Dim. 2 3.833152 1.70139 0.886942 -0.785304 0.904782  
## Partly Cloudy Rain Scattered Clouds Snow  
## Mass 0.053804 0.010871 0.04493 0.004376  
## ChiDist 0.064818 0.203850 0.11931 0.453735  
## Inertia 0.000226 0.000452 0.00064 0.000901  
## Dim. 1 0.512216 3.318525 -1.21399 -7.329588  
## Dim. 2 -0.925001 1.465850 -2.01617 2.226647  
##   
##   
## Columns:  
## Arts & Entertainment College & University Event Food  
## Mass 0.104464 0.018005 0.005104 0.294412  
## ChiDist 0.069371 0.089178 0.167862 0.034082  
## Inertia 0.000503 0.000143 0.000144 0.000342  
## Dim. 1 0.968852 -1.102140 -1.050339 0.221491  
## Dim. 2 0.503647 0.543120 -2.320821 -0.616601  
## Nightlife Spot Outdoors & Recreation Professional & Other Places  
## Mass 0.120769 0.065992 0.099320  
## ChiDist 0.106456 0.127843 0.125242  
## Inertia 0.001369 0.001079 0.001558  
## Dim. 1 1.790745 -0.819744 -1.895650  
## Dim. 2 0.578032 -2.427488 1.445576  
## Residence Shop & Service Travel & Transport  
## Mass 0.02717 0.157100 0.107664  
## ChiDist 0.09787 0.043942 0.059912  
## Inertia 0.00026 0.000303 0.000386  
## Dim. 1 0.52009 -0.539896 -0.412600  
## Dim. 2 1.05165 -0.442939 1.103566

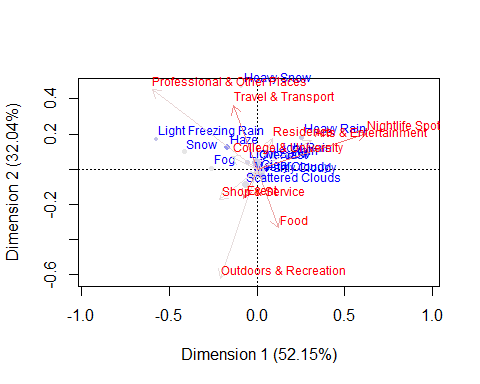
summary(fit) # extended results

##   
## Principal inertias (eigenvalues):  
##   
## dim value % cum% scree plot   
## 1 0.003175 52.2 52.2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
## 2 0.001950 32.0 84.2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## 3 0.000545 8.9 93.1 \*\*\*\*   
## 4 0.000175 2.9 96.0 \*   
## 5 8.4e-050 1.4 97.4 \*   
## 6 6.3e-050 1.0 98.4   
## 7 5.7e-050 0.9 99.4   
## 8 2.7e-050 0.4 99.8   
## 9 1.1e-050 0.2 100.0   
## -------- -----   
## Total: 0.006087 100.0   
##   
##   
## Rows:  
## name mass qlt inr k=1 cor ctr k=2 cor ctr   
## 1 | Cler | 484 588 67 | 12 175 22 | -19 414 86 |  
## 2 | Fog | 2 622 33 | -258 622 39 | 6 0 0 |  
## 3 | Haze | 24 961 184 | -171 629 222 | 124 332 190 |  
## 4 | HvyR | 5 924 84 | 254 615 99 | 180 309 81 |  
## 5 | HvyS | 0 157 5 | -86 5 0 | 472 152 3 |  
## 6 | LgFR | 0 742 32 | -574 682 42 | 169 59 6 |  
## 7 | LghR | 35 952 90 | 97 594 103 | 75 358 101 |  
## 8 | LghS | 8 245 24 | -56 164 7 | 39 81 6 |  
## 9 | MstC | 80 721 42 | -33 342 27 | -35 379 49 |  
## 10 | Ovrc | 249 891 75 | -5 15 2 | 40 876 204 |  
## 11 | PrtC | 54 595 37 | 29 198 14 | -41 397 46 |  
## 12 | Rain | 11 942 74 | 187 841 120 | 65 101 23 |  
## 13 | SctC | 45 886 105 | -68 329 66 | -89 557 183 |  
## 14 | Snow | 4 875 148 | -413 828 235 | 98 47 22 |  
##   
## Columns:  
## name mass qlt inr k=1 cor ctr k=2 cor ctr   
## 1 | ArtE | 104 722 83 | 55 619 98 | 22 103 26 |  
## 2 | CllU | 18 557 24 | -62 485 22 | 24 72 5 |  
## 3 | Evnt | 5 497 24 | -59 124 6 | -102 373 27 |  
## 4 | Food | 294 772 56 | 12 134 14 | -27 638 112 |  
## 5 | NghS | 121 956 225 | 101 898 387 | 26 58 40 |  
## 6 | OtdR | 66 834 177 | -46 131 44 | -107 703 389 |  
## 7 | PrOP | 99 987 256 | -107 727 357 | 64 260 208 |  
## 8 | Rsdn | 27 315 43 | 29 90 7 | 46 225 30 |  
## 9 | ShpS | 157 677 50 | -30 479 46 | -20 198 31 |  
## 10 | TrvT | 108 812 63 | -23 151 18 | 49 662 131 |

plot(fit) # symmetric map



plot(fit, mass = TRUE, contrib = "absolute", map =  
 "rowgreen", arrows = c(FALSE, TRUE)) # asymmetric map



#### 2. (temporal weighted) sequential factor

Now consider the sequential factor. The hypothesis is two categories can be more connected if they are usually checked in consecutively. Basically, this should be done by 2nd level category, right? because the 1st level cannot give too much information.

The foundation approach here is Markov chain.

first, we should create a data frame that describe the sequences:

sequence.list = lapply(split(checkin.global, checkin.global$user\_id), function(i){  
 if(nrow(i)>30)  
 copylastcheckinrec(i, samesize=FALSE)  
})  
sequence.list[sapply(sequence.list, is.null)] = NULL  
sequence.global <- as.data.frame( do.call("rbind", sequence.list) )  
sequence.global$weight <- exp( -2 \* sequence.global$time\_diff / 60 )  
#temp <- lapply(sequence.list, function(i) sequence.global<<-rbind(sequence.global, i))

Note: do.call("rbind", sequence.list) combines all the data frame in the list into a data frame.

markov chain is based on contingency table; we'd like to do similarly, but the contingency table should be temporally weighted:

unique.cates <- sort(unique(sequence.global$last\_cate))  
levels.cates <- levels(checkin.global$cate\_l2)  
transition.matrix <- matrix(0,nrow=length(unique.cates), ncol=length(unique.cates),   
 dimnames=list(levels.cates[unique.cates],  
 levels.cates[unique.cates]));  
temp = lapply(split(sequence.global, sequence.global$last\_cate), function(lst){  
 total.sum <- sum(lst$weight)  
 transition.matrix[ levels.cates[unique(lst$last\_cate)],   
 levels.cates[sort(unique(lst$cate))] ] <<-   
 sapply( split(lst, lst$cate), function(cur){   
 if(total.sum!=0) { p = sum(cur$weight) / total.sum }  
 else { p = 1/length(unique(lst$cate)) } ## for all zeros  
 p  
 })  
 NA  
})  
rm(temp)

# just in case:  
# save(transition.matrix,file="D:\\Experiments\\R\\data\\transition.matrix\_0814.Rda")  
load("D:\\Experiments\\R\\data\\transition.matrix\_0814.Rda")  
library(markovchain)  
mcSequence <- new("markovchain",  
 states = levels.cates[unique(sequence.global$last\_cate)],  
 transitionMatrix = transition.matrix,  
 name = "sequence")  
predict(object = mcSequence, newdata = c("American Restaurant"),n.ahead = 3)