Predicting Household Composition with TV Viewing Data

Generating Features of TV Viewing

Rafael Lüchinger

11 Februar 2019

Introduction

TV audiance in Switzerland is measured by Mediapulse AG. A representative panel of roughly 2000 households is constantly under measurement. These homes were carfully selected by a complex sampling design and all householdmembers have agreed to be part of the study. The TV viewing of each householdmember is individually recorded and detailed demografics are known for each person. This allows the market to target TV audiances by relevant characteristics like age gender and many more.

One issue with the panel approach is poor granularity. That means sometimes the system can not provide any audiance figures for a specific channel or airtime. It is likely that in the Swiss population of about 3.5 Mio. households at least a few people are watching even exotic programs at exotic times of the day. However, out of a panel of 2000 households chances are high that no one was watching that content. This is not a bias of the measurement but poor resolution.

A solution to this problem could be the inclusion of third party data. Set-Top-Boxes (STB) of TV-provider (Swisscom, UPC, etc.) are automatically recording the TV consumption in millions of Swiss homes and the data is returned to the providers servers (return path data, RPD). There are still many issues with these data that are currently adressed.

One major issue of RPD is that the viewing data is on household level, not on indvidual level. Household-level data is of little use to the market. Because it gives no insight in target groups based on age and gender and alike.

It is unlikely that RPD provider will ever measure the individual viewing or survey individual demografics within the subscribers homes. Apart from region code, the only information about the home is the viwing data itself. So the question arises if it is possible to predict the household composition based on viewing behavior.

The aim of this study is to explore the possiblity to predict the household composition within a household using TV viewing data. It seems to be a two-step-problem, first to find the number of householdmembers and then to assign age and gender to the individuals.

We will use the *Mediapulse TV-Panel* and its viewing data to study the subject. For all households in the panel its composition including household size and age and sex of each person is known. For each panel home the viewing data will be aggregated to household level. Different supervised machine learning algorithms will be fed with features extracted from that houesehold viewing data.

Target: Household Composition

Data Import

A R-Package `tv` is used to import the raw data of the *Mediapulse-TV-Panel*. The setup functions allows to specify the data to be read from disk into R. The import functions by default returns three data.tables:

1. `dem`: all individuals with their demografics
2. `view`: the TV viewing
3. `prog`: the program timetable with genre information

Date Range

TV viewing behavior is known to differ by season as well as by weekdays. During cold months TV viwing is more popular than during Sommer months. Similarly, on weekends people watch more TV than during the rest of the week. Not only differs the total viewing duration. Also the individual preferences for channels and programs might differ between weekend and workdays.

To extract freatures of TV viewing, we will consider a range of 8 weeks during 2017. This should be long enough to reflect the individual viewing patterns. We focus on cold months and a period free of holidays or special TV events (FIFA Wolrdcup, etc.). We make sure to get an equal number of each weekday.

```
dayx <- as.Date('2017-11-12')
days <- seq(dayx - 28, dayx + 27, by = "day")
table(weekdays(days))
```

```
##
##   Dienstag Donnerstag    Freitag   Mittwoch     Montag    Samstag
##          8          8          8          8          8          8
##    Sonntag
##          8
```

Household Composition

A Household is not necessarily under measurement on every day of our 8 weeks. Sometimes a household leaves the panel, then a new household will join the panel. Also for technical reasons a household may drop the panel just for a couple of days.

To create a dataset of households not all households within the 8 weeks were included. Rather, the sample of a single specific day is choosen. This day is a sunday and exactly in the middle of the 8 weeks. The difference is small anyway, e.g. routhly a 2000 versus 2100 households.

```
library('tv')
id <- setup(days, obs = "ind", dem.var = c("sg","hhsize","age","sex"),
            dem.day = dayx, dem.uni = FALSE, view = FALSE, prg = FALSE)
import(id)
```

```
(dem <- dem[(!guest)]) # excluding guests
```

```
##              day   hh ind    pin weight guest age sex hw sg hhsize
##    1: 2017-11-12    6   1    601 1.0831 FALSE  70   1  2  2      2
##    2: 2017-11-12    6   2    602 1.3365 FALSE  67   2  1  2      2
##    3: 2017-11-12    9   1    901 0.9552 FALSE  55   1  2  2      4
##    4: 2017-11-12    9   2    902 0.9935 FALSE  50   2  1  2      4
##    5: 2017-11-12    9   4    904 1.5404 FALSE  21   1  2  2      4
##   ---
## 4384: 2017-11-12 6201   2 620102 1.8489 FALSE  73   2  1  1      2
## 4385: 2017-11-12 6204   1 620401 0.7449 FALSE  52   1  2  2      4
## 4386: 2017-11-12 6204   2 620402 0.9444 FALSE  47   2  1  2      4
```

```
## 4387: 2017-11-12 6204   3 620403 1.2214 FALSE  17   1  2  2      4
## 4388: 2017-11-12 6204   4 620404 0.9262 FALSE  13   2  2  2      4
```

On our sample day 2017-11-12 the TV-Panel was formed by 2006 households and 4388 individuals living in these households. This gives a average householdsize of 2.19.

A note on the variale *hhsize*. Household size is not constant over time, the number of people living in a household can change by natural reasons like birth, death, moving in or out. Also the variable *hhsize* is not necessarily equal to the sum of individuals for the following reasons:

- babys 0-2 years old are not part of the panel
- guests are part of the data but not counted for household size
- houshold size is counted 1, 2, ..., 5+, 5+ meaning housholds with 5 or more members

A simple transformation of the `dem` data.table presents for each household on a row its composition. There are 2006 households. We call the household ID `pin` and `sg` is the linguistic region (german, french, italian). Age and Sex of up to 8 householdmembers. There is no missing data.

```
hh <- dcast(dem, day + hh + sg + hhsize ~ ind, value.var = c("age","sex"), fill = 0L)
setnames(hh, 'hh', 'pin')
rm(id, dem)
hh
```

```
##              day  pin sg hhsize age_1 age_2 age_3 age_4 age_5 age_6 age_7
##    1: 2017-11-12    6  2      2    70    67     0     0     0     0     0
##    2: 2017-11-12    9  2      4    55    50     0    21    17     0     0
##    3: 2017-11-12   14  1      2    71    72     0     0     0     0     0
##    4: 2017-11-12   20  1      2    59    49     0     0     0     0     0
##    5: 2017-11-12   21  1      2    63    52     0     0     0     0     0
##   ---
## 2002: 2017-11-12 6196  1      2    74    76     0     0     0     0     0
## 2003: 2017-11-12 6197  1      2    40    47     0     0     0     0     0
## 2004: 2017-11-12 6200  1      2    63    72     0     0     0     0     0
## 2005: 2017-11-12 6201  1      2    71    73     0     0     0     0     0
## 2006: 2017-11-12 6204  2      4    52    47    17    13     0     0     0
##       age_8 sex_1 sex_2 sex_3 sex_4 sex_5 sex_6 sex_7 sex_8
##    1:     0     1     2     0     0     0     0     0     0
##    2:     0     1     2     0     1     2     0     0     0
##    3:     0     1     2     0     0     0     0     0     0
##    4:     0     1     2     0     0     0     0     0     0
##    5:     0     1     2     0     0     0     0     0     0
##   ---
## 2002:     0     2     1     0     0     0     0     0     0
## 2003:     0     2     2     0     0     0     0     0     0
## 2004:     0     2     1     0     0     0     0     0     0
## 2005:     0     1     2     0     0     0     0     0     0
## 2006:     0     1     2     1     2     0     0     0     0
```

Features: Viewing Behavior

Data Import

We use our knowledge and intuiton about TV viewing to generate features we believe would carry information about the household composition.

1. Dimension time
    - Weekend vs. Workingdays
    - daytime
2. Dimension content (genre)
    - type of channel
    - type of program

To split the data by weekend vs workingdays we do not specify anything but later simply use the date variable.

To split the data by daytime the `tv` package allows us to specify so called timebands.

We are interested in the viewing on household level. The `tv` package allows to specify this with the `setup(obs = "hh")`. Simply summing up all individuals viewing within a household doe not mean household level. If people watch together than this viewing counts only once.

```
id <- setup(
  day = days,
  guest = FALSE,
  obs = "hh",
  dem.var = "sg",
  tmb = list(
    '02to06' = c(start = '02:00:00', end = '05:59:59'),
    '06to08' = c(start = '06:00:00', end = '07:59:59'),
    '08to11' = c(start = '08:00:00', end = '10:59:59'),
    '11to13' = c(start = '11:00:00', end = '12:59:59'),
    '13to17' = c(start = '13:00:00', end = '16:59:59'),
    '17to20' = c(start = '17:00:00', end = '19:59:59'),
    '20to22' = c(start = '20:00:00', end = '21:59:59'),
    '22to24' = c(start = '22:00:00', end = '23:59:59'),
    '24to02' = c(start = '24:00:00', end = '25:59:59')
  )
  )
import(id)
```

sum viewing by weekpart and daytime

The 'tv' package provides a function 'dem.add' to add weekday and more calendar variables given the date in a column.

```
dem.add(dem, 'calendar')
nday <- dem[, .(N = uniqueN(day)), k=.(pin, wend)]
```

```
res <- calc(
  dt = view[dem, on=c('day',"pin")],
  by = c("day","tmb","pin"),
  period = "tmb.dur"
)

dem.add(res, 'calendar')

res <- res[, sum(dur), k=c("wend","tmb","pin")]
```

```
res[, nday := nday[res, on=c('wend','pin'), N]]
# na.omit(res, invert = TRUE)
res[, mean.dur := V1 / nday]
res[, wend := id$lab$wend[res, on='wend', label]]

X.tmb <- dcast(res, pin ~ wend + tmb, value.var = "mean.dur")
X.tmb <- X.tmb[hh[,.(pin)], on="pin"]
na.to.0(X.tmb) # na.omit(X.tmb, invert = TRUE)
setnames(X.tmb, -1, paste0("day_",tolower(names(X.tmb)[-1])))
```

sum viewing by channel groups

The tv package provides a mapping table for channels including channel names but also information on the type (genre) country of origin and language of each channel

```
id$lab$sta
```

```
cols <- c("chn.type","chn.country","chn.lang")
view[, (cols) := id$lab$sta[view, on=c(id="base"), mget(cols)]]
ordercol(view, cols, "chn.name")
# view[, any(is.na(chn.type))]
```

channel types

```
res.type <- calc(
  dt = view[dem, on=c('day',"pin")],
  by = c("day","pin","chn.type"),
  period = "day"
)
res.type <- res.type[, sum(dur), k=c("chn.type","pin")]
res.type[, nday := nday[, .(N=sum(N)), k=pin][res.type, on='pin', N]]
# na.omit(res.type, invert = TRUE)
res.type[, mean.dur := V1 / nday]

X.chn.type <- dcast(res.type, pin ~ chn.type, value.var = "mean.dur")
X.chn.type <- X.chn.type[hh[,.(pin)], on="pin"]
na.to.0(X.chn.type) # na.omit(X.chn.type, invert = TRUE)
```

channel country of origin

```
res.country <- calc(
  dt = view[dem, on=c("day","pin")],
  by = c("day","pin","chn.country"),
  period = "day"
)
res.country <- res.country[, sum(dur), k=c("chn.country","pin")]
res.country[, nday := nday[, .(N=sum(N)), k=pin][res.country, on='pin', N]]
# na.omit(res.country, invert = TRUE)
```

```
res.country[, mean.dur := V1 / nday]

X.chn.country <- dcast(res.country, pin ~ chn.country, value.var = "mean.dur")
X.chn.country <- X.chn.country[hh[,.(pin)], on="pin"]
na.to.0(X.chn.country) # na.omit(X.chn.country, invert = TRUE)
```

channel language

```
res.lang <- calc(
  dt = view[dem, on=c("day","pin")],
  by = c("day","pin","chn.lang"),
  period = "day"
)
res.lang <- res.lang[, sum(dur), k=c("chn.lang","pin")]
res.lang[, nday := nday[, .(N=sum(N)), k=pin][res.lang, on='pin', N]]
# na.omit(res.lang, invert = TRUE)
res.lang[, mean.dur := V1 / nday]

X.chn.lang <- dcast(res.lang, pin ~ chn.lang, value.var = "mean.dur")
X.chn.lang <- X.chn.lang[hh[,.(pin)], on="pin"]
na.to.0(X.chn.lang) # na.omit(X.chn.lang, invert = TRUE)
```

put together channel features

```
# these are nested left joins:
X.chn <- X.chn.type[,-"0"][X.chn.country[,-"0"][X.chn.lang[,-"0"], on="pin"], on="pin"]
setnames(X.chn, -1, paste0("chn_",tolower(names(X.chn)[-1])))
```

sum viewing by program genre

The 'tv' package provides a function `overlap.join` for joined overlap of time segements. This splits all viewing statements by the corresponding program schedule by day and by channel.

```
view <- overlap.join(view, prog, type='prg')

res.genre <- calc(
  dt = view[dem, on=c("day","pin")],
  by = c("day","pin","genre"),
  period = "day"
)
res.genre <- res.genre[, sum(dur), k=c("genre","pin")]
res.genre[, nday := nday[, .(N=sum(N)), k=pin][res.genre, on='pin', N]]
# na.omit(res.genre, invert = TRUE)
res.genre[, mean.dur := V1 / nday]

X.genre <- dcast(res.genre, pin ~ genre, value.var = "mean.dur")
X.genre <- X.genre[hh[,.(pin)], on="pin"]
na.to.0(X.genre) # na.omit(X.genre, invert = TRUE)
setnames(X.genre, -1, paste0("prg_",tolower(names(X.genre)[-1])))
```

combine all Features in one data.table

```
predictors <- X.tmb[X.chn[X.genre, on="pin"], on="pin"] # nested left joins
setnames(predictors, 'pin', 'hh')
setorder(predictors, 'hh')
```

View Features

```
predictors[, 1:4]
```

```
##         hh day_weekend_02to06 day_weekend_06to08 day_weekend_08to11
##    1:    6             0.00000              0.0000         372.875000
##    2:    9            88.31250             20.7500         621.500000
##    3:   14           328.12500             39.2500          12.000000
##    4:   20          1019.66667            555.1333         824.466667
##    5:   21           607.37500            917.2500        3143.500000
##   ---
## 2002: 6196             0.00000              0.0000         907.125000
## 2003: 6197             0.00000              0.0000           8.222222
## 2004: 6200             0.00000              0.0000          48.363636
## 2005: 6201           102.33333              0.0000         492.111111
## 2006: 6204            38.88889             50.0000        2802.888889
```

```
matrix(names(predictors), ncol = 3)
```

```
##       [,1]                [,2]                     [,3]
##  [1,] "hh"                "day_workday_24to02"     "chn_french"
##  [2,] "day_weekend_02to06" "chn_arts"               "chn_german"
##  [3,] "day_weekend_06to08" "chn_generalistprivate" "chn_italian"
##  [4,] "day_weekend_08to11" "chn_generalistpublic"  "chn_other"
##  [5,] "day_weekend_11to13" "chn_kids"               "prg_commercial"
##  [6,] "day_weekend_13to17" "chn_livestileindoor"   "prg_info"
##  [7,] "day_weekend_17to20" "chn_livestileoutdoor"  "prg_kids"
##  [8,] "day_weekend_20to22" "chn_local"              "prg_missing"
##  [9,] "day_weekend_22to24" "chn_movieseries"       "prg_movie"
## [10,] "day_weekend_24to02" "chn_music"              "prg_music"
## [11,] "day_workday_02to06" "chn_nature"             "prg_news"
## [12,] "day_workday_06to08" "chn_news"               "prg_other"
## [13,] "day_workday_08to11" "chn_paytv"              "prg_series"
## [14,] "day_workday_11to13" "chn_religion"           "prg_service"
## [15,] "day_workday_13to17" "chn_sport"              "prg_show"
## [16,] "day_workday_17to20" "chn_foreign"            "prg_sport"
## [17,] "day_workday_20to22" "chn_swiss"              "prg_talk"
## [18,] "day_workday_22to24" "chn_english"            "prg_trailer"
```

Export Data

```r
hh.composition <- hh[, day := NULL]
setnames(hh.composition, 'pin', 'hh')
setorder(hh.composition, 'hh')

save(hh.composition, predictors, file = '~/diplom/data/data_predictors.RData')
```

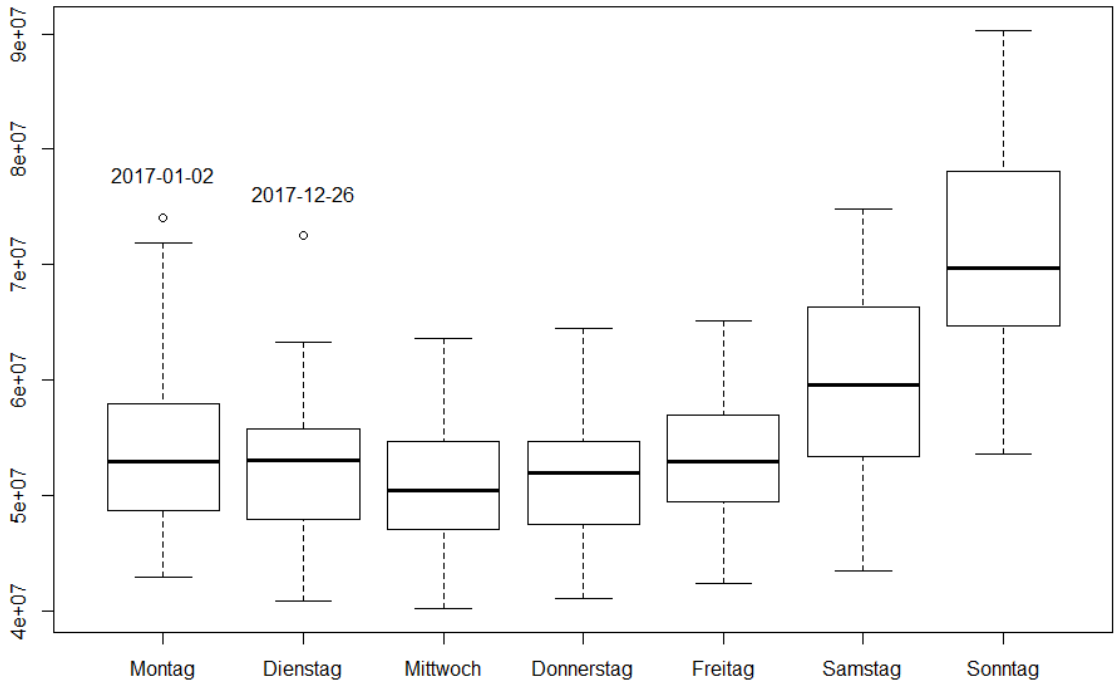

Figure 1: Amount of TV viewing by weekdays during 2017. More viewing on weekends, festival days behave like sundays

Appendix

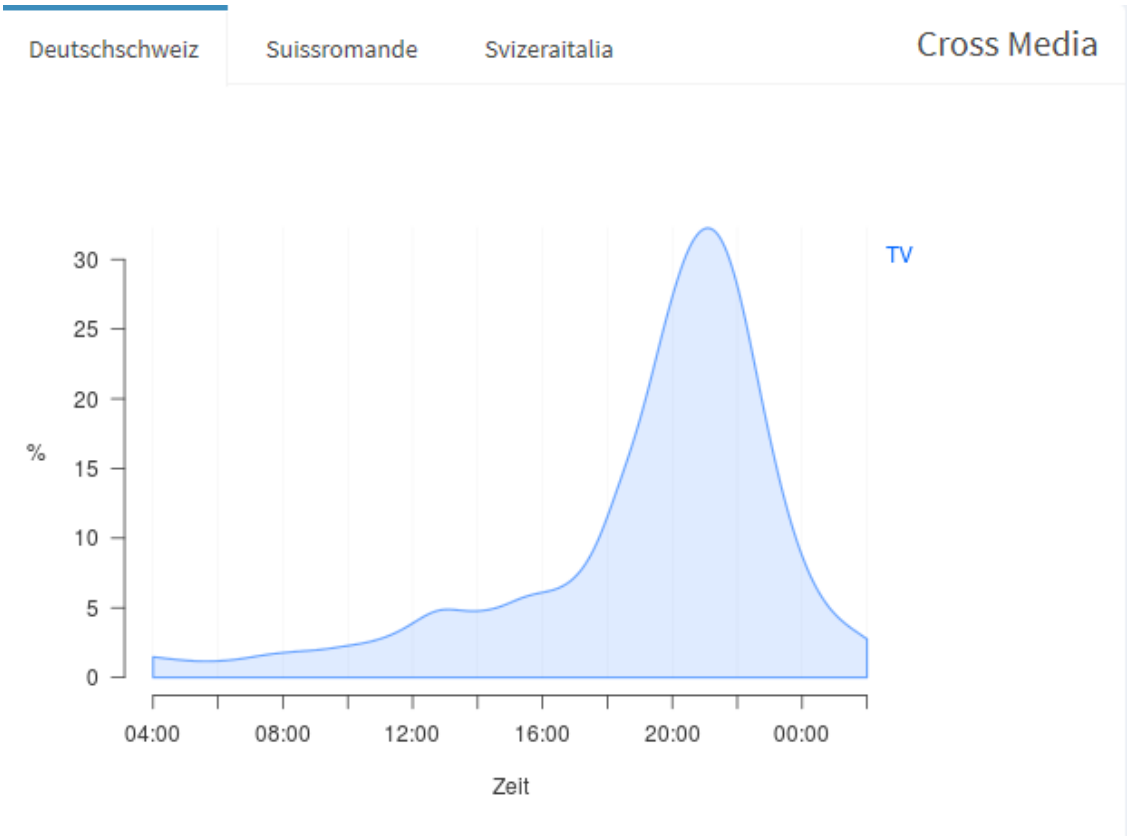

Figure 2: Relative amount of TV viewing during a day. Averaged across one Year.