Introduction to REMs (student)

Bianchi, Boschi, Filippi-Mazzola, Juozaitiene, Wit

2024-06-20

Demo 1: Simulation of a small relational event process

In the lecture, we encountered a simple relational event process that consisted of 5 people asking for two reference letters from among the group. It was assumed that the *reference letter request* process depended merely on the difference in work experience (WE) between the person s making the request and the person requested r,

$$\lambda_{sr}(t) = Y_{sr}(t)e^{-1.5|WE_s - WE_r|}.$$

where $Y_{sr}(t)$ was either 1, if s had not yet received two reference letters, and 0 otherwise.

```r
set.seed(2972)

#number of individuals
p<-5

#interest in cars
WE<-c(0,3,5,9,11)
if (length(WE)!=p){stop()}

needed<-2

#collected letters so far
n.letters<-rep(0,p)

#risk set
riskset<-cbind(rep(1:p,each=p),rep(1:p,p))
riskset<-riskset[riskset[,1]!=riskset[,2],]


# dat
dat<-NULL
tm<-0

while (min(n.letters)<needed){
  #hazard
  hazard<- exp(-1.5*abs(WE[riskset[,1]]-WE[riskset[,2]]))

  #sample new event time
  tm<-tm+rexp(1,sum(hazard))
  event.id<-sample(length(hazard),1, prob=hazard/sum(hazard))
  dat<-rbind(dat,c(tm,riskset[event.id,]))
  #update number of reference letters
  n.letters[riskset[event.id,1]]<-n.letters[riskset[event.id,1]]+1
  if (n.letters[riskset[event.id,1]]==needed){
    riskset<-riskset[riskset[,1]!=riskset[event.id,1],]
  } else {
    riskset<-riskset[-event.id,]
  }
}
colnames(dat)<-c("tm","solicitor","receiver")
dat
```

```
##                tm solicitor receiver
##  [1,]   6.020988         2        3
##  [2,]  12.112763         1        2
##  [3,]  14.732286         2        1
##  [4,]  22.733545         4        5
##  [5,]  27.228996         5        4
##  [6,]  31.093684         3        2
##  [7,]  62.156751         4        3
##  [8,] 266.940177         1        3
##  [9,] 683.657476         3        4
## [10,] 955.662792         5        3
```

```
require(igraph)
```

```
## Loading required package: igraph
```

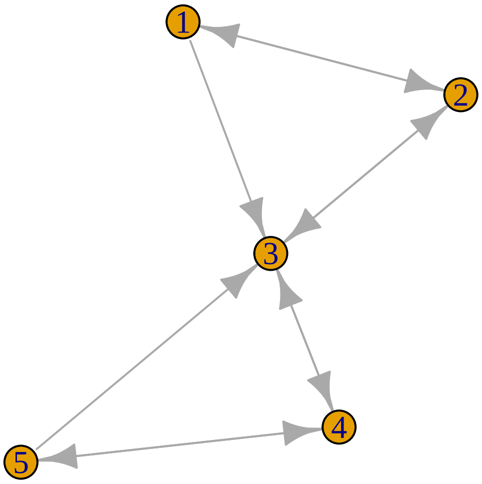
```
## Warning: package 'igraph' was built under R version 4.3.1
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(igraph)
g<-graph_from_edgelist(dat[,2:3])
plot(g)
```



So, person 2 asks person 3 for a letter after only 6 minutes, whereas it takes person 5 more almost 16 hours (i.e., 956 minutes) to ask person 3 for a reference letter.

Demo 2: Inference of REM

Step 1: Sampling non-events

During the lecture, we saw that it is possible to estimate the parameters from the relational event model by sampling for each event a non-event. A non-event (s^*_k, r^*_k) at time t_k is defined as an event that did not happen at time t_k, but which could in principle have happened at that time point. In real data situations, you will get the event data, like the one shown above. Then after that, you have to go through each time point and randomly select one that could have happened, but did not. Then you join the data together in a single dataset.

```
n.event <- nrow(dat)
V <- sort(unique(c(dat[,2:3])))
p <- length(V)

#collected letters so far
n.letters<-rep(0,p)

#risk set
riskset<-cbind(rep(V,each=p),rep(V,p))
riskset<-riskset[riskset[,1]!=riskset[,2],]

#sampled data
sampled.dat <- NULL
```

```
for (i in 1:n.event){
  event.id<- which((riskset[,1]==dat[i,2])&(riskset[,2]==dat[i,3]))
  non.event.id<-sample(setdiff(1:nrow(riskset),event.id),1)
  sampled.dat<-rbind(sampled.dat,c(dat[i,],riskset[non.event.id,]))
  #update number of reference letters
  n.letters[riskset[event.id,1]]<-n.letters[riskset[event.id,1]]+1
  if (n.letters[riskset[event.id,1]]==needed){
    riskset<-riskset[riskset[,1]!=riskset[event.id,1],]
  } else {
    riskset<-riskset[-event.id,]
  }
}
colnames(sampled.dat)<-c("tm","solicitor","receiver","non.solicitor","non.receiver")
sampled.dat
```

```
##                 tm solicitor receiver non.solicitor non.receiver
##  [1,]   6.020988         2        3             2            1
##  [2,]  12.112763         1        2             1            3
##  [3,]  14.732286         2        1             5            4
##  [4,]  22.733545         4        5             1            3
##  [5,]  27.228996         5        4             4            3
##  [6,]  31.093684         3        2             3            1
##  [7,]  62.156751         4        3             3            1
##  [8,] 266.940177         1        3             5            2
##  [9,] 683.657476         3        4             3            5
## [10,] 955.662792         5        3             5            2
```

Step 2: Preparing the covariate and response for logistic regression

For the 10 observed relational events combined with the 10 sampled relational non-events, we can calculate the *sampled partial likelihood*,

$$L(\beta)=\prod_{i=1}^{10}\frac{e^{\beta|WE_{s_i}-WE_{r_i}|}}{e^{\beta|WE_{s_i}-WE_{r_i}|}+e^{\beta|WE_{s_i^*}-WE_{r_i^*}|}}=\prod_{i=1}^{10}\frac{e^{\beta\left(|WE_{s_i}-WE_{r_i}|-|WE_{s_i^*}-WE_{r_i^*}|\right)}}{1+e^{\beta\left(|WE_{s_i}-WE_{r_i}|-|WE_{s_i^*}-WE_{r_i^*}|\right)}}$$

This is exactly equal to the likelihood of a logistic regression

- with response equal to 1 for all 10 observations
- with covariate equal to $|WE_{s_i}-WE_{r_i}|-|WE_{s_i^*}-WE_{r_i^*}|$ $(i=1,...,10)$

```
WE.diff<-abs(WE[sampled.dat[,2]]-WE[sampled.dat[,3]]) - abs(WE[sampled.dat[,4]]-WE[sampled.dat[,5]])
one<-rep(1,length(WE.diff))
full.dat<-cbind(sampled.dat,WE.diff,one)
colnames(full.dat)<-c("tm","solicitor","receiver","non.solicitor","non.receiver","WE.diff","one")
full.dat<-as.data.frame(full.dat)
full.dat
```

```
##             tm solicitor receiver non.solicitor non.receiver WE.diff one
## 1     6.020988         2        3             2            1      -1   1
## 2    12.112763         1        2             1            3      -2   1
## 3    14.732286         2        1             5            4       1   1
## 4    22.733545         4        5             1            3      -3   1
## 5    27.228996         5        4             4            3      -2   1
## 6    31.093684         3        2             3            1      -3   1
## 7    62.156751         4        3             3            1      -1   1
## 8  266.940177          1         3              5             2       -3   1
## 9  683.657476          3         4              3             5       -2   1
## 10 955.662792          5         3              5             2       -2   1
```

Step 3: performing logistic regression

Given the data, we can estimate the parameter β by maximizing the sampled partial likelihood,

$$\hat{\beta} = \arg\max \prod_{i=1}^{10} \frac{e^{\beta|WE_{s_i}-WE_{r_i}|}}{e^{\beta|WE_{s_i}-WE_{r_i}|} + e^{\beta|WE_{s_i^*}-WE_{r_i^*}|}},$$

which involves using a logistic regression function in your favorite statistical software package.

```
require(mgcv)
```

```
## Loading required package: mgcv

## Warning: package 'mgcv' was built under R version 4.3.3

## Loading required package: nlme

## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
library(mgcv)
letter.lr <- gam(one ~ -1 + WE.diff, family = binomial,data=full.dat)
summary(letter.lr)
```

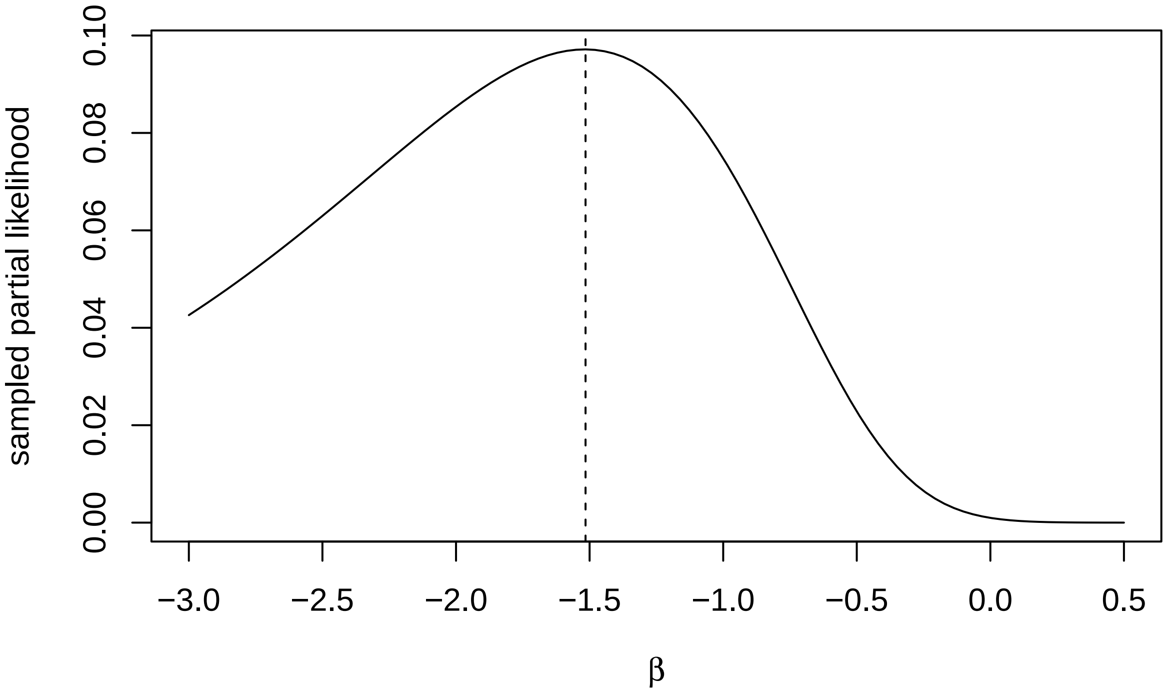
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## one ~ -1 + WE.diff
##
## Parametric coefficients:
##         Estimate Std. Error z value Pr(>|z|)
## WE.diff  -1.5177     0.8389  -1.809   0.0704 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = -0.3337  Scale est. = 1         n = 10
```

So, we find that the best estimate for the effect of work experience is equal to **-1.5177**. Note that this is very close to the -1.5 that we used to simulate the data.

PS. Check that we indeed found the maximum

In this simple case, there is only one parameter to estimate, so it is easy check visually that the `gam` function for logistic regression indeed found the maximum of the sampled partial likelihood.

```
beta<-seq(-3,.5,length=100)
lk<-sapply(beta,function(b,x){sum(x*b-log(1+exp(x*b)))},x=WE.diff)
plot(beta,exp(lk),type="l",xlab=expression(beta),ylab="sampled partial likelihood")
abline(v=beta[which.max(lk)],lty=2)
```



From the figure it is indeed clear that the maximum is found close to -1.5, as found by the `gam` function in the `mgcv` package that is able to fit general logistic regressions (and more). This function will be used throughout this tutorial.

Exercise 1. Classroom

Data for this exercise are based on streaming interaction data collected by Daniel McFarland on students in classrooms. Time-stamped interactions in each classroom were recorded, with information on the 'sender' and 'receiver' of the interaction, as well as the nature of the interaction. Interactions could be social or task-based, for example, although we focus just on social interactions in this exercise. We focus on the interactions that were recorded during one class hour in classroom 182.

There are 279 interactions between 17 predominantly female students and 1 male, black teacher. There is an even proportion of white and black students in the room.

```
load("00-Data/class.RData")
head(class$rev)
```

```
##   sender receiver  time
## 1     11        2 0.143
## 2      2       11 0.286
## 3      2        5 0.429
## 4      5        2 0.571
## 5      9        8 0.714
## 6      8        9 0.857
```

```
class$info
```

```
## $male
##  [1] 0 0 0 0 0 1 0 1 1 0 0 1 0 0 1 0 0 1
##
## $teacher
##  [1] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
##
## $white
##  [1] 1 0 0 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0
```

1.1. Sample non-events

```
# Riskset
p<-length(class$info$male)
riskset<-cbind(rep(1:p,p),rep(1:p,each=p))
riskset<-riskset[riskset[,1]!=riskset[,2],]
riskset<-cbind(riskset,1:nrow(riskset))
colnames(riskset)<-c("sender","receiver","id")

# Sample non-events
set.seed(1)
n<-nrow(class$rev)
non.events<-NULL

for (i in 1:n){
  snd<-class$rev[i,1]
  rcv<-class$rev[i,1]
  non.id<-sample(riskset[riskset[,1]!=snd | riskset[,2]!=rcv,3],1)
  non.snd<-riskset[non.id,1]
  non.rcv<-riskset[non.id,2]
  non.events<-rbind(non.events,c(non.snd,non.rcv))
}
colnames(non.events)<-c("non.sender","non.receiver")
class$rev<-cbind(class$rev,non.events)
```

1.2. Model formulation: does gender play a role?

The question we want to answer is to what extend do the interactions between the individuals depend on their gender $x^{(1)}$, race $x^{(2)}$ and their role in the classroom $x^{(3)}$. Even with these three convariates, there are a number of effects that could be plausible. For example, with the effect of gender:

- **Sender.** Either boys or girls could be more often the initiator of an interaction. This corresponds to a sender gender effect $x_s^{(1)}$.
- **Receiver.** Either boys or girls could be more on the receiving end of a interaction. This corresponds to a reiver gender effect $x_r^{(1)}$.
- **Homophily.** Interactions may be more common between the same genders (or not). This corresponds to an homophily effect, $h_{sr}^{(1)} = 1_{\{x_s^{(1)}=x_r^{(1)}\}}$.

Focusing just on gender, we consider the log-linear hazard formulation,

$$\lambda_{sr}(t) = Y(t) \times \lambda_0(t) e^{\beta_1 x_s^{(1)} + \beta_2 x_r^{(1)} + \beta_3 h_{sr}^{(1)}}.$$

How can we extend the model including the other two covariates?

1.3. Preparing the covariates

Our aim is to estimate the parameters β from the data in order to understand the importance of the various social mechanisms in the classroom setting. We do this by maximizing the sampled partial likelihood. For the gender covariates only, this means

$$L^{PS}(\beta) = \prod_{i=1}^{279} \frac{e^{\beta_1\left(x^{(1)}_{s_i} - x^{(1)}_{s_i^*}\right) + \beta_1\left(x^{(1)}_{r_i} - x^{(1)}_{r_i^*}\right) + \beta_1\left(h^{(1)}_{s_i r_i} - x^{(1)}_{s_i^* r_i^*}\right)}}{1 + e^{\beta_1\left(x^{(1)}_{s_i} - x^{(1)}_{s_i^*}\right) + \beta_1\left(x^{(1)}_{r_i} - x^{(1)}_{r_i^*}\right) + \beta_1\left(h^{(1)}_{s_i r_i} - x^{(1)}_{s_i^* r_i^*}\right)}}$$

This means that for each observed event (s_i, r_i, t_i) and non-event (s_i^*, r_i^*, t_i) we need to add a row to the data containing,

$$\left(x^{(1)}_{s_i} - x^{(1)}_{s_i^*}\right), \quad \left(x^{(1)}_{r_i} - x^{(1)}_{r_i^*}\right), \quad \left(h^{(1)}_{s_i r_i} - x^{(1)}_{s_i^* r_i^*}\right)$$

This can be done as follows.

```
sender.gender <- class$info$male[class$rev$sender] - class$info$male[class$rev$non.sender]
receiver.gender <- class$info$male[class$rev$receiver] - class$info$male[class$rev$non.receiver]
same.gender <- (class$info$male[class$rev$sender] == class$info$male[class$rev$receiver]) -        (cla

class$rev <- cbind(class$rev,sender.gender,receiver.gender,same.gender)
```

Now you add the code for the other two covariates.

```
# Define sender.race, receiver.race, same.race, sender.teacher, receiver.teacher
```

1.4. Fit the Relational Event Model

Graphical maximizing of the sampled partial likelihood is not possible anymore as there are multiple effects. Instead, the likelihood matches the likelihood of a logistic regression with responses equal to one for all the events. Considering only gender, we obtain:

```
library(mgcv)
one<-rep(1,n)
class$rev<- cbind(class$rev,one)
class1.glm <- gam(one~-1+sender.gender+receiver.gender+same.gender, family = binomial, data=class$rev)
summary(class1.glm)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## one ~ -1 + sender.gender + receiver.gender + same.gender
##
## Parametric coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## sender.gender   -0.39578    0.20644  -1.917  0.05522 .
## receiver.gender -0.08011    0.20954  -0.382  0.70223
## same.gender      0.54443    0.18999   2.866  0.00416 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = 0.32918  Scale est. = 1         n = 279
```

We interpret the output as follows:

- The coefficient $\hat{\beta}_1 = -0.396$ from sender.gender is negative, which means that boys tend to be 33% $(=1 - \exp(-0.396))$ less likely than girls to initiate a relational event in a classroom.

- Although the receiver.gender effect $\hat{\beta}_2 = -0.080$ is also negative, it is not significant (p-value = 0.7022). Although it may be tempting to remove this variable, because it is not significant, this would be incorrect. As the dyadic same.gender effect is important, we need to include the monadic gender effects as well: this is called the *hierarchy principle*.
- A strong driver of interactions is gender homophily: as $\hat{\beta}_3 = 0.544$, it is 72% ($= \exp(0.544) - 1$) more likely that interactions occur between individuals of the same gender.

Now perform the following exercises yourself:

- Add the race and teacher covariates to the model and interpret the results.
- Do some model selection. Consider using significance tests and/or AIC consideration. However, make sure that your selected model does not violate the *hierarchy principle*.

```
# Fit a degenerate logistic regression on all covariates and perform model selection
```

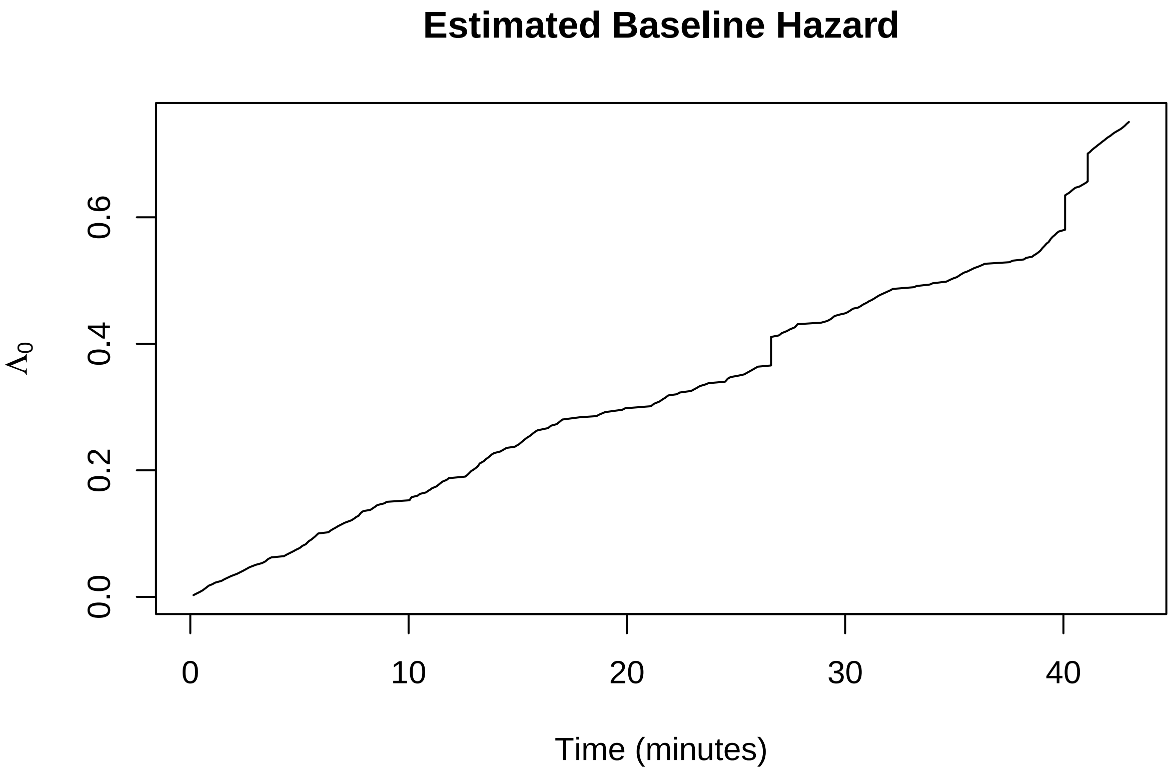
1.5. Visualizing the unmodelled part: the baseline hazard

Although our model captures part of the dynamic in the classroom interactions, there is probably some unexplained variation left. In our model, these terms are collected in the nuisance parameter $\lambda_0(t)$, referred to as the baseline hazard. In order to visualize this baseline hazard, it is possible to estimate its cumulative version $\Lambda_0(t) = \int_0^t \lambda_0(u)du$ *after* fitting the model,

$$\widehat{\Lambda}_0(t) = \sum_{t_i \leq t} \frac{2/|\mathcal{R}(t_i)|}{\exp(\hat{\beta}^t x_{s_i r_i}) + \exp(\hat{\beta}^t x_{s_i^* r_i^*})}$$

We prepare the baseline hazard for the reduced model with only gender. It is your task to calculate the baseline hazard for the final model.

```
tms <- class$rev$time
Lambda0<-NULL
absR<-nrow(riskset)
event.x<-cbind(class$info$male[class$rev$sender], class$info$male[class$rev$receiver],
               1*(class$info$male[class$rev$sender]==class$info$male[class$rev$receiver]))
non.event.x<-cbind(class$info$male[class$rev$non.sender], class$info$male[class$rev$non.receiver],
               1*(class$info$male[class$rev$non.sender]==class$info$male[class$rev$non.receiver]))
bhat<-coef(class1.glm)
Lambda0<-2/absR*cumsum(1/(exp(event.x%*%bhat)+exp(non.event.x%*%bhat)))
plot(tms,Lambda0,type="l",xlab="Time (minutes)", ylab=expression(Lambda[0]),
     main="Estimated Baseline Hazard")
```



If the plot is entirely linear, then most of the variation in the timings has been accounted for. Although not bad, this is not entirely the case when accounting just for gender. Calculate and visualize the plot for the baseline hazard for the final model including gender, race and teacher/student status.

```
# Estimate the cumulative baseline hazard function for your selected model
```