T2DM CKD Study

Updated Data Simulation

Nicolas Gonzalez Granda

August 1, 2024

Simulating EHR Data

We will generate a simulated training/testing dataset to use with LATTE, made up of N=5000 samples, each with p=59 covariates

```
N=5000
p=59
n=200
```

Event Time Generation and Censoring

First, we generate the baseline predictors for each patient from a multivariate normal

```
set.seed(16)
rho=0.1
covmat=(1-rho)*diag(p)+rep(rho,p)
mus=rep(0,p)
baseline=mvrnorm(N,mus,covmat)
```

Then, we convert these to the datatypes most commonly found in EHR datasets (25 count + 25 binary + 9 continuous covariates)

```
set.seed(16)
simdata=matrix(rep(0,N*p),N,p)
tau=0.3
simdata[,1:25]=baseline[,1:25]>tau
lambdas=2*pnorm(baseline[,26:50])
simdata[,26:50]=rpois(25*N,lambdas)
simdata[,51:59]=pnorm(baseline[,51:59])
```

Including the above covariates in a Cox PH model, we can simulate the clinical event times for each patient based on their features

```
set.seed(16)
baserisk=0.05*12
#Fine tuned to yield 50% censoring
b0=-2
beta=0.5*c(1,-1,-2,rep(c(1,-1,-2,2),14))
risks=baserisk*exp(b0+simdata%*%beta)
times=rexp(N,rate=risks)
summary(times)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.000e+00 1.000e+00 1.800e+01 2.754e+07 5.520e+02 1.266e+11
```

Now, we sample the censoring time from a discrete uniform, yielding about a 50% censoring rate

```
set.seed(16)
t0=8
tf=34
censor=sample(t0:tf,N,replace=TRUE)
delta=censor>=times
mean(delta)
```

```
[1] 0.506
```

Surrogate data (Gold and Silver standard labels)

We simulate periods=20 timepoints for each subject, defining bool_data as whether or not the patient's censoring time is greater than the "current" timepoint. Similarly, the Gold standard labels Y are defined as whether or not the "current" time is greater than the event time for each patient. Finally, the silver standard labels are defined as a mixture beta, with fixed coefficients for all timepoints before the event time, and time-varying coefficients for all timepoints after the event time. We evaluate the predictive value of these silver standard labels for the gold standard labels by means of the ROC curve.

```
set.seed(16)
periods=20
curr_Times=rep(1:periods,N)
bool_data=(curr_Times<=rep(censor,each=periods))
Y=ifelse(rep(times,each=periods)<curr_Times,1,0)
alphas=-rep(times,each=periods)+curr_Times
silver=rbeta(N*periods,ifelse(rep(times,each=periods)>curr_Times,1,1+alphas/2),ifelse(rep(times,each=periods)>
annotated=sample(c(1:N),n)
```

Annotated Subjects

In order to mimic annotation of a small subset of patient records, we will sample n patient IDs without replacement from the population size N and select these to be annotated. The others will be masked as NAs for their Y variable (gold-standard label) but their ground truth will be conserved separately for later validation.

```
Y_ann=rep(NA,N*periods)
for (t in annotated){
  Y_ann[seq(from=t*periods-periods+1,to=t*periods)]=Y[seq(from=t*periods-periods+1,to=t*periods)]
}
```

Count Data

We will additionally generate longitudinal Poisson (time-varying rates) data relating to EHR utitilization, CUI counts, and NLP mentions, the latter two's rates depending on the former. The following function will allow us to generate the count data, with rates being inputed afterwards.

Now, the normally distributed rate coefficients for each subject and time period will be generated i.e. before event time, shortly after event time, and afterwards for each patient. The variance-covariance matrix is randomly generated and positive-semidefinite

```
set.seed(16)
sigma=matrix(runif(49),7)
sigma=0.1*t(sigma)%*%sigma
alpha_means=c(0.5,0.05,0.2,0.1,0.3,0.15,0.4)
beta_means=c(2.5,3.05,6.2,4.1,7.3,5.15,6.4)
gamma_means=c(1.5,2.05,4.2,2.1,4.3,2.15,3.4)
halphas=mvrnorm(N,rep(0,7),sigma)
hbetas=mvrnorm(N,rep(0,7),sigma)
hgammas=mvrnorm(N,rep(0,7),sigma)
alphas=2*pnorm(halphas)*matrix(rep(alpha_means,N),nrow=N,byrow=TRUE)
betas=2*pnorm(hbetas)*matrix(rep(beta_means,N),nrow=N,byrow=TRUE)
gammas=2*pnorm(hgammas)*matrix(rep(gamma_means,N),nrow=N,byrow=TRUE)
```

Finally, each of the count data will be generated

Semantic Embeddings

We first compute the co-occurrence matrix C:

```
cooc=matrix(nrow=6,ncol=6)
for (i in 1:6){
  for (j in 1:6){
    cooc[i,j]=sum(counts[,i]*counts[,j])
  }
}
cooc
```

```
        [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
[1,] 1658898 2999977 1732607 3291751 1866038 2612588
[2,] 2999977 6692353 3558923 6741895 3931495 5469005
[3,] 1732607 3558923 2311075 3955322 2268896 3141261
[4,] 3291751 6741895 3955322 7950054 4261592 5995704
[5,] 1866038 3931495 2268896 4261592 2835448 3527628
[6,] 2612588 5469005 3141261 5995704 3527628 5252434
```

Then, the SPPMI matrix:

```
D=sum(cooc)
SPPMI=matrix(nrow=6,ncol=6)
for (i in 1:6){
  for (j in 1:6){
    val=log(cooc[i,j])+log(D)-log(sum(cooc[i,1:6]))-log(sum(cooc[j,1:6]))
    SPPMI[i,j]=ifelse(val>0,val,0)
  }
}
SPPMI
```

```
          [,1]       [,2]       [,3]       [,4]      [,5]      [,6]
[1,] 0.1280152 0.00000000 0.00000000 0.00000000 0.0000000 0.0000000
[2,] 0.0000000 0.06237477 0.00000000 0.00000000 0.0000000 0.0000000
[3,] 0.0000000 0.00000000 0.09801088 0.00000000 0.0000000 0.0000000
[4,] 0.0000000 0.00000000 0.00000000 0.05244096 0.0000000 0.0000000
[5,] 0.0000000 0.00000000 0.00000000 0.00000000 0.1090718 0.0000000
[6,] 0.0000000 0.00000000 0.00000000 0.00000000 0.0000000 0.0655711
```

```
sqrt(diag(SPPMI))
```

```
[1] 0.3577922 0.2497494 0.3130669 0.2289999 0.3302601 0.2560686
```

Formatting data

We will finally compile all above simulated values into a single dataset that includes event times, censoring times and status, surrogate data, and all 59 covariates for each patient. This should later be changed into a format similar to that in the LATTE protocol i.e. several EHR visits for the same patient, with correct identification of the true event time for n=200 patients (annotated). Additionally, variable lists should be formatted, and semantic embeddings extracted and compiled in a separate file.

```
ID=rep(1:N,each=periods)
Utilization=as.vector(t(utilization))
CUI1=as.vector(t(CUI1))
NLP1=as.vector(t(NLP1))
CUI2=as.vector(t(CUI2))
NLP2=as.vector(t(NLP2))
CUI3=as.vector(t(CUI3))
NLP3=as.vector(t(NLP3))
EHR_data=data.frame(ID,curr_Times,bool_data,Y_ann,silver,Utilization,CUI1,NLP1,CUI2,NLP2,CUI3,NLP3)
library(knitr)
kable(head(EHR_data,80), row.names = F,digits=3)
```

ID	curr_Times	bool_data	Y_ann	silver	Utilization	CUI1	NLP1	CUI2	NLP2	CUI3	NLP3
1	1	TRUE	NA	0.134	0	0	0	0	0	0	0
1	2	TRUE	NA	0.289	0	0	0	0	0	0	0
1	3	TRUE	NA	0.050	0	0	0	0	0	0	0
1	4	TRUE	NA	0.046	0	0	0	0	0	0	0
1	5	TRUE	NA	0.415	1	0	1	0	0	0	0
1	6	TRUE	NA	0.231	0	0	0	0	0	0	0
1	7	TRUE	NA	0.057	0	0	0	0	0	0	0
1	8	TRUE	NA	0.116	1	0	1	0	0	0	0
1	9	FALSE	NA	0.218	1	0	0	0	0	0	0
1	10	FALSE	NA	0.530	0	0	0	0	0	0	0
1	11	FALSE	NA	0.093	0	0	0	0	0	0	0
1	12	FALSE	NA	0.116	0	0	0	0	0	0	0
1	13	FALSE	NA	0.135	0	0	0	0	0	0	0
1	14	FALSE	NA	0.047	0	0	0	0	0	0	0
1	15	FALSE	NA	0.355	0	0	0	0	0	0	0
1	16	FALSE	NA	0.309	0	0	0	0	0	0	0
1	17	FALSE	NA	0.025	1	0	0	0	1	0	1
1	18	FALSE	NA	0.022	0	0	0	0	0	0	0
1	19	FALSE	NA	0.054	0	0	0	0	0	0	0
1	20	FALSE	NA	0.056	0	0	0	0	0	0	0
2	1	TRUE	NA	0.077	2	0	0	0	1	3	0
2	2	TRUE	NA	0.191	0	0	0	0	0	0	0
2	3	TRUE	NA	0.212	0	0	0	0	0	0	0
2	4	TRUE	NA	0.272	1	0	0	0	1	1	0
2	5	TRUE	NA	0.615	2	0	2	0	1	0	0
2	6	TRUE	NA	0.648	0	0	0	0	0	0	0
2	7	TRUE	NA	0.175	1	0	1	0	0	0	0
2	8	TRUE	NA	0.121	0	0	0	0	0	0	0
2	9	TRUE	NA	0.213	1	0	0	0	0	0	0
2	10	TRUE	NA	0.175	1	0	1	0	0	0	0
2	11	TRUE	NA	0.348	1	0	0	0	0	0	0
2	12	TRUE	NA	0.185	1	0	0	1	0	0	1
2	13	TRUE	NA	0.369	1	0	0	0	1	0	1
2	14	TRUE	NA	0.482	1	0	0	0	0	0	0
2	15	TRUE	NA	0.175	0	0	0	0	0	0	0
2	16	TRUE	NA	0.215	2	0	0	0	0	0	0
2	17	TRUE	NA	0.118	0	0	0	0	0	0	0
2	18	TRUE	NA	0.117	1	0	1	0	0	0	0
2	19	TRUE	NA	0.278	0	0	0	0	0	0	0
2	20	TRUE	NA	0.187	0	0	0	0	0	0	0
3	1	TRUE	NA	0.089	0	0	0	0	0	0	0
3	2	TRUE	NA	0.074	0	0	0	0	0	0	0
3	3	TRUE	NA	0.077	0	0	0	0	0	0	0
3	4	TRUE	NA	0.099	0	0	0	0	0	0	0
3	5	TRUE	NA	0.044	1	0	0	0	0	0	1
3	6	TRUE	NA	0.720	0	0	0	0	0	0	0
3	7	TRUE	NA	0.501	0	0	0	0	0	0	0
3	8	TRUE	NA	0.395	1	0	0	0	0	1	1
3	9	TRUE	NA	0.454	1	0	0	1	0	0	1
3	10	TRUE	NA	0.713	0	0	0	0	0	0	0
3	11	TRUE	NA	0.522	0	0	0	0	0	0	0
3	12	TRUE	NA	0.504	0	0	0	0	0	0	0
3	13	TRUE	NA	0.072	0	0	0	0	0	0	0
3	14	TRUE	NA	0.432	0	0	0	0	0	0	0
3	15	TRUE	NA	0.755	0	0	0	0	0	0	0
3	16	TRUE	NA	0.285	2	0	1	0	1	1	1
3	17	TRUE	NA	0.072	0	0	0	0	0	0	0
3	18	TRUE	NA	0.245	1	0	1	0	0	0	0

ID	curr_Times	bool_data	Y_ann	silver	Utilization	CUI1	NLP1	CUI2	NLP2	CUI3	NLP3
3	19	TRUE	NA	0.085	1	0	0	0	0	0	0
3	20	TRUE	NA	0.229	0	0	0	0	0	0	0
4	1	TRUE	NA	0.299	1	0	1	0	0	0	0
4	2	TRUE	NA	0.066	0	0	0	0	0	0	0
4	3	TRUE	NA	0.020	2	0	1	0	0	0	2
4	4	TRUE	NA	0.881	0	0	0	0	0	0	0
4	5	TRUE	NA	0.931	2	5	8	3	6	8	10
4	6	TRUE	NA	0.530	1	3	4	4	3	1	3
4	7	TRUE	NA	0.686	5	10	26	9	26	11	15
4	8	TRUE	NA	0.716	1	1	3	0	4	2	2
4	9	TRUE	NA	0.935	2	4	11	4	6	5	3
4	10	TRUE	NA	0.775	3	4	10	2	5	7	4
4	11	TRUE	NA	0.976	1	6	2	2	3	1	4
4	12	TRUE	NA	0.649	0	0	0	0	0	0	0
4	13	TRUE	NA	0.903	0	0	0	0	0	0	0
4	14	TRUE	NA	0.939	1	1	4	0	3	3	4
4	15	TRUE	NA	0.947	1	1	5	1	1	1	3
4	16	TRUE	NA	0.527	1	1	3	3	2	1	3
4	17	TRUE	NA	0.957	1	0	7	0	3	1	2
4	18	TRUE	NA	0.959	2	0	9	8	7	9	9
4	19	FALSE	NA	0.733	5	8	24	8	12	10	12
4	20	FALSE	NA	0.992	2	7	10	3	7	3	5