Sample size for prediction of quantitative and binary outcomes based on cohort study

ZWu

28 November, 2021

For EPPIC_2021 application (PI Jean King. Project/Core PIs: Korkin, Ruiz).

Consider a cohort study of three groups: MBSR, acupuncture, and control, each has same sample size n. The total sample size is 3n.

Predict quantiative outcome based on regression model assumptions

Sample size for pediction accuracy of quantiative outcomes based on simulations.

Consider a linear mixed effect model as the true underlying model:

$$Y_{ij} = \eta_{ij}(X) + \epsilon_{ij}$$

, where

$$\eta_{ij}(X) = \beta_0 + b_i + \beta_{acup} * acup_{ij} + \beta_{mbsr} * mbsr_{ij} + \beta_{sex} * sex_{ij} + \beta_{age} * age_{ij} + \beta_{edu} * edu_{ij} + \beta_{len} * len_{ij} + \beta_{base} * base_{ij} + \beta_{conc} * conc_{ij} + \sum_k$$

and

$$b_i \sim N(0, \sigma_i^2) \quad \text{and} \quad \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2).$$

Interpretation:

- Y_{ij}: quantitative response variable of pain reduction (i.e., TreatmentImpact, after a given period of time since the beginning of treatment) for the jth individual in the ith racial group. Positive value means pain reduces.
- β_0: a "natural" pain reduction corresponds to no-treatment (no acupuncture nor MBSR) and all other covariates being 0.
- $b_i \sim N(0, \sigma_i^2)$: The mixed-effect (clustering effect) of the ith racial group. Individuals in the same racial group have the same b_i value so that their pair-wise covariance is σ_i^2.
- $acup = 1$ if acupuncture treatment, $= 0$ o.w. β_{acup} is the effectof acupuncture.
- $mbsr = 1$ if MBSR treatment, $= 0$ o.w. β_{mbsr} is the effectof MBSR.
- $sex = 1$ if male, $= 0$ if female. β_{sex} is the effect of sex.
- age: standardized age value (mean 0 and sd 1). β_{age} is the effect of age.
- edu: education level: 0 / 1, 50% each.
- len: duration length of pain. short or long: 0 / 1, 50% each.
- bas: Baseline pain score. standardized.

- *conc*: presence of certain concomitant diseases: 0/1, 50% each.
- x_k: standardized value of the kth PainMarker, $k = 1, ..., p$. β_k's are the corresponding effects. They are from OMICS- and biopsychosocial PainMarkers.
- z_{lij} and z_{mij} are modifiers for acupuncture and MBSR, respectively. They are from OMICS- and biopsychosocial PainMarkers.
- The error SD σ can be used to adjust the relative effects (or signal-to-noise ratio) regarding β parameters, and the variation explained model (R^2).
- Cohen's

$$f^2 = R^2/(1-R^2) = \frac{\sigma_i^2 + \beta_{acup}^2 Var(acup) + \beta_{mbsr}^2 Var(mbsr) + \beta_{sex}^2 Var(sex) + \beta_{age}^2 Var(age) + \sum_{k=1}^{p} \beta_k^2 Var(x_k)}{\sigma^2}$$

, where $R^2 is the coefficient of determination, the proportion of the variation in the dependent variable that is predictable from the$

Considerations:

- b_i: consider 3 groups, roughly equal numbers in the sample.
- β_{acup}, β_{mbsr}, β_{sex}, and β_{sex} are used to set/control their R^2, i.e., the percentage of variation explained by these "basic" factors.
- β_{sex}: Consider males are easier to reduce pain than females. Assume 50
- β_k, γ_l, γ_m are used to set the percentage of variation explained by these extra factors.
- Interaction terms are for Hypothesis 1 saying that the effects of treatments are "modulated by biopsychosocial factors".
- This study does address the aim of clustering patients ("Clustering and discovery of EPPIC-TreatmentPhenotypes" in Project 3). If some factors/markers are positively and some are negatively interacted with acupuncture/MBSR, then we could decide which treatment is better for them based on their markers. [Check Hong's project for AbbVie on subgrouping patients for drug treatment.]

Predict binary outcome based on logistic model assumptions

Consider a generalized linear mixed effect model as the true underlying model:

$$E(P(Y_{ij} = 1|X)) = \frac{1}{1 + exp(-\eta_{ij}(X))},$$

where X is the matrix of covariate data, and $\eta_{ij}(X)$ is same as above (except that the coefficient values could be different)

Interpretation:

- Y_{ij}: binary response variable of TreatmentResponse for the jth individual in the ith racial group. 1 for pain relief, 0 for no relief.
- β_0: the baseline pain relieve probability when no interventions and all other covariates being 0. Together with othe terms, we can adjust β_0 to control the prevalence of Y.
- We considered the ORs of some predictors based on Tables 2 and 3 of [**?**] "Patient characteristics and variation in treatment outcomes: which patients benefit most from acupuncture for chronic pain?"

Considerations:

We use AUC to measure how well the factors explains / contributes to the model. AUC and Cox and Snell R2 are connected (https://onlinelibrary.wiley.com/doi/full/10.1002/sim.8806). Both are used to represent how well the factors explains the model (Cox and Snell R2 is an extension of R2 for measuring the percentage

of variation explained in regression). Both are interchangeably used for calculating the minimum sample size based on the criterion regarding Nagelkerke's R-squared value. See The formula by given in Fig 5 of paper [?]: https://www.research.manchester.ac.uk/portal/files/161373531/bmj.m441.full.pdf

```
a=1;
#
#   library(MASS)
#   library(nlme);
#   source("/Users/zheyangwu/ResearchDoc/Computation/CodingLibraries/myRLibrary/Prediction/Lib_Predicti
#   source("/Users/zheyangwu/ResearchDoc/Computation/CodingLibraries/myRLibrary/Simulations/Simulate_Da
#   source("/Users/zheyangwu/ResearchDoc/Computation/CodingLibraries/myRLibrary/Prediction/generate_res
#
#
#
#
# ###Parameter setting
#
#   ###Parameters on the effects / coefficients
#   beta0 = -0.5; #For binary trait, the risk at X=0 is 1/(1+exp(-beta0)). The average risk (prevalence
#                       # Y = array(NA, length(Xbeta));
#                       #  for (i in 1:length(Y)) {
#                       #    Py1x = 1/(1 + exp(-Xbeta[i]) - beta0); # P(y=1 | x) based on logistic model
#                       #    Y[i] = ifelse(runif(1) < Py1x, 1, 0);
#                       #  }
#                       #  mean(Y); #prevalance of Y.
#
#   #beta0 = -7; # for binary trait. Risk when X=0 is 1/(1+exp(7))=0.0009110512. Also, at the given coe
#   #beta0 = -3; # for binary trait. Risk when X=0 is 1/(1+exp(3))=0.04742587. Also, at the given coeff
#
#
#   raceN = 3; #Number of racial groups
#   sigma.race = 1; #pair-wise covariance among individuals in the same racial group
#   beta.race = 1; #treat the "coefficient" of the mix-effect b_i be 1.
#
#   beta.acup = log(4.9); #6; #Coeff of acupuncture
#   beta.mbsr = log(4.9); #6; #Coeff of mbsr
#   beta.sex = log(1.1); #1;
#   beta.age = log(1.25); #-1;
#   beta.edu = log(1.26); #1;
#   beta.len = log(1.13); #1;
#   beta.bas = log(0.80);#1;
#   beta.conc = log(0.77); #1;
#
#   markerN = 10; #The # of PainMarkers
#   beta.biom.v = 2; #=1; #The value of the coeff of the biomarkers
#   beta.biom = rep(beta.biom.v, markerN); #Coeff of the biomarkers
#   names.xbiom = paste("xbiom", 1:markerN, sep=""); #variable names for biomarkers
#
#   mdfN.acup = 4; #The # of modifiers for acupuncture
#   beta.mdf.acup.v=2; #The value of the coeff of accupunctur's modifiers
#   beta.mdf.acup = rep(beta.mdf.acup.v, mdfN.acup); #Vector of coeffs of accupunctur's modifiers
#   gama.acup.v = 2; #The value of the coeff of the modifier*acupucture interaction terms.
#   gama.acup = rep(gama.acup.v, mdfN.acup); #Vector of coeffs of the modifier*acupucture interaction t
#
```

```
#   names.mdf.acup = paste("zmdfAcup", 1:mdfN.acup, sep=""); #variable names of the modifiers for acupu
#
#   mdfN.mbsr = 4; #The # of modifiers for MBSR
#   beta.mdf.mbsr.v=2; #The value of the coeff of MBSR's modifiers
#   beta.mdf.mbsr = rep(beta.mdf.mbsr.v, mdfN.mbsr); #Vector of coeffs of accupunctur's modifiers
#   gama.mbsr.v = 2; #The value of the coeff of the modifier*MBSR interaction terms.
#   gama.mbsr = rep(gama.mbsr.v, mdfN.mbsr); #Vector of coeffs of the modifier*MBSR interaction terms.
#
#   names.mdf.mbsr = paste("zmdfMBSR", 1:mdfN.mbsr, sep=""); #variable names of the modifiers for MBSR
#
#   errSD=1; #The SD of error term
#
# ###Parametters on prodiction process
#   isRandomCV=T; #Random cross-validation in prediction
#   nfold=5; #The number of folds in cross-validation
#   nrepeat=2; #number of repeats of cross-validation
#
# ###Parameters on simulations
#   simuN = 100; #The number of simulations.
#
# ###Data simulation and prediction outcomes
#   predProp = c(0, 0.25, 0.5, 0.75, 1); #Proportion of true predictors besides names.basic that are in
#   models = vector(mode = "list", length(predProp)); #prediction models .
#   outputs = vector(mode = "list", length(predProp)); #prediction outputs .
#
# ###Parameters on data
# groupSampleSizes = seq(20, 400, by=20);
# #groupSampleSizes = seq(500, 2000, by=100);
# #groupSampleSizes = seq(150, 500, by=50);
# AUC = array(NA, dim=c(length(groupSampleSizes), length(predProp))); #Store AUC over sample sizes and
# for(gi in 1:length(groupSampleSizes)){
#   groupSampleSize = groupSampleSizes[gi]; #sample size for each of the three groups: control, mbsr, a
#
# ###Looping through simulations
#   R2.controls = array(NA, simuN); #Store the R2 of the controlling predictors.
#   for(i in 1:simuN) {
#     ###Generate data
#     x0 = rep(1, groupSampleSize*3);
#
#     #The mixed-effect term for racial group
#     xrace = sample(1:raceN, size=groupSampleSize*3, replace=T, prob=rep(1/raceN, raceN));
#       #Assume equal chance for each racial group to be sampled.
#     b.xrace = array(NA, dim=groupSampleSize*3); #b.xrace is the vector of b_i values.
#     for (racei in 1:raceN){ b.xrace[which(xrace==racei)] = rnorm(1, sd=sigma.race); }
#       #assign the same b_i value for the all in the ith racial group.
#
#     #The "basic" factors
#     xacup = c(rep(0, groupSampleSize*2), rep(1, groupSampleSize)); #acupuncture group indicator
#     xmbsr = c(rep(0, groupSampleSize), rep(1, groupSampleSize), rep(0, groupSampleSize)); #MBSR group
#     xsex = rbinom(n=groupSampleSize*3, size=1, prob=0.5); #50% recruited are males??
#     xage = rnorm(n=groupSampleSize*3, mean=0, sd=1); #standardized age.
#     xedu = rbinom(n=groupSampleSize*3, size=1, prob=0.5);
#     xlen = rbinom(n=groupSampleSize*3, size=1, prob=0.5);
```

```
#     xbas = rnorm(n=groupSampleSize*3, mean=0, sd=1);
#     xconc = rbinom(n=groupSampleSize*3, size=1, prob=0.5);
#
#     #PainMarker data
#     xbiom = data.frame(matrix(rnorm(n=groupSampleSize*3*markerN), ncol=markerN)); #Assume biomarker v
#     names(xbiom) = names.xbiom;
#
#     #acupuncture-modifier data
#     zmdfAcup = data.frame(matrix(rnorm(n=groupSampleSize*3*mdfN.acup), ncol=mdfN.acup)); #Assume acup
#     names(zmdfAcup) = names.mdf.acup;
#
#     #mbsr-modifier data
#     zmdfMBSR = data.frame(matrix(rnorm(n=groupSampleSize*3*mdfN.mbsr), ncol=mdfN.mbsr)); #Assume acup
#     names(zmdfMBSR) = names.mdf.mbsr;
#
#     Xmatrix = cbind(x0, b.xrace, xacup, xmbsr, xsex, xage, xedu, xlen, xbas, xconc, xbiom, zmdfAcup,
#
#     #### Generate response
#     names.mainEff  = c("x0", "b.xrace", "xacup", "xmbsr", "xsex", "xage", "xedu", "xlen", "xbas", "xc
#     coeffs.mainEff = c(beta0, beta.race, beta.acup, beta.mbsr, beta.sex, beta.age, beta.edu, beta.len
#     names.trt = c("xacup", "xmbsr"); #variable names of the treatments
#     names.mdf = list(names.mdf.acup, names.mdf.mbsr); #variable names of the modifiers corresponding
#     coeffs.interaction = list(gama.acup, gama.mbsr); #coeffficients of the treatment-modifier interact
#
#
#     # ####----Quantitative Response-----
#     # resp = get.Y.reg(XData=Xmatrix, names.mainEff, coeffs.mainEff, names.trt, names.mdf, coeffs.int
#     # #print(resp$R2); #proportion of variation explained by all predictors
#     #
#     # #Calculate the R^2 of the controlling predictors
#     # vars.control = c(varBeta0=0, varRace=sigma.race^2, varAcup=(1/3)*(1-1/3), varMbsr=(1/3)*(1-1/3)
#     # betas.control = c(beta0, beta.race, beta.acup, beta.mbsr, beta.sex, beta.age, beta.edu, beta.le
#     # R2.controls[i] = sum(vars.control*betas.control^2)/var(resp$Y);
#
#     ####----Binary Response-----
#     resp = get.Y.logit(XData=Xmatrix, names.mainEff, coeffs.mainEff, names.trt, names.mdf, coeffs.int
#
#     ####Combine data for analysis
#     xrace = as.factor(xrace); #Convert race into factor variable, which is used in data analysis.
#     dat = data.frame(Y=resp$Y, Xmatrix, xrace);
#
#
#     ####Predictive analysis
#     names.control = c("xrace", "xacup", "xmbsr", "xsex", "xage", "xedu", "xlen", "xbas", "xconc"); #C
#     names.trt.all =  c("xacup", "xmbsr"); #All possible treatments that could have interaction effect
#     for (mi in 1:length(predProp)){
#       ##Create model formula based on proportion of predictors used.
#       xbiom.used    = round(length(names.xbiom)*predProp[mi]);
#       mdf.acup.used = round(length(names.mdf.acup)*predProp[mi]);
#       mdf.mbsr.used = round(length(names.mdf.mbsr)*predProp[mi]);
#       names.main=c(names.control, names.xbiom[0:xbiom.used], names.mdf.acup[0:mdf.acup.used], names.m
#       if (predProp[mi]==0) {
#         names.trt = NULL;
```

```
#       } else{
#         names.trt = names.trt.all;
#         names.mdf = list(names.mdf.acup[0:mdf.acup.used], names.mdf.mbsr[0:mdf.mbsr.used]);
#       }
#       models[[mi]]= formula.f.r(names.main=names.main, names.trt=names.trt, names.mdf=names.mdf, name
#
#       # ### ---- Predict quantitative outcome ----
#       # #out = meanPredEvaluCV.lme(fixed=models[[mi]]$fixed, dat=dat, randomf=models[[mi]]$random, mo
#       # out = meanPredEvaluCV.lme(fixed=models[[mi]]$fixed, dat=dat, randomf=NULL, model_R='lm', loop
#       # outputs[[mi]] = rbind(outputs[[mi]], t(c(MSE=out[1], L2normRatio=out[2], L1normRatio=out[3],
#
#
#       ### ---- Predict quantitative outcome ----
#       out = predEvaluCV.glm(formula=models[[mi]]$fixed, dat=dat, nfold=nfold, nrepeat=nrepeat, isRand
#       #outputs[[mi]] = rbind(outputs[[mi]], t(c(prob=out[1], sensi=out[2], speci=out[3], AUC=out[4]))
#       outputs[[mi]] = rbind(outputs[[mi]], t(unlist(out)));
#
#     }
#   }
#
#   # #True underlying model
#   print(c(sampleSize=groupSampleSize*3, raceN=raceN, sigma.race = sigma.race,  beta.acup = beta.acup,
#     beta.mbsr = beta.mbsr, beta.sex = beta.sex, beta.age = beta.age, beta.edu=beta.edu, beta.len=beta
#     mdfN.acup=mdfN.acup, beta.mdf.acup.v=beta.mdf.acup.v, gama.acup.v=gama.acup.v,
#     mdfN.mbsr=mdfN.mbsr, beta.mdf.mbsr.v=beta.mdf.mbsr.v, gama.mbsr.v=gama.mbsr.v));
#
#   # #R2 of controlling predictors
#   # mean(R2.controls);
#
#   #Prediction accuracies
#   for (mi in 1:length(predProp)){
#     # print(predProp[mi]);
#     # print(models[[mi]]$fixed);
#     # print(apply(outputs[[mi]], 2, mean));
#     # print(apply(outputs[[mi]], 2, quantile, probs=c(0.05, 0.95)));
#
#     AUC[gi, mi] = apply(outputs[[mi]], 2, mean)[4];
#   }
# }
# cbind(groupSampleSizes*3, AUC);
#
```