Multiple Imputation, WAIC and LOO, Misc

David Carlson

March 4, 2021

Multiple imputation - Assumptions

Nonignorable missing data is data in which the missingness depends on unobserved data. In order to impute data, we need to assume that we can model the missing data based on data that is observed. There is no standard tool to deal with this problem. We can model the missing data using various predictors and compare the results to see if which predictors are used dramatically changes our findings. The assumption we are making is that the model we use to impute the data is appropriate and is not biasing our interpretations of the data.

An example function

```
#function summarizing missingness in matrix or dataframe X
#param X matrix or dataframe to be summarized
#param row.crit critical proportion to drop rows that exceed this proportion of missingness - default i
#param col.crit critical proportion to drop columns that exceed this proportion of missingness - defaul
#return list of the following elements:
# prop.miss Proportion of missing data
# summary.cols Matrix of summary of missingness by variable
# problem.cols Vector of column names with too much missingness
# problem.rows Vector of row indexes with too much missingness
# cor.tuples Vector of pairwise columns with too much correlation - names separated by period
# X.dropped Dataframe with problematic rows and columns dropped
missingness <- function(X, row.crit = .5, col.crit = .5){
  #give unnamed variables in X the number corresponding to location
  colnames(X) <- colnames(X, do.NULL = FALSE, prefix='col')
  #proportion of missing data
  prop.miss <- sum(is.na(X))/prod(dim(X))
  #missingness by column
  summary.cols <- matrix(NA, nrow=2, ncol=dim(X)[2], dimnames = list(c('Number Missing', 'Proportion Mis
  summary.cols[1,] <- apply(X, 2, function(x) sum(is.na(x)))
  summary.cols[2,] <- summary.cols[1,]/dim(X)[1]
  #problematic columns
  problem.cols <- names(which(summary.cols[2,]>col.crit))
  #problematic rows
  problem.rows <- which(apply(X, 1, function(x){
    sum(is.na(x)) > row.crit*dim(X)[2]
  }))
  #pairwise variables with high correlation
  cor.mat <- cor(X, use='pair')
  cor.mat[is.na(cor.mat)] <- 0
```

```
  cor.mat[lower.tri(cor.mat, diag=TRUE)] <- 0
  cor.tuples <- names(unlist(apply(cor.mat, 2, function(x) which(x >= .8))))
  #dataframe dropping problematic rows and columns
  X.dropped <- as.data.frame(X)[-problem.rows, colnames(X)!=problem.cols]
  return(list(prop.miss=prop.miss, summary.cols=summary.cols, problem.cols=problem.cols, problem.rows=pr
}
#example
set.seed(1313)
X <- matrix(rnorm(50), ncol=5, nrow=10)
#add 2 correlated columns
X <- cbind(X, X[,1]+rnorm(1))
X <- cbind(X, X[,3]+rnorm(1))
#add missingness
(X <- matrix(sample(c(NA, 1), 70, replace=TRUE, prob=c(.4,.6)), nrow=10)*X)
```

```
##             [,1]        [,2]       [,3]       [,4]       [,5]       [,6]
##  [1,]         NA -0.40333656  1.0926134         NA -0.8628869         NA
##  [2,]  0.5513848  0.81858632         NA -0.7687996 -1.5642445 -0.8854046
##  [3,]         NA -0.47022214 -0.1359219 -0.6195070  0.4320451 -1.0831551
##  [4,] -0.7039707 -0.58408678  0.6119054         NA         NA -2.1407601
##  [5,]         NA  0.04865912 -0.4522788 -2.4054508         NA         NA
##  [6,] -0.4236903          NA         NA -0.5576746 -0.6069164 -1.8604797
##  [7,] -0.2177472  1.34418537 -0.2085219         NA -0.8071041 -1.6545365
##  [8,]         NA          NA         NA         NA  0.2077456 -0.8925098
##  [9,]  1.9105651          NA         NA  1.0581734 -1.2657258  0.4737758
## [10,]  0.9588508  1.02474942         NA         NA         NA -0.4779386
##             [,7]
##  [1,]  0.2906410
##  [2,]         NA
##  [3,] -0.9378942
##  [4,]         NA
##  [5,]         NA
##  [6,]         NA
##  [7,] -1.0104943
##  [8,]  0.9699928
##  [9,]         NA
## [10,]         NA
```

```
missingness(X)
```

```
## $prop.miss
## [1] 0.4
##
## $summary.cols
##                    col1 col2 col3 col4 col5 col6 col7
## Number Missing      4.0  3.0  5.0  5.0  3.0  2.0  6.0
## Proportion Missing  0.4  0.3  0.5  0.5  0.3  0.2  0.6
##
## $problem.cols
## [1] "col7"
##
## $problem.rows
## [1]  5  8 10
##
```

```
## $cor.tuples
## [1] "col4.col1" "col4.col3" "col6.col1" "col6.col4" "col7.col3"
##
## $X.dropped
##         col1       col2       col3       col4       col5       col6
## 1         NA -0.4033366  1.0926134         NA -0.8628869         NA
## 2  0.5513848  0.8185863         NA -0.7687996 -1.5642445 -0.8854046
## 3         NA -0.4702221 -0.1359219 -0.6195070  0.4320451 -1.0831551
## 4 -0.7039707 -0.5840868  0.6119054         NA         NA -2.1407601
## 6 -0.4236903         NA         NA -0.5576746 -0.6069164 -1.8604797
## 7 -0.2177472  1.3441854 -0.2085219         NA -0.8071041 -1.6545365
## 9  1.9105651         NA         NA  1.0581734 -1.2657258  0.4737758
```

Replication

Table 1 shows the regression with omitting NAs. Table 2 shows the pooled regression aftermultiple imputation to 5 data frames. The missingness was randomly generated on all variables with probability .1 of being missing. All of the variables were used in the imputationp rocess, despite only using a selection of the variables in the regression. Maximizing the amount of information used to model the missing data tends to decrease bias in the estimations. We see that party ID and med.time switched signs in the pooled multiple imputation regression, though both estimates have 95% confidence intervals bounding zero. Category 3 in the model omitting NAs was not estimated, likely due to singularities when the NAs wereomitted. State ID has a smaller magnitude in the imputed regression, but the confidenceinterval does not bound zero. Leg influence gained in magnitude and the confidence intervalin the imputed regression does not bound zero, while the NA omitting model does. Election board is significant in both models, but has a smaller magnitude in the MICE regression. There are other variables in which the magnitude changes, but the signs are consistent. Generally, significance levels increased in the MICE imputation, which isn't terribly surprisingas the sample size is larger. The change in magnitude and the change in signs of some ofthe estimates is a little surprising, since the data is missing completely at random. Thismeans the data is MCAR, and omitting NAs shouldn't drastically change our substantiveinterpretations. This highlights the importance of not ignoring missing data. Even when itwas randomly missing, the interpretations changed.

```
asap <- read.table('asap.individual.dat.txt', header=TRUE)
set.seed(1919)
missing <- matrix(sample(c(NA,1), prod(dim(asap)), replace=TRUE, prob=c(.1, .9)), ncol=dim(asap)[2])
asap <- asap*missing
asap.lm <- lm(gov.influence ~ . - medt.contr - grp.influence, asap, na.action=na.omit)
library(xtable)
xtable(cbind(summary(asap.lm)$coef, confint(asap.lm)[-10,]), caption='Linear Model Omitting NAs')
```

% latex table generated in R 3.6.3 by xtable 1.8-4 package % Wed Mar 3 19:55:48 2021

```
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

	Estimate	Std. Error	t value	Pr(>|t|)	2.5 %	97.5 %
(Intercept)	16.62	4.15	4.00	0.00	8.26	24.98
state.id	-0.05	0.03	-1.78	0.08	-0.11	0.01
contracting	0.15	0.27	0.55	0.59	-0.40	0.70
leg.influence	0.21	0.16	1.32	0.19	-0.11	0.52
elect.board	-4.28	1.17	-3.66	0.00	-6.63	-1.93
years.tenure	-0.06	0.07	-0.85	0.40	-0.21	0.08
education	-0.78	0.60	-1.31	0.20	-1.98	0.42
party.ID	-0.20	0.29	-0.68	0.50	-0.78	0.39
category2	5.69	2.10	2.71	0.01	1.47	9.91
category4	2.84	1.69	1.68	0.10	-0.56	6.24
category5	2.99	2.24	1.34	0.19	-1.52	7.49
category6	2.93	2.01	1.46	0.15	-1.11	6.97
category7	1.06	1.61	0.66	0.51	-2.18	4.30
category8	3.95	1.58	2.49	0.02	0.76	7.13
category9	2.42	1.84	1.31	0.20	-1.29	6.13
category10	4.33	1.91	2.27	0.03	0.50	8.17
category11	1.98	1.72	1.15	0.26	-1.48	5.45
category12	1.88	3.61	0.52	0.61	-5.39	9.15
med.time	0.60	1.01	0.59	0.56	-1.44	2.63

Table 1: Linear Model Omitting NAs

```
asap.mids <- mice(asap, print=FALSE)
asap.mice <- lm.mids(gov.influence ~ . - medt.contr - grp.influence, asap.mids)
```

```
## Warning: Use with(imp, lm(yourmodel)).
```

```
xtable(summary(pool(asap.mice)), caption='Linear Model with MICE Imputation')
```

% latex table generated in R 3.6.3 by xtable 1.8-4 package % Wed Mar 3 19:55:57 2021

Example analysis comparison

The paper uses a probit link and explanatory variables POLITICS, READPAP, PTYTHNK, IDSTRNG, TAXLESS, DEATHPEN, LORDS, SCENGBEN, SCOPREF1, RSEX, RAGE, RSOCCLA2, TENURE1, PRESB, and INDPAR. I will use a logit link and I will drop SCENGBEN, which indicates those that think economic policies benefit Scotland more than England, and INDPAR, which is whether they favor independence, because these estimates were anomolies in the findings. I will also drop PRESB, as it doesn't appear in the data. Table 3 shows the results while omitting NAs. Table 4 shows the pooled results after mulitple imputation to 5 dataframes. All of the variables were used to model the missing data to decrease bias and increase the amount of information used in determining imputed values. Between the two models, the intercept, though substantively unimportant, is changed rather dramatically. POLITICS, READPAP, and PTYTHHINK are fairly close both in magnitude and their 95% confidence intervals. IDSTRONG decreases in magnitude but maintains significance. TAXLESS and DEATHPEN decrease in magnitude and also lose significance. LORDS and SCOPREF1 are similar in magnitude and confidence intervals. RSEX decreases in magnitude but the confidence intervals for both models bound zero. RAGE increases in magnitude but both confidence intervals bound zero. RSOCCLA2 increases slightly in magnitude, but gains significance (in the NA omitting model the confidence interval bounds zero but it does not in the MICE pooled regression). Finally, TENURE1 decreases in magnitude, and neither confidence interval bounds zero. In sum, there are not large differences in substantive interpretations, with some changes in magnitude and the confidence intervals. Also, the standard errors tend to be smaller in the imputed regression, which is sensible as the number of observations is larger.

	term	estimate	std.error	statistic	df	p.value
1	(Intercept)	8.80	1.17	7.51	49.99	0.00
2	state.id	-0.02	0.01	-1.79	50.57	0.08
3	contracting	0.03	0.08	0.44	185.27	0.66
4	leg.influence	0.43	0.05	9.34	43.97	0.00
5	elect.board	-2.90	0.42	-6.92	53.10	0.00
6	years.tenure	-0.06	0.03	-2.05	29.98	0.05
7	education	-0.04	0.16	-0.27	47.41	0.79
8	party.ID	0.10	0.09	1.05	53.04	0.30
9	category2	3.10	0.72	4.31	398.90	0.00
10	category3	2.51	0.93	2.70	32.90	0.01
11	category4	1.72	0.60	2.85	296.14	0.00
12	category5	2.99	0.64	4.64	571.75	0.00
13	category6	2.27	0.77	2.95	89.59	0.00
14	category7	1.02	0.59	1.74	282.04	0.08
15	category8	2.75	0.73	3.78	50.74	0.00
16	category9	3.16	0.74	4.29	38.70	0.00
17	category10	1.84	0.63	2.91	168.17	0.00
18	category11	0.82	0.60	1.36	109.76	0.18
19	category12	1.18	0.79	1.49	139.11	0.14
20	med.time	-0.15	0.31	-0.48	93.84	0.63

Table 2: Linear Model with MICE Imputation

```
scot <- read.table('http://jgill.wustl.edu/data/scotland.1997.missing.dat')
scot$VOTE <- ifelse(scot$VOTE==2, 1, 0)
scot.glm <- glm(VOTE ~ POLITICS + READPAP + PTYTHNK + IDSTRNG + TAXLESS + DEATHPEN + LORDS + SCOPREF1 +
xtable(cbind(summary(scot.glm)$coef, confint(scot.glm)), caption='Logit Omitting NAs')
scot.mids <- mice(scot, print=FALSE)
scot.mice <- glm.mids(VOTE ~ POLITICS + READPAP + PTYTHNK + IDSTRNG + TAXLESS + DEATHPEN + LORDS + SCOP
xtable(summary(pool(scot.mice)), caption='Pooled Logit with Multiple Imputation')
```

WAIC and LOO (leave-one-out)

Leave-one-out cross-validation (LOO) and the widely applicable information criterion (WAIC) are methods for estimating pointwise out-of-sample prediction accuracy from a fitted Bayesianmodel using the log-likelihood evaluated at the posterior simulations of the parameter values. LOO and WAIC have various advantages over simpler estimates of predictive error such as AIC and DIC but are less used in practice because they involve additional computational steps. Here we lay out fast and stable computations for LOO and WAIC that can be performed using existing simulation draws. We introduce an efficient computation of LOO using Pareto-smoothed importance sampling (PSIS), a new procedure for regularizing importance weights. Although WAIC is asymptotically equal to LOO, we demonstrate that PSIS-LOO is more robust in the finite case with weak priors or influential observations. As a byproduct of our calculations, we also obtain approximate standard errors for estimated predictive errors and for comparing of predictive errors between two models. We implement the computations in an R package called loo and demonstrate using models fit with the Bayesian inference package Stan.

loo is an R package that allows users to compute efficient approximate leave-one-out cross-validation for fitted Bayesian models, as well as model weights that can be used to average predictive distributions. The loo package implements the fast and stable computations for approximate LOO-CV and WAIC.

From existing posterior simulation draws, we compute approximate LOO-CV using Pareto smoothed im-

portance sampling (PSIS), a new procedure for regularizing importance weights. As a byproduct of our calculations, we also obtain approximate standard errors for estimated predictive errors and for comparing predictive errors between two models. We recommend PSIS-LOO-CV instead of WAIC, because PSIS provides useful diagnostics and effective sample size and Monte Carlo standard error estimates.

```
#model selection
library(rstan)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)

load('merged.Rdata')

mergedY = merged[!is.na(merged$avgAA),]
mergedY[,4:dim(mergedY)[2]] = apply(mergedY[,4:dim(mergedY)[2]], 2, scale)
y = mergedY$avgAA
g=model.matrix(~ pais -1, mergedY)
X = cbind(as.matrix(mergedY[,c('inflation', 'exports', 'imports', 'aid', 'propEmig')]),g)
K=dim(X)[2]
N=dim(X)[1]
t=mergedY$year-1995
X_corr=cbind(X,t)
M=dim(X_corr)[2]

model_code = '
data {
int<lower=1> N;
int<lower=1> K;
int<lower=1> M;
matrix[N,K] X;
matrix[N,M] X_corr;
vector[N] y;
}
parameters {
real<lower=0> nug;
real<lower=0> sig_sq;
vector<lower=0>[M] d1;
vector<lower=0>[M] d2;
vector[K] b;
}
model {
matrix[N,N] Sigma;
vector[N] mu;
matrix[N,K] Mu;
vector[M] d;

for(m in 1:M){
d1[m] ~ gamma(1,20);
d2[m] ~ gamma(10,10);
d[m] = .5*(d1[m] + d2[m]);
}
for (i in 1:(N-1)) {
for (j in (i+1):N) {
vector[M] summand;
for(m in 1:M){
```

```
summand[m] = -pow(X_corr[i,m] - X_corr[j,m],2)/d[m];
}
Sigma[i,j] = exp(sum(summand));
Sigma[j,i] = Sigma[i,j];
}
}
for (i in 1:N){
for(k in 1:K){
Mu[i,k] = X[i,k]*b[k];
}
mu[i]=sum(Mu[i,1:K]);
}
for (i in 1:N)
Sigma[i,i] = 1 + nug; // + jitter

sig_sq ~ inv_gamma(1,1);
nug ~ exponential(1);

b ~ normal(0,10);

y ~ multi_normal(mu,sig_sq*Sigma);
}
generated quantities {
   vector[N] log_lik;
   for (n in 1:N){
      log_lik[n] = normal_lpdf(y[n] | X[n]*b, sig_sq);
   }
}'


N = dim(X)[1]

set.seed(990)
fit_fit <- stan(model_code=model_code, data=list(X=X,N=N,K=K,y=y,M=M,X_corr=X_corr),
                iter=1000, chains=2)
save(fit_fit, file='modelSel1.Rdata')

X2=X[,-1]
X_corr2 = X_corr[,-1]
N2=dim(X2)[1]
K2=dim(X2)[2]
M2=dim(X_corr2)[2]
set.seed(9898)
fit_fit2 <- stan(model_code=model_code, data=list(X=X2,N=N2,K=K2,y=y,M=M2,X_corr=X_corr2),
                iter=1000, chains=2)
save(fit_fit2, file='modelSel2.Rdata')
```

```
load('modelSel1.Rdata')
load('modelSel2.Rdata')

library(loo)
```

```
## This is loo version 2.2.0
```

```
## - Online documentation and vignettes at mc-stan.org/loo
## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' ar
log_lik1 <- extract_log_lik(fit_fit)
log_lik2 <- extract_log_lik(fit_fit2)
(waic1 <- waic(log_lik1))

## Warning:
## 115 (44.9%) p_waic estimates greater than 0.4. We recommend trying loo instead.
##
## Computed from 1000 by 256 log-likelihood matrix
##
##           Estimate   SE
## elpd_waic   -558.3 43.5
## p_waic       231.4 22.4
## waic        1116.7 87.1
##
## 115 (44.9%) p_waic estimates greater than 0.4. We recommend trying loo instead.
(waic2 <- waic(log_lik2))

## Warning:
## 135 (52.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
##
## Computed from 1000 by 256 log-likelihood matrix
##
##           Estimate    SE
## elpd_waic   -651.7  52.9
## p_waic       305.9  29.5
## waic        1303.3 105.8
##
## 135 (52.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
print(compare(waic1, waic2), digits = 2)

## Warning: 'compare' is deprecated.
## Use 'loo_compare' instead.
## See help("Deprecated")
## elpd_diff        se
##    -93.31     18.38
loo1 = loo(log_lik1)

## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
loo2 = loo(log_lik2)

## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.

## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```r
print(compare(loo1,loo2))
```

```
## Warning: 'compare' is deprecated.
## Use 'loo_compare' instead.
## See help("Deprecated")
```

```
## elpd_diff        se
##     -95.9      18.8
```

Additional reading

http://www.stat.columbia.edu/~gelman/research/unpublished/loo_stan.pdf