Bayesian Psychometric Model Fit Methods

Lecture 5

Today's Lecture Objectives

1. Show how to use PPMC to evaluate absolute model fit in Bayesian psychometric models

2. Show how to use LOO and WAIC for relative model fit in Bayesian psychometric models

Example Data: Conspiracy Theories

Today's example is from a bootstrap resample of 177 undergraduate students at a large state university in the Midwest. The survey was a measure of 10 questions about their beliefs in various conspiracy theories that were being passed around the internet in the early 2010s. Additionally, gender was included in the survey. All items responses were on a 5-point Likert scale with:

1. Strongly Disagree

2. Disagree

3. Neither Agree or Disagree

4. Agree

5. Strongly Agree

PLEASE NOTE, THE PURPOSE OF THIS SURVEY WAS TO STUDY INDIVIDUAL BELIEFS REGARDING CONSPIRACIES. THE QUESTIONS CAN PROVOKE SOME STRONG EMOTIONS GIVEN THE WORLD WE LIVE IN CURRENTLY. ALL QUESTIONS WERE APPROVED BY UNIVERSITY IRB PRIOR TO THEIR USE.

Our purpose in using this instrument is to provide a context that we all may find relevant as many of these conspiracy theories are still prevalent today.

Conspiracy Theory Questions 1-5

Questions:

1. The U.S. invasion of Iraq was not part of a campaign to fight terrorism, but was driven by oil companies and Jews in the U.S. and Israel.

2. Certain U.S. government officials planned the attacks of September 11, 2001 because they wanted the United States to go to war in the Middle East.

3. President Barack Obama was not really born in the United States and does not have an authentic Hawaiian birth certificate.

4. The current financial crisis was secretly orchestrated by a small group of Wall Street bankers to extend the power of the Federal Reserve and further their control of the world's economy.

5. Vapor trails left by aircraft are actually chemical agents deliberately sprayed in a clandestine program directed by government officials.

Conspiracy Theory Questions 6-10

Questions:

6. Billionaire George Soros is behind a hidden plot to destabilize the American government, take control of the media, and put the world under his control.

7. The U.S. government is mandating the switch to compact fluorescent light bulbs because such lights make people more obedient and easier to control.

8. Government officials are covertly Building a 12-lane "NAFTA superhighway" that runs from Mexico to Canada through America's heartland.

9. Government officials purposely developed and spread drugs like crack-cocaine and diseases like AIDS in order to destroy the African American community.

10. God sent Hurricane Katrina to punish America for its sins.

Model Setup Today

Today, we will revert back to the graded response model assumptions to discuss how to estimate the latent variable standard deviation

$$P\left(Y_{ic} = c \mid \theta_p\right) = \begin{cases} 1 - P\left(Y_{i1} > 1 \mid \theta_p\right) & \text{if } c = 1 \\ P\left(Y_{ic-1} > c - 1 \mid \theta_p\right) - P\left(Y_{ic} > c \mid \theta_p\right) & \text{if } 1 < c < C_i \\ P\left(Y_{iC_i-1} > C_i - 1 \mid \theta_p\right) & \text{if } c = C_i \end{cases}$$

Where:

$$P\left(Y_{ic} > c \mid \theta\right) = \frac{\exp(-\tau_{ic} + \lambda_i\theta_p)}{1 + \exp(-\tau_{ic} + \lambda_i\theta_p)}$$

With:

- $C_i - 1$ Ordered thresholds: $\tau_1 < \tau_2 < \ldots < \tau_{C_i-1}$

We can convert thresholds to intercepts by multiplying by negative one: $\mu_c = -\tau_c$

Model Comparisons: One vs. Two Dimensions

We will fit two models to the data:

1. A unidimensional model **(see Lecture 4d here)**

2. A two-dimensionsal model **(see Lecture 4e here)**

   We know from previous lectures that the two-dimensional model had a correlation very close to one

- Such a high correlation made it seem implausible there were two dimensions

Posterior Predictive Model Checking for Absolute Fit in Bayesian Psychometric Models

Psychometric Model PPMC

Psychometric models can use posterior predictive model checking (PPMC) to assess how well they fit the data in an absolute sense

- Each iteration of the chain, each item is simulated using model parameter values from that iteration
- Summary statistics are used to evaluate model fit
  - Univariate measures (each item, separately):
    - Item mean
    - Item χ^2 (comparing observed data with each simulated data set)
  - Not very useful unless:
    - Parameters have "obscenely" informative priors
    - There are cross-item constraints on some parameters (such as all loadings are equal)
  - Bivariate measures (each pair of items)
    - For binary data: Tetrachoric correlations
    - For polytomous data: Polychoric correlations (but difficult to estimate with small samples), pearson correlations
    - For other types of data: Pearson correlations

Problems with PPMC

Problems with PPMC include

- No uniform standard for which statistics to use

    - Tetrachoric correlations? Pearson correlations?

- No uniform standard by which data should fit, absolutely

    - Jihong Zhang has some work on this topic, though:

        - Paper in _**Structural Equation Modeling**_

        - Dissertation on PPMC with M2 statistics (working on publishing)

- No way to determine if a model is overparameterized (too complicated)

    - Fit only improves to a limit

Implementing PPMC in Stan (one θ)

```stan
 1  generated quantities{
 2
 3    // for PPMC:
 4    array[nItems, nObs] int<lower=0> simY;
 5
 6    for (item in 1:nItems){
 7      for (obs in 1:nObs){
 8        // generate data based on distribution and model
 9        simY[item, obs] = ordered_logistic_rng(lambda[item]*theta[obs], thr[item]);
10
11      }
12    }
13  }
```

Notes:

- Generated quantities block is where to implement PPMC

- Each type of distribution also has a random number generator

  - Here, `ordered_logistic_rng` goes with `ordered_logistic`

- Each may have some issue in types of inputs (had to go person-by-person in this block)

- Rather than have Stan calculate statistics, I will do so in R

Implementing PPMC in Stan (two θs)

```
1  generated quantities{
2    // for PPMC:
3    array[nItems, nObs] int<lower=0> simY;
4
5    for (item in 1:nItems){
6      for (obs in 1:nObs){
7        // generate data based on distribution and model
8        simY[item, obs] = ordered_logistic_rng(thetaMatrix[obs,]*lambdaMatrix[item,1:nFactors]', thr[item]);
9
10     }
11   }
12 }
```

Notes:

- Very similar to one dimension–just using the syntax from the model block within the `ordered_logistic_rng` function

PPMC Processing

Stan generated a lot of data–but now we must take it from the format of Stan and process it:

```
1  # setting up PPMC
2  simData = modelOrderedLogit_samples$draws(variables = "simY", format = "draws_matrix")
3  colnames(simData)
4  dim(simData)
5
6
7  # set up object for storing each iteration's PPMC data
8  nPairs = choose(10, 2)
9  pairNames = NULL
10 for (col in 1:(nItems-1)){
11   for (row in (col+1):nItems){
12     pairNames = c(pairNames, paste0("item", row, "_item", col))
13   }
14 }
```

PPMC Calculating in R

```r
1  PPMCsamples = list()
2
3  PPMCsamples$correlation = NULL
4  PPMCsamples$mean = NULL
5
6  # loop over each posterior sample's simulated data
7
8  for (sample in 1:nrow(simData)){
9
10   # create data frame that has observations (rows) by items (columns)
11   sampleData = data.frame(matrix(data = NA, nrow = nObs, ncol = nItems))
12
13   for (item in 1:nItems){
14     itemColumns = colnames(simData)[grep(pattern = paste0("simY\\[", item, "\\,"), x = colnames(simData))]
15     sampleData[,item] = t(simData[sample, itemColumns])
16   }
17   # with data frame created, apply functions of the data:
18
19   # calculate item means
20   PPMCsamples$mean = rbind(PPMCsamples$mean, apply(X = sampleData, MARGIN = 2, FUN = mean))
21
22   # calculate pearson correlations
23   temp=cor(sampleData)
24   PPMCsamples$correlation = rbind(PPMCsamples$correlation, temp[lower.tri(temp)])
25
26
27 }
```
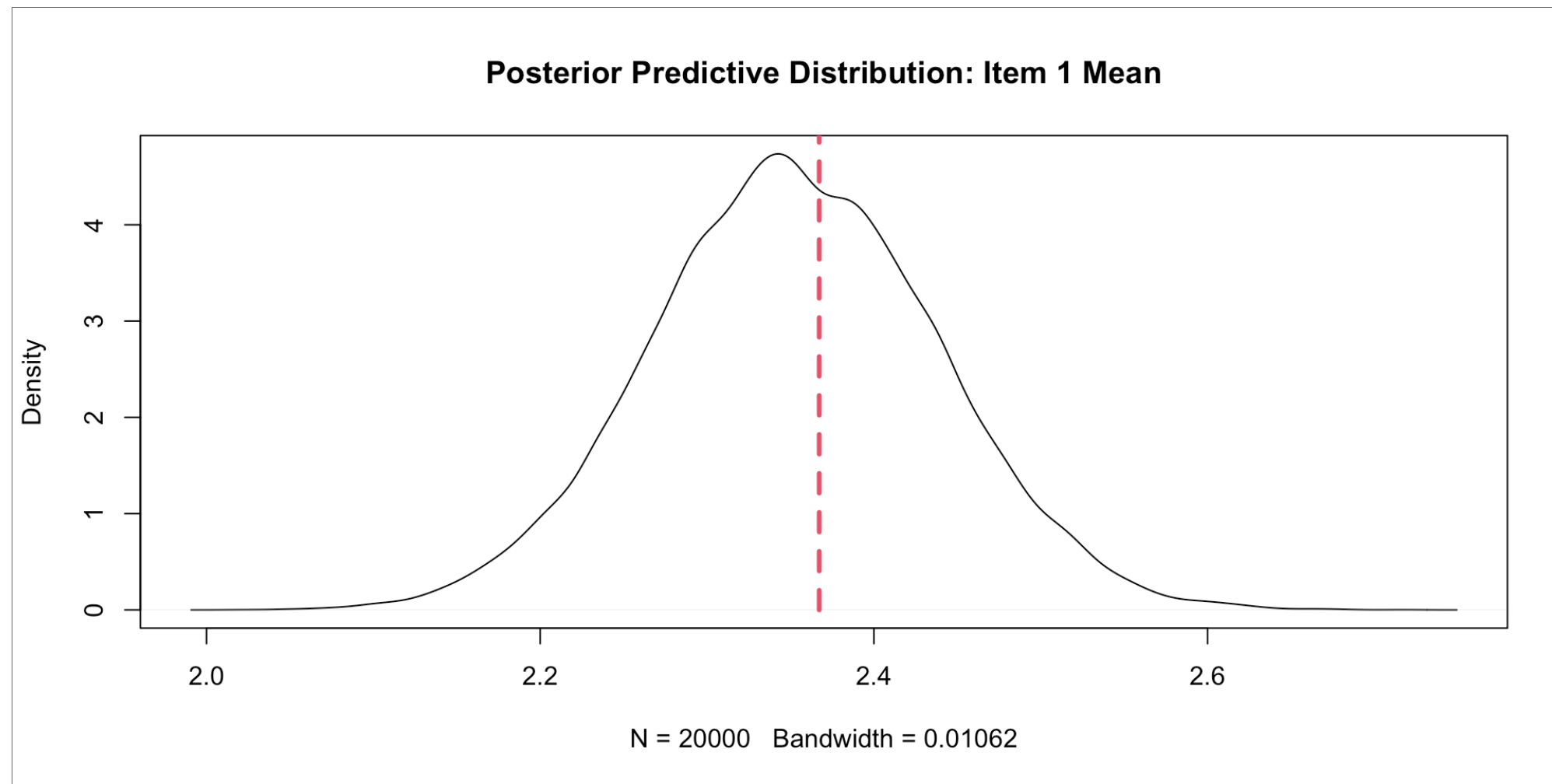
PPMC Results Tabulation in R (Mean)

```r
# next, build distributions for each type of statistic
meanSummary = NULL

# for means
for (item in 1:nItems){

  tempDist = ecdf(PPMCsamples$mean[,item])
  ppmcMean = mean(PPMCsamples$mean[,item])
  observedMean = mean(conspiracyItems[,item])
  meanSummary = rbind(
    meanSummary,
    data.frame(
      item = paste0("Item", item),
      ppmcMean = ppmcMean,
      observedMean = observedMean,
      residual = observedMean - ppmcMean,
      observedMeanPCT = tempDist(observedMean)
    )
  )

}
```
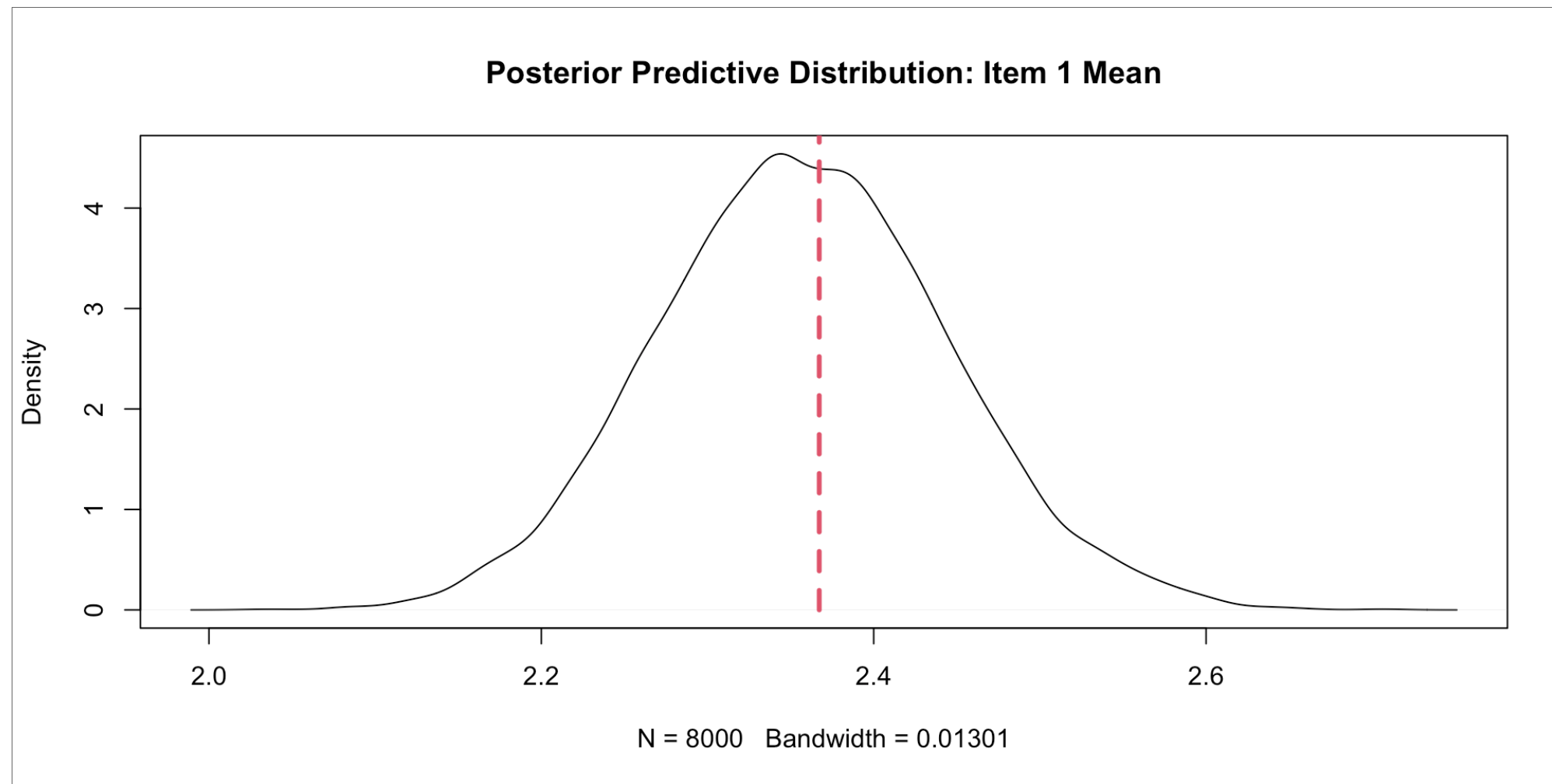
PPMC Results Tabulation in R (Correlation)

```r
1   # for pearson correlations
2   corrSummary = NULL
3
4   # for means
5   for (column in 1:ncol(PPMCsamples$correlation)){
6
7     # get item numbers from items
8     items = unlist(strsplit(x = colnames(PPMCsamples$correlation)[column], split = "_"))
9     item1num = as.numeric(substr(x = items[1], start = 5, stop = nchar(items[1])))
10    item2num = as.numeric(substr(x = items[2], start = 5, stop = nchar(items[2])))
11
12    tempDist = ecdf(PPMCsamples$correlation[,column])
13    ppmcCorr = mean(PPMCsamples$correlation[,column])
14    observedCorr = cor(conspiracyItems[,c(item1num, item2num)])[1,2]
15    pct = tempDist(observedCorr)
16    if (pct > .975 | pct < .025){
17      inTail = TRUE
18    } else {
19      inTail = FALSE
20    }
21    corrSummary = rbind(
22      corrSummary,
23      data.frame(
24        item1 = paste0("Item", item1num),
25        item2 = paste0("Item", item2num),
26        ppmcCorr = ppmcCorr,
27        observedCorr = observedCorr,
28        residual = observedCorr - ppmcCorr,
29        observedCorrPCT = pct,
30        inTail = inTail
```

PPMC Mean Results: Example Item (One θ)

Posterior Predictive Distribution: Item 1 Mean

N = 20000   Bandwidth = 0.01062

PPMC Mean Results: Example Item (Two θs)

Posterior Predictive Distribution: Item 1 Mean

N = 8000   Bandwidth = 0.01301

PPMC Mean Results

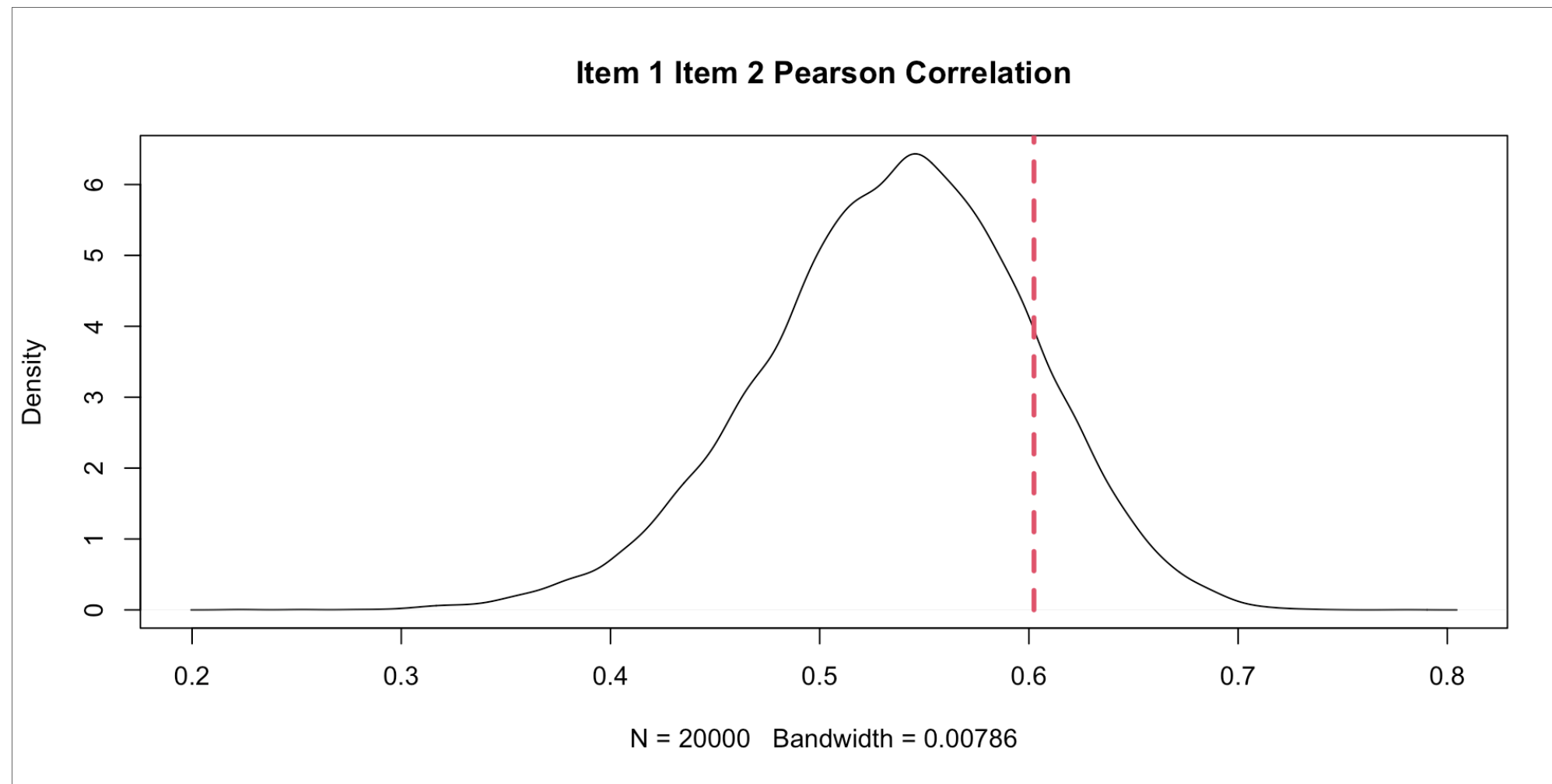## One θ

```
        item ppmcMean observedMean       residual observedMeanPCT
1    Item1 2.352560     2.367232  0.014671186        0.58340
2    Item2 1.958342     1.954802 −0.003539831        0.47495
3    Item3 1.862719     1.875706  0.012987571        0.58610
4    Item4 2.008068     2.011299  0.003231638        0.53915
5    Item5 1.978207     1.983051  0.004843503        0.53005
6    Item6 1.880974     1.892655  0.011681356        0.61725
7    Item7 1.748335     1.723164 −0.025171469        0.37290
8    Item8 1.840580     1.841808  0.001228249        0.51550
9    Item9 1.825317     1.807910 −0.017407062        0.43230
10 Item10 1.535198     1.519774 −0.015424294        0.45225
```
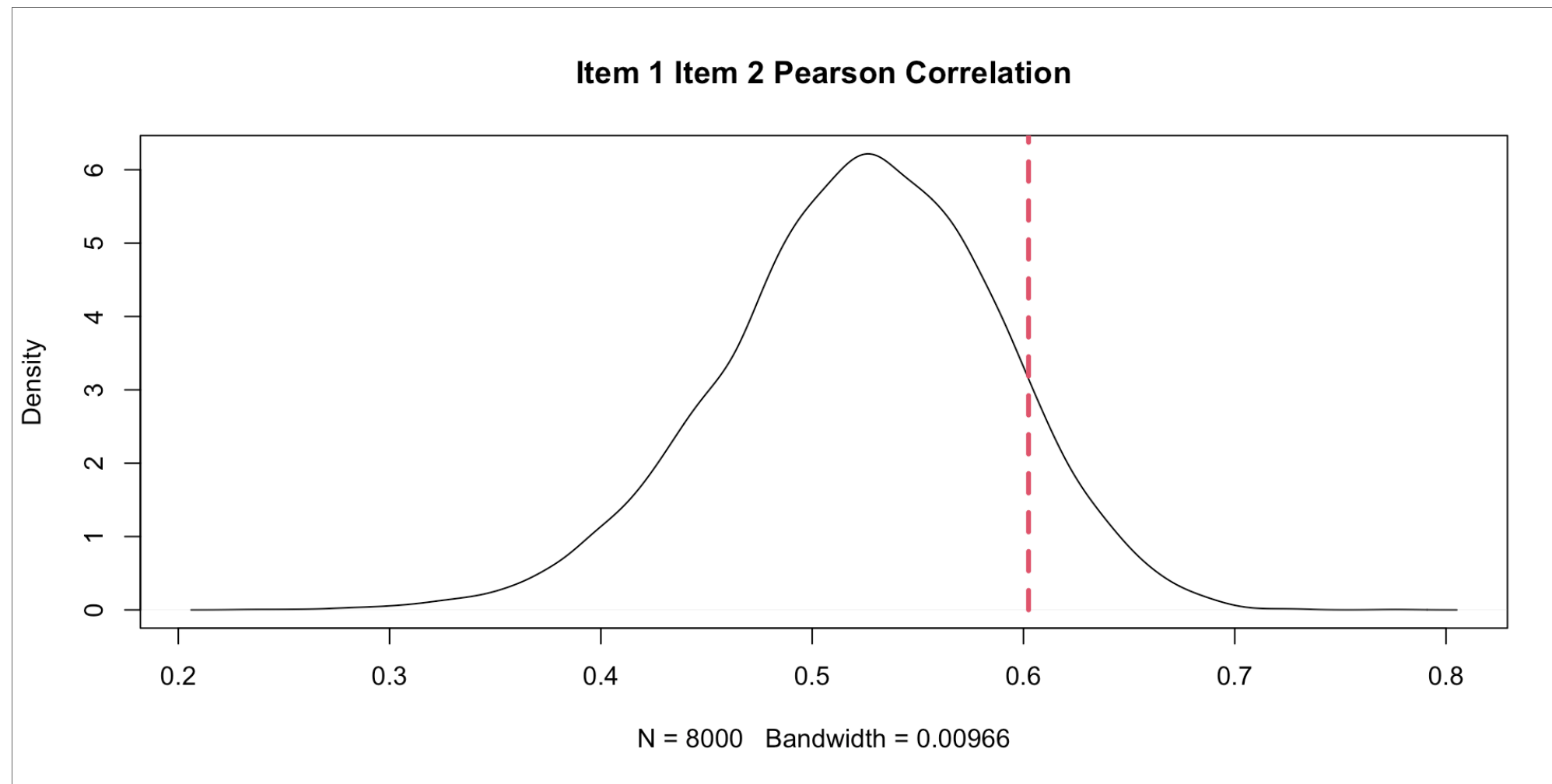
Two θs

```
        item ppmcMean observedMean       residual observedMeanPCT
1    Item1 2.357626     2.367232  0.0096059322       0.559500
2    Item2 1.969614     1.954802 −0.0148114407       0.418500
3    Item3 1.874785     1.875706  0.0009216102       0.528000
4    Item4 2.016655     2.011299 −0.0053552260       0.493000
5    Item5 1.988220     1.983051 −0.0051687853       0.465375
6    Item6 1.895842     1.892655 −0.0031864407       0.500375
7    Item7 1.758951     1.723164 −0.0357874294       0.326125
8    Item8 1.853244     1.841808 −0.0114357345       0.424500
9    Item9 1.835992     1.807910 −0.0280819209       0.373625
10 Item10 1.553063     1.519774 −0.0332888418       0.376125
```

PPMC Correlation Results: Example Item Pair (One θ)

Item 1 Item 2 Pearson Correlation

N = 20000   Bandwidth = 0.00786

PPMC Correlation Results: Example Item Pair (Two θs)

Item 1 Item 2 Pearson Correlation

N = 8000   Bandwidth = 0.00966

PPMC Correlation Results

	item1	item2	observedCorr	ppmcCorr.x	residual.x	observedCorrPCT.x	inTail.x
1	Item10	Item1	0.2568334	0.4131131	-0.156279686	0.01655	TRUE
2	Item10	Item2	0.4121264	0.4917030	-0.079576626	0.14375	FALSE
3	Item10	Item3	0.5670156	0.4522000	0.114815615	0.93835	FALSE
4	Item10	Item4	0.4516124	0.4890239	-0.037411565	0.28790	FALSE
5	Item10	Item5	0.5822203	0.5567640	0.025456304	0.62005	FALSE
6	Item10	Item6	0.5092003	0.5501939	-0.040993565	0.25890	FALSE
7	Item10	Item7	0.4420205	0.4992130	-0.057192487	0.22315	FALSE
8	Item10	Item8	0.5457127	0.5434373	0.002275470	0.48760	FALSE
9	Item10	Item9	0.4244916	0.4940797	-0.069588089	0.18525	FALSE
10	Item2	Item1	0.6024186	0.5357365	0.066682114	0.85585	FALSE
11	Item3	Item1	0.3473072	0.4931202	-0.145812943	0.01745	TRUE
12	Item3	Item2	0.5176819	0.5569093	-0.039227387	0.26420	FALSE
13	Item4	Item1	0.5700901	0.5439901	0.026099947	0.64290	FALSE
14	Item4	Item2	0.5815808	0.6090530	-0.027472228	0.30375	FALSE
15	Item4	Item3	0.4973987	0.5615192	-0.064120482	0.15380	FALSE
16	Item5	Item1	0.5526073	0.6008370	-0.048229709	0.19820	FALSE
17	Item5	Item2	0.6512768	0.6835423	-0.032265485	0.25645	FALSE
18	Item5	Item3	0.5942163	0.6288536	-0.034637387	0.26495	FALSE
19	Item5	Item4	0.6267160	0.6877014	-0.060985378	0.11495	FALSE
20	Item6	Item1	0.5435771	0.5996223	-0.056045208	0.16460	FALSE
21	Item6	Item2	0.6835282	0.6802492	0.003279028	0.49990	FALSE
22	Item6	Item3	0.6601292	0.6276426	0.032486611	0.68855	FALSE
23	Item6	Item4	0.6251455	0.6853371	-0.060191526	0.11720	FALSE
24	Item6	Item5	0.7366890	0.7696183	-0.032929309	0.18445	FALSE
25	Item7	Item1	0.4084280	0.5196226	-0.111184764	0.04695	FALSE

Count of Items in Misfitting Pairs

One θ

```
badItems
 Item1  Item3 Item10  Item4  Item7  Item9
    2      2      1      1      1      1
```

Two θs

```
badItems2
 Item1 Item10
     1      1
```

Plotting Misfitting Items (One θ)
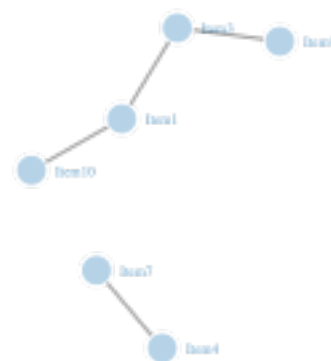
Relative Model Fit in Bayesian Psychometric Models

Relative Model Fit in Bayesian Psychometric Models

As with other Bayesian models, we can use WAIC and LOO to compare the model fit of two Bayesian models

- Of note: There is some debate as to whether or not we should marginalize across the latent variables

  - We won't do that here as that would involve a numeric integral

- What is needed: The conditional log likelihood for each observation at each step of the chain

  - Here, we have to sum the log likelihood across all items

  - There are built-in functions in Stan to do this

Implementing ELPD in Stan (one θ)

```stan
1  generated quantities{
2
3    // for LOO/WAIC:
4    vector[nObs] personLike = rep_vector(0.0, nObs);
5
6    for (item in 1:nItems){
7      for (obs in 1:nObs){
8        // calculate conditional data likelihood for LOO/WAIC
9        personLike[obs] = personLike[obs] + ordered_logistic_lpmf(Y[item, obs] | lambda[item]*theta[obs], thr[item]);
10     }
11   }
12 }
```

Notes:

- `ordered_logistic_lpmf` needs the observed data to work (first argument)

- `vector[nObs] personLike = rep_vector(0.0, nObs);` is needed to set the values to zero at each iteration

Implementing ELPD in Stan (two θs)

```
 1  generated quantities{
 2
 3    // for LOO/WAIC:
 4    vector[nObs] personLike = rep_vector(0.0, nObs);
 5
 6    for (item in 1:nItems){
 7      for (obs in 1:nObs){
 8        // calculate conditional data likelihood for LOO/WAIC
 9        personLike[obs] = personLike[obs] + ordered_logistic_lpmf(Y[item, obs] | thetaMatrix[obs,]*lambdaMatrix[item,1:nFactors]', thr[item]);
10      }
11    }
12  }
```

Notes:

- `ordered_logistic_lpmf` needs the observed data to work (first argument)

- `vector[nObs] personLike = rep_vector(0.0, nObs);` is needed to set the values to zero at each iteration

Comparing WAIC Values

```
1  # model comparisons
2  waic(x = modelOrderedLogit_samples$draws("personLike"))
```

```
Computed from 20000 by 177 log-likelihood matrix

          Estimate    SE
elpd_waic  -1482.6  70.4
p_waic       137.7   5.1
waic        2965.2 140.8

173 (97.7%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

```
1  waic(x = modelOrderedLogit2D_samples$draws("personLike"))
```

```
Computed from 8000 by 177 log-likelihood matrix

          Estimate    SE
elpd_waic  -1490.6  68.0
p_waic       142.6   4.6
waic        2981.1 136.1

174 (98.3%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```

Smaller is better, so the unidimensional model wins (but very high SE for both)

Comparing LOO Values

```
1  modelOrderedLogit_samples$loo(variables = "personLike")
```

```
Computed from 20000 by 177 log-likelihood matrix

          Estimate    SE
elpd_loo  -1542.0   71.0
p_loo       197.1    5.8
looic      3084.0  141.9
------
Monte Carlo SE of elpd_loo is NA.

Pareto k diagnostic values:
                         Count  Pct.    Min. n_eff
(-Inf, 0.5]   (good)        0    0.0%   <NA>
 (0.5, 0.7]   (ok)          4    2.3%   426
   (0.7, 1]   (bad)       156   88.1%    31
   (1, Inf)   (very bad)   17    9.6%    10
See help('pareto-k-diagnostic') for details.
```

```
1  modelOrderedLogit2D_samples$loo(variables = "personLike")
```

```
Computed from 8000 by 177 log-likelihood matrix

          Estimate    SE
elpd_loo  -1541.3   68.1
p_loo       193.2    5.1
looic      3082.5  136.2
------
Monte Carlo SE of elpd_loo is NA.

Pareto k diagnostic values:
                         Count  Pct.    Min. n_eff
(-Inf, 0.5]   (good)        1    0.6%   743
 (0.5, 0.7]   (ok)         19   10.7%    42
   (0.7, 1]   (bad)       135   76.3%     4
   (1, Inf)   (very bad)   22   12.4%     4
See help('pareto-k-diagnostic') for details.
```

LOO seems to (slightly) prefer the two-dimensional model (but by warnings about bad PSIS)

Comparing LOO Values

```
1 loo_compare(list(unidimensional = modelOrderedLogit_samples$loo(variables = "personLike"),
2                   twodimensional = modelOrderedLogit2D_samples$loo(variables = "personLike")))
```

```
               elpd_diff se_diff
twodimensional  0.0        0.0
unidimensional -0.7        4.9
```

Again, both models are very close in fit

Wrapping Up

Wrapping Up

Model fit is complicated for psychometric models

- Bayesian model fit methods are even more complicated than non-Bayesian methods

- Open area for research!

Thank you for a great semester