Bayesian Psychometric Model Fit Methods

Lecture 4f

https://jonathantemplin.com/bayesian-psychometric-modeling-fall-2024/

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: $\mu_c = -\tau_c$

6

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

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)

```
generated quantities{

  // for PPMC:
  array[nItems, nObs] int<lower=0> simY;

  for (item in 1:nItems){
    for (obs in 1:nObs){
      // generate data based on distribution and model
      simY[item, obs] = ordered_logistic_rng(lambda[item]*theta[obs], thr[item]);

    }
  }
}
```

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)

```
generated quantities{
  // for PPMC:
  array[nItems, nObs] int<lower=0> simY;

  for (item in 1:nItems){
    for (obs in 1:nObs){
      // generate data based on distribution and model
      simY[item, obs] = ordered_logistic_rng(thetaMatrix[obs,]*lambdaMatrix[item,1:nFactors]', thr[item]);

    }
  }
}
```

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:

```
# setting up PPMC
simData = modelOrderedLogit_samples$draws(variables = "simY", format = "draws_matrix")
colnames(simData)
dim(simData)


# set up object for storing each iteration's PPMC data
nPairs = choose(10, 2)
pairNames = NULL
for (col in 1:(nItems-1)){
  for (row in (col+1):nItems){
    pairNames = c(pairNames, paste0("item", row, "_item", col))
  }
}
```

PPMC Calculating in R

```r
PPMCsamples = list()

PPMCsamples$correlation = NULL
PPMCsamples$mean = NULL

# loop over each posterior sample's simulated data

for (sample in 1:nrow(simData)){

  # create data frame that has observations (rows) by items (columns)
  sampleData = data.frame(matrix(data = NA, nrow = nObs, ncol = nItems))

  for (item in 1:nItems){
    itemColumns = colnames(simData)[grep(pattern = paste0("simY\\[", item, "\\,"), x = colnames(simData))]
    sampleData[,item] = t(simData[sample, itemColumns])
  }
  # with data frame created, apply functions of the data:

  # calculate item means
  PPMCsamples$mean = rbind(PPMCsamples$mean, apply(X = sampleData, MARGIN = 2, FUN = mean))

  # calculate pearson correlations
  temp=cor(sampleData)
  PPMCsamples$correlation = rbind(PPMCsamples$correlation, temp[lower.tri(temp)])


}
```

14

PPMC Results Tabulation in R (Mean)

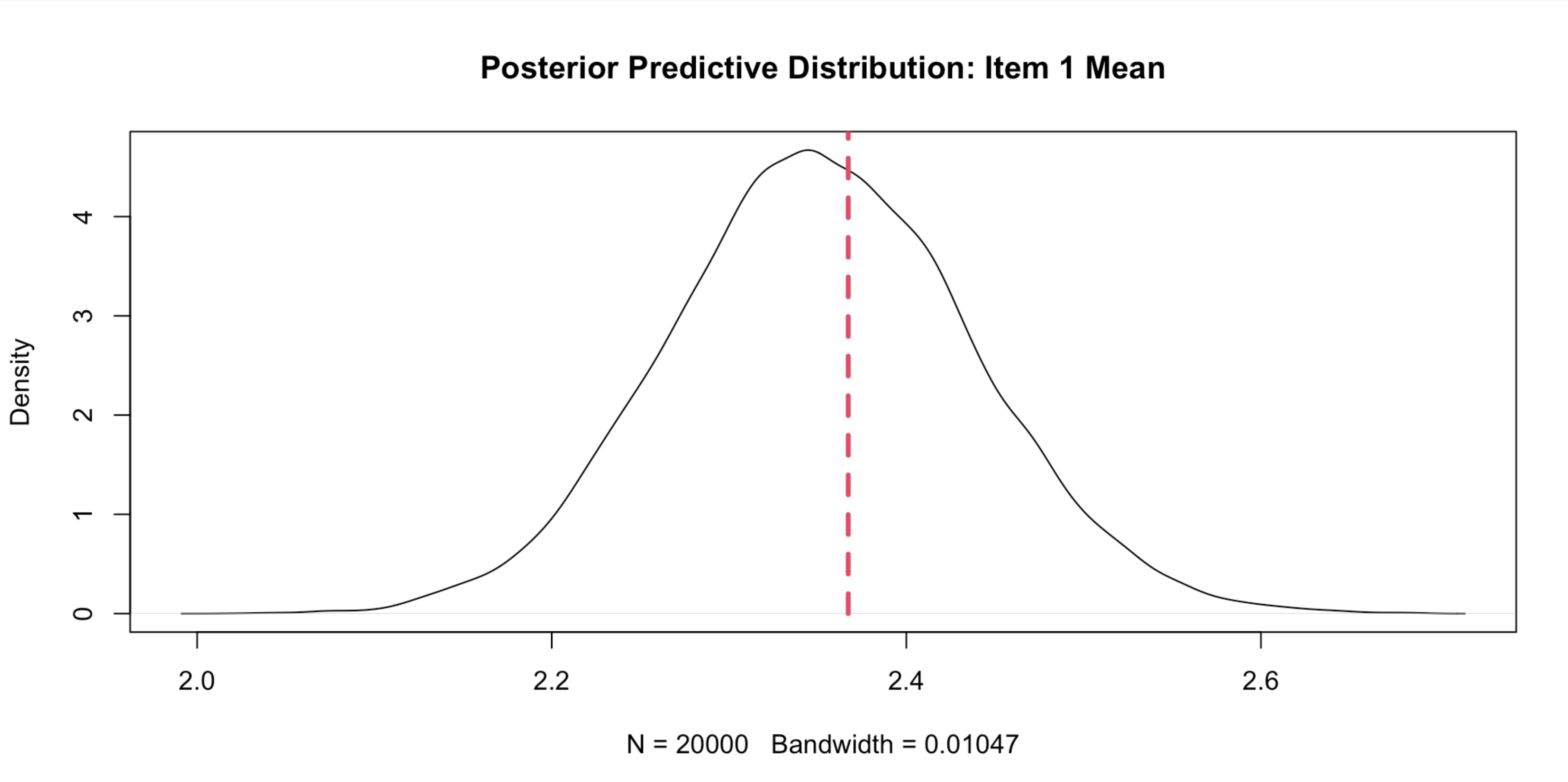
```r
# next, build distributions for each type of statistic
meanSummary = NULL

# 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)

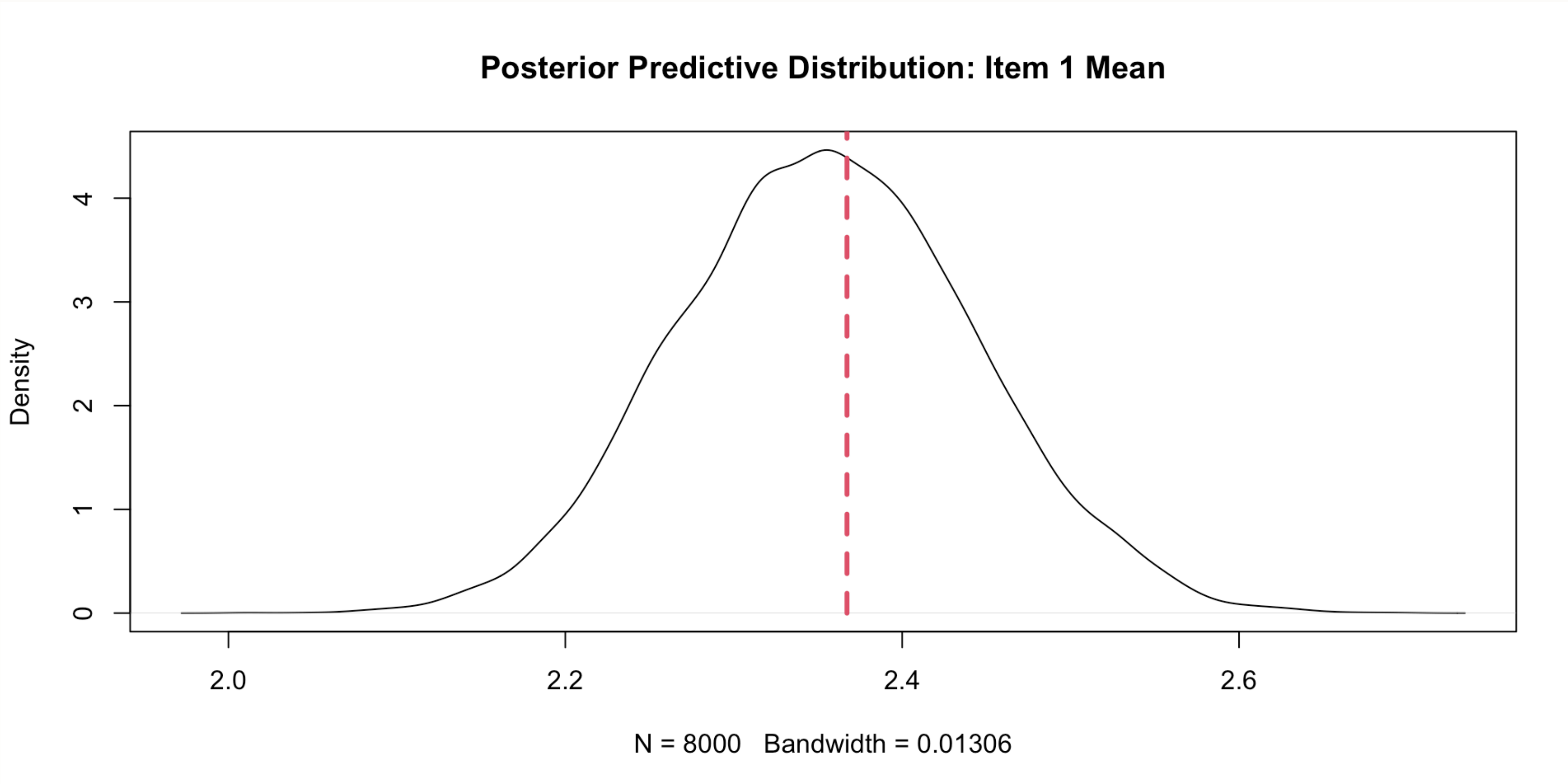
```
# for pearson correlations
corrSummary = NULL

# for means
for (column in 1:ncol(PPMCsamples$correlation)){

  # get item numbers from items
  items = unlist(strsplit(x = colnames(PPMCsamples$correlation)[column], split = "_"))
  item1num = as.numeric(substr(x = items[1], start = 5, stop = nchar(items[1])))
  item2num = as.numeric(substr(x = items[2], start = 5, stop = nchar(items[2])))

  tempDist = ecdf(PPMCsamples$correlation[,column])
  ppmcCorr = mean(PPMCsamples$correlation[,column])
  observedCorr = cor(conspiracyItems[,c(item1num, item2num)])[1,2]
  pct = tempDist(observedCorr)
  if (pct > .975 | pct < .025){
    inTail = TRUE
  } else {
    inTail = FALSE
  }
  corrSummary = rbind(
    corrSummary,
    data.frame(
      item1 = paste0("Item", item1num),
      item2 = paste0("Item", item2num),
      ppmcCorr = ppmcCorr,
      observedCorr = observedCorr,
      residual = observedCorr - ppmcCorr,
      observedCorrPCT = pct,
      inTail = inTail
```

16

PPMC Mean Results: Example Item (One θ)



PPMC Mean Results: Example Item (Two θs)



PPMC Mean Results

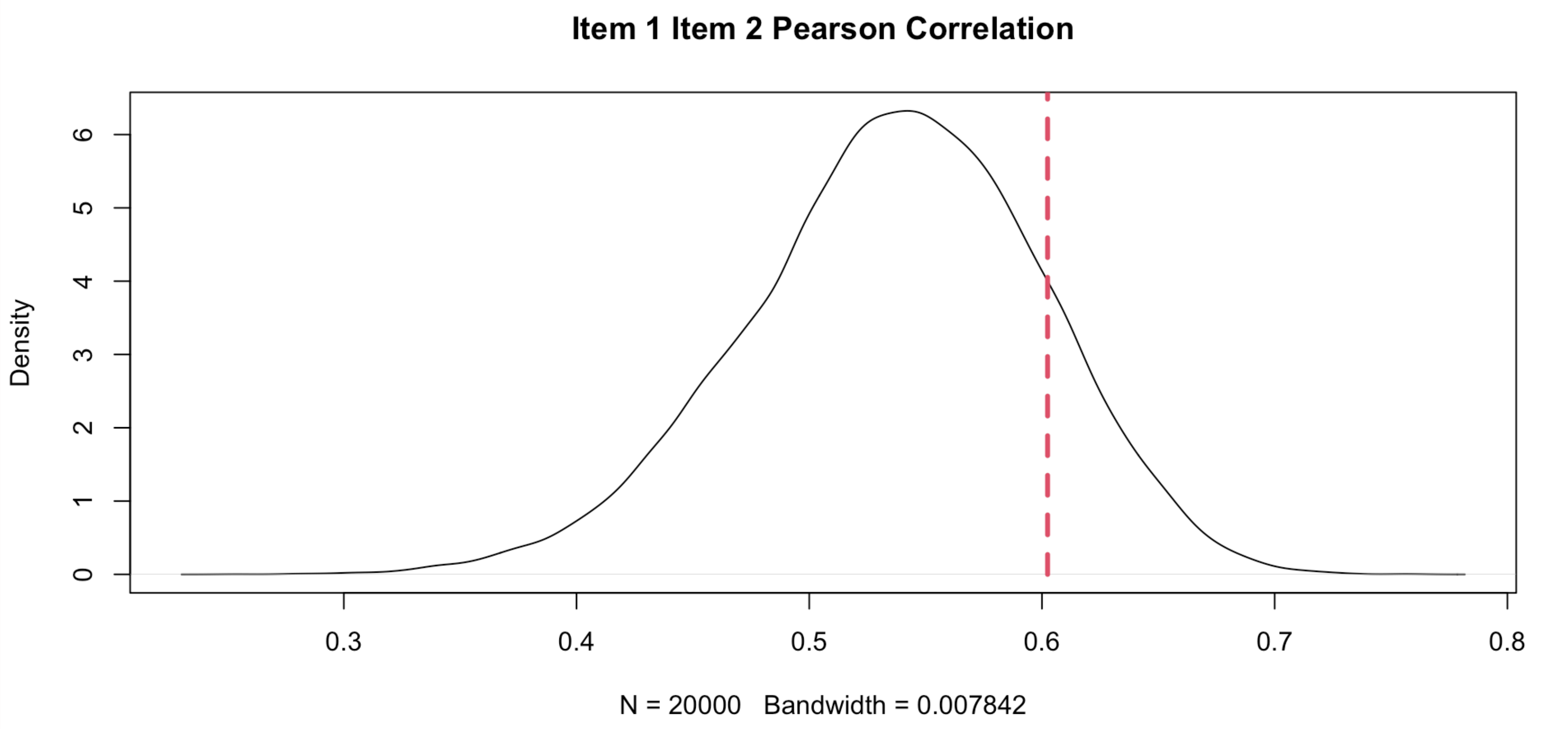
One θ

```
     item ppmcMean observedMean     residual observedMeanPCT
1   Item1 2.351961     2.367232  0.015270904         0.58750
2   Item2 1.958019     1.954802 -0.003216949         0.47760
3   Item3 1.862903     1.875706  0.012803672         0.58730
4   Item4 2.007206     2.011299  0.004093220         0.54505
5   Item5 1.978496     1.983051  0.004554802         0.52495
6   Item6 1.880662     1.892655  0.011992938         0.61725
7   Item7 1.748742     1.723164 -0.025578249         0.37475
8   Item8 1.840632     1.841808  0.001175989         0.51105
9   Item9 1.826041     1.807910 -0.018131638         0.42525
10 Item10 1.534652     1.519774 -0.014878249         0.45995
```

Two θs

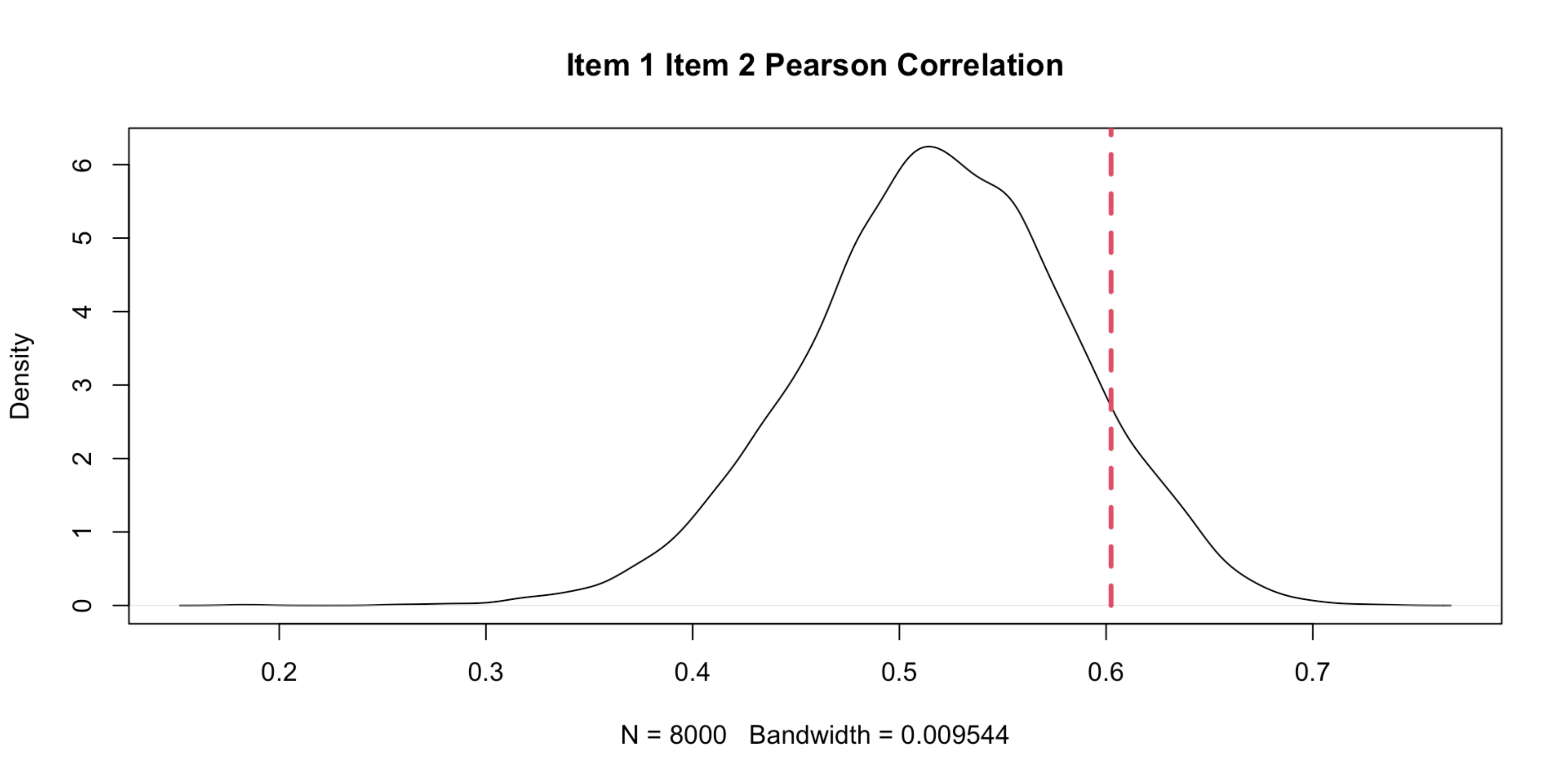
```
     item ppmcMean observedMean      residual observedMeanPCT
1   Item1 2.355617     2.367232  0.0116151130        0.567375
2   Item2 1.966982     1.954802 -0.0121800847        0.425375
3   Item3 1.875570     1.875706  0.0001362994        0.521375
4   Item4 2.017831     2.011299 -0.0065310734        0.487875
5   Item5 1.989239     1.983051 -0.0061878531        0.457625
6   Item6 1.895312     1.892655 -0.0026567797        0.499625
7   Item7 1.758679     1.723164 -0.0355148305        0.325875
8   Item8 1.853364     1.841808 -0.0115557910        0.422500
9   Item9 1.836593     1.807910 -0.0286829096        0.374125
10 Item10 1.552025     1.519774 -0.0322507062        0.376625
```

PPMC Correlation Results: Example Item Pair (One θ)



20

PPMC Correlation Results: Example Item Pair (Two θs)



21

PPMC Correlation Results

```
    item1 item2 observedCorr ppmcCorr.x   residual.x observedCorrPCT.x inTail.x
1  Item10 Item1    0.2568334  0.4138354 -0.157002024           0.01610     TRUE
2  Item10 Item2    0.4121264  0.4922525 -0.080126104           0.13835    FALSE
3  Item10 Item3    0.5670156  0.4528339  0.114181723           0.93580    FALSE
4  Item10 Item4    0.4516124  0.4885576 -0.036945276           0.29200    FALSE
5  Item10 Item5    0.5822203  0.5574134  0.024806856           0.61855    FALSE
6  Item10 Item6    0.5092003  0.5503082 -0.041107914           0.25660    FALSE
7  Item10 Item7    0.4420205  0.4994767 -0.057456184           0.22105    FALSE
8  Item10 Item8    0.5457127  0.5433053  0.002407421           0.48650    FALSE
9  Item10 Item9    0.4244916  0.4940490 -0.069557415           0.18350    FALSE
10  Item2 Item1    0.6024186  0.5359091  0.066509519           0.85330    FALSE
11  Item3 Item1    0.3473072  0.4935820 -0.146274751           0.01650     TRUE
12  Item3 Item2    0.5176819  0.5576809 -0.039998980           0.26080    FALSE
13  Item4 Item1    0.5700901  0.5428928  0.027197305           0.65445    FALSE
14  Item4 Item2    0.5815808  0.6093644 -0.027783572           0.30365    FALSE
15  Item4 Item3    0.4973987  0.5611649 -0.063766138           0.15725    FALSE
16  Item5 Item1    0.5526073  0.6009026 -0.048295276           0.20115    FALSE
17  Item5 Item2    0.6512768  0.6835498 -0.032272935           0.25520    FALSE
18  Item5 Item3    0.5942163  0.6295476 -0.035331390           0.26010    FALSE
19  Item5 Item4    0.6267160  0.6870688 -0.060352830           0.11680    FALSE
20  Item6 Item1    0.5435771  0.5995125 -0.055935394           0.16770    FALSE
21  Item6 Item2    0.6835282  0.6808187  0.002709457           0.49480    FALSE
22  Item6 Item3    0.6601292  0.6278075  0.032321713           0.69385    FALSE
23  Item6 Item4    0.6251455  0.6850634 -0.059917805           0.11470    FALSE
```

Count of Items in Misfitting Pairs

One θ

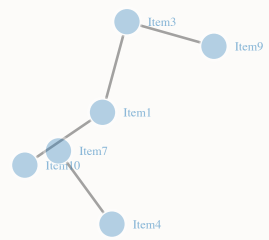
```
badItems
 Item1  Item3 Item10  Item4  Item7  Item9 
     2      2      1      1      1      1 
```

Two θs

```
badItems2
 Item1 Item10  Item3 
     2      1      1 
```

23

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)

```
generated quantities{

  // for LOO/WAIC:
  vector[nObs] personLike = rep_vector(0.0, nObs);

  for (item in 1:nItems){
    for (obs in 1:nObs){
      // calculate conditional data likelihood for LOO/WAIC
      personLike[obs] = personLike[obs] + ordered_logistic_lpmf(Y[item, obs] | lambda[item]*theta[obs], thr[item]);
    }
  }
}
```

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)

```
generated quantities{

  // for LOO/WAIC:
  vector[nObs] personLike = rep_vector(0.0, nObs);

  for (item in 1:nItems){
    for (obs in 1:nObs){
      // calculate conditional data likelihood for LOO/WAIC
      personLike[obs] = personLike[obs] + ordered_logistic_lpmf(Y[item, obs] | thetaMatrix[obs,]*lambdaMatrix[item,1:nFact
    }
  }
}
```

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

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

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

          Estimate    SE
elpd_waic  -1482.8  70.4
p_waic       137.8   5.1
waic        2965.6 140.8

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

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

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

          Estimate    SE
elpd_waic  -1491.6  68.1
p_waic       145.7   5.0
waic        2983.2 136.1

175 (98.9%) 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

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

```

Computed from 20000 by 177 log-likelihood matrix.

         Estimate    SE
elpd_loo  -1541.2  70.9
p_loo       196.2   5.8
looic      3082.4 141.9
------
MCSE of elpd_loo is NA.
MCSE and ESS estimates assume MCMC draws (r_eff in [0.5, 1.3]).

Pareto k diagnostic values:
                         Count Pct.    Min. ESS
(-Inf, 0.7]   (good)       6    3.4%   531
   (0.7, 1]   (bad)      151   85.3%   <NA>
   (1, Inf)   (very bad)  20   11.3%   <NA>
See help('pareto-k-diagnostic') for details.
```

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

```

Computed from 8000 by 177 log-likelihood matrix.

         Estimate    SE
elpd_loo  -1543.7  68.3
p_loo       197.8   5.7
looic      3087.3 136.6
------
MCSE of elpd_loo is NA.
MCSE and ESS estimates assume MCMC draws (r_eff in [0.1, 1.3]).

Pareto k diagnostic values:
                         Count Pct.    Min. ESS
(-Inf, 0.7]   (good)      14    7.9%   220
   (0.7, 1]   (bad)      141   79.7%   <NA>
   (1, Inf)   (very bad)  22   12.4%   <NA>
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

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

```
               elpd_diff se_diff
unidimensional  0.0       0.0   
twodimensional -2.5       5.0   
```

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!