Item Parameter Estimation III

(largely) Albert (1992):
Markov chain Monte Carlo (MCMC)

Back to the Normal Ogive model, with slightly different notation: *d* as the negative intercept—Albert refers to it as γ. Here we choose *d* to keep it roman, and distinguish it from other *c*s.

$$T_i(1_i|\theta) = \Phi(a_i\theta - d_i)$$

Albert also constrains the slope parameters to be positive; it is not clear to me why, so we don't do that here.

Rebuilding the model: This is the Lord & Novick derivation of the Normal Ogive model, but developed by Albert for a statistical purpose:

$$\mathrm{x}_{ij} = \begin{cases} 0 & \text{if} \quad Z_{ij} \le 0 \\ 1 & \text{if} \quad Z_{ij} > 0 \end{cases}$$

$$Z_{ij} \sim N(a_i\theta_j - d_i, 1) = N(\eta_{ij}, 1)$$

Collect the item parameters into: $\boldsymbol{\xi}_i = [a_i, d_i]$

Then the joint posterior density is:

$$\pi(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Z}|\mathbf{x}) = C \prod_{j=1}^{npersons} \left(\prod_{i=1}^{nitems} \Phi(Z_{ij}, 1)^{x_i} [1 - \Phi(Za_{ij}, 1)]^{1-x_i} \right) \phi(0,1)$$

Gibbs sampling:
Partition the parameter set to produce a list of conditional densities:

$$\begin{gathered} p_1(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_r, \text{data}) \\ p_2(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \boldsymbol{\theta}_3, \ldots, \boldsymbol{\theta}_r, \text{data}) \\ \vdots \\ p_r(\boldsymbol{\theta}_r|\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{r-1}, \text{data}) \end{gathered}$$

(Note that theta may represent an individual parameter or a set of parameters.)

Then, for the *j*th iteration:

1. Generate $\boldsymbol{\theta}_1^{(j)}$ from $p_1(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2^{(j-1)}, \ldots, \boldsymbol{\theta}_r^{(j-1)}, \text{data})$
2. Generate $\boldsymbol{\theta}_2^{(j)}$ from $p_2(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1^{(j)}, \boldsymbol{\theta}_3^{(j-1)}, \ldots, \boldsymbol{\theta}_r^{(j-1)}, \text{data})$

   $\vdots$

r. Generate $\boldsymbol{\theta}_r^{(j)}$ from $p_r(\boldsymbol{\theta}_r|\boldsymbol{\theta}_1^{(j)}, \ldots, \boldsymbol{\theta}_{r-1}^{(j)}, \text{data})$

At the end of each iteration one has a random draw from the (intractable, full) distribution. The draws are autocorrelated, but that is handled with "thinning" or some statistical adjustment.

The idea is to make the conditional distributions "easy to draw from" (aka, what's already programmed).

(almost) Albert's DA Gibbs for the Normal Ogive model:

1. Draw *Z*s from truncated normal distributions:

$$Z_{ij}|\boldsymbol{\xi}_i, \theta_j, x_{ij} \propto \begin{cases} \phi(z:\eta_{ij}, 1)I(z \le 0) & \text{if} \quad x_{ij} = 0 \\ \phi(z:\eta_{ij}, 1)I(z > 0) & \text{if} \quad x_{ij} = 1 \end{cases}$$

2. Draw θs ... see the next page ...

3. Draw ξs from a bivariate normal distribution:

$$\boldsymbol{\xi}_i|\mathbf{Z}, \boldsymbol{\Theta}, \mathbf{X} \sim N(\hat{\boldsymbol{\xi}}_i, \mathbf{U'U}^{-1})$$

where $\hat{\boldsymbol{\xi}}_i = (\mathbf{U'U})^{-1}\mathbf{U'z}_i$ and $\mathbf{U} = [\boldsymbol{\theta}, \mathbf{-1}]$

2. Draw θs:

Re-describe the model as regression:

$$Z_{ij} + d_i = a_i\theta_j + \epsilon_{ij} \qquad \epsilon_{ij} \sim N(0,1)$$

Then the mean (LS estimate) of θ given the zs is:

$$\hat{\theta}_j = \frac{\sum_i a_i(Z_{ij} + d_i)}{\sum_i a_i^2} \qquad v_j = \frac{1}{\sum_i a_i^2}$$

And θ is distributed:

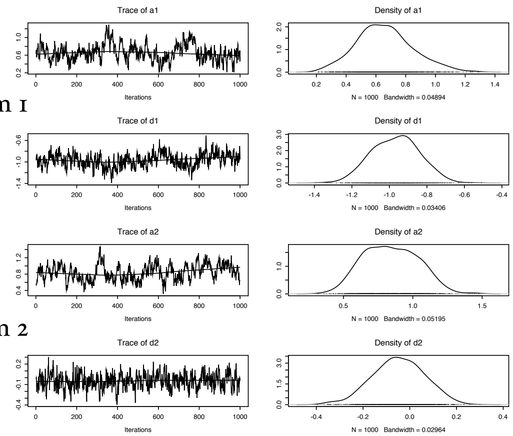
$$\theta_j \sim N\left(\frac{\hat{\theta}_j/v + 0/1}{1/v + 1/1}, \frac{1}{1/v + 1/1}\right) = N\left(\frac{\sum_i a_i(Z_{ij} + d_i)}{1/v + 1}, \frac{1}{1/v + 1}\right)$$

which is a weighted combination of the estimate given the Zs and the population mean, both weighted by the inverse of their variances.

Stouffer-Toby data, Albert MCMC, 1000 iterations

Item 1

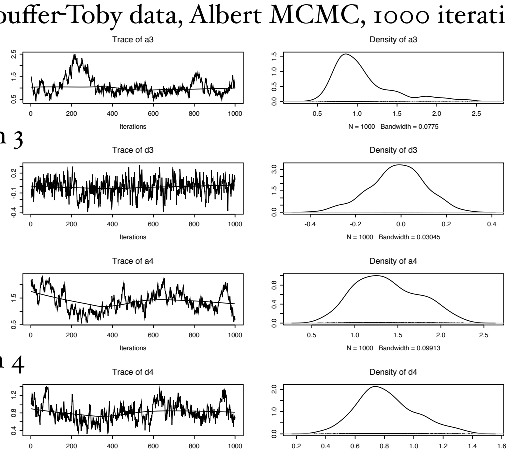
Item 2



Stouffer-Toby data, Albert MCMC, 1000 iterations

Item 3

Item 4



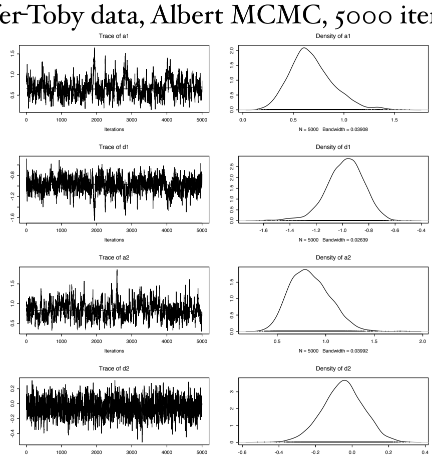
Stouffer-Toby data, Albert MCMC, 5000 iterations

```
[1] 2853.15  123.83 2992.60    0.00    0.00
is 48 minutes 1.25 GHz G4
```


Stouffer-Toby data, Albert MCMC, 5000 iterations

Item 1

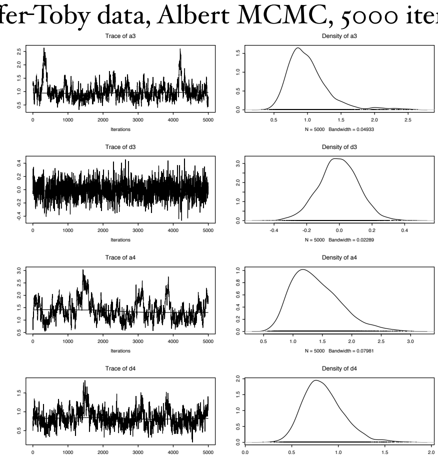
Item 2



Stouffer-Toby data, Albert MCMC, 5000 iterations

Item 3

Item 4



Stouffer-Toby data, Albert MCMC, 5000 iterations

```
Iterations = 1:4981
Thinning interval = 20
Number of chains = 1
Sample size per chain = 250

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:
                                                     ML       ML
          Mean     SD Naive SE Time-series SE  Estimate    S.E.
a1  0.697878 0.1770 0.011193        0.02065     0.647   0.188
d1 -0.961008 0.1199 0.007584        0.01329    -0.969   0.137
a2  0.848209 0.1695 0.010722        0.01873     0.810   0.201
d2 -0.044146 0.1134 0.007175        0.01168    -0.053   0.110
a3  0.994152 0.2489 0.015740        0.02808     0.940   0.243
d3  0.004158 0.1086 0.006871        0.01314    -0.009   0.117
a4  1.567595 0.3104 0.019633        0.03551     1.215   0.383
d4  0.883221 0.1889 0.011945        0.02139     0.767   0.197
```

13

Stouffer-Toby data, Albert MCMC, 5000 iterations

```
Iterations = 1000:5000
Thinning interval = 20
Number of chains = 1
Sample size per chain = 201

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:
                                                     ML       ML
          Mean     SD Naive SE Time-series SE  Estimate    S.E.
a1  0.701117 0.1892 0.013343        0.02434     0.647   0.188
d1 -0.964800 0.1212 0.008548        0.01621    -0.969   0.137
a2  0.826087 0.1626 0.011470        0.02004     0.810   0.201
d2 -0.035134 0.1087 0.007669        0.01291    -0.053   0.110
a3  1.004867 0.2644 0.018646        0.03287     0.940   0.243
d3 -0.001862 0.1123 0.007920        0.01531    -0.009   0.117
a4  1.634554 0.2905 0.020487        0.03737     1.215   0.383
d4  0.921751 0.1855 0.013082        0.02373     0.767   0.197
```

14

Stouffer-Toby data, Albert MCMC, 5000 iterations

```
> thin <- 1
> ResultsMCMC <- mcmc(Results, 1, niter, thin)
> autocorr(ResultsMCMC)
, , a1

              a1  ...
Lag 0  1.00000000  ...
Lag 1  0.89348938  ...
Lag 5  0.64665126  ...
Lag 10 0.48471071  ...
Lag 50 0.08532213  ...

, , d1

  ...          d1  ...
Lag 0    ... 1.00000000  ...
Lag 1    ... 0.77813724  ...
Lag 5    ... 0.46134274  ...
Lag 10   ... 0.30738820  ...
Lag 50   ... 0.03392549  ...
...
```

15

Patz & Junker (1999) describe a complete alternative set of (two) steps that eschew augmentation (the Zs) and use "Metropolis-Hastings within Gibbs" to directly draw:

1) $\theta_j \sim \pi(\theta|\boldsymbol{\xi}, \mathbf{X})$

2) $\boldsymbol{\xi}_i \sim \pi(\boldsymbol{\xi}_i|\boldsymbol{\theta}, \mathbf{X})$

They provide S-Plus code, an expanded version of which is at http://lib.stat.cmu.edu/DOS/S/

16

Hybrids (Data Augmentation and M-H within Gibbs) are possible, and have been used.

For the IRT model (alone), the advantages of MCMC estimation are unclear.

MCMC does provide a "complete" description of the posterior density of the parameters, but at great computational cost. There appears to be little other advantage for such a "simple" model.

17

A real advantage may be implementation of estimation systems for more complex models, or *ad hoc* models.

For example, a multilevel model that involves IRT models for categorical item responses of persons within classes, with repeated measurements and a regression model for the latent variable on some exogenous variables, might be challenging to estimate with ML—but in the MCMC approach that is "just extra steps." We'll get back to you on this one.