Estimation of the Reparameterized Unified Model Using MCMC

Jonathan Templin

University of Illinois at Urbana-Champaign

Cognitive Diagnosis Models

- The goal of cognitive diagnosis is to identify the set of cognitive skills (known as attributes) an examinee possesses.

Cognitive Diagnosis Models

- The goal of cognitive diagnosis is to identify the set of cognitive skills (known as attributes) an examinee possesses.
- Each item is created to measure a specific set of attributes.

Cognitive Diagnosis Models

- The goal of cognitive diagnosis is to identify the set of cognitive skills (known as attributes) an examinee possesses.
- Each item is created to measure a specific set of attributes.
- Central to models of cognitive diagnosis is the notion of the matrix indicating each attribute thought to be measured by each item, called the Q-matrix.

Cognitive Diagnosis Models

- The goal of cognitive diagnosis is to identify the set of cognitive skills (known as attributes) an examinee possesses.
- Each item is created to measure a specific set of attributes.
- Central to models of cognitive diagnosis is the notion of the matrix indicating each attribute thought to be measured by each item, called the Q-matrix.
- A concern of cognitive diagnosis is the accuracy or completeness of the Q-matrix.

Cognitive Diagnosis Models

- The goal of cognitive diagnosis is to identify the set of cognitive skills (known as attributes) an examinee possesses.
- Each item is created to measure a specific set of attributes.
- Central to models of cognitive diagnosis is the notion of the matrix indicating each attribute thought to be measured by each item, called the Q-matrix.
- A concern of cognitive diagnosis is the accuracy or completeness of the Q-matrix.
- Popular models include the DINA and NIDA give only cursory information regarding the completeness of the Q-matrix.

Unified Model

- A preliminary cognitive diagnosis model that attempts to measure the completeness of the Q-matrix is the Unified Model (DiBello, Stout, and Roussos, 1995):
$P\left(X_{ij}=1|\boldsymbol{\alpha}_i,\theta_i\right)=\left(1-p\right)\times$
$\left[d_j\prod_{k=1}^{K}\pi_{jk}^{\alpha_{ik}\times q_{jk}}r_{jk}^{(1-\alpha_{ik})\times q_{jk}}P_j\left(\theta_i+\Delta c_j\right)+\left(1-d_j\right)P_j\left(\theta_i\right)\right]$

Unified Model

- A preliminary cognitive diagnosis model that attempts to measure the completeness of the Q-matrix is the Unified Model (DiBello, Stout, and Roussos, 1995):
$$P\left(X_{ij}=1|\boldsymbol{\alpha}_i,\theta_i\right)=(1-p)\times$$
$$\left[d_j\prod_{k=1}^{K}\pi_{jk}^{\alpha_{ik}\times q_{jk}}r_{jk}^{(1-\alpha_{ik})\times q_{jk}}P_j\left(\theta_i+\Delta c_j\right)+\left(1-d_j\right)P_j\left(\theta_i\right)\right]$$
- The completeness of an item is measured by c_j

Unified Model

- A preliminary cognitive diagnosis model that attempts to measure the completeness of the Q-matrix is the Unified Model (DiBello, Stout, and Roussos, 1995):
$$P\left(X_{ij}=1|\boldsymbol{\alpha}_i,\theta_i\right)=\left(1-p\right)\times$$
$$\left[d_j\prod_{k=1}^{K}\pi_{jk}^{\alpha_{ik}\times q_{jk}}r_{jk}^{(1-\alpha_{ik})\times q_{jk}}P_j\left(\theta_i+\Delta c_j\right)+\left(1-d_j\right)P_j\left(\theta_i\right)\right]$$
- The completeness of an item is measured by c_j
- The parameters of this model were shown to be unidentified.

The Reparameterized Unified Model

- The RUM is an identified reparameterization of the Unified Model:

$$P\left(X_{ij}=1|\boldsymbol{\alpha}_i,\theta_i\right)=$$

$$\pi_j^*\prod_{k=1}^{K} r_{jk}^{*(1-\alpha_{ik})\times q_{jk}}\left(\frac{e^{1.701(c_j+\theta_i)}}{1+e^{1.701(c_j+\theta_i)}}\right)$$

The Reparameterized Unified Model

- The RUM is an identified reparameterization of the Unified Model:

$$P\left(X_{ij} = 1 | \boldsymbol{\alpha}_i, \theta_i\right) =$$

$$\pi_j^* \prod_{k=1}^{K} r_{jk}^{*(1-\alpha_{ik}) \times q_{jk}} \left(\frac{e^{1.701(c_j + \theta_i)}}{1 + e^{1.701(c_j + \theta_i)}} \right)$$

The Reparameterized Unified Model

- The RUM is an identified reparameterization of the Unified Model:

$$P\left(X_{ij}=1|\boldsymbol{\alpha}_i,\theta_i\right)=$$
$$\pi_j^*\prod_{k=1}^{K} r_{jk}^{*(1-\alpha_{ik})\times q_{jk}}\left(\frac{e^{1.701(c_j+\theta_i)}}{1+e^{1.701(c_j+\theta_i)}}\right)$$

- Unlike other cognitive diagnosis models, the RUM incorporates a continuous examinee variable θ, which, along with the c item parameter attempts to account for the lack of completeness of an item's Q-matrix entries.

Fusion Model

- In order to estimate the RUM, an additional model was created, called the Fusion Model.

Fusion Model

- In order to estimate the RUM, an additional model was created, called the Fusion Model.
- The Fusion Model is nearly identical to the RUM with the exception of the way the attributes are defined:

$$\alpha_{ik} = I(\tilde{\alpha}_{ik} > \phi^{-1}(1 - p_k))$$

Fusion Model

- In order to estimate the RUM, an additional model was created, called the Fusion Model.
- The Fusion Model is nearly identical to the RUM with the exception of the way the attributes are defined:

$$\alpha_{ik} = I(\tilde{\alpha}_{ik} > \phi^{-1}(1 - p_k))$$

- $I(\cdot)$ is the binary indicator function.

Fusion Model

- In order to estimate the RUM, an additional model was created, called the Fusion Model.
- The Fusion Model is nearly identical to the RUM with the exception of the way the attributes are defined:

$$\alpha_{ik} = I(\tilde{\alpha}_{ik} > \phi^{-1}(1 - p_k))$$

- $I(\cdot)$ is the binary indicator function.
- ϕ^{-1} is the inverse of the standard normal CDF.

Fusion Model

- In order to estimate the RUM, an additional model was created, called the Fusion Model.
- The Fusion Model is nearly identical to the RUM with the exception of the way the attributes are defined:

$$\alpha_{ik} = I(\tilde{\alpha}_{ik} > \phi^{-1}(1 - p_k))$$

- $I(\cdot)$ is the binary indicator function.
- ϕ^{-1} is the inverse of the standard normal CDF.
- $\tilde{\alpha}_{ik}$ is a continuous variable with joint distribution:

$$(\boldsymbol{\alpha}, \boldsymbol{\theta}) \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

Fusion Model

- In order to estimate the RUM, an additional model was created, called the Fusion Model.
- The Fusion Model is nearly identical to the RUM with the exception of the way the attributes are defined:

$$\alpha_{ik} = I(\tilde{\alpha}_{ik} > \phi^{-1}(1 - p_k))$$

- $I(\cdot)$ is the binary indicator function.
- ϕ^{-1} is the inverse of the standard normal CDF.
- $\tilde{\alpha}_{ik}$ is a continuous variable with joint distribution:

$$(\boldsymbol{\alpha}, \boldsymbol{\theta}) \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

- To estimate the Fusion Model the program *Arpeggio* was created.

Markov Chain Monte Carlo

- Due to the complexity of the Fusion Model, MCMC was chosen as an estimation method.

Markov Chain Monte Carlo

- Due to the complexity of the Fusion Model, MCMC was chosen as an estimation method.
- The MCMC algorithm is a stochastic process whereby the posterior distribution of the parameters given the data is sampled.

Markov Chain Monte Carlo

- Due to the complexity of the Fusion Model, MCMC was chosen as an estimation method.
- The MCMC algorithm is a stochastic process whereby the posterior distribution of the parameters given the data is sampled.
- Different methods of sampling from the posterior distribution exist.

Markov Chain Monte Carlo

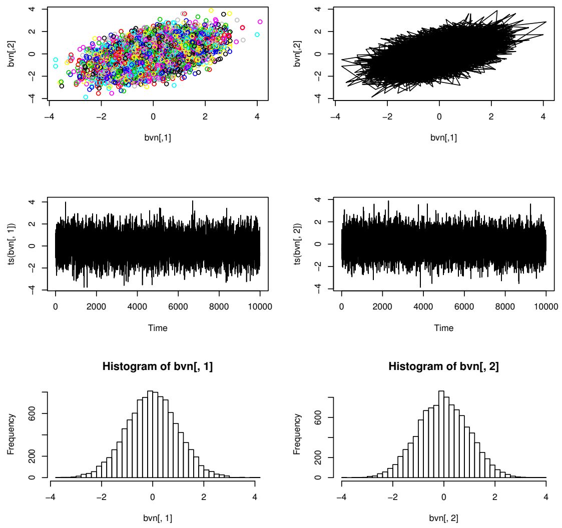
- Due to the complexity of the Fusion Model, MCMC was chosen as an estimation method.
- The MCMC algorithm is a stochastic process whereby the posterior distribution of the parameters given the data is sampled.
- Different methods of sampling from the posterior distribution exist.
- Two of the most practical sampling algorithms are the Gibbs sampler and the Metropolis-Hastings within Gibbs.

Markov Chain Monte Carlo

- Due to the complexity of the Fusion Model, MCMC was chosen as an estimation method.
- The MCMC algorithm is a stochastic process whereby the posterior distribution of the parameters given the data is sampled.
- Different methods of sampling from the posterior distribution exist.
- Two of the most practical sampling algorithms are the Gibbs sampler and the Metropolis-Hastings within Gibbs.
- To illustrate the use of both algorithms, consider sampling from the conditional distributions of two normally distributed variables, X and Y, both with unit variance, and with correlation ρ .

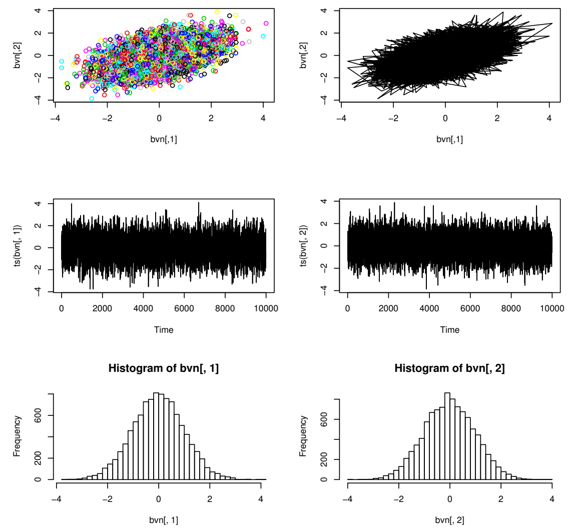
MCMC: Gibbs Sampling

- Note the distribution of $Y|X \sim N(0, \sqrt{1-\rho^2})$, and $X|Y \sim N(0, \sqrt{1-\rho^2})$



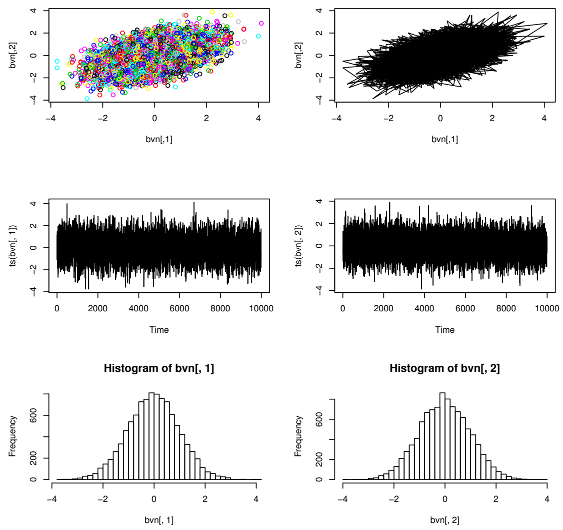
MCMC: Gibbs Sampling

- Note the distribution of $Y|X \sim N(0, \sqrt{1-\rho^2})$, and $X|Y \sim N(0, \sqrt{1-\rho^2})$
- With $\rho = 0.5$, sampling from the conditional distributes produces the data on the right.



MCMC: Gibbs Sampling

- Note the distribution of $Y|X \sim N(0, \sqrt{1-\rho^2})$, and $X|Y \sim N(0, \sqrt{1-\rho^2})$
- With $\rho = 0.5$, sampling from the conditional distributes produces the data on the right.
- In practice, the conditional distribution is usually not easily obtainable, since normalizing constants typically require integration.



MCMC: Metropolis Hastings

- The Metropolis-Hastings within Gibbs algorithm uses rejection sampling to draw from the conditional distribution.

MCMC: Metropolis Hastings

- The Metropolis-Hastings within Gibbs algorithm uses rejection sampling to draw from the conditional distribution.
- Proposed values are drawn from a simple distribution.

MCMC: Metropolis Hastings

- The Metropolis-Hastings within Gibbs algorithm uses rejection sampling to draw from the conditional distribution.
- Proposed values are drawn from a simple distribution.
- The likelihood of the proposed values is evaluated using a likelihood proportional to that of the conditional distribution.

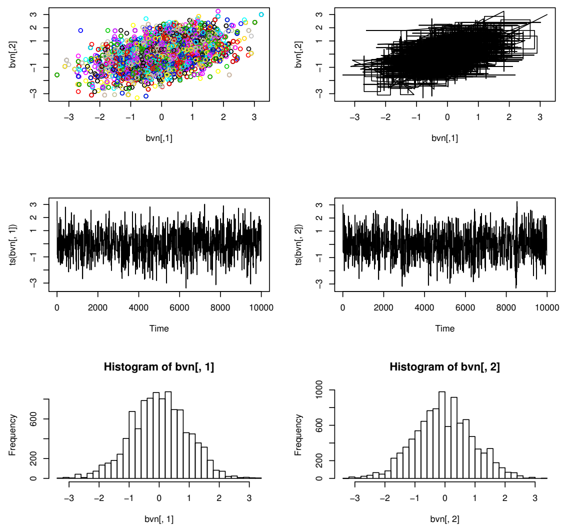
MCMC: Metropolis Hastings

- The Metropolis-Hastings within Gibbs algorithm uses rejection sampling to draw from the conditional distribution.
- Proposed values are drawn from a simple distribution.
- The likelihood of the proposed values is evaluated using a likelihood proportional to that of the conditional distribution.
- The likelihood is then compared with that of the current value of the parameter, and accepted with probability:

$$r_{MH} = \min\left[\frac{P(\mathbf{X}|\tau_j^*)P(\tau_j^*)q(\tau_j^{t-1}|\tau_j^*)}{P(\mathbf{X}|\tau_j^{t-1})P(\tau_j^{t-1})q(\tau_j^*|\tau_j^{t-1})}, 1\right]$$

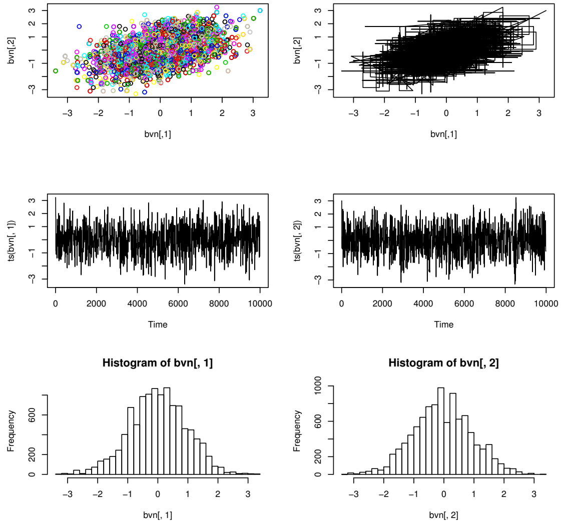
MCMC: Metropolis Hastings

- Initial values: $X = -3$, $Y = 3$.



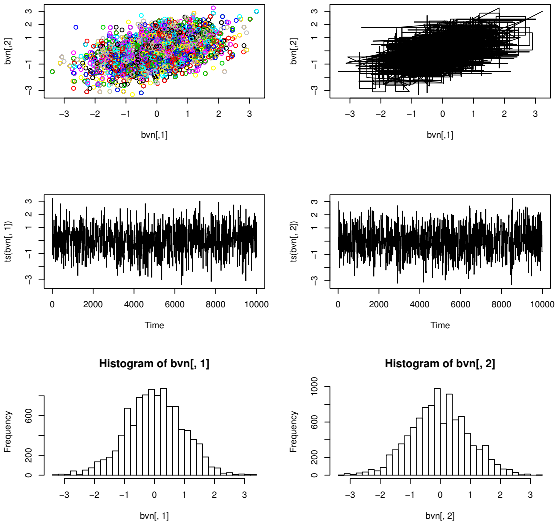
MCMC: Metropolis Hastings

- Initial values: $X = -3$, $Y = 3$.
- $\rho = 0.5$.



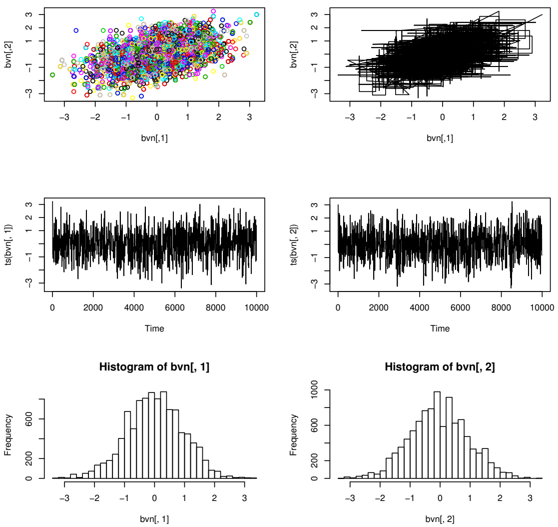
MCMC: Metropolis Hastings

- Initial values: $X = -3$, $Y = 3$.
- $\rho = 0.5$.
- Proposal values drawn from $U(-10, 10)$.



MCMC: Metropolis Hastings

- Initial values: $X = -3$, $Y = 3$.
- $\rho = 0.5$.
- Proposal values drawn from $U(-10, 10)$.
- Normal likelihood evaluated without $\frac{1}{\sqrt{2\pi\sigma^2}}$ constant.



MCMC Comparison

Figure 1: Gibbs

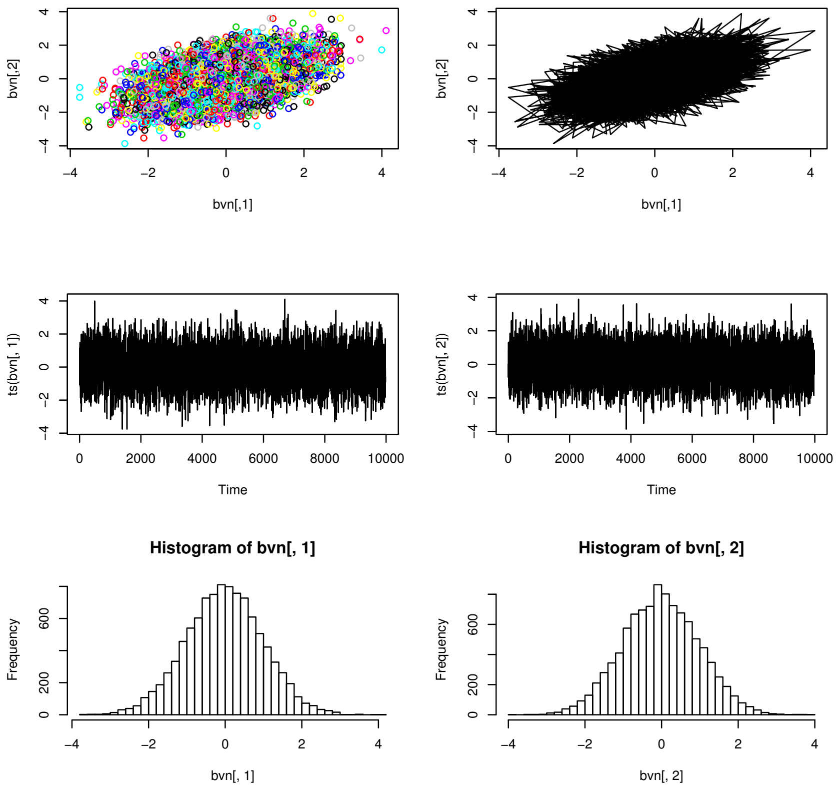
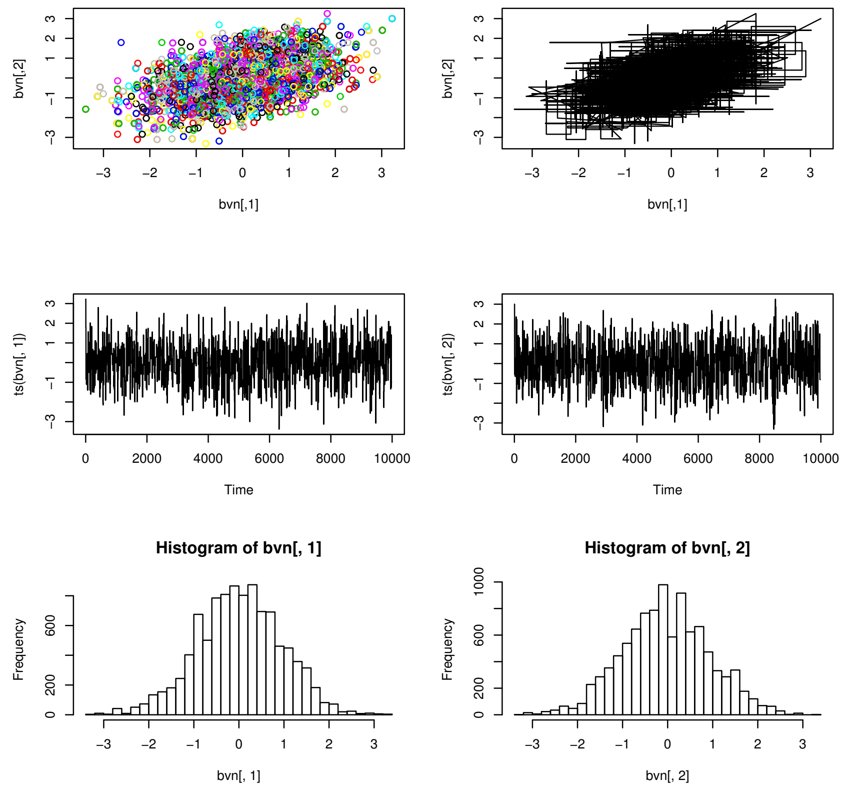


Figure 2: MHG



MCMC in Practice

- Parameter set blocking.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.
- Chain convergence.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.
- Chain convergence.
  - Chain length.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.
- Chain convergence.
  - Chain length.
  - Burnin size.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.
- Chain convergence.
  - Chain length.
  - Burnin size.
  - Convergence diagnostics.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.
- Chain convergence.
  - Chain length.
  - Burnin size.
  - Convergence diagnostics.
- Starting values.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.
- Chain convergence.
  - Chain length.
  - Burnin size.
  - Convergence diagnostics.
- Starting values.
- Posterior estimates.

MCMC in Practice

- Parameter set blocking.
- Choice of proposal distribution: symmetric, moving window (Henson, Templin, and Porch, 2003), other.
- Chain convergence.
  - Chain length.
  - Burnin size.
  - Convergence diagnostics.
- Starting values.
- Posterior estimates.
- Latent variable estimates.

Arpeggio MCMC Algorithm

- Estimate p_k parameter for each attribute k.

Arpeggio MCMC Algorithm

- Estimate p_k parameter for each attribute k.
- For each item j, estimate π^* and r^* parameters simultaneously.

Arpeggio MCMC Algorithm

- Estimate p_k parameter for each attribute k.
- For each item j, estimate π^* and r^* parameters simultaneously.
- For each item j, estimate c parameter.

Arpeggio MCMC Algorithm

- Estimate p_k parameter for each attribute k.
- For each item j, estimate π^* and r^* parameters simultaneously.
- For each item j, estimate c parameter.
- Estimate examinee attribute parameters, jointly, for each examinee i (really estimate $\tilde{\alpha}_i$).

Arpeggio MCMC Algorithm

- Estimate p_k parameter for each attribute k.
- For each item j, estimate π^* and r^* parameters simultaneously.
- For each item j, estimate c parameter.
- Estimate examinee attribute parameters, jointly, for each examinee i (really estimate $\tilde{\boldsymbol{\alpha}}_i$).
- Estimate examinee θ parameters for each examinee i.

Arpeggio Item Parameter Steps

- Use Metropolis-Hastings within Gibbs.

Arpeggio Item Parameter Steps

- Use Metropolis-Hastings within Gibbs.
- Proposal values generated from moving window family with user-specified maximum width.

Arpeggio Item Parameter Steps

- Use Metropolis-Hastings within Gibbs.
- Proposal values generated from moving window family with user-specified maximum width.
- Uniform priors for all item parameters (previously was improper Beta).

Arpeggio Item Parameter Steps

- Use Metropolis-Hastings within Gibbs.
- Proposal values generated from moving window family with user-specified maximum width.
- Uniform priors for all item parameters (previously was improper Beta).
- Likelihood for each parameter set is based on the responses of all I examinees for the specific item j.

Arpeggio Examinee Variable Steps

- Estimate examinee attribute parameters using MHG.

Arpeggio Examinee Variable Steps

- Estimate examinee attribute parameters using MHG.

  1. Proposal distribution is $\tilde{\boldsymbol{\alpha}}_i$ from $N(\mathbf{0}, \rho^*)$.

Arpeggio Examinee Variable Steps

- Estimate examinee attribute parameters using MHG.
  1. Proposal distribution is $\tilde{\boldsymbol{\alpha}}_i$ from $N(\mathbf{0}, \rho^*)$.
  2. Accept or reject proposals for each examinee.

Arpeggio Examinee Variable Steps

- Estimate examinee attribute parameters using MHG.
  1. Proposal distribution is $\tilde{\boldsymbol{\alpha}}_i$ from $N(\mathbf{0}, \rho^*)$.
  2. Accept or reject proposals for each examinee.
  3. Compute order statistics of examinee variables: for each attribute, k, order $\tilde{\alpha}_{ik}$ into $\tilde{\alpha}_{ikm}$ where m indicates the order statistic for examinee i.

Arpeggio Examinee Variable Steps

- Estimate examinee attribute parameters using MHG.
  1. Proposal distribution is $\tilde{\boldsymbol{\alpha}}_i$ from $N(\mathbf{0}, \rho^*)$.
  2. Accept or reject proposals for each examinee.
  3. Compute order statistics of examinee variables: for each attribute, k, order $\tilde{\alpha}_{ik}$ into $\tilde{\alpha}_{ikm}$ where m indicates the order statistic for examinee i.
  4. For each examinee, assign attribute value of 1 if $m > N(1 - p_k)$, where N represents the number of examinees.

Arpeggio Examinee Variable Steps

- Estimate examinee attribute parameters using MHG.
  1. Proposal distribution is $\tilde{\boldsymbol{\alpha}}_i$ from $N(\mathbf{0}, \rho^*)$.
  2. Accept or reject proposals for each examinee.
  3. Compute order statistics of examinee variables: for each attribute, k, order $\tilde{\alpha}_{ik}$ into $\tilde{\alpha}_{ikm}$ where m indicates the order statistic for examinee i.
  4. For each examinee, assign attribute value of 1 if $m > N(1 - p_k)$, where N represents the number of examinees.
- Estimate examinee θ parameters.

Arpeggio Examinee Variable Steps

- Estimate examinee attribute parameters using MHG.
  1. Proposal distribution is $\tilde{\boldsymbol{\alpha}}_i$ from $N(\mathbf{0}, \rho^*)$.
  2. Accept or reject proposals for each examinee.
  3. Compute order statistics of examinee variables: for each attribute, k, order $\tilde{\alpha}_{ik}$ into $\tilde{\alpha}_{ikm}$ where m indicates the order statistic for examinee i.
  4. For each examinee, assign attribute value of 1 if $m > N(1 - p_k)$, where N represents the number of examinees.
- Estimate examinee θ parameters.
- Independently rescale all $\tilde{\alpha}_{ik}$ and θ for each examinee.

Analysis of Arpeggio Algorithm

- The Arpeggio algorithm is very good, with only minor issues.

Analysis of Arpeggio Algorithm

- The Arpeggio algorithm is very good, with only minor issues.
- Slow convergence of algorithm.

Analysis of Arpeggio Algorithm

- The Arpeggio algorithm is very good, with only minor issues.
- Slow convergence of algorithm.
- Unnecessary rescaling of examinee parameters at the end of each step.

Analysis of Arpeggio Algorithm

- The Arpeggio algorithm is very good, with only minor issues.
- Slow convergence of algorithm.
- Unnecessary rescaling of examinee parameters at the end of each step.
- Unnecessary coercion of examinee attributes to match p_k at the end of each step.

Analysis of Arpeggio Algorithm

- The Arpeggio algorithm is very good, with only minor issues.
- Slow convergence of algorithm.
- Unnecessary rescaling of examinee parameters at the end of each step.
- Unnecessary coercion of examinee attributes to match p_k at the end of each step.
- Lack of ability to model $\tilde{\alpha}_{ik}$ as a function of covariates or higher order traits.

General Examinee Estimation

Within an MCMC algorithm, for a single attribute k of an examinee i, use a Gibbs sampler to draw examinee variables from:

$$P(\alpha_{ik} = 1|\mathbf{X}_i, \boldsymbol{\beta}) \sim B(1, p_{ik}),$$

where:

$$p_{ik} = p(\alpha_{ik} = 1|\mathbf{X}_i, \boldsymbol{\beta}) =$$

$$\frac{P(\mathbf{X}_i|\alpha_{ij}{=}1,\boldsymbol{\beta})P(\alpha_{ik}{=}1)}{P(\mathbf{X}_i|\alpha_{ik}{=}1,\boldsymbol{\beta})P(\alpha_{ik}{=}1)+P(\mathbf{X}_i|\alpha_{ik}{=}0,\boldsymbol{\beta})P(\alpha_{ik}{=}0)},$$

with:

$\boldsymbol{\beta} = (\pi_1^*, \pi_2^*, \ldots, \pi_J^*, r_{11}^*, r_{21}^*, \ldots, r_{JK}^*, c_1, c_2, \ldots, c_J, \theta_i, \alpha_{i,l \neq k})'$

Uniform Prior for Attribute Mastery

- Within the formula for p_{ik} is $P(\alpha_{ik} = 1)$, which is the prior probability of examinee i mastering attribute k.

Uniform Prior for Attribute Mastery

- Within the formula for p_{ik} is $P(\alpha_{ik} = 1)$, which is the prior probability of examinee i mastering attribute k.
- A simple prior for attribute mastery would be where that set the probability of mastery equal for all attributes:
  $p_{11} = p_{12} = \ldots = p_{1k} = \ldots = p_{ik}$

Uniform Prior for Attribute Mastery

- Within the formula for p_{ik} is $P(\alpha_{ik} = 1)$, which is the prior probability of examinee i mastering attribute k.
- A simple prior for attribute mastery would be where that set the probability of mastery equal for all attributes:
  $p_{11} = p_{12} = \ldots = p_{1k} = \ldots = p_{ik}$
- For dichotomous attributes, this translates into:
  $p_{ik} = 0.5$

Uniform Prior Initial Study

Parameter	HCS RMSE	LCS RMSE
π^*	0.061	0.117
r^*	0.050	0.233
c	0.442	0.438
θ	0.660	0.575

RMSE - Root Mean Squared Error:

$$RMSE(\tau) = \sqrt{\sum_{t=1}^{T} (\hat{\tau}_t - \tau_0)^2}$$

HCS - High Cognitive Structure parameter set.
LCS - Low Cognitive Structure parameter set.

Uniform Prior Initial Study

Parameter	HCS RMSE	HCS CR	LCS RMSE	LCS CR
α_1	0.265	0.913	0.377	0.794
α_2	0.289	0.901	0.452	0.700
α_3	0.272	0.913	0.375	0.801
α_4	0.304	0.900	0.419	0.760
α_5	0.289	0.898	0.479	0.637
α_6	0.279	0.915	0.467	0.704
α_7	0.271	0.902	0.473	0.685
α_8	0.277	0.925	0.414	0.766

CR - Classification Rate.

General Prior for Attribute Mastery

- Consider a general model for the prior probability of attribute mastery:

$$P(\alpha_{ik} = 1) = P(\tilde{\alpha}_{ik} > \kappa_k | \boldsymbol{\beta}_k, \mathbf{Y}_i, \lambda_k, G_i),$$

where,

$$\tilde{\alpha}_{ik} = \boldsymbol{\beta}_k \mathbf{Y}_i + \lambda_k G_i + E_{ik}$$

General Prior for Attribute Mastery

- Consider a general model for the prior probability of attribute mastery:

$$P(\alpha_{ik} = 1) = P(\tilde{\alpha}_{ik} > \kappa_k | \boldsymbol{\beta}_k, \mathbf{Y}_i, \lambda_k, G_i),$$

  where,

$$\tilde{\alpha}_{ik} = \boldsymbol{\beta}_k \mathbf{Y}_i + \lambda_k G_i + E_{ik}$$

- The distribution of $(\tilde{\alpha}_{ik} | \mathbf{Y}_i, G_i)$ is fixed at $N(\boldsymbol{\beta}_k \mathbf{Y}_i + \lambda_k G_i, 1 - \lambda_k^2)$.

General Prior for Attribute Mastery

- Consider a general model for the prior probability of attribute mastery:

$$P(\alpha_{ik} = 1) = P(\tilde{\alpha}_{ik} > \kappa_k | \boldsymbol{\beta}_k, \mathbf{Y}_i, \lambda_k, G_i),$$

  where,

$$\tilde{\alpha}_{ik} = \boldsymbol{\beta}_k \mathbf{Y}_i + \lambda_k G_i + E_{ik}$$

- The distribution of $(\tilde{\alpha}_{ik} | \mathbf{Y}_i, G_i)$ is fixed at $N(\boldsymbol{\beta}_k \mathbf{Y}_i + \lambda_k G_i, 1 - \lambda_k^2)$.
- Also, let E_{ik} be uncorrelated normal variables with mean zero and variance $Var(E_k) = \sigma_k^2 = 1 - \lambda_k^2$.

General Prior for Attribute Mastery

- Consider a general model for the prior probability of attribute mastery:

$$P(\alpha_{ik} = 1) = P(\tilde{\alpha}_{ik} > \kappa_k | \boldsymbol{\beta}_k, \mathbf{Y}_i, \lambda_k, G_i),$$

  where,

$$\tilde{\alpha}_{ik} = \boldsymbol{\beta}_k \mathbf{Y}_i + \lambda_k G_i + E_{ik}$$

- The distribution of $(\tilde{\alpha}_{ik} | \mathbf{Y}_i, G_i)$ is fixed at $N(\boldsymbol{\beta}_k \mathbf{Y}_i + \lambda_k G_i, 1 - \lambda_k^2)$.
- Also, let E_{ik} be uncorrelated normal variables with mean zero and variance $Var(E_k) = \sigma_k^2 = 1 - \lambda_k^2$.
- The function transforming $\tilde{\alpha}_{ik}$ to a probability will be the probit, denoted by $\Phi(\cdot)$, which is the standard normal CDF.

General Prior for θ

- Similar to the general model for the prior probability of attribute mastery, the prior for the continuous examinee variable of the RUM, θ:

$$\theta_i = \boldsymbol{\beta}_\theta \mathbf{Y}_i + \lambda_\theta G_i + E_{i\theta}$$

General Prior for θ

- Similar to the general model for the prior probability of attribute mastery, the prior for the continuous examinee variable of the RUM, θ:

$$\theta_i = \boldsymbol{\beta}_\theta \mathbf{Y}_i + \lambda_\theta G_i + E_{i\theta}$$

- By definition, the distribution of $(\theta_i | \mathbf{Y}_i, G_i)$ is $N(\boldsymbol{\beta}_\theta \mathbf{Y}_i + \lambda_\theta G_i, 1 - \lambda_\theta^2)$.

General Prior for θ

- Similar to the general model for the prior probability of attribute mastery, the prior for the continuous examinee variable of the RUM, θ:

$$\theta_i = \boldsymbol{\beta}_\theta \mathbf{Y}_i + \lambda_\theta G_i + E_{i\theta}$$

- By definition, the distribution of $(\theta_i | \mathbf{Y}_i, G_i)$ is $N(\boldsymbol{\beta}_\theta \mathbf{Y}_i + \lambda_\theta G_i, 1 - \lambda_\theta^2)$.
- Also, let $E_{i\theta}$ be an uncorrelated normal variable with mean zero and variance $Var(E_\theta) = \sigma_\theta^2 = 1 - \lambda_\theta^2$.

General Prior for θ

- Similar to the general model for the prior probability of attribute mastery, the prior for the continuous examinee variable of the RUM, θ:

$$\theta_i = \boldsymbol{\beta}_\theta \mathbf{Y}_i + \lambda_\theta G_i + E_{i\theta}$$

- By definition, the distribution of $(\theta_i | \mathbf{Y}_i, G_i)$ is $N(\boldsymbol{\beta}_\theta \mathbf{Y}_i + \lambda_\theta G_i, 1 - \lambda_\theta^2)$.
- Also, let $E_{i\theta}$ be an uncorrelated normal variable with mean zero and variance $Var(E_\theta) = \sigma_\theta^2 = 1 - \lambda_\theta^2$.
- As an analog to the general prior for each attribute, the prior for θ uses an identity link function.

Attribute-wise Prior

- An attribute-wise prior is equivalent across all examinees for each attribute k, but can differ from attribute to attribute:

$$p_{1k} = p_{1k} = \ldots = p_{ik}$$

Attribute-wise Prior

- An attribute-wise prior is equivalent across all examinees for each attribute k, but can differ from attribute to attribute:

$$p_{1k} = p_{1k} = \ldots = p_{ik}$$

- Expressed in terms of the general prior:

$$\tilde{\alpha}_{ik} = E_{ik}$$

Attribute-wise Prior

- An attribute-wise prior is equivalent across all examinees for each attribute k, but can differ from attribute to attribute:

$$p_{1k} = p_{1k} = \ldots = p_{ik}$$

- Expressed in terms of the general prior:

$$\tilde{\alpha}_{ik} = E_{ik}$$

- Again, $\tilde{\alpha}_{ik} \sim N(0, 1)$, and $E_{ik} \sim N(0, \sigma_k^2 = 1)$.

Attribute-wise Prior

- An attribute-wise prior is equivalent across all examinees for each attribute k, but can differ from attribute to attribute:

$$p_{1k} = p_{1k} = \ldots = p_{ik}$$

- Expressed in terms of the general prior:

$$\tilde{\alpha}_{ik} = E_{ik}$$

- Again, $\tilde{\alpha}_{ik} \sim N(0,1)$, and $E_{ik} \sim N(0, \sigma_k^2 = 1)$.
- The prior for the attribute is then given by:

$$P(\alpha_{ik} = 1) = P(\tilde{\alpha}_{ik} > \kappa_k) = P(E_{ik} > \kappa_k) =$$
$$P(E_{ik} < -\kappa_k) = \Phi(-\kappa_k)$$

Attribute-wise Prior for θ

- The prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.

$$\theta_i = E_{i\theta}$$

Attribute-wise Prior for θ

- The prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.

$$\theta_i = E_{i\theta}$$

- Again, $\theta_i \sim N(0, 1)$, and $E_{i\theta} \sim N(0, \sigma_\theta^2 = 1)$.

Attribute-wise Prior for θ

- The prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.

$$\theta_i = E_{i\theta}$$

- Again, $\theta_i \sim N(0,1)$, and $E_{i\theta} \sim N(0, \sigma_\theta^2 = 1)$.
- The likelihood function for the prior for θ is given by the ordinate of the standard normal distribution evaluated at θ:

$$f(\theta_i) = \frac{1}{\sqrt{2\pi}} e^{\left(\frac{-\theta_i^2}{2}\right)}$$

Higher Order Prior

- A higher order trait for the attributes can be used as a prior:

$$\tilde{\alpha}_{ik} = \lambda_k G_i + E_{ik}$$

Higher Order Prior

- A higher order trait for the attributes can be used as a prior:

$$\tilde{\alpha}_{ik} = \lambda_k G_i + E_{ik}$$

- Again, $(\tilde{\alpha}_{ik}|G_i) \sim N(\lambda_k G_i, 1 - \lambda_k^2)$, and $E_{ik} \sim N(0, \sigma_k^2)$, with $\sigma_k^2 = 1 - \lambda_k^2$.

Higher Order Prior

- A higher order trait for the attributes can be used as a prior:

$$\tilde{\alpha}_{ik} = \lambda_k G_i + E_{ik}$$

- Again, $(\tilde{\alpha}_{ik}|G_i) \sim N(\lambda_k G_i, 1 - \lambda_k^2)$, and $E_{ik} \sim N(0, \sigma_k^2)$, with $\sigma_k^2 = 1 - \lambda_k^2$.
- The prior for the attribute is then given by: $P(\alpha_{ik} = 1) = P(\tilde{\alpha}_{ik} > \kappa_k|\lambda_k, G_i) =$ $P(\lambda_k G_i + E_{ik} > \kappa_k|\lambda_k, G_i) = P(E_{ik} < \lambda_k G_i - \kappa_k|\lambda_k, G_i) =$

$$\Phi(\frac{\lambda_k G_i - \kappa_k}{\sqrt{1-\lambda^2}})$$

Higher Order Prior for θ

- The higher order prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.

Higher Order Prior for θ

- The higher order prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.
- The model for θ is:

$$\theta_i = \lambda_\theta G_i + E_{i\theta}$$

Higher Order Prior for θ

- The higher order prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.
- The model for θ is:

$$\theta_i = \lambda_\theta G_i + E_{i\theta}$$

- By definition, $(\theta_i | G_i) \sim N(\lambda_\theta G_i, 1 - \lambda_\theta^2)$, and by constraint, $E_{ik} \sim N(0, \sigma_\theta^2)$, with $\sigma_\theta^2 = 1 - \lambda_\theta^2$.

Higher Order Prior for θ

- The higher order prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.
- The model for θ is:

$$\theta_i = \lambda_\theta G_i + E_{i\theta}$$

- By definition, $(\theta_i | G_i) \sim N(\lambda_\theta G_i, 1 - \lambda_\theta^2)$, and by constraint, $E_{ik} \sim N(0, \sigma_\theta^2)$, with $\sigma_\theta^2 = 1 - \lambda_\theta^2$.
- The prior for θ is then the ordinate of the normal distribution with mean $\lambda_\theta G_i$, and variance σ_θ^2, evaluated at $theta_i$.

Higher Order Prior Initial Study

Parameter	HCS RMSE	LCS RMSE
π^*	0.032	0.042
r^*	0.042	0.128
c	0.584	0.216
θ	0.792	0.585

RMSE - Root Mean Squared Error:

$$RMSE(\tau) = \sqrt{\sum_{t=1}^{T} (\hat{\tau}_t - \tau_0)^2}$$

HCS - High Cognitive Structure parameter set.
LCS - Low Cognitive Structure parameter set.

Higher Order Prior Initial Study

	HCS		LCS	
Parameter	RMSE	CR	RMSE	CR
α_1	0.198	0.952	0.325	0.859
α_2	0.263	0.902	0.429	0.735
α_3	0.217	0.943	0.333	0.843
α_4	0.251	0.924	0.391	0.787
α_5	0.255	0.919	0.423	0.735
α_6	0.216	0.937	0.396	0.777
α_7	0.213	0.939	0.360	0.813
α_8	0.201	0.947	0.380	0.791

CR - Classification Rate.

Covariate Prior

- A model for covariates for the attributes can be used as a prior:

$$\tilde{\alpha}_{ik} = \boldsymbol{\beta}_k \mathbf{Y}_i + E_{ik}$$

Covariate Prior

- A model for covariates for the attributes can be used as a prior:

$$\tilde{\alpha}_{ik} = \boldsymbol{\beta}_k \mathbf{Y}_i + E_{ik}$$

- Fix, $(\tilde{\alpha}_{ik} | \mathbf{Y}_i) \sim N(\boldsymbol{\beta}_k \mathbf{Y}_i, 1)$.

Covariate Prior

- A model for covariates for the attributes can be used as a prior:

$$\tilde{\alpha}_{ik} = \boldsymbol{\beta}_k \mathbf{Y}_i + E_{ik}$$

- Fix, $(\tilde{\alpha}_{ik}|\mathbf{Y}_i) \sim N(\boldsymbol{\beta}_k \mathbf{Y}_i, 1)$.
- The prior for the attribute is then given by: $P(\alpha_{ik} = 1) = P(\tilde{\alpha}_{ik} > \kappa_k|\boldsymbol{\beta}_k, \mathbf{Y}_i) = P(\boldsymbol{\beta}_k \mathbf{Y}_i + E_{ik} > \kappa_k|\boldsymbol{\beta}_k, \mathbf{Y}_i) = P(E_{ik} < \boldsymbol{\beta}_k \mathbf{Y}_i - \kappa_k|\boldsymbol{\beta}_k, \mathbf{Y}_i) =$

$$\Phi(\boldsymbol{\beta}_k \mathbf{Y}_i - \kappa_k)$$

Covariate Prior for θ

- The covariate prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.

Covariate Prior for θ

- The covariate prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.
- The model for θ is:

$$\theta_i = \boldsymbol{\beta}_\theta \mathbf{Y}_i + E_{i\theta}$$

Covariate Prior for θ

- The covariate prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.
- The model for θ is:

$$\theta_i = \boldsymbol{\beta}_\theta \mathbf{Y}_i + E_{i\theta}$$

- By definition, $(\theta_i | \mathbf{Y}_i \sim N(\boldsymbol{\beta}_\theta \mathbf{Y}_i, 1)$.

Covariate Prior for θ

- The covariate prior for θ uses the same underlying variable model as that for the attribute variables, but with an identity link.
- The model for θ is:

$$\theta_i = \boldsymbol{\beta}_\theta \mathbf{Y}_i + E_{i\theta}$$

- By definition, $(\theta_i | \mathbf{Y}_i \sim N(\boldsymbol{\beta}_\theta \mathbf{Y}_i, 1)$.
- The prior for θ is then the ordinate of the normal distribution with mean $\boldsymbol{\beta}_k \mathbf{Y}_i$, and variance 1, evaluated at θ_i.

Acknowledgments

Stout

Acknowledgments

Stout

Douglas

Acknowledgments

Stout

Douglas

Roussos

Acknowledgments

Stout

Douglas

Roussos

Henson

Acknowledgments

Stout

Douglas

Roussos

Henson

Karelitz

Acknowledgments

Stout

Douglas

Roussos

Henson

Karelitz

Templin

Acknowledgments

Stout

Douglas

Roussos

Henson

Karelitz

Templin

F. Porch

Acknowledgments

Stout

Douglas

Roussos

Henson

Karelitz

Templin

F. Porch

Bibby