Inference for BART with Multinomial Outcomes

Yizhen Xu

Johns Hopkins University

December, 2020

Inference for BART with Multinomial Outcomes

- Motivation
  - Probit model is a natural platform to use Bayesian Additive Regression Trees (BART) on multinomial outcomes
  - Existing algorithm for posterior inference in multinomial probit BART (MPBART) can be unstable
    - ⋆ Difficulty with unbalanced categories
    - ⋆ Sensitive to choice of reference level
    - ⋆ Existing R package crashes frequently
- Contributions
  - Developed new posterior sampling algorithm for MPBART
  - Proved new algorithms have better mixing rate
  - Constructed an R package. Available on Github at "yizhenxu/GcompBART"

Inference for BART with Multinomial Outcomes

- Results
  - Faster MCMC convergence over different data structures
  - Better mixing rate under stationarity
  - Better predictive accuracy on simulated and real data

Overview

- Multinomial probit models (MNP)
  - Latent variable formulation
  - BART for multinomial probit models
- Bayesian inference in MNP
  - Data augmentation in standard MNP
  - Data augmentation and Gibbs sampling for MPBART
- Results
  - Comparison of mixing rate across algorithms
  - Simulation and application

Latent Variable Formulation for Multinomial Models

- Consider the 3 category model where $S \in \{1, 2, 3\}$

- S arises from a bivariate latent utility

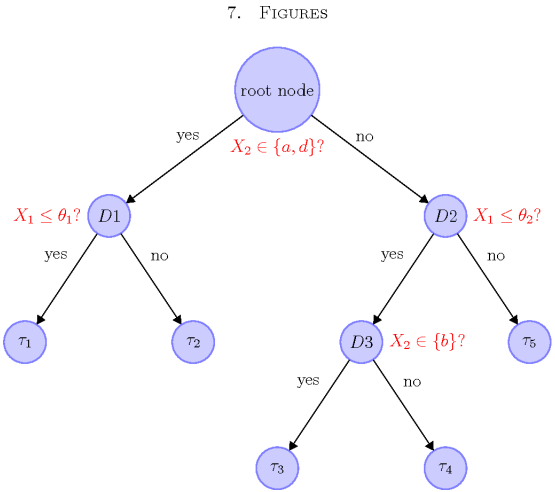
$$\begin{pmatrix} W_1 \\ W_2 \end{pmatrix} \sim N\left[\begin{pmatrix} G_1 \\ G_2 \end{pmatrix}, \Sigma\right], \quad \Sigma = \begin{pmatrix} \sigma_{11} & \\ \sigma_{12} & \sigma_{22} \end{pmatrix}$$

$$S = \begin{cases} 1 & \text{if } \max(W) = W_1 \geq 0 \\ 2 & \text{if } \max(W) = W_2 \geq 0 \\ 3 & \text{if } \max(W) < 0, \end{cases}$$

- Multinomial probit model: $G_k(X; \beta) = X\beta_k$
- MPBART: $G_k(X; \theta) = \sum_j g(X; \theta_{kj})$

BART for Multinomial Models

$g(X; \theta_{jk})$: Binary tree



Bayesian Inference for Multinomial Models and MPBART

- Bayesian methods for MNP are well developed
  - e.g., Imai and van Dyk (2005 J Econometrics)
- Kindo et al. (2016 Stat) adapted one of these methods for implementing MPBART
- This algorithm (and R package) has limitations
  - Sensitive to choice of reference level
  - Slow MCMC convergence under unbalanced categories
  - Generates trees that are too large
    - ⋆ Goes against general idea of BART: use large number of small trees
    - ⋆ Leads to overfitting
- Solution: Sample the sum-of-trees based on latent utilities W under a constraint on the covariance matrix Σ

Identifiability of Multinomial Probit Models

- For identifiability, need to impose constraints on scale of the latent variable distribution
- Reason: for any scalar $\alpha > 0$

$$S(W) \quad = \quad S(\alpha W)$$

- Examples of constraints
  - $\text{trace}(\Sigma) = \dim(W)$
  - $\sigma_{11} = 1$

Parameter Expansion for Data Augmentation in MNP

- Goal: Posterior inference about (θ, Σ)
  - Involves Gibbs sampling of $(W, \theta, \Sigma) \,|\, \text{Data}$
- Major challenges
  - Hard to derive full conditionals under scale constraint
  - Also hard to derive conjugate priors and posteriors
- Solution: Parameter expansion and data augmentation
  - Add α to (W, θ, Σ)
  - Rescale latent variables via $\widetilde{W} = \alpha W$
  - Enables sampling from unconstrained space $(\widetilde{W}, \widetilde{\theta}, \widetilde{\Sigma})$ – much easier
  - Can translate back to get samples from constrained space
- Specifically: $\alpha^2 = \text{tr}(\widetilde{\Sigma})/\dim(W)$

Gibbs Sampling for MPBART - General Strategy

- Step 0: Specify priors $p(\alpha \mid \Sigma)$, $p(\theta)$, $p(\Sigma)$
  - Note $\mu = G(X; \theta)$
- Step 1: Draw W and α, rescale W
- Step 2: Sampling for mean (θ and μ)
- Step 3: Sampling for Σ

Our modification of Kindo algorithm

- Do not use parameter expansion for sampling of trees
- Only use parameter expansion for sampling Σ

Gibbs Sampling for MPBART

	Algorithm 1 (Kindo et al. 2016)	Algorithm 2
1	$\alpha_1^2 \sim f(\alpha^2\|\Sigma), W \sim f(W\|\mu,\Sigma)$ $\widetilde{W} = \alpha_1 W$	$\alpha_1^2 \sim f(\alpha^2\|\Sigma), W \sim f(W\|\mu,\Sigma)$ $\widetilde{W} = \alpha_1 W$
2	$\widetilde{\theta} \sim f(\widetilde{\theta}\|\widetilde{W}, \alpha_1^2, \Sigma)$ $\mu = G(X;\widetilde{\theta})/\alpha_1$	$\theta \sim f(\theta\|W,\Sigma)$ $\mu = G(X;\theta)$
3	$\Sigma \sim \int f(\Sigma, \alpha^2\|\widetilde{W}, \alpha_1^2, \mu) d\alpha^2$	$\Sigma \sim \int f(\Sigma, \alpha^2\|\widetilde{W}, \alpha_1^2, \mu) d\alpha^2$

Gibbs Sampling for MPBART

Algorithm 2

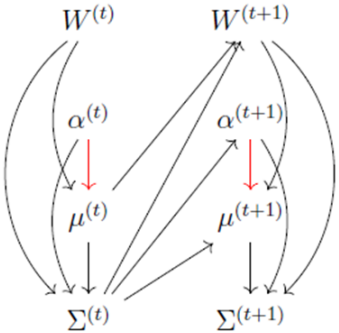
1. $\alpha_1^2 \sim f(\alpha^2|\Sigma), W \sim f(W|\mu, \Sigma)$
   $\widetilde{W} = \alpha_1 W$
2. $\theta \sim f(\theta|W, \Sigma)$
   $\mu = G(X; \theta)$
3. $\Sigma \sim \int f(\Sigma, \alpha^2|\widetilde{W}, \alpha_1^2, \mu) d\alpha^2$

Algorithm 3

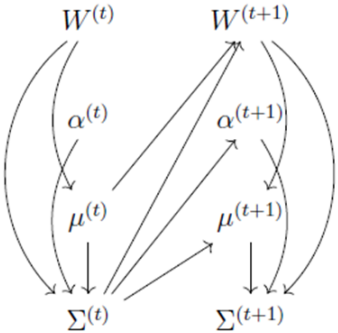
1. $W \sim f(W|\mu, \Sigma)$
2. $\theta \sim f(\theta|W, \Sigma)$
   $\mu = G(X; \theta)$
3. $\Sigma \sim \int f(\Sigma, \alpha^2|W, \mu) d\alpha^2$

Dependency Diagrams

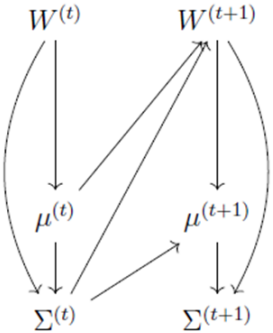
Algorithm 1



Algorithm 2



Algorithm 3



Theoretical Result: Provably Better Mixing Rates

- A common measure for quantifying the mixing rate of a Markov chain is the lag1 autocorrelation; lower autocorrelation indicates better mixing rate
- For Algorithm [k], define

$$\rho_{\mu}^{(k)} = \text{corr}(\mu^{(t)}, \mu^{(t+1)}) \qquad \rho_{\Sigma}^{(k)} = \text{corr}(\Sigma^{(t)}, \Sigma^{(t+1)})$$

  where t indexes posterior draws

- Key result: When the Markov chain is stationary,

$$\rho_{\mu}^{(3)} = \rho_{\mu}^{(2)} \leq \rho_{\mu}^{(1)},$$
$$\rho_{\Sigma}^{(3)} \leq \rho_{\Sigma}^{(2)} = \rho_{\Sigma}^{(1)}.$$

Empirical Result: More Accurate Predictions

- J posterior samples, N subjects
- Posterior mean accuracy: the average accuracy across all posterior predictions,

$$\frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J}1\{\hat{S}_i^{(j)} = S_i\}, \tag{1}$$

Simulation

$$
\begin{aligned}
&U = (U_1, \ldots, U_5) \sim \text{Uniform}(0, 1) \\
&V \sim \text{Uniform}(0, 2) \\
&G_1 = 15 \sin(\pi U_1 U_2) + (U_3 - 0.5)^2 - 10 U_4 - 5 U_5 \\
&G_2 = (U_3 - 0.5)^2 - U_4 U_5 + 4V \\
&G^T = (G_1, G_2), \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \\
&W = (W_1, W_2)^T \sim \text{MVN}(G, \Sigma) \\
&S = \begin{cases} 1 & \text{if } W_1 \geq \max\{0, W_2\} \\ 2 & \text{if } W_2 \geq \max\{0, W_1\} \\ 3 & \text{if } W_1 < 0 \text{ and } W_2 < 0 \end{cases}
\end{aligned}
$$

The proportions of $S = 1, 2$, and 3 are 31%, 66%, and 3%.

MCMC Convergence

	Algorithms		
Accuracy	1	2	3
Train	0.65	0.91	0.90
Test	0.63	0.88	0.88

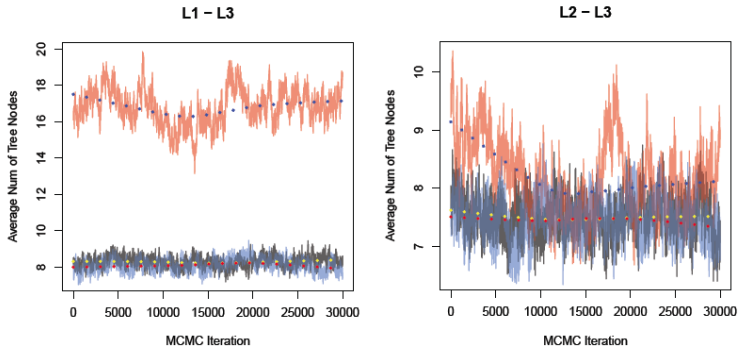


Figure: 100 trees, 50,000 burn-in's, and 30,000 posterior draws.

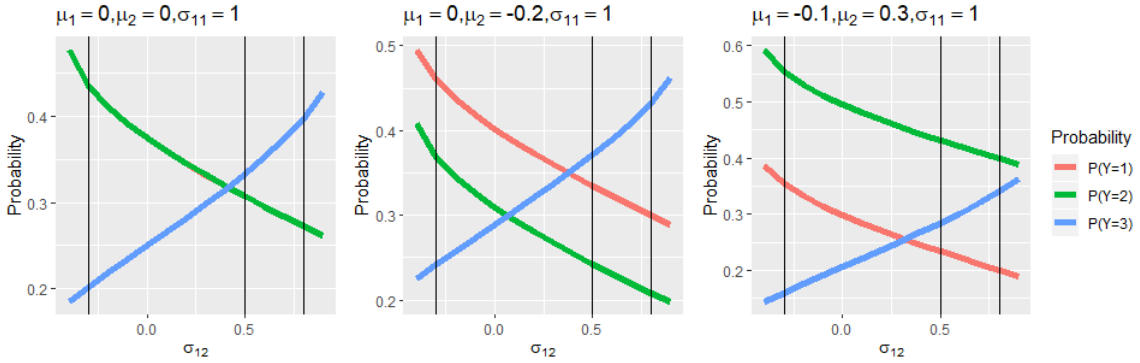
Posterior Distribution of Variance Components

True values: $\sigma_{11} = 1, \sigma_{12} = 0.5$

	Algorithm 1	Algorithm 2	Algorithm 3
σ_{11}	.88 (.64, 1.17)	.79 (.66,.92)	.75 (.63,.88)
σ_{12}	−.32 (−.46, −.18)	.82 (.72, .89)	.81 (.73, .87)

- The sign from our methods agrees with the true value
- Covariance parameter is important for prediction

Prediction Accuracy Depends on Covariance



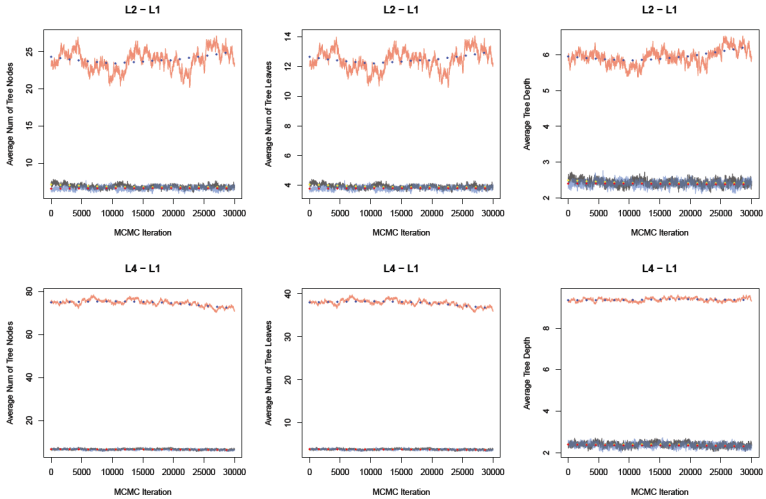
Application - AMPATH Data

- Outcome model at the first time interval $P(S_1|X_1, A_0, V; \theta)$

	Algorithms		
Accuracy	1	2	3
Train	.63	.77	.77
Test	.63	.78	.78

Diagnostic Plot

Red: Kindo et al.'s algorithm (Algorithm 1); Blue/black: our proposals



Future Works

- Embed BART in a framework with less distributional assumptions
- Account for cluster effect (clinic etc.)

Acknowledgments

PhD Advisor: Joseph Hogan - Brown University

Collaborators:

- Liu, Tao - Brown University
- Daniels, Michael - University of Florida
- Marshall, Brandon - Brown University
- Kantor, Rami - Brown University
- Omodi, Victor - Moi University / AMPATH
- Mwangi, Ann - Moi University

This work is supported by NIH grant R01 AI 108441

Latent Variable Formulation for Multinomial Models

- $S \in \{1, 2, 3\}$, $S(Z) = \text{argmax}\{Z_1, Z_2, Z_3\}$
  Normalization for identifiability : $W_k = Z_k - Z_3, k = 1, 2$
- $$S(W) = \begin{cases} 1 & \text{if } \max(W) = W_1 \geq 0 \\ 2 & \text{if } \max(W) = W_2 \geq 0 \\ 3 & \text{if } \max(W) < 0, \end{cases}$$
- Latent utilities $(W_1, W_2) = (G_1, G_2) + \epsilon$, where $G_k(X; \theta) = X\theta_k$
  - Multinomial logistic: $\epsilon_k \overset{\text{iid}}{\sim} \textit{Logistic}(0, 1)$
  - Multinomial probit (MNP): $\epsilon \sim \textit{MVN}(\mathbf{0}, \Sigma)$

    $\Rightarrow$ MPBART (Kindo et al. 2016): $G_k(X; \theta) = \sum_j g(X; \theta_{kj})$

Grouping and Collapsing (Meng and van Dyk 1999, Liu et al. 1994)

- Standard data augmentation scheme:

$$x \sim f(x|y), \quad y \sim f(y|x)$$

- Parameter expansion: overparameterize $f(x, y)$ to $f(x, y, \alpha)$
- Grouping:

$$(x, \alpha) \sim f(x, \alpha|y), \quad y \sim f(y|x, \alpha)$$

- Collapsing:

$$x \sim f(x|y) = \int f(x, \alpha|y) d\alpha = \int f(\alpha|y) f(x|\alpha, y) d\alpha$$

by $\alpha \sim f(\alpha|y)$, $x \sim f(x|\alpha, y)$, and discard α.

Gibbs Sampling for MPBART

	lvD1	Algorithm 1 (Kindo et al. 2016)
1	$\alpha_1^2 \sim f(\alpha^2\|\Sigma), \widetilde{W} \sim f(\widetilde{W}\|\alpha_1^2, \theta, \Sigma)$ discard α_1^2	$\alpha_1^2 \sim f(\alpha^2\|\Sigma), \widetilde{W} \sim f(\widetilde{W}\|\alpha_1^2, \mu, \Sigma)$
2	$\alpha_2^2 \sim f(\alpha^2\|\widetilde{W}, \Sigma), \widetilde{\theta} \sim f(\widetilde{\theta}\|\widetilde{W}, \alpha_2^2, \Sigma)$ $\theta = \widetilde{\theta}/\alpha_2$	$\widetilde{\theta} \sim f(\theta\|\widetilde{W}, \alpha_1^2, \Sigma)$ $\mu = G(X; \widetilde{\theta})/\alpha_1$
3	$\Sigma \sim \int f(\Sigma, \alpha^2\|\widetilde{W}, \widetilde{\theta}) d\alpha^2$	$\Sigma \sim \int f(\Sigma, \alpha^2\|\widetilde{W}, \alpha_1^2, \mu) d\alpha^2$

- Step 2 in Algorithm 1,
  - $f(\alpha^2|\widetilde{W}, \Sigma) = \int f(\alpha^2, \theta|\widetilde{W}, \Sigma) d\theta$ is hard to derive for trees $\Rightarrow$ set $\alpha_2^2 = \alpha_1^2$
  - sampling $\mu \sim f(\mu|\widetilde{W}, \alpha_1^2, \Sigma)$

Gibbs Sampling for MPBART

lvD2

1. $\alpha_1^2 \sim f(\alpha^2|\Sigma), W \sim f(W|\theta, \Sigma)$
2. $\theta \sim f(\theta|W, \Sigma)$
3. $\Sigma \sim \int f(\Sigma, \alpha^2|W, \alpha_1^2, \theta) d\alpha^2$

Algorithm 2

1. $\alpha_1^2 \sim f(\alpha^2|\Sigma), W \sim f(W|\mu, \Sigma)$
2. $\theta \sim f(\theta|W, \Sigma), \mu = G(X; \theta)$
3. $\Sigma \sim \int f(\Sigma, \alpha^2|W, \alpha_1^2, \mu) d\alpha^2$

Algorithm 3

1. $W \sim f(W|\mu, \Sigma)$
2. $\theta \sim f(\theta|W, \Sigma), \mu = G(X; \theta)$
3. $\Sigma \sim \int f(\Sigma, \alpha^2|W, \mu) d\alpha^2$

Marginal Augmentation

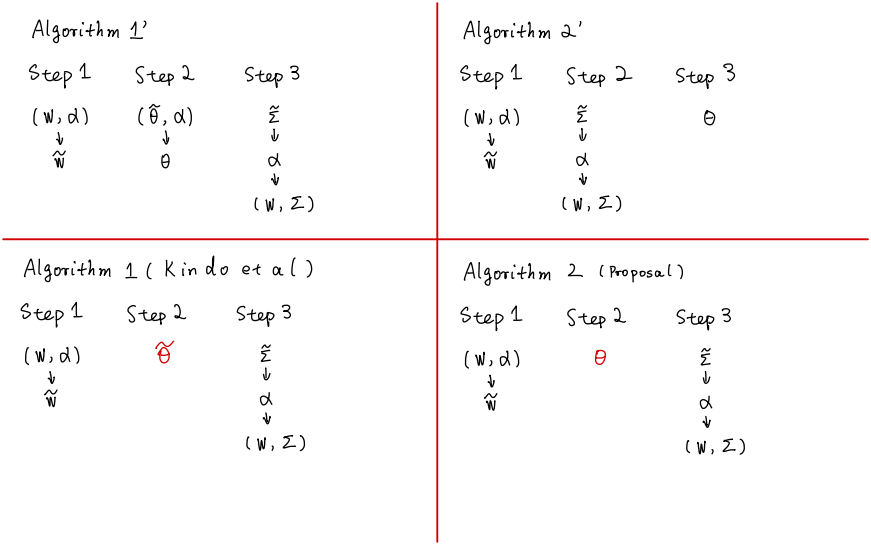
Imai and van Dyk (2005)

- Data augmentation (DA) algorithm: sample $p(\theta, W|S)$ by iterative posterior sampling of $p(\theta|W, S)$ and $p(W|\theta, S)$
- Marginal augmentation: $L(\theta|S) \propto \int[\int p(S, W|\theta, \alpha)p(\alpha|\theta)d\alpha]dW$; Meng and van Dyk (1999) theoretically proved that this can improve the geometric rate of convergence of the DA algorithm
- "using unidentifiable parameters within a Markov chain is the key to the substantial computational gains offered by marginal augmentation."
- The constraint on Σ is made to be sure the model parameters (θ, Σ) are identified; parameter α is unidentifiable. Even with the constraint, model parameters may be unidentifiable without certain conditions on X and S.

Connection of our Proposal to Imai and van Dyk (2005)

- Imai and van Dyk (2005) provided two algorithms (1' and 2') for implementing MNP, and they expected algorithm 1' to outperform algorithm 2', because algorithm 1' is a complete marginal augmentation procedure while 2' is not.
- In Step 2, algorithm 1' updates α first and then samples θ conditional on the updated α, while algorithm 2' samples θ without conditioning on α
- Kindo et al (2016) employed the algorithm 1' for extending MNP to incorporate BART, skipping the sampling of α in Step 2 and updating θ conditional the α from Step 1; they called this sampling procedure a "semi marginal augmentation"
- Our proposal is somehow similar to the algorithm 2' of Imai and van Dyk (2005), sampling θ from its conditional distribution that does not depend on α, i.e. updating θ conditional on the constrained latent utilities W

Connection of our Proposal to Imai and van Dyk (2005)



Gibbs sampling of (W, θ, Σ)
Linear model specification: $G(X;\theta) = X\theta$

Algorithm 0

1. Sample $W, \alpha^* | S, \mu, \Sigma \quad \Rightarrow \quad \tilde{W} = \alpha^* W, \tilde{\Sigma} = (\alpha^*)^2 \Sigma$
2. Sample $\tilde{\theta}, \alpha^* | \tilde{W}, \tilde{\Sigma}, X \quad \Rightarrow \quad \theta = \tilde{\theta}/\alpha^*$
3. Sample $\tilde{\Sigma}, \alpha | \tilde{W} - X\tilde{\theta} \quad \Rightarrow \quad \theta = \tilde{\theta}/\alpha$, $\Sigma = \tilde{\Sigma}/\alpha^2$, and $W = \tilde{W}/\alpha$.

$G(X;\tilde{\theta}) = \alpha G(X;\theta)$

Existing BART Packages

	bartMachine	BayesTree	BART	mpbart
Language	Java	C++	C++	C++
Tree Proposal Types	3	4	2	2
Binary/Continuous Y	Yes	Yes	Yes	No
Multinomial Y	No	No	Yes	Yes
Preferable on Big Data	No	Yes	Yes	No

- Posterior samplings are not formatted for Bayesian G-comp
- Categories in the engagement outcome are correlated
  - BART package: multinomial BART with logistic latents assumes **independence** among alternatives – not applicable to our analysis
  - **mpbart package**: multinomial probit BART accounts for **correlation** among multinomial categories

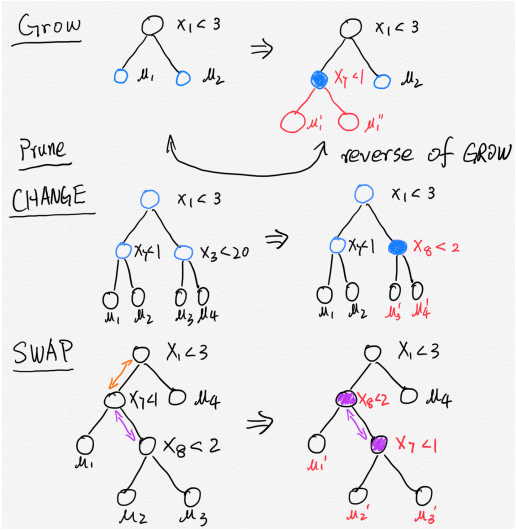
Existing BART Packages

- Problems in the mpbart package
  - Memory leaks
  - Conditional moments for gibbs sampling of multivariate Gaussian latent variables
  - Data augmention of the covariance matrix of the latent variables
  - Posterior sampling of the covariance matrix of the latent variables
  - Data processing of choice-specific covariates etc.
- Tree proposal types: GROW, PRUNE, SWAP, CHANGE
  - The Bayesian CART paper [Chipman 1998] demonstrated the importance of SWAP and CHANGE
  - BayesTree is the only package that uses all the four tree proposal types

Package GcompBART

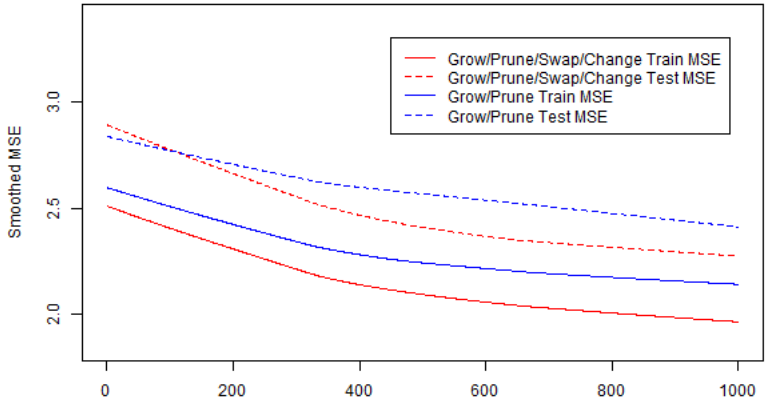
- Can be applied to data with continuous, binary, or multinomial outcomes (accounting for correlation among categories)
- MPBART implementation allows different set of covariates and number of trees for each latent utility
- Provides posterior predictions in both the regular (as in other packages) and the dynamic G-computation format
- Uses all the four tree proposal types
- Available for download on Github at "yizhenxu/GcompBART"

Posterior Trees Sampling



Performance with/without SWAP and CHANGE

Simulation setting from the Friedman MARS paper



Inclusion Proportions of Covariates

Outcome at $t = 1$