Bayesian model configuration, selection and averaging in complex regression contexts

Aliaksandr Hubin

Department of Mathematics
University of Oslo

aliaksah@math.uio.no

UiO : Universitetet i Oslo

Oslo
09.11.2018

Outline

1. Introduction;
2. The Deep Bayesian regression model (DBRM):
   - Feature engineering;
   - Bayesian model specification;
   - Links to GLM(M)s and logic regressions;
3. (R)(G)MJMCMC algorithm;
4. Applications:
   - Prediction;
   - Inference.

Outline

1. Introduction;
2. The Deep Bayesian regression model (DBRM):
   - Feature engineering;
   - Bayesian model specification;
   - Links to GLM(M)s and logic regressions;
3. (R)(G)MJMCMC algorithm;
4. Applications:
   - Prediction;
   - Inference.

Outline

1. Introduction;
2. The Deep Bayesian regression model (DBRM):
   - Feature engineering;
   - Bayesian model specification;
   - Links to GLM(M)s and logic regressions;
3. (R)(G)MJMCMC algorithm;
4. Applications:
   - Prediction;
   - Inference.

Outline

1. Introduction;
2. The Deep Bayesian regression model (DBRM):
    - Feature engineering;
    - Bayesian model specification;
    - Links to GLM(M)s and logic regressions;
3. (R)(G)MJMCMC algorithm;
4. Applications:
    - Prediction;
    - Inference.

Complex regression models

Mapping explanatory variables X to the responses Y via $Y \sim F(X, \theta)$.

Purposes:

1. Inference:
   - Explain **how** and **why** X influences Y.
2. Prediction:
   - Identify **unobserved** Y for the **observed** X as precisely as possible.

Complex regression models

Mapping explanatory variables X to the responses Y via $Y \sim F(X, \theta)$.

Purposes:

1. Inference:
   - Explain **how** and **why** X influences Y.
2. Prediction:
   - Identify **unobserved** Y for the **observed** X as precisely as possible.

Complex regression models

Mapping explanatory variables X to the responses Y via $Y \sim F(X, \theta)$.

Purposes:

1. Inference:
   - Explain **how** and **why** X influences Y.
2. Prediction:
   - Identify **unobserved** Y for the **observed** X as precisely as possible.

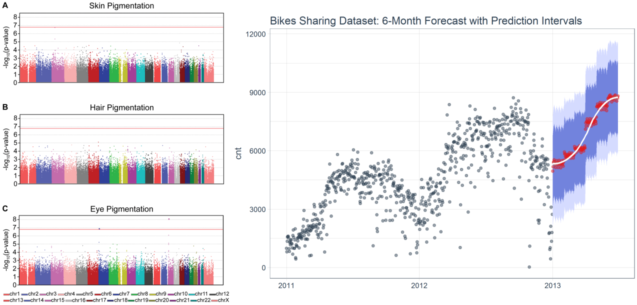


Figure: Prediction and inference illustrations

Applications

1. Regression problems:
   - Prediction of house prices;
   - Explaining which factors influence the amount of rainfall.
2. Classification problems:
   - Classify objects from the pictures;
   - Explain why a particular person is rejected a mortgage.
3. Ranking problems:
   - Rank a set of articles w.r.t. their relevance to a given person;
   - Identify which factors cause a particular SP Rating of a country.
4. Time to event problems:
   - Predict how likely an insurance event will happen within a given horizon;
   - Identify which factors increase the probability of survival of a patient within a given horizon.

Applications

1. Regression problems:
   - Prediction of house prices;
   - Explaining which factors influence the amount of rainfall.
2. Classification problems:
   - Classify objects from the pictures;
   - Explain why a particular person is rejected a mortgage.
3. Ranking problems:
   - Rank a set of articles w.r.t. their relevance to a given person;
   - Identify which factors cause a particular SP Rating of a country.
4. Time to event problems:
   - Predict how likely an insurance event will happen within a given horizon;
   - Identify which factors increase the probability of survival of a patient within a given horizon.

Applications

1. Regression problems:
   - Prediction of house prices;
   - Explaining which factors influence the amount of rainfall.
2. Classification problems:
   - Classify objects from the pictures;
   - Explain why a particular person is rejected a mortgage.
3. Ranking problems:
   - Rank a set of articles w.r.t. their relevance to a given person;
   - Identify which factors cause a particular SP Rating of a country.
4. Time to event problems:
   - Predict how likely an insurance event will happen within a given horizon;
   - Identify which factors increase the probability of survival of a patient within a given horizon.

Applications

1. Regression problems:
   - Prediction of house prices;
   - Explaining which factors influence the amount of rainfall.
2. Classification problems:
   - Classify objects from the pictures;
   - Explain why a particular person is rejected a mortgage.
3. Ranking problems:
   - Rank a set of articles w.r.t. their relevance to a given person;
   - Identify which factors cause a particular SP Rating of a country.
4. Time to event problems:
   - Predict how likely an insurance event will happen within a given horizon;
   - Identify which factors increase the probability of survival of a patient within a given horizon.

Models

1. Linear:
   - Linear regression (LR);
   - Generalized linear model (GLM);
   - Mixed linear models (LMM, GLMM);
   - Gaussian processes (GP);
   - Support vector machines, etc.

2. Nonlinear:
   - Classification and regression trees (CART);
   - Generalized additive models (GAM);
   - Generalized additive mixed models (GAMM);
   - Deep Gaussian processes (DGP);
   - Artificial neural networks (ANN), etc.

Models

1. Linear:
   - Linear regression (LR);
   - Generalized linear model (GLM);
   - Mixed linear models (LMM, GLMM);
   - Gaussian processes (GP);
   - Support vector machines, etc.

2. Nonlinear:
   - Classification and regression trees (CART);
   - Generalized additive models (GAM);
   - Generalized additive mixed models (GAMM);
   - Deep Gaussian processes (DGP);
   - Artificial neural netweorks (ANN), etc.

Problems and goals

Problems

1. Models for inference and predictions do not coincide;
2. Manual selection of an optimal model and a set of explanatory variables from a huge set of possibilities is hard and requires strong statistical expertise.

Main aims

1. Develop automatic model selection and configuration for both good inference and predictions;
2. Remain rigorous mathematically.

Problems and goals

Problems

1. Models for inference and predictions do not coincide;
2. Manual selection of an optimal model and a set of explanatory variables from a huge set of possibilities is hard and requires strong statistical expertise.

Main aims

1. Develop automatic model selection and configuration for both good inference and predictions;
2. Remain rigorous mathematically.

Bayesian approach as a natural solution

1. Consider a class of models $\Omega : m_1(Y|X,\theta_1),...,m_k(Y|X,\theta_k)$;
2. Put priors for all models $p(m_1),...,p(m_k)$ and their parameters $p(\theta_1|m_1),...,p(\theta_k|m_k)$;
3. Obtain the joint posterior distribution of models and parameters $p(m_1,\theta_1|D),...,p(m_k,\theta_k|D)$;
4. Make inference on Δ in the joint space of models and parameters: $p(\Delta|D) = \int_\Omega p(m|D)\int_\Theta p(\Delta|m,\theta,D)p(\theta|m,D)d\theta dm$;
5. Easy to extend by means of considering several classes of models $\Omega_1,...,\Omega_r$ with priors $p(\Omega_1),...,p(\Omega_r)$.

Bayesian approach as a natural solution

1. Consider a class of models $\Omega : m_1(Y|X,\theta_1), ..., m_k(Y|X,\theta_k)$;
2. Put priors for all models $p(m_1), ..., p(m_k)$ and their parameters $p(\theta_1|m_1), ..., p(\theta_k|m_k)$;
3. Obtain the joint posterior distribution of models and parameters $p(m_1,\theta_1|D), ..., p(m_k,\theta_k|D)$;
4. Make inference on Δ in the joint space of models and parameters: $p(\Delta|D) = \int_\Omega p(m|D) \int_\Theta p(\Delta|m,\theta,D)p(\theta|m,D)d\theta dm$;
5. Easy to extend by means of considering several classes of models $\Omega_1, ..., \Omega_r$ with priors $p(\Omega_1), ..., p(\Omega_r)$.

Bayesian approach as a natural solution

1. Consider a class of models $\Omega : m_1(Y|X,\theta_1), ..., m_k(Y|X,\theta_k)$;
2. Put priors for all models $p(m_1), ..., p(m_k)$ and their parameters $p(\theta_1|m_1), ..., p(\theta_k|m_k)$;
3. Obtain the joint posterior distribution of models and parameters $p(m_1,\theta_1|D), ..., p(m_k,\theta_k|D)$;
4. Make inference on Δ in the joint space of models and parameters: $p(\Delta|D) = \int_\Omega p(m|D) \int_\Theta p(\Delta|m,\theta,D)p(\theta|m,D)d\theta dm$;
5. Easy to extend by means of considering several classes of models $\Omega_1, ..., \Omega_r$ with priors $p(\Omega_1), ..., p(\Omega_r)$.

Bayesian approach as a natural solution

1. Consider a class of models $\Omega : m_1(Y|X,\theta_1), ..., m_k(Y|X,\theta_k)$;
2. Put priors for all models $p(m_1), ..., p(m_k)$ and their parameters $p(\theta_1|m_1), ..., p(\theta_k|m_k)$;
3. Obtain the joint posterior distribution of models and parameters $p(m_1,\theta_1|D), ..., p(m_k,\theta_k|D)$;
4. Make inference on Δ in the joint space of models and parameters: $p(\Delta|D) = \int_\Omega p(m|D) \int_\Theta p(\Delta|m,\theta,D)p(\theta|m,D)d\theta dm$;
5. Easy to extend by means of considering several classes of models $\Omega_1, ..., \Omega_r$ with priors $p(\Omega_1), ..., p(\Omega_r)$.

Bayesian approach as a natural solution

1. Consider a class of models $\Omega : m_1(Y|X,\theta_1), ..., m_k(Y|X,\theta_k)$;
2. Put priors for all models $p(m_1), ..., p(m_k)$ and their parameters $p(\theta_1|m_1), ..., p(\theta_k|m_k)$;
3. Obtain the joint posterior distribution of models and parameters $p(m_1,\theta_1|D), ..., p(m_k,\theta_k|D)$;
4. Make inference on Δ in the joint space of models and parameters: $p(\Delta|D) = \int_\Omega p(m|D) \int_\Theta p(\Delta|m,\theta,D)p(\theta|m,D)d\theta dm$;
5. Easy to extend by means of considering several classes of models $\Omega_1, ..., \Omega_r$ with priors $p(\Omega_1), ..., p(\Omega_r)$.

List of papers

1. Hubin, A., Storvik G. (2018). **Mode jumping MCMC for Bayesian variable selection in GLMM.** *Journal of Computational Statistics and Data Analysis; 2018 November; 127:281-297*.
2. Hubin, A., Storvik G., Frommlet F. (2018). **A novel algorithmic approach to Bayesian Logic Regression**. *Submitted to **Bayesian analysis** for publication.*
3. Hubin, A., Storvik G., Frommlet F. (2018). **Deep Bayesian regression models.** *Submitted to **JASA** for publication.*
4. Hubin, A., Hagmann M., Bodenstorfer B., Gola A., Bogdan M., Frommlet F. (2018). **A comprehensive study of Bayesian approaches to Genome-Wide Association Studies.** *Work in progress.*
5. Hubin, A., Storvik G. (2016). **Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA).** *Technical report; arXiv:1611.01450.*

List of papers

1. Hubin, A., Storvik G. (2018). **Mode jumping MCMC for Bayesian variable selection in GLMM.** *Journal of Computational Statistics and Data Analysis; 2018 November; 127:281-297*.
2. Hubin, A., Storvik G., Frommlet F. (2018). **A novel algorithmic approach to Bayesian Logic Regression**. *Submitted to **Bayesian analysis** for publication*.
3. Hubin, A., Storvik G., Frommlet F. (2018). **Deep Bayesian regression models.** *Submitted to **JASA** for publication.*
4. Hubin, A., Hagmann M., Bodenstorfer B., Gola A., Bogdan M., Frommlet F. (2018). **A comprehensive study of Bayesian approaches to Genome-Wide Association Studies.** *Work in progress.*
5. Hubin, A., Storvik G. (2016). **Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA).** *Technical report; arXiv:1611.01450.*

List of papers

1. Hubin, A., Storvik G. (2018). **Mode jumping MCMC for Bayesian variable selection in GLMM.** *Journal of Computational Statistics and Data Analysis; 2018 November; 127:281-297*.
2. Hubin, A., Storvik G., Frommlet F. (2018). **A novel algorithmic approach to Bayesian Logic Regression**. *Submitted to **Bayesian analysis** for publication*.
3. Hubin, A., Storvik G., Frommlet F. (2018). **Deep Bayesian regression models.** *Submitted to **JASA** for publication.*
4. Hubin, A., Hagmann M., Bodenstorfer B., Gola A., Bogdan M., Frommlet F. (2018). **A comprehensive study of Bayesian approaches to Genome-Wide Association Studies.** *Work in progress.*
5. Hubin, A., Storvik G. (2016). **Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA).** *Technical report; arXiv:1611.01450.*

List of papers

1. Hubin, A., Storvik G. (2018). **Mode jumping MCMC for Bayesian variable selection in GLMM.** *Journal of Computational Statistics and Data Analysis; 2018 November; 127:281-297*.
2. Hubin, A., Storvik G., Frommlet F. (2018). **A novel algorithmic approach to Bayesian Logic Regression**. *Submitted to **Bayesian analysis** for publication*.
3. Hubin, A., Storvik G., Frommlet F. (2018). **Deep Bayesian regression models.** *Submitted to **JASA** for publication.*
4. Hubin, A., Hagmann M., Bodenstorfer B., Gola A., Bogdan M., Frommlet F. (2018). **A comprehensive study of Bayesian approaches to Genome-Wide Association Studies.** *Work in progress*.
5. Hubin, A., Storvik G. (2016). **Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA).** *Technical report; arXiv:1611.01450*.

List of papers

1. Hubin, A., Storvik G. (2018). **Mode jumping MCMC for Bayesian variable selection in GLMM.** *Journal of Computational Statistics and Data Analysis; 2018 November; 127:281-297*.
2. Hubin, A., Storvik G., Frommlet F. (2018). **A novel algorithmic approach to Bayesian Logic Regression**. *Submitted to **Bayesian analysis** for publication*.
3. Hubin, A., Storvik G., Frommlet F. (2018). **Deep Bayesian regression models.** *Submitted to **JASA** for publication.*
4. Hubin, A., Hagmann M., Bodenstorfer B., Gola A., Bogdan M., Frommlet F. (2018). **A comprehensive study of Bayesian approaches to Genome-Wide Association Studies.** *Work in progress*.
5. Hubin, A., Storvik G. (2016). **Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA).** *Technical report; arXiv:1611.01450*.

Deep Bayesian Regression Model, Paper III

Sample of observations $i = 1, \dots, n$

- Y_i ... response data;
- $\boldsymbol{x}_i = (x_{i1}, \dots, x_{ip})$ ... p-dimensional vector of input covariates.

Specification of the model

From input variables a huge (but finite) number of features can be generated: $F_j(\boldsymbol{x}_i)$, $j = 1, ..., q$ (consider ordering w.r.t. complexity)
The **model** is then specified as GLM:

$$Y_i|\mu_i, \phi \sim \text{f}(y|\mu_i, \phi)\ , i = 1, ..., n; \tag{1}$$

$$\text{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i)\ . \tag{2}$$

- $\text{f}(\cdot|\mu, \phi)$ - density from exponential family with mean μ_i and dispersion parameter ϕ;
- h - link function;
- $\beta_j \in \mathbb{R}$ - regression coefficients of j-th feature;
- $\gamma_j \in \{0, 1\}$ - indicator variable for j-th feature.

Deep Bayesian Regression Model, Paper III

Sample of observations $i = 1, \dots, n$

- Y_i ...response data;
- $\boldsymbol{x}_i = (x_{i1}, \dots, x_{ip})$ ...p-dimensional vector of input covariates.

Specification of the model

From input variables a huge (but finite) number of features can be generated: $F_j(\boldsymbol{x}_i)$, $j = 1, ..., q$ (consider ordering w.r.t. complexity)
The **model** is then specified as GLM:

$$Y_i|\mu_i, \phi \sim \mathfrak{f}(y|\mu_i, \phi)\ , i = 1, ..., n; \tag{1}$$

$$\mathrm{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i)\ . \tag{2}$$

- $\mathfrak{f}(\cdot|\mu, \phi)$ - density from exponential family with mean μ_i and dispersion parameter ϕ;
- h - link function;
- $\beta_j \in \mathbb{R}$ - regression coefficients of j-th feature;
- $\gamma_j \in \{0, 1\}$ - indicator variable for j-th feature.

Deep Bayesian Regression Model, Paper III

Sample of observations $i = 1, \dots, n$

- Y_i ... response data;
- $\boldsymbol{x}_i = (x_{i1}, \dots, x_{ip})$ ... p-dimensional vector of input covariates.

Specification of the model

From input variables a huge (but finite) number of features can be generated: $F_j(\boldsymbol{x}_i)$, $j = 1, ..., q$ (consider ordering w.r.t. complexity)
The **model** is then specified as GLM:

$$Y_i|\mu_i, \phi \sim \mathfrak{f}(y|\mu_i, \phi)\ , i = 1, ..., n; \tag{1}$$

$$\mathrm{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i)\ . \tag{2}$$

- $\mathfrak{f}(\cdot|\mu, \phi)$ - density from exponential family with mean μ_i and dispersion parameter ϕ;
- h - link function;
- $\beta_j \in \mathbb{R}$ - regression coefficients of j-th feature;
- $\gamma_j \in \{0, 1\}$ - indicator variable for j-th feature.

Hierarchy of the features

A feature $F_j(\boldsymbol{x}), j \in \{p+1, ..., q\}$ can be constructed recursively through:

$$F_j(\boldsymbol{x}) = \begin{cases} v(F_k(\boldsymbol{x})), & \text{for a modification;} \\ F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}), & \text{for a crossover;} \\ v(\boldsymbol{\alpha}^T \mathbf{F}(\boldsymbol{x})), & \text{for a projection;} \end{cases}$$

- $F_k(\boldsymbol{x})$ and $F_l(\boldsymbol{x})$ are previously defined features $(k, l < j)$;
- $v \in \mathcal{G}$ is one of the allowed basic function from set $\mathcal{G}$;
- $\mathbf{F}(\boldsymbol{x})$ is a sub-vector of all possible features with indexes $1, ..., j-1$;
- A constraint on the complexity of feature $F_j(\boldsymbol{x})$ is defined by a finite number q of all possible features;
- Projections include modifications and crossovers as particular cases.

Hierarchy of the features

A feature $F_j(\boldsymbol{x}), j \in \{p+1, ..., q\}$ can be constructed recursively through:

$$F_j(\boldsymbol{x}) = \begin{cases} v(F_k(\boldsymbol{x})), & \text{for a modification;} \\ F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}), & \text{for a crossover;} \\ v(\boldsymbol{\alpha}^T \mathbf{F}(\boldsymbol{x})), & \text{for a projection;} \end{cases}$$

- $F_k(\boldsymbol{x})$ and $F_l(\boldsymbol{x})$ are previously defined features $(k, l < j)$;
- $v \in \mathcal{G}$ is one of the allowed basic function from set $\mathcal{G}$;
- $\mathbf{F}(\boldsymbol{x})$ is a sub-vector of all possible features with indexes $1, ..., j-1$;
- A constraint on the complexity of feature $F_j(\boldsymbol{x})$ is defined by a finite number q of all possible features;
- Projections include modifications and crossovers as particular cases.

Hierarchy of the features

A feature $F_j(\boldsymbol{x}), j \in \{p+1, \dots, q\}$ can be constructed recursively through:

$$F_j(\boldsymbol{x}) = \begin{cases} v(F_k(\boldsymbol{x})), & \text{for a modification;} \\ F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}), & \text{for a crossover;} \\ v(\boldsymbol{\alpha}^T \mathbf{F}(\boldsymbol{x})), & \text{for a projection;} \end{cases}$$

- $F_k(\boldsymbol{x})$ and $F_l(\boldsymbol{x})$ are previously defined features $(k, l < j)$;
- $v \in \mathcal{G}$ is one of the allowed basic function from set $\mathcal{G}$;
- $\mathbf{F}(\boldsymbol{x})$ is a sub-vector of all possible features with indexes $1, \dots, j-1$;
- A constraint on the complexity of feature $F_j(\boldsymbol{x})$ is defined by a finite number q of all possible features;
- Projections include modifications and crossovers as particular cases.

Hierarchy of the features

A feature $F_j(\boldsymbol{x}), j \in \{p+1, ..., q\}$ can be constructed recursively through:

$$F_j(\boldsymbol{x}) = \begin{cases} v(F_k(\boldsymbol{x})), & \text{for a modification;} \\ F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}), & \text{for a crossover;} \\ v(\boldsymbol{\alpha}^T \mathbf{F}(\boldsymbol{x})), & \text{for a projection;} \end{cases}$$

- $F_k(\boldsymbol{x})$ and $F_l(\boldsymbol{x})$ are previously defined features $(k, l < j)$;
- $v \in \mathcal{G}$ is one of the allowed basic function from set $\mathcal{G}$;
- $\mathbf{F}(\boldsymbol{x})$ is a sub-vector of all possible features with indexes $1, ..., j-1$;
- A constraint on the complexity of feature $F_j(\boldsymbol{x})$ is defined by a finite number q of all possible features;
- Projections include modifications and crossovers as particular cases.

Hierarchy of the features

A feature $F_j(\boldsymbol{x}), j \in \{p+1, ..., q\}$ can be constructed recursively through:

$$F_j(\boldsymbol{x}) = \begin{cases} v(F_k(\boldsymbol{x})), & \text{for a modification;} \\ F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}), & \text{for a crossover;} \\ v(\boldsymbol{\alpha}^T \mathbf{F}(\boldsymbol{x})), & \text{for a projection;} \end{cases}$$

- $F_k(\boldsymbol{x})$ and $F_l(\boldsymbol{x})$ are previously defined features $(k, l < j)$;
- $v \in \mathcal{G}$ is one of the allowed basic function from set $\mathcal{G}$;
- $\mathbf{F}(\boldsymbol{x})$ is a sub-vector of all possible features with indexes $1, ..., j-1$;
- A constraint on the complexity of feature $F_j(\boldsymbol{x})$ is defined by a finite number q of all possible features;
- Projections include modifications and crossovers as particular cases.

Other remarks

Types and meaning of functions in $\mathcal{G}$

- Neural Networks: $\text{logit}(x)$, $\tanh(x)$, $\text{erf}(x)$, $\text{ReLU}(x)$;
- Polynomials: $F_k(\mathbf{x}) * F_l(\mathbf{x}) = \exp\left(\log(F_k(\mathbf{x})) + \log(F_l(\mathbf{x}))\right);$
- CART: $\mathrm{I}(x \geq 1)$;
- MARS: $\max\{0, x - t\}$ and $\max\{0, t - x\}$;
- Fractional polynomials: $x^{\frac{1}{a}} = \exp\left(b \log(x)\right), b = \frac{1}{a};$
- **Logical *AND*, *OR* and *NOT*:** $L_k \wedge L_l = L_k * L_l$, $L_k \vee L_l = L_k + L_l - L_k * L_l$, **and** $\overline{L}_k = 1 - L_k$.

Potential extension of DBRM

Include Gaussian latent variables $\boldsymbol{\delta_k} = (\delta_{1k}, ..., \delta_{nk}) \sim N_n\left(\mathbf{0}, \boldsymbol{\Sigma}_k\right)$ to model correlation structure or overdispersion for GLM

$$\mathrm{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\mathbf{x}_i) + \sum_{k=1}^{r} \lambda_k \delta_{ik}.$$

Other remarks

Types and meaning of functions in $\mathcal{G}$

- Neural Networks: $\text{logit}(x)$, $\tanh(x)$, $\text{erf}(x)$, $\text{ReLU}(x)$;
- Polynomials: $F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}) = \exp\left(\log(F_k(\boldsymbol{x})) + \log(F_l(\boldsymbol{x}))\right);$
- CART: $\text{I}(x \geq 1)$;
- MARS: $\max\{0, x - t\}$ and $\max\{0, t - x\}$;
- Fractional polynomials: $x^{\frac{1}{a}} = \exp\left(b \log(x)\right), b = \frac{1}{a}$;
- **Logical *AND*, *OR* and *NOT*:** $L_k \wedge L_l = L_k * L_l$, $L_k \vee L_l = L_k + L_l - L_k * L_l$, **and** $\overline{L}_k = 1 - L_k$.

Potential extension of DBRM

Include Gaussian latent variables $\boldsymbol{\delta_k} = (\delta_{1k}, ..., \delta_{nk}) \sim N_n\left(\mathbf{0}, \boldsymbol{\Sigma}_k\right)$ to model correlation structure or overdispersion for GLM

$$\text{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i) + \sum_{k=1}^{r} \lambda_k \delta_{ik}.$$

Other remarks

Types and meaning of functions in $\mathcal{G}$

- Neural Networks: $\mathrm{logit}(x)$, $\tanh(x)$, $\mathrm{erf}(x)$, $\mathrm{ReLU}(x)$;
- Polynomials: $F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}) = \exp\left(\log(F_k(\boldsymbol{x})) + \log(F_l(\boldsymbol{x}))\right);$
- CART: $\mathrm{I}(x \geq 1)$;
- MARS: $\max\{0, x - t\}$ and $\max\{0, t - x\}$;
- Fractional polynomials: $x^{\frac{1}{a}} = \exp\left(b \log(x)\right), b = \frac{1}{a}$;
- **Logical *AND*, *OR* and *NOT*:** $L_k \wedge L_l = L_k * L_l$, $L_k \vee L_l = L_k + L_l - L_k * L_l$, **and** $\overline{L}_k = 1 - L_k$.

Potential extension of DBRM

Include Gaussian latent variables $\boldsymbol{\delta_k} = (\delta_{1k}, ..., \delta_{nk}) \sim N_n\left(\mathbf{0}, \boldsymbol{\Sigma}_k\right)$ to model correlation structure or overdispersion for GLM

$$\mathrm{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i) + \sum_{k=1}^{r} \lambda_k \delta_{ik}.$$

Other remarks

Types and meaning of functions in $\mathcal{G}$

- Neural Networks: $\text{logit}(x)$, $\tanh(x)$, $\text{erf}(x)$, $\text{ReLU}(x)$;
- Polynomials: $F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}) = \exp\left(\log(F_k(\boldsymbol{x})) + \log(F_l(\boldsymbol{x}))\right);$
- CART: $\text{I}(x \geq 1)$;
- MARS: $\max\{0, x - t\}$ and $\max\{0, t - x\}$;
- Fractional polynomials: $x^{\frac{1}{a}} = \exp\left(b \log(x)\right), b = \frac{1}{a}$;
- **Logical *AND*, *OR* and *NOT*:** $L_k \wedge L_l = L_k * L_l$, $L_k \vee L_l = L_k + L_l - L_k * L_l$, **and** $\overline{L}_k = 1 - L_k$.

Potential extension of DBRM

Include Gaussian latent variables $\boldsymbol{\delta_k} = (\delta_{1k}, ..., \delta_{nk}) \sim N_n\left(\boldsymbol{0}, \boldsymbol{\Sigma}_k\right)$ to model correlation structure or overdispersion for GLM

$$\text{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i) + \sum_{k=1}^{r} \lambda_k \delta_{ik}.$$

Other remarks

Types and meaning of functions in $\mathcal{G}$

- Neural Networks: $\text{logit}(x)$, $\tanh(x)$, $\text{erf}(x)$, $\text{ReLU}(x)$;
- Polynomials: $F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}) = \exp\left(\log(F_k(\boldsymbol{x})) + \log(F_l(\boldsymbol{x}))\right);$
- CART: $\text{I}(x \geq 1)$;
- MARS: $\max\{0, x - t\}$ and $\max\{0, t - x\}$;
- Fractional polynomials: $x^{\frac{1}{a}} = \exp\left(b \log(x)\right), b = \frac{1}{a};$
- **Logical *AND*, *OR* and *NOT*:** $L_k \wedge L_l = L_k * L_l$, $L_k \vee L_l = L_k + L_l - L_k * L_l$, **and** $\overline{L}_k = 1 - L_k$.

Potential extension of DBRM

Include Gaussian latent variables $\boldsymbol{\delta_k} = (\delta_{1k}, ..., \delta_{nk}) \sim N_n(\boldsymbol{0}, \boldsymbol{\Sigma}_k)$ to model correlation structure or overdispersion for GLM

$$\text{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i) + \sum_{k=1}^{r} \lambda_k \delta_{ik}.$$

Other remarks

Types and meaning of functions in $\mathcal{G}$

- Neural Networks: $\text{logit}(x)$, $\tanh(x)$, $\text{erf}(x)$, $\text{ReLU}(x)$;
- Polynomials: $F_k(\boldsymbol{x}) * F_l(\boldsymbol{x}) = \exp\left(\log(F_k(\boldsymbol{x})) + \log(F_l(\boldsymbol{x}))\right);$
- CART: $\mathrm{I}(x \geq 1)$;
- MARS: $\max\{0, x - t\}$ and $\max\{0, t - x\}$;
- Fractional polynomials: $x^{\frac{1}{a}} = \exp\left(b \log(x)\right), b = \frac{1}{a}$;
- **Logical *AND*, *OR* and *NOT*:** $L_k \wedge L_l = L_k * L_l$, $L_k \vee L_l = L_k + L_l - L_k * L_l$, **and** $\overline{L}_k = 1 - L_k$.

Potential extension of DBRM

Include Gaussian latent variables $\boldsymbol{\delta_k} = (\delta_{1k}, \ldots, \delta_{nk}) \sim N_n(\mathbf{0}, \boldsymbol{\Sigma}_k)$ to model correlation structure or overdispersion for GLM

$$\mathrm{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\boldsymbol{x}_i) + \sum_{k=1}^{r} \lambda_k \delta_{ik}.$$

Other remarks

Types and meaning of functions in $\mathcal{G}$

- Neural Networks: $\text{logit}(x)$, $\tanh(x)$, $\text{erf}(x)$, $\text{ReLU}(x)$;
- Polynomials: $F_k(\mathbf{x}) * F_l(\mathbf{x}) = \exp\left(\log(F_k(\mathbf{x})) + \log(F_l(\mathbf{x}))\right);$
- CART: $\text{I}(x \geq 1)$;
- MARS: $\max\{0, x - t\}$ and $\max\{0, t - x\}$;
- Fractional polynomials: $x^{\frac{1}{a}} = \exp\left(b \log(x)\right), b = \frac{1}{a}$;
- **Logical *AND*, *OR* and *NOT*:** $L_k \wedge L_l = L_k * L_l$**,** $L_k \vee L_l = L_k + L_l - L_k * L_l$**, and** $\overline{L}_k = 1 - L_k$**.**

Potential extension of DBRM

Include Gaussian latent variables $\boldsymbol{\delta_k} = (\delta_{1k}, ..., \delta_{nk}) \sim N_n\left(\mathbf{0}, \boldsymbol{\Sigma}_k\right)$ to model correlation structure or overdispersion for GLM

$$\text{h}(\mu_i) = \beta_0 + \sum_{j=1}^{q} \gamma_j \beta_j F_j(\mathbf{x}_i) + \sum_{k=1}^{r} \lambda_k \delta_{ik}.$$

Prior specification

Model priors

Model topology defined by $\mathfrak{m} = (\gamma_1, \dots, \gamma_q)$
Priors of $\mathfrak{m}$ should guarantee regularization (parsimonious models)

$$p(\mathfrak{m}) \propto \mathrm{I}\left(|\gamma_{1:q}| \leq Q\right) \prod_{j=1}^{q} a^{\gamma_j c(F_j(\mathbf{x}))}, \quad 0 < a < 1 \tag{3}$$

- $c(F_j(\mathbf{x})) \geq 0$ non-decreasing **complexity measure** of feature $F_j(\mathbf{x})$;
- Q is the maximal number of features per given model.

Example: Total width (sum of width of all nested features involved);
GLMM : boils down to a simple prior $p(\mathfrak{m}) \propto \prod_{j=1}^{q} a^{\gamma_j}$;
BLR: $a^{c(L_j)} = \frac{1}{N(s_j)}$, $s_j \leq C_{max}$, $N(s_j)$ - number of trees with s_j leaves.

Parameter priors

Priors for β_j (and ϕ): Problem-specific and computational considerations
$\Rightarrow$ Conjugate priors, Jeffrey's priors, robust g priors, etc.

Prior specification

Model priors

Model topology defined by $\mathfrak{m} = (\gamma_1, \ldots, \gamma_q)$
Priors of $\mathfrak{m}$ should guarantee regularization (parsimonious models)

$$p(\mathfrak{m}) \propto \mathrm{I}\left(|\gamma_{1:q}| \leq Q\right) \prod_{j=1}^{q} a^{\gamma_j c(F_j(\mathbf{x}))}, \quad 0 < a < 1 \tag{3}$$

- $c(F_j(\mathbf{x})) \geq 0$ non-decreasing **complexity measure** of feature $F_j(\mathbf{x})$;
- Q is the maximal number of features per given model.

Example: Total width (sum of width of all nested features involved);
GLMM : boils down to a simple prior $p(\mathfrak{m}) \propto \prod_{j=1}^{q} a^{\gamma_j}$;
BLR: $a^{c(L_j)} = \frac{1}{N(s_j)}$, $s_j \leq C_{max}$, $N(s_j)$ - number of trees with s_j leaves.

Parameter priors

Priors for β_j (and ϕ): Problem-specific and computational considerations
$\Rightarrow$ Conjugate priors, Jeffrey's priors, robust g priors, etc.

Prior specification

Model priors

Model topology defined by $\mathfrak{m} = (\gamma_1, \dots, \gamma_q)$
Priors of $\mathfrak{m}$ should guarantee regularization (parsimonious models)

$$p(\mathfrak{m}) \propto \mathrm{I}\left(|\gamma_{1:q}| \leq Q\right) \prod_{j=1}^{q} a^{\gamma_j c(F_j(\mathbf{x}))}\,, \quad 0 < a < 1 \tag{3}$$

- $c(F_j(\mathbf{x})) \geq 0$ non-decreasing **complexity measure** of feature $F_j(\mathbf{x})$;
- Q is the maximal number of features per given model.

Example: Total width (sum of width of all nested features involved);
GLMM : boils down to a simple prior $p(\mathfrak{m}) \propto \prod_{j=1}^{q} a^{\gamma_j}$;
BLR: $a^{c(L_j)} = \frac{1}{N(s_j)}$, $s_j \leq C_{max}$, $N(s_j)$ - number of trees with s_j leaves.

Parameter priors

Priors for β_j (and ϕ): Problem-specific and computational considerations
$\Rightarrow$ Conjugate priors, Jeffrey's priors, robust g priors, etc.

Computation of model posterior

Marginal likelihood $P(D|\mathfrak{m}) = \int_{\theta_{\mathfrak{m}} \in \Theta_{\mathfrak{m}}} p(D|\theta_{\mathfrak{m}}, \mathfrak{m}) p(\theta_{\mathfrak{m}}|\mathfrak{m}) d\theta_{\mathfrak{m}}$
Conjugate priors, Laplace approximation, INLA, Variational inference, etc.

$$P(\mathfrak{m}|D) = \frac{P(D|\mathfrak{m})p(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega} P(D|\mathfrak{m}')p(\mathfrak{m}')}\ , \tag{4}$$

Denominator

Model space Ω extremely large

- Classical MCMC: use relative frequency of models in Markov chain;
- Alternative: approximate by summing over set of visited models Ω^*

$$P(\mathfrak{m}|D) \approx \frac{P(D|\mathfrak{m})P(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega^*} P(D|\mathfrak{m}')P(\mathfrak{m}')} \quad \text{for} \quad \mathfrak{m} \in \Omega^*. \tag{5}$$

Computation of model posterior

Marginal likelihood $P(D|\mathfrak{m}) = \int_{\theta_\mathfrak{m} \in \Theta_\mathfrak{m}} p(D|\theta_\mathfrak{m}, \mathfrak{m}) p(\theta_\mathfrak{m}|\mathfrak{m}) d\theta_\mathfrak{m}$
Conjugate priors, Laplace approximation, INLA, Variational inference, etc.

$$P(\mathfrak{m}|D) = \frac{P(D|\mathfrak{m})p(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega} P(D|\mathfrak{m}')p(\mathfrak{m}')} \ , \tag{4}$$

Denominator

Model space Ω extremely large

- Classical MCMC: use relative frequency of models in Markov chain;
- Alternative: approximate by summing over set of visited models Ω^*

$$P(\mathfrak{m}|D) \approx \frac{P(D|\mathfrak{m})P(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega^*} P(D|\mathfrak{m}')P(\mathfrak{m}')} \quad \text{for} \quad \mathfrak{m} \in \Omega^*. \tag{5}$$

Computation of model posterior

Marginal likelihood $P(D|\mathfrak{m}) = \int_{\theta_\mathfrak{m} \in \Theta_\mathfrak{m}} p(D|\theta_\mathfrak{m}, \mathfrak{m}) p(\theta_\mathfrak{m}|\mathfrak{m}) d\theta_\mathfrak{m}$
Conjugate priors, Laplace approximation, INLA, Variational inference, etc.

$$P(\mathfrak{m}|D) = \frac{P(D|\mathfrak{m})p(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega} P(D|\mathfrak{m}')p(\mathfrak{m}')} \;, \tag{4}$$

Denominator

Model space Ω extremely large

- Classical MCMC: use relative frequency of models in Markov chain;
- Alternative: approximate by summing over set of visited models Ω^*

$$P(\mathfrak{m}|D) \approx \frac{P(D|\mathfrak{m})P(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega^*} P(D|\mathfrak{m}')P(\mathfrak{m}')} \quad \text{for} \quad \mathfrak{m} \in \Omega^*. \tag{5}$$

Computation of model posterior

Marginal likelihood $P(D|\mathfrak{m}) = \int_{\theta_{\mathfrak{m}} \in \Theta_{\mathfrak{m}}} p(D|\theta_{\mathfrak{m}}, \mathfrak{m}) p(\theta_{\mathfrak{m}}|\mathfrak{m}) d\theta_{\mathfrak{m}}$
Conjugate priors, Laplace approximation, INLA, Variational inference, etc.

$$P(\mathfrak{m}|D) = \frac{P(D|\mathfrak{m})p(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega} P(D|\mathfrak{m}')p(\mathfrak{m}')} \quad , \tag{4}$$

Denominator

Model space Ω extremely large

- Classical MCMC: use relative frequency of models in Markov chain;
- Alternative: approximate by summing over set of visited models Ω^*

$$P(\mathfrak{m}|D) \approx \frac{P(D|\mathfrak{m})P(\mathfrak{m})}{\sum_{\mathfrak{m}' \in \Omega^*} P(D|\mathfrak{m}')P(\mathfrak{m}')} \quad \text{for} \quad \mathfrak{m} \in \Omega^*. \tag{5}$$

Algorithmic Details. MCMC

Hubin and Storvik (2018)

Variable selection with p potential exploratory variables

- 2^p potential models;
- Multimodality $\Rightarrow$ MCMC trapped by local maxima or extremely low acceptance ratio.

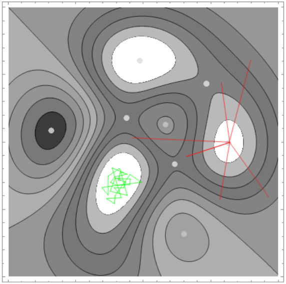


Figure: MCMC with either small (green) or large (red) proposals

Algorithmic Details. **MJ**MCMC

Mode jumping MCMC (MJMCMC)

- After certain number of MCMC steps make a jump
  $\Rightarrow$ Random model plus local improvement creates new proposal;
- Get valid acceptance probability for Metropolis-Hastings.

Algorithmic Details. MJMCMC

Mode jumping MCMC (MJMCMC)

- After certain number of MCMC steps make a jump
  $\Rightarrow$ Random model plus local improvement creates new proposal;
- Get valid acceptance probability for Metropolis-Hastings.

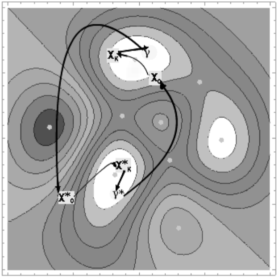


Figure: Locally optimized with randomization proposals

The protein activity data. 2^{88} models. Multiple modes

Comparison to other algorithms. On 2^{20} unique models visited for MJMCMC, ESS and BAS and 88×2^{20} iterations of RS.

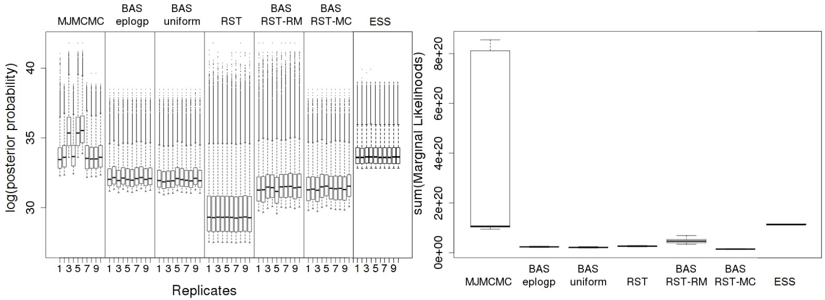


Figure: 100000 best mliks found (left) and posterior masses captured (right). Bayesian linear regression with a g-prior is addressed

Algorithmic Details.**GMJ**MCMC

Problem for DBRM (Logic Regressions)

- Diffcult to fully specify the model space Ω;
- Also q too large for MJMCMC.

GMJMCMC Solution

- MJMCMC is embedded in a genetic algorithm which updates a finite population of features of size $d \ll q$;
- Populations $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{T_{max}} \quad \Rightarrow \quad$ each $\mathcal{S}_t$ is a set of d features;
- Initialization of $\mathcal{S}_1$ using best original input variables;
- Then use evolutionary dynamic to generate further populations using filtration, modification, crossover and projection operators.

Algorithmic Details.**GMJ**MCMC

Problem for DBRM (Logic Regressions)

- Diffcult to fully specify the model space Ω;
- Also q too large for MJMCMC.

GMJMCMC Solution

- MJMCMC is embedded in a genetic algorithm which updates a finite population of features of size $d \ll q$;
- Populations $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{T_{max}} \quad \Rightarrow \quad$ each $\mathcal{S}_t$ is a set of d features;
- Initialization of $\mathcal{S}_1$ using best original input variables;
- Then use evolutionary dynamic to generate further populations using filtration, modification, crossover and projection operators.

Algorithmic Details.**GMJ**MCMC

Problem for DBRM (Logic Regressions)

- Diffcult to fully specify the model space Ω;
- Also q too large for MJMCMC.

GMJMCMC Solution

- MJMCMC is embedded in a genetic algorithm which updates a finite population of features of size $d \ll q$;
- Populations $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{T_{max}} \quad \Rightarrow \quad$ each $\mathcal{S}_t$ is a set of d features;
- Initialization of $\mathcal{S}_1$ using best original input variables;
- Then use evolutionary dynamic to generate further populations using filtration, modification, crossover and projection operators.

Algorithmic Details.**GMJ**MCMC

Problem for DBRM (Logic Regressions)

- Diffcult to fully specify the model space Ω;
- Also q too large for MJMCMC.

GMJMCMC Solution

- MJMCMC is embedded in a genetic algorithm which updates a finite population of features of size $d \ll q$;
- Populations $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{T_{max}} \quad \Rightarrow \quad$ each $\mathcal{S}_t$ is a set of d features;
- Initialization of $\mathcal{S}_1$ using best original input variables;
- Then use evolutionary dynamic to generate further populations using filtration, modification, crossover and projection operators.

Algorithmic Details.**GMJ**MCMC

Problem for DBRM (Logic Regressions)

- Diffcult to fully specify the model space Ω;
- Also q too large for MJMCMC.

GMJMCMC Solution

- MJMCMC is embedded in a genetic algorithm which updates a finite population of features of size $d \ll q$;
- Populations $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{T_{max}} \quad \Rightarrow \quad$ each $\mathcal{S}_t$ is a set of d features;
- Initialization of $\mathcal{S}_1$ using best original input variables;
- Then use evolutionary dynamic to generate further populations using filtration, modification, crossover and projection operators.

Algorithm. RGMJMCMC

Reversible GMJMCMC (RGMJMCMC)

- For every search space we suggest running MJMCMC for a sufficient amount of time;
- Otherwise a transition $\mathfrak{m} \to \mathfrak{m}^* \to \mathfrak{m}^1 \to ... \to \mathfrak{m}^k \to \mathfrak{m}'$ is considered with a given probability kernel:
  - $q(\mathfrak{m}^*|\mathfrak{m})$ is the proposal for the solution in the new search space induced by $\mathcal{S}'$;
  - Transitions $\mathfrak{m}^1 \to ... \to \mathfrak{m}^k$ are generated by local MJMCMC in the new search space induced by $\mathcal{S}'$;
  - Transition $\mathfrak{m}^k \to \mathfrak{m}'$ is some randomization at the end of the procedure;
- Acceptance probability for such a procedure is $r_m = \min\{1, \alpha_m\}$

$$\alpha_m = \frac{\pi(\mathfrak{m}')q(\mathfrak{m}|\mathfrak{m}'_k)}{\pi(\mathfrak{m})q(\mathfrak{m}'|\mathfrak{m}_k)}. \tag{6}$$

Algorithm. **RGMJM**CMC

Reversible GMJMCMC (RGMJMCMC)

- For every search space we suggest running MJMCMC for a sufficient amount of time;
- Otherwise a transition $\mathfrak{m} \to \mathfrak{m}^* \to \mathfrak{m}^1 \to \ldots \to \mathfrak{m}^k \to \mathfrak{m}'$ is considered with a given probability kernel:
  - $q(\mathfrak{m}^*|\mathfrak{m})$ is the proposal for the solution in the new search space induced by $\mathcal{S}'$;
  - Transitions $\mathfrak{m}^1 \to \ldots \to \mathfrak{m}^k$ are generated by local MJMCMC in the new search space induced by $\mathcal{S}'$;
  - Transition $\mathfrak{m}^k \to \mathfrak{m}'$ is some randomization at the end of the procedure;
- Acceptance probability for such a procedure is $r_m = \min\{1, \alpha_m\}$

$$\alpha_m = \frac{\pi(\mathfrak{m}')q(\mathfrak{m}|\mathfrak{m}'_k)}{\pi(\mathfrak{m})q(\mathfrak{m}'|\mathfrak{m}_k)}. \tag{6}$$

Algorithm. **RGMJM**CMC

Reversible GMJMCMC (RGMJMCMC)

- For every search space we suggest running MJMCMC for a sufficient amount of time;
- Otherwise a transition $\mathfrak{m} \to \mathfrak{m}^* \to \mathfrak{m}^1 \to ... \to \mathfrak{m}^k \to \mathfrak{m}'$ is considered with a given probability kernel:
  - $q(\mathfrak{m}^*|\mathfrak{m})$ is the proposal for the solution in the new search space induced by $\mathcal{S}'$;
  - Transitions $\mathfrak{m}^1 \to ... \to \mathfrak{m}^k$ are generated by local MJMCMC in the new search space induced by $\mathcal{S}'$;
  - Transition $\mathfrak{m}^k \to \mathfrak{m}'$ is some randomization at the end of the procedure;
- Acceptance probability for such a procedure is $r_m = \min\{1, \alpha_m\}$

$$\alpha_m = \frac{\pi(\mathfrak{m}')q(\mathfrak{m}|\mathfrak{m}'_k)}{\pi(\mathfrak{m})q(\mathfrak{m}'|\mathfrak{m}_k)}. \tag{6}$$

Algorithm. **RGMJM**CMC

Reversible GMJMCMC (RGMJMCMC)

- For every search space we suggest running MJMCMC for a sufficient amount of time;
- Otherwise a transition $\mathfrak{m} \to \mathfrak{m}^* \to \mathfrak{m}^1 \to \ldots \to \mathfrak{m}^k \to \mathfrak{m}'$ is considered with a given probability kernel:
  - $q(\mathfrak{m}^*|\mathfrak{m})$ is the proposal for the solution in the new search space induced by $\mathcal{S}'$;
  - Transitions $\mathfrak{m}^1 \to \ldots \to \mathfrak{m}^k$ are generated by local MJMCMC in the new search space induced by $\mathcal{S}'$;
  - Transition $\mathfrak{m}^k \to \mathfrak{m}'$ is some randomization at the end of the procedure;
- Acceptance probability for such a procedure is $r_m = \min\{1, \alpha_m\}$

$$\alpha_m = \frac{\pi(\mathfrak{m}')q(\mathfrak{m}|\mathfrak{m}'_k)}{\pi(\mathfrak{m})q(\mathfrak{m}'|\mathfrak{m}_k)}. \tag{6}$$

Algorithm. **RGMJM**CMC

Reversible GMJMCMC (RGMJMCMC)

- For every search space we suggest running MJMCMC for a sufficient amount of time;
- Otherwise a transition $\mathfrak{m} \rightarrow \mathfrak{m}^* \rightarrow \mathfrak{m}^1 \rightarrow \ldots \rightarrow \mathfrak{m}^k \rightarrow \mathfrak{m}'$ is considered with a given probability kernel:
  - $q(\mathfrak{m}^*|\mathfrak{m})$ is the proposal for the solution in the new search space induced by $\mathcal{S}'$;
  - Transitions $\mathfrak{m}^1 \rightarrow \ldots \rightarrow \mathfrak{m}^k$ are generated by local MJMCMC in the new search space induced by $\mathcal{S}'$;
  - Transition $\mathfrak{m}^k \rightarrow \mathfrak{m}'$ is some randomization at the end of the procedure;
- Acceptance probability for such a procedure is $r_m = \min\{1, \alpha_m\}$

$$\alpha_m = \frac{\pi(\mathfrak{m}')q(\mathfrak{m}|\mathfrak{m}'_k)}{\pi(\mathfrak{m})q(\mathfrak{m}'|\mathfrak{m}_k)}. \tag{6}$$

Algorithm. **RGMJM**CMC

Reversible GMJMCMC (RGMJMCMC)

- For every search space we suggest running MJMCMC for a sufficient amount of time;
- Otherwise a transition $\mathfrak{m} \to \mathfrak{m}^* \to \mathfrak{m}^1 \to ... \to \mathfrak{m}^k \to \mathfrak{m}'$ is considered with a given probability kernel:
  - $q(\mathfrak{m}^*|\mathfrak{m})$ is the proposal for the solution in the new search space induced by $\mathcal{S}'$;
  - Transitions $\mathfrak{m}^1 \to ... \to \mathfrak{m}^k$ are generated by local MJMCMC in the new search space induced by $\mathcal{S}'$;
  - Transition $\mathfrak{m}^k \to \mathfrak{m}'$ is some randomization at the end of the procedure;
- Acceptance probability for such a procedure is $r_m = \min\{1, \alpha_m\}$

$$\alpha_m = \frac{\pi(\mathfrak{m}')q(\mathfrak{m}|\mathfrak{m}'_k)}{\pi(\mathfrak{m})q(\mathfrak{m}'|\mathfrak{m}_k)}. \tag{6}$$

Parallelization

1. Method 1
   - Embarrassing parallelization of B (R)(G)MJMCMCs on different CPUs;
   - Combine all unique models visited by all B chains into Ω^*;
   - Evaluate the posterior of interest as:

$$\widehat{p}(\Delta|D) = \sum_{\mathfrak{m}\in\Omega^*} p(\Delta|\mathfrak{m}, D)\widehat{p}(\mathfrak{m}|D) \ . \tag{7}$$

2. Method 2
   - A memory efficient alternative is:

$$\tilde{p}(\Delta|D) = \sum_{b=1}^{B} u_b\tilde{p}_b(\Delta|D) \ , \tag{8}$$

   - Here u_b is a set of arbitrary normalized weights;
   - $\tilde{p}_b(\Delta|Y)$ are the posteriors from individual runs.

Parallelization

1. Method 1
   - Embarrassing parallelization of B (R)(G)MJMCMCs on different CPUs;
   - Combine all unique models visited by all B chains into Ω^*;
   - Evaluate the posterior of interest as:

$$\widehat{p}(\Delta|D) = \sum_{\mathfrak{m}\in\Omega^*} p(\Delta|\mathfrak{m}, D)\widehat{p}(\mathfrak{m}|D)\ . \tag{7}$$

2. Method 2
   - A memory efficient alternative is:

$$\tilde{p}(\Delta|D) = \sum_{b=1}^{B} u_b \tilde{p}_b(\Delta|D)\ , \tag{8}$$

   - Here u_b is a set of arbitrary normalized weights;
   - $\tilde{p}_b(\Delta|Y)$ are the posteriors from individual runs.

Examples on Prediction

Comparison of algorithms

1. DBRM with different settings:
   - DBRM_G and DBRM_G_PAR: Using GMJMCMC with 1 or 32 threads;
   - DBRM_R and DBRM_R_PAR: Using RGMJMCMC with 1 or 32 threads;
   - LBRM_G: DBRM algorithm but without accepting non-linear features.
2. Competing algorithms:
   - DEEPNETS: Deep dense neural networks;
   - RFOREST: Random forest;
   - TXGBOOST: Tree based gradient boosting;
   - LXGBOOST: Linear gradient boosting;
   - LASSO and RIDGE;
   - LR: simple logistic regression;
   - NBAYES: Naive Bayes classifyer.

Examples on Prediction

Comparison of algorithms

1. DBRM with different settings:
   - DBRM_G and DBRM_G_PAR: Using GMJMCMC with 1 or 32 threads;
   - DBRM_R and DBRM_R_PAR: Using RGMJMCMC with 1 or 32 threads;
   - LBRM_G: DBRM algorithm but without accepting non-linear features.
2. Competing algorithms:
   - DEEPNETS: Deep dense neural networks;
   - RFOREST: Random forest;
   - TXGBOOST: Tree based gradient boosting;
   - LXGBOOST: Linear gradient boosting;
   - LASSO and RIDGE;
   - LR: simple logistic regression;
   - NBAYES: Naive Bayes classifyer.

Examples for Prediction

Three binary classification tasks (Deep logistic regression)

- Asteroids data: Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude;
- Breast cancer data: Radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension;
- Spam emails: 58 characteristics, including 57 continuous and 1 nominal variable, where most of these are concerned with the frequency of particular words or characters. 3 provide different measurements on the sequence length of consecutive capital letters.

Measures for evaluation

- Accuracy of predictions (ACC);
- False positive rate (FPR);
- False negative rate (FNR).

Examples for Prediction

Three binary classification tasks (Deep logistic regression)

- Asteroids data: Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude;
- Breast cancer data: Radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension;
- Spam emails: 58 characteristics, including 57 continuous and 1 nominal variable, where most of these are concerned with the frequency of particular words or characters. 3 provide different measurements on the sequence length of consecutive capital letters.

Measures for evaluation

- Accuracy of predictions (ACC);
- False positive rate (FPR);
- False negative rate (FNR).

Examples for Prediction

Three binary classification tasks (Deep logistic regression)

- Asteroids data: Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude;
- Breast cancer data: Radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension;
- Spam emails: 58 characteristics, including 57 continuous and 1 nominal variable, where most of these are concerned with the frequency of particular words or characters. 3 provide different measurements on the sequence length of consecutive capital letters.

Measures for evaluation

- Accuracy of predictions (ACC);
- False positive rate (FPR);
- False negative rate (FNR).

Examples for Prediction

Three binary classification tasks (Deep logistic regression)

- Asteroids data: Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude;
- Breast cancer data: Radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension;
- Spam emails: 58 characteristics, including 57 continuous and 1 nominal variable, where most of these are concerned with the frequency of particular words or characters. 3 provide different measurements on the sequence length of consecutive capital letters.

Measures for evaluation

- Accuracy of predictions (ACC);
- False positive rate (FPR);
- False negative rate (FNR).

Examples for Prediction

Three binary classification tasks (Deep logistic regression)

- Asteroids data: Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude;
- Breast cancer data: Radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension;
- Spam emails: 58 characteristics, including 57 continuous and 1 nominal variable, where most of these are concerned with the frequency of particular words or characters. 3 provide different measurements on the sequence length of consecutive capital letters.

Measures for evaluation

- Accuracy of predictions (ACC);
- False positive rate (FPR);
- False negative rate (FNR).

Deep Bayesian logsitic regression

$$y_i = y|\rho_i \sim \text{Binom}(1, \rho_i); \tag{9}$$

$$\rho_i = \frac{e^{\gamma_0\beta_0 + \sum_{j=1}^{q} \gamma_j\beta_j F_j(\mathbf{x}_i)}}{1 + e^{\gamma_0\beta_0 + \sum_{j=1}^{q};\gamma_j\beta_j F_j(\mathbf{x}_i)}}; \tag{10}$$

$$p(\gamma) \propto \mathrm{I}\left(|\gamma_{1:q}| \leq 20\right) \prod_{j=1}^{q} \exp(-\gamma_j 2c(F_j(x))); \tag{11}$$

$$p(\boldsymbol{\beta}|\gamma) = |J_n^{\gamma}(\boldsymbol{\beta})|^{\frac{1}{2}} \; ; \tag{12}$$

$$\mathcal{G} = \{\text{sigmoid}(x), \mathrm{I}(x > 1), \text{ReLu}(x), x^{\frac{1}{3}}, x^{\frac{1}{5}}\}. \tag{13}$$

Example 1

Asteroid classification

Training sample: $n = 64$, Test sample: $n_p = 20702$

Algorithm	ACC	FNR	FPR
LBRM	0.9999 (0.9999,0.9999)	0.0001	0.0002
DBRM_G_PAR	0.9998 (0.9986,1.0000)	0.0002	0.0000
DBRM_R_PAR	0.9998 (0.9964,0.9999)	0.0002	0.0000
DBRM_R	0.9998 (0.9946,1.0000)	0.0002	0.0002
DBRM_G	0.9998 (0.9942,1.0000)	0.0002	0.0002
LASSO	0.9991 (-,-)	0.0013	0.0000
RIDGE	0.9982 (-,-)	0.0026	0.0000
LXGBOOST	0.9980 (0.9980,0.9980)	0.0029	0.0000
LR	0.9963 (-,-)	0.0054	0.0000
DEEPNETS	0.9728 (0.8979,0.9979)	0.0384	0.0000
TXGBOOST	0.8283 (0.8283,0.8283)	0.0005	0.3488
RFOREST	0.8150 (0.6761,0.9991)	0.1972	0.0162
NBAYES	0.6471 (-,-)	0.0471	0.4996

Example 2

Breast cancer data

Training sample: $n = 142$, Test sample: $n_p = 427$

Algorithm	ACC	FNR	FPR
DBRM_R_PAR	0.9765 (0.9695,0.9812)	0.0479	0.0074
DBRM_G_PAR	0.9742 (0.9695,0.9812)	0.0479	0.0111
RIDGE	0.9742 (-,-)	0.0592	0.0037
LBRM	0.9718 (0.9648,0.9765)	0.0592	0.0074
DBRM_G	0.9695 (0.9554,0.9789)	0.0536	0.0148
DEEPNETS	0.9695 (0.9225,0.9789)	0.0674	0.0074
DBRM_R	0.9671 (0.9577,0.9812)	0.0536	0.0148
LR	0.9671 (-,-)	0.0479	0.0220
LASSO	0.9577 (-,-)	0.0756	0.0184
LXGBOOST	0.9554 (0.9554,0.9554)	0.0809	0.0184
TXGBOOST	0.9531 (0.9484,0.9601)	0.0647	0.0326
RFOREST	0.9343 (0.9038,0.9624)	0.0914	0.0361
NBAYES	0.9272 (-,-)	0.0305	0.0887

Example 3

Spam data

Training sample: $n = 1536$, Test sample: $n_p = 3065$

Algorithm	ACC	FNR	FPR
TXGBOOST	0.9465 (0.9442,0.9481)	0.0783	0.0320
RFOREST	0.9328 (0.9210,0.9413)	0.0814	0.0484
DEEPNETS	0.9292 (0.9002,0.9357)	0.0846	0.0531
DBRM_R_PAR	0.9268 (0.9162,0.9390)	0.0897	0.0538
DBRM_G_PAR	0.9251 (0.9139,0.9377)	0.0897	0.0552
DBRM_G	0.9243 (0.9113,0.9328)	0.0927	0.0552
DBRM_R	0.9237 (0.9106,0.9351)	0.0917	0.0557
LR	0.9194 (-,-)	0.0681	0.0788
LBRM	0.9178 (0.9168,0.9188)	0.1090	0.0528
LASSO	0.9171 (-,-)	0.1077	0.0548
RIDGE	0.9152 (-,-)	0.1288	0.0415
LXGBOOST	0.9139 (0.9139,0.9139)	0.1083	0.0591
NBAYES	0.7811 (-,-)	0.0801	0.2342

Summary prediction examples (DBRM)

Example1: Asteroid

complexity	G	R	G_PAR	R_PAR	LBRM
1	8.96	8.97	9.00	9.00	9.00
2	2.58	2.62	0.05	0.15	0.00
Total	11.54	11.59	9.05	9.15	9.00

Example2: Breast cancer

complexity	G	R	G_PAR	R_PAR	LBRM
1	11.30	11.73	14.20	10.79	29.83
2	3.09	3.06	0.04	0.21	0.00
3	0.30	0.00	0.00	0.00	0.00
6	0.00	0.01	0.00	0.00	0.00
7	0.00	0.01	0.00	0.00	0.00
Total	14.42	14.81	14.24	11.00	29.83

- Non-linear features don't play an important role;
- Still DBRM performs very well $\Rightarrow$ not too much overfitting;
- Parallel version much better performance.

Summary prediction examples (DBRM)

Example1: Asteroid

complexity	G	R	G_PAR	R_PAR	LBRM
1	8.96	8.97	9.00	9.00	9.00
2	2.58	2.62	0.05	0.15	0.00
Total	11.54	11.59	9.05	9.15	9.00

Example2: Breast cancer

complexity	G	R	G_PAR	R_PAR	LBRM
1	11.30	11.73	14.20	10.79	29.83
2	3.09	3.06	0.04	0.21	0.00
3	0.30	0.00	0.00	0.00	0.00
6	0.00	0.01	0.00	0.00	0.00
7	0.00	0.01	0.00	0.00	0.00
Total	14.42	14.81	14.24	11.00	29.83

- Non-linear features don't play an important role;
- Still DBRM performs very well $\Rightarrow$ not too much overfitting;
- Parallel version much better performance.

Summary prediction examples (DBRM)

Example1: Asteroid

complexity	G	R	G_PAR	R_PAR	LBRM
1	8.96	8.97	9.00	9.00	9.00
2	2.58	2.62	0.05	0.15	0.00
Total	11.54	11.59	9.05	9.15	9.00

Example2: Breast cancer

complexity	G	R	G_PAR	R_PAR	LBRM
1	11.30	11.73	14.20	10.79	29.83
2	3.09	3.06	0.04	0.21	0.00
3	0.30	0.00	0.00	0.00	0.00
6	0.00	0.01	0.00	0.00	0.00
7	0.00	0.01	0.00	0.00	0.00
Total	14.42	14.81	14.24	11.00	29.83

- Non-linear features don't play an important role;
- Still DBRM performs very well $\Rightarrow$ not too much overfitting;
- Parallel version much better performance.

Summary prediction examples (DBRM)

Example3: Spam mail

complexity	G	R	G_PAR	R_PAR	LBRM
1	36.34	36.09	39.87	39.17	49.83
2	14.45	14.83	21.47	22.43	0.00
3	2.83	3.17	5.24	5.81	0.00
4	0.69	0.57	1.36	1.36	0.00
5	1.15	1.09	1.56	1.68	0.00
6	0.92	0.74	1.24	1.07	0.00
7	0.37	0.40	0.57	0.42	0.00
8	0.25	0.22	0.33	0.17	0.00
9	0.04	0.08	0.16	0.11	0.00
$\geq$10	0.15	0.11	0.11	0.18	0.00
Total	57.190	57.300	71.910	72.400	49.830

- Non-linear features are important for spam filter;
- Parallel version gives here more complex features;
- **Interpretability**: Certain non-linear features appear quite reproducible by DBRM over 100 independent runs.

Summary prediction examples (DBRM)

Example3: Spam mail

complexity	G	R	G_PAR	R_PAR	LBRM
1	36.34	36.09	39.87	39.17	49.83
2	14.45	14.83	21.47	22.43	0.00
3	2.83	3.17	5.24	5.81	0.00
4	0.69	0.57	1.36	1.36	0.00
5	1.15	1.09	1.56	1.68	0.00
6	0.92	0.74	1.24	1.07	0.00
7	0.37	0.40	0.57	0.42	0.00
8	0.25	0.22	0.33	0.17	0.00
9	0.04	0.08	0.16	0.11	0.00
$\geq$10	0.15	0.11	0.11	0.18	0.00
Total	57.190	57.300	71.910	72.400	49.830

- Non-linear features are important for spam filter;
- Parallel version gives here more complex features;
- **Interpretability**: Certain non-linear features appear quite reproducible by DBRM over 100 independent runs.

Summary prediction examples (DBRM)

Example3: Spam mail

complexity	G	R	G_PAR	R_PAR	LBRM
1	36.34	36.09	39.87	39.17	49.83
2	14.45	14.83	21.47	22.43	0.00
3	2.83	3.17	5.24	5.81	0.00
4	0.69	0.57	1.36	1.36	0.00
5	1.15	1.09	1.56	1.68	0.00
6	0.92	0.74	1.24	1.07	0.00
7	0.37	0.40	0.57	0.42	0.00
8	0.25	0.22	0.33	0.17	0.00
9	0.04	0.08	0.16	0.11	0.00
$\geq$10	0.15	0.11	0.11	0.18	0.00
Total	57.190	57.300	71.910	72.400	49.830

- Non-linear features are important for spam filter;
- Parallel version gives here more complex features;
- **Interpretability**: Certain non-linear features appear quite reproducible by DBRM over 100 independent runs.

Examples on Inference

Deep Gaussian regression model

Dataset: Ten physical parameters of $n = 223$ exoplanets. We want to recover 2 basic physical laws.

Input variables include: *Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude*.

Example 4: Planetary mass

$$m_p \approx K_1 R_p^3 \times \rho_p.$$

Planetary mass m_p is proportional to cube of radius R_p times the density of the planet ρ_p

Example 5: Kepler's third law

The square of the orbital period P of a planet is directly proportional to the cube of the semi-major axis a of its orbit:

$$a \approx K_2 \left(P^2 M_h\right)^{\frac{1}{3}}.$$

Examples on Inference

Deep Gaussian regression model

Dataset: Ten physical parameters of $n = 223$ exoplanets. We want to recover 2 basic physical laws.
Input variables include: *Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude*.

Example 4: Planetary mass

$$m_p \approx K_1 R_p^3 \times \rho_p.$$

Planetary mass m_p is proportional to cube of radius R_p times the density of the planet ρ_p

Example 5: Kepler's third law

The square of the orbital period P of a planet is directly proportional to the cube of the semi-major axis a of its orbit:

$$a \approx K_2 \left(P^2 M_h\right)^{\frac{1}{3}}.$$

Examples on Inference

Deep Gaussian regression model

Dataset: Ten physical parameters of $n = 223$ exoplanets. We want to recover 2 basic physical laws.
Input variables include: *Mean anomaly, Inclination, Argument of perihelion, Longitude of the ascending node, Rms residual, Semi major axis, Eccentricity, Mean motion, Absolute magnitude*.

Example 4: Planetary mass

$$m_p \approx K_1 R_p^3 \times \rho_p.$$

Planetary mass m_p is proportional to cube of radius R_p times the density of the planet ρ_p

Example 5: Kepler's third law

The square of the orbital period P of a planet is directly proportional to the cube of the semi-major axis a of its orbit:

$$a \approx K_2 \left(P^2 M_h\right)^{\frac{1}{3}}.$$

Deep Bayesian Gaussian regression

$$Y_i|\mu_i,\sigma^2 \sim N(\mu_i,\sigma^2), i \in \{1,...,n\}; \tag{14}$$

$$\mu_i = \gamma_0\beta_0 + \sum_{j=1}^{q} \gamma_j\beta_j F_j(\boldsymbol{x}_i); \tag{15}$$

$$p(\gamma) \propto \mathrm{I}\left(|\gamma_{1:q}| \leq 15\right) \prod_{j=1}^{q} \exp(-2\log n\gamma_j c(F_j(x))); \tag{16}$$

$$p(\boldsymbol{\beta}|\gamma) = |J_n^{\gamma}(\boldsymbol{\beta})|^{\frac{1}{2}} \ ; \tag{17}$$

$$\pi(\sigma^2) = \sigma^{-2}; \tag{18}$$

$$\mathcal{G} = \{\text{sigmoid}(x), \cos(x), \tanh(x), \text{atan}(x), |x|^{\frac{1}{3}}\}. \tag{19}$$

Examples on Inference

Measures of success

- Example 4: only feature $R_p^3 \times \rho_p$ counted as true positive (TP);
- Example 5: $(P^2M_h)^{\frac{1}{3}}, (P^2T_h)^{\frac{1}{3}}, (P^2FE_h)^{\frac{1}{3}}$ counted as TP;
- Power, FDR and average number of false positives (FP) estimated by 100 runs of DBRM;
- Comparison of DBRM_G and DBRM_R;
- Different numbers of parallel runs.

Examples on Inference

Measures of success

- Example 4: only feature $R_p^3 \times \rho_p$ counted as true positive (TP);
- Example 5: $(P^2 M_h)^{\frac{1}{3}}, (P^2 T_h)^{\frac{1}{3}}, (P^2 FE_h)^{\frac{1}{3}}$ counted as TP;
- Power, FDR and average number of false positives (FP) estimated by 100 runs of DBRM;
- Comparison of DBRM_G and DBRM_R;
- Different numbers of parallel runs.

Examples on Inference

Measures of success

- Example 4: only feature $R_p^3 \times \rho_p$ counted as true positive (TP);
- Example 5: $(P^2 M_h)^{\frac{1}{3}}, (P^2 T_h)^{\frac{1}{3}}, (P^2 FE_h)^{\frac{1}{3}}$ counted as TP;
- Power, FDR and average number of false positives (FP) estimated by 100 runs of DBRM;
- Comparison of DBRM_G and DBRM_R;
- Different numbers of parallel runs.

Examples on Inference

Measures of success

- Example 4: only feature $R_p^3 \times \rho_p$ counted as true positive (TP);
- Example 5: $\left(P^2 M_h\right)^{\frac{1}{3}}, \left(P^2 T_h\right)^{\frac{1}{3}}, \left(P^2 FE_h\right)^{\frac{1}{3}}$ counted as TP;
- Power, FDR and average number of false positives (FP) estimated by 100 runs of DBRM;
- Comparison of DBRM_G and DBRM_R;
- Different numbers of parallel runs.

Examples on Inference

Results, Example 4

	DBRM_G_PAR			DBRM_R_PAR		
Threads	Power	FP	FDR	Power	FP	FDR
16	1.00	0.00	0.00	0.97	0.06	0.03
4	0.79	0.40	0.21	0.61	0.73	0.39
1	0.42	1.21	0.58	0.33	1.63	0.67

- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Examples on Inference

Results, Example 4

	DBRM_G_PAR			DBRM_R_PAR		
Threads	Power	FP	FDR	Power	FP	FDR
16	1.00	0.00	0.00	0.97	0.06	0.03
4	0.79	0.40	0.21	0.61	0.73	0.39
1	0.42	1.21	0.58	0.33	1.63	0.67

- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Examples on Inference

Results, Example 4

	DBRM_G_PAR			DBRM_R_PAR		
Threads	Power	FP	FDR	Power	FP	FDR
16	1.00	0.00	0.00	0.97	0.06	0.03
4	0.79	0.40	0.21	0.61	0.73	0.39
1	0.42	1.21	0.58	0.33	1.63	0.67

- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Examples on Inference

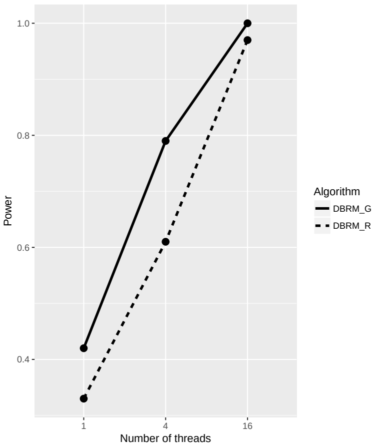
Results, Example 4

	DBRM_G_PAR			DBRM_R_PAR		
Threads	Power	FP	FDR	Power	FP	FDR
16	1.00	0.00	0.00	0.97	0.06	0.03
4	0.79	0.40	0.21	0.61	0.73	0.39
1	0.42	1.21	0.58	0.33	1.63	0.67

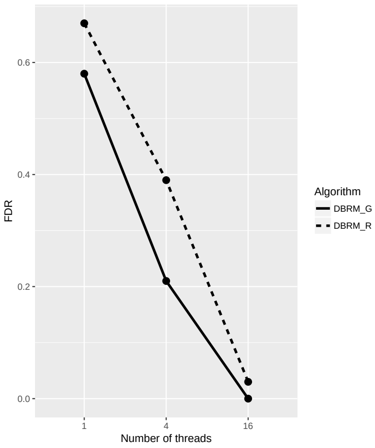
- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Results, Example 4

Power



FDR



Examples on Inference

Results, Example 5

	DBRM_G_PAR						DBRM_R_PAR					
Thr	F_1	F_2	F_3	Pow	FP	FDR	F_1	F_2	F_3	Pow	FP	FDR
64	81	71	1	1.00	0.02	0.01	78	75	2	0.99	0.03	0.01
16	34	41	32	0.84	0.46	0.18	31	38	18	0.79	0.68	0.25
1	6	5	3	0.14	0.65	0.86	6	4	2	0.12	1.81	0.88

- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Examples on Inference

Results, Example 5

	DBRM_G_PAR						DBRM_R_PAR					
Thr	F_1	F_2	F_3	Pow	FP	FDR	F_1	F_2	F_3	Pow	FP	FDR
64	81	71	1	1.00	0.02	0.01	78	75	2	0.99	0.03	0.01
16	34	41	32	0.84	0.46	0.18	31	38	18	0.79	0.68	0.25
1	6	5	3	0.14	0.65	0.86	6	4	2	0.12	1.81	0.88

- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Examples on Inference

Results, Example 5

	DBRM_G_PAR						DBRM_R_PAR					
Thr	F_1	F_2	F_3	Pow	FP	FDR	F_1	F_2	F_3	Pow	FP	FDR
64	81	71	1	1.00	0.02	0.01	78	75	2	0.99	0.03	0.01
16	34	41	32	0.84	0.46	0.18	31	38	18	0.79	0.68	0.25
1	6	5	3	0.14	0.65	0.86	6	4	2	0.12	1.81	0.88

- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Examples on Inference

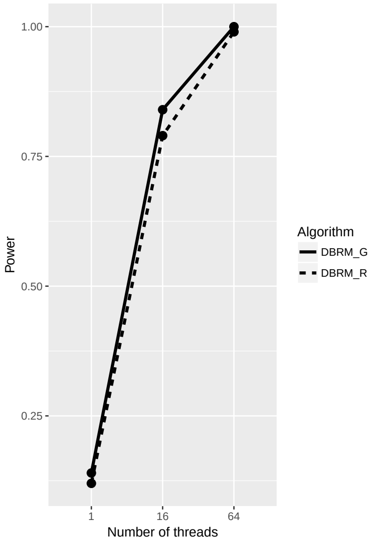
Results, Example 5

	DBRM_G_PAR						DBRM_R_PAR					
Thr	F_1	F_2	F_3	Pow	FP	FDR	F_1	F_2	F_3	Pow	FP	FDR
64	81	71	1	1.00	0.02	0.01	78	75	2	0.99	0.03	0.01
16	34	41	32	0.84	0.46	0.18	31	38	18	0.79	0.68	0.25
1	6	5	3	0.14	0.65	0.86	6	4	2	0.12	1.81	0.88

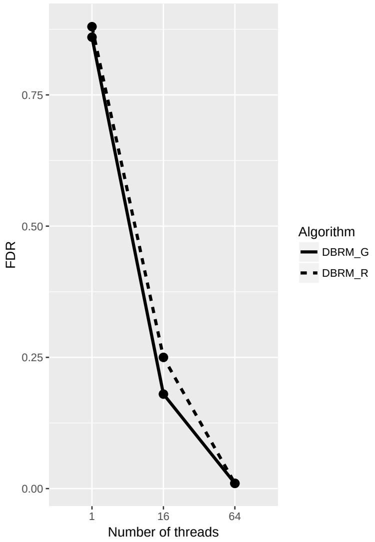
- Power increases with number of parallel threads;
- FP and FDR decrease with number of parallel threads;
- GMJMCMC slightly better than RGMJMCMC.

Results, Example 5

Power



FDR



Examples on Inference

Bayesian logic regression domain

For Example 6 we generated $N = 100$ datasets with $n = 1000$ observations and $p = 50$ binary covariates. The covariates were assumed to be independent and were simulated as $X_j \sim \text{Bernoulli}(0.5)$ for $j \in \{1, \ldots, 50\}$.

Continuous responses

Gaussian observations with error variance $\sigma^2 = 1$ and individual expectations specified as follows for the different scenarios:

Example 6:

$E(Y|X) = 1 + 1.5\, X_7 + 1.5\, X_8 + 6.6\, X_{18} \wedge X_{21} + 3.5\, X_2 \wedge X_9 + 9\, X_{12} \wedge X_{20} \wedge X_{37} + 7\, X_1 \wedge X_3 \wedge X_{27} + 7\, X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30} + 7\, X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$.

Examples on Inference

Bayesian logic regression domain

For Example 6 we generated $N = 100$ datasets with $n = 1000$ observations and $p = 50$ binary covariates. The covariates were assumed to be independent and were simulated as $X_j \sim$ Bernoulli(0.5) for $j \in \{1, \ldots, 50\}$.

Continuous responses

Gaussian observations with error variance $\sigma^2 = 1$ and individual expectations specified as follows for the different scenarios:

Example 6:

$$E(Y|X) = 1 + 1.5\, X_7 + 1.5\, X_8 + 6.6\, X_{18} \wedge X_{21} + 3.5\, X_2 \wedge X_9 + 9\, X_{12} \wedge X_{20} \wedge X_{37} + 7\, X_1 \wedge X_3 \wedge X_{27} + 7\, X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30} + 7\, X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}.$$

Examples on Inference

Bayesian logic regression domain

For Example 6 we generated $N = 100$ datasets with $n = 1000$ observations and $p = 50$ binary covariates. The covariates were assumed to be independent and were simulated as $X_j \sim \text{Bernoulli}(0.5)$ for $j \in \{1, \ldots, 50\}$.

Continuous responses

Gaussian observations with error variance $\sigma^2 = 1$ and individual expectations specified as follows for the different scenarios:

Example 6:

$$E(Y|X) = 1+1.5\,X_7+1.5\,X_8+6.6\,X_{18}\wedge X_{21}+3.5\,X_2\wedge X_9+9\,X_{12}\wedge X_{20}\wedge X_{37} +7\,X_1 \wedge X_3 \wedge X_{27} + 7\,X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30} + 7\,X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}.$$

Examples on Inference

Table: Examples 6. Inference. 32 Threads

	GMJ	RGMJ	GMJ(logic)
X_7	1.0000	1.0000	0.9900
X_8	1.0000	1.0000	1.0000
$X_2 \wedge X_9$	1.0000	0.9600	1.0000
$X_{18} \wedge X_{21}$	1.0000	1.0000	0.9600
$X_1 \wedge X_3 \wedge X_{27}$	1.0000	1.0000	1.0000
$X_{12} \wedge X_{20} \wedge X_{37}$	1.0000	1.0000	0.9900
$X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.9900	0.9200	0.9100
$X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$	0.9800	0.8900	0.3800
Overall Power	0.9963	0.9712	0.9038
FP	0.5100	1.1400	1.0900
FDP	0.0601	0.1279	0.1310

- Power is the best for DBRM with GMJMCMC;
- FP and FDR are the best for DBRM with GMJMCMC.

Examples on Inference

Table: Examples 6. Inference. 32 Threads

	GMJ	RGMJ	GMJ(logic)
X_7	1.0000	1.0000	0.9900
X_8	1.0000	1.0000	1.0000
$X_2 \wedge X_9$	1.0000	0.9600	1.0000
$X_{18} \wedge X_{21}$	1.0000	1.0000	0.9600
$X_1 \wedge X_3 \wedge X_{27}$	1.0000	1.0000	1.0000
$X_{12} \wedge X_{20} \wedge X_{37}$	1.0000	1.0000	0.9900
$X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.9900	0.9200	0.9100
$X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$	0.9800	0.8900	0.3800
Overall Power	0.9963	0.9712	0.9038
FP	0.5100	1.1400	1.0900
FDP	0.0601	0.1279	0.1310

- Power is the best for DBRM with GMJMCMC;
- FP and FDR are the best for DBRM with GMJMCMC.

Examples on Inference

Table: Examples 6. Inference. 32 Threads

	GMJ	RGMJ	GMJ(logic)
X_7	1.0000	1.0000	0.9900
X_8	1.0000	1.0000	1.0000
$X_2 \wedge X_9$	1.0000	0.9600	1.0000
$X_{18} \wedge X_{21}$	1.0000	1.0000	0.9600
$X_1 \wedge X_3 \wedge X_{27}$	1.0000	1.0000	1.0000
$X_{12} \wedge X_{20} \wedge X_{37}$	1.0000	1.0000	0.9900
$X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.9900	0.9200	0.9100
$X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$	0.9800	0.8900	0.3800
Overall Power	0.9963	0.9712	0.9038
FP	0.5100	1.1400	1.0900
FDP	0.0601	0.1279	0.1310

- Power is the best for DBRM with GMJMCMC;
- FP and FDR are the best for DBRM with GMJMCMC.

Concluding remarks

- We introduced the (R)(G)MJMCMC algorithm for various regresson contexts capable of:
  - Performing model configuration;
  - Estimating posterior model probabilities;
  - Bayesian model averaging and selection.
- *EMJMCMC* R-package is available:
  - `http://aliaksah.github.io/EMJMCMC2016/`;
  - Flexibility in the choice of methods for:
    - Marginal likelihoods;
    - Model selection criteria;
  - Extensive parallel computing is available;
  - Vectorized predictions with NA handling is incorporated.
- Results showed that (R)(G)MJMCMC:
  - Performs well in terms of the search speed and quality;
  - Provides nice predictive and inferential performance in the applications.

Literature

1. Hubin, A., Storvik G. (2018). **Mode jumping MCMC for Bayesian variable selection in GLMM.** *Journal of Computational Statistics and Data Analysis; 2018 November; 127:281-297*.
2. Hubin, A., Storvik G., Frommlet F. (2018). **A novel algorithmic approach to Bayesian Logic Regression**. *Submitted to **Bayesian analysis** for publication*.
3. Hubin, A., Storvik G., Frommlet F. (2018). **Deep Bayesian regression models.** *Submitted to **JASA** for publication*.
4. Hubin, A., Hagmann M., Bodenstorfer B., Gola A., Bogdan M., Frommlet F. (2018). **A comprehensive study of Bayesian approaches to Genome-Wide Association Studies.** *Work in progress*.
5. Hubin, A., Storvik G. (2016). **Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA).** *Technical report; arXiv:1611.01450*.