Fast Discovery of Pairwise Interactions in High Dimensions using Bayes

Tamara Broderick
Associate Professor
EECS, MIT

Raj Agrawal, Jonathan H. Huggins, Brian L. Trippe
Gene expression levels

Person 1
Person 2

Person N
Gene expression levels

Environmental factors

Person 1

Person 2

... 

Person N
Gene expression levels

Person 1

Person 2

Person N

Environmental factors

Blood pressure
Gene expression levels

Environmental factors

Blood pressure

Person 1

Person 2

Person N

- Which genes/factors are associated with a health issue?
• Which genes/factors are associated with a health issue?
• Want small subset of $p (> N)$ covariates
Gene expression levels

Environmental factors

Blood pressure

Person 1

Person 2

Person N

- Which genes/factors are associated with a health issue?
- Want small subset of $p > N$ covariates (cf. LASSO)
Which genes/factors are associated with a health issue?
Want small subset of $p (> N)$ covariates (cf. LASSO)
Additive model often not enough: need interactions
Gene expression levels

Environmental factors

Blood pressure

• Which genes/factors are associated with a health issue?
• Want small subset of $p (> N)$ covariates (cf. LASSO)
• Additive model often not enough: need interactions (now $p^2$ dims!)
Pairwise interactions in high dimensions

- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions (now $p^2$ dims!)
Pairwise interactions in high dimensions

- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions (now $p^2$ dims!)
- **We provide**: Fast, accurate (Bayes) method for interaction discovery
Pairwise interactions in high dimensions

- Which genes/factors are associated with a health issue?
- Want small subset of $p (> N)$ covariates (cf. LASSO)
- Additive model often not enough: need interactions (now $p^2$ dims!)
- **We provide**: Fast, accurate (Bayes) method for interaction discovery
  - Better scaling in $p$ & better accuracy than LASSO-based methods.
  - Orders of magnitude faster than naive Bayesian inference
Roadmap
Roadmap

- Setup: Discovering main and interaction effects
Roadmap

• Setup: Discovering main and interaction effects
• Our method
Roadmap

• Setup: Discovering main and interaction effects
• Our method
  • A Bayesian generative model
Roadmap

• Setup: Discovering main and interaction effects
• Our method
  • A Bayesian generative model
  • Fast inference
Roadmap

- Setup: Discovering main and interaction effects
- Our method
  - A Bayesian generative model
  - Fast inference
  - Fast reporting of results
Roadmap

• Setup: Discovering main and interaction effects
• Our method
  • A Bayesian generative model
  • Fast inference
  • Fast reporting of results
• Experiments on simulated and real data
Roadmap

• Setup: Discovering main and interaction effects
  • Our method
    • A Bayesian generative model
    • Fast inference
    • Fast reporting of results
  • Experiments on simulated and real data
Discovering main and interaction effects

Gene expression levels

Person 1

Person 2

Blood pressure
Discovering main and interaction effects

Gene expression levels

Person 1

$\begin{array}{c}
\text{x1} \\
\text{...}
\end{array}$

Person 2

$\begin{array}{c}
\text{x1} \\
\text{...}
\end{array}$

Blood pressure

$\begin{array}{c}
\text{...} \\
\text{xp}
\end{array}$
Discovering main and interaction effects

Gene expression levels

Person 1

Person 2

Blood pressure

\[ x_1 \quad \ldots \quad x_p \quad y \]
Discovering main and interaction effects

Gene expression levels

Person 1

$\begin{align*}
 & x_1 \quad \ldots \\
 & x^\top := [x_1, \ldots, x_p]
\end{align*}$

Person 2

Blood pressure

$y$
Discovering main and interaction effects

Gene expression levels

Person 1

Person 2

\[ x_{1} \ldots \]

\[ x^\top := [x_{1}, \ldots, x_{p}] \]

\[ y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]

Blood pressure
Discovering main and interaction effects

Gene expression levels

Person 1

Person 2

\[ x_\top := [1, x_1, \ldots, x_p] \]

\[ y^{(n)} = \theta \top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]
Discovering main and interaction effects

Gene expression levels

Person 1

$\begin{bmatrix} x_1 & \cdots \end{bmatrix}$

$y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}$, \hspace{1cm} \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

Blood pressure

Person 2

$\begin{bmatrix} \cdots & x_p \end{bmatrix}$

$y$
Discovering main and interaction effects

Gene expression levels

Person 1

$\Phi_2(x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2]$ 

$y^{(n)} = \theta^\top x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$
Discovering main and interaction effects

Gene expression levels

Blood pressure

\[
\Phi_2^T (x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2]
\]

\[
y^{(n)} = \theta^T x^{(n)} + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)
\]
Discovering main and interaction effects

Gene expression levels

Person 1

Person 2

\[ \Phi_2^\top(x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2] \]

\[ y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]
Discovering main and interaction effects

**Gene expression levels**

<table>
<thead>
<tr>
<th>Person 1</th>
<th>Person 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$x_1$</td>
</tr>
<tr>
<td>$\cdots$</td>
<td>$\cdots$</td>
</tr>
</tbody>
</table>

$\Phi_2^T(x) := [1, x_1, \ldots, x_p, x_1x_2, \ldots, x_{p-1}x_p, x_1^2, \ldots, x_p^2]$  

$y^{(n)} = \theta^T \Phi_2(x^{(n)}) + \epsilon^{(n)}$,  

$\epsilon^{(n)} \sim \mathcal{N}(0, \sigma^2)$
Discovering main and interaction effects

Gene expression levels

Person 1

Person 2

\[
\Phi_2^\top (x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2]
\]

\[
y^{(n)} = \theta^\top \Phi_2 (x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)
\]

- **Goal**: Parameter selection/estimation
Discovering main and interaction effects

Gene expression levels

Person 1

Person 2

Blood pressure

\[ \Phi_2^T(x) := [1, x_1, \ldots, x_p, x_1x_2, \ldots, x_{p-1}x_p, x_1^2, \ldots, x_p^2] \]

\[ y(n) = \theta^T \Phi_2(x(n)) + \epsilon(n), \quad \epsilon(n) \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]

- **Goal**: Parameter selection/estimation under assumptions:
Discovering main and interaction effects

Gene expression levels

Person 1

\[ x_1 \ldots \]

\[ \Phi_2^\top(x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2] \]

\[ y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]

• **Goal**: Parameter selection/estimation under assumptions:
  • **Sparsity**: most main effects are negligible (interpretable)
Discovering main and interaction effects

Gene expression levels

Person 1

Gene expression levels

Person 2

Blood pressure

\[ \Phi_2^\top(x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2] \]

\[ y^{(n)} = \theta^\top \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]

• **Goal**: Parameter selection/estimation under assumptions:
  • *Sparsity*: most main effects are negligible (interpretable)
  • *Strong hierarchy*: Interaction only if main effects are present

[Chipman 1996; Wu et al 2009; Bien et al 2013; Lim, Hastie 2015; Nakagawa et al 2016; Griffin, Brown 2017]
Discovering main and interaction effects

Gene expression levels

Person 1

\[ x_1 \quad \ldots \quad x_p \]

\[ \Phi_2^T(x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2] \]

\[ y^{(n)} = \theta^T \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]

**Goal**: Parameter selection/estimation under assumptions:

- **Sparsity**: most main effects are negligible (interpretable)
- **Strong hierarchy**: Interaction only if main effects are present
- \( p^2 \) covariates: large \( p \) \( \Rightarrow \) statistical & computational challenge

[Chipman 1996; Wu et al 2009; Bien et al 2013; Lim, Hastie 2015; Nakagawa et al 2016; Griffin, Brown 2017]
Discovering main and interaction effects

Gene expression levels

Person 1

\[ \Phi_2^T(x) := [1, x_1, \ldots, x_p, x_1 x_2, \ldots, x_{p-1} x_p, x_1^2, \ldots, x_p^2] \]

\[ y^{(n)} = \theta^T \Phi_2(x^{(n)}) + \epsilon^{(n)}, \quad \epsilon^{(n)} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \]

• **Goal**: Parameter selection/estimation under assumptions:
  • **Sparsity**: most main effects are negligible (interpretable)
  • **Strong hierarchy**: Interaction only if main effects are present

• \( p^2 \) covariates: large \( p \) $\rightarrow$ statistical & computational challenge

• **Our solution**: using structure in covariates + sparsity assumptions to reduce to a problem *linear* in \( p \)
Roadmap

• Setup: Discovering main and interaction effects
  • Our method
    • A Bayesian generative model
    • Fast inference
    • Fast reporting of results
  • Experiments on simulated and real data
Roadmap

• Setup: Discovering main and interaction effects
• Our method
  • A Bayesian generative model
  • Fast inference
  • Fast reporting of results
• Experiments on simulated and real data
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model

2. Compute posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model

2. Compute posterior

3. Report relevant summaries of the posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. Choose generative model

2. Compute posterior

3. Report relevant summaries of the posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: Sparse Kernel Interaction Model (SKIM) to encode sparsity and strong hierarchy

2. Compute posterior

3. Report relevant summaries of the posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. Compute posterior

3. Report relevant summaries of the posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. Compute posterior

3. Report relevant summaries of the posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration

3. Report relevant summaries of the posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration

3. Report relevant summaries of the posterior
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration

3. **Kernel Interaction Trick (KIT)**: Use kernel trick to report all non-negligible main and interaction effects in $O(p)$ time
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. **Kernel Interaction Sampler (KIS)**: Use kernel trick to run MCMC in $O(p)$ time per iteration

3. **Kernel Interaction Trick (KIT)**: Use kernel trick to report all non-negligible main and interaction effects in $O(p)$ time
Our approach

A Bayesian method: expert information, uncertainty quantification, regularization

1. New Bayesian generative model: **Sparse Kernel Interaction Model (SKIM)** to encode sparsity and strong hierarchy [Carvalho et al 2009; Piironen, Vehtari 2017; Chipman 1996, Griffin & Brown 2017]

2. **Kernel Interaction Sampler (KIS):** Use kernel trick to run MCMC in $O(p)$ time per iteration

3. **Kernel Interaction Trick (KIT):** Use kernel trick to report all non-negligible main and interaction effects in $O(p)$ time

Not just for SKIM
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample $\theta$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample $\theta$ ($p^2$ parameters)
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample $\theta$ ($p^2$ parameters)
  - Time cost: $O(p^2N)$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample $\theta$ ($p^2$ parameters)
  - Time cost: $O(p^2N)$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 1: sample $\theta$ ($p^2$ parameters)
  - Time cost: $O(p^2N)$

- Mixing (1000 iters Stan):
  - Option #1: all $\hat{R} > 1.05$
  - Our method: all $\hat{R} < 1.05$
Kernel Interaction Sampler vs. Naive MCMC
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert
    $$X^\top X$$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert
    $$X^\top X + \text{prior precision matrix}$$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert
    $$X^\top X$$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert
    \[
    \Phi_2(X)^\top \Phi_2(X)
    \]
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert
    $$\Phi_2(X)^\top \Phi_2(X)$$
    $X$: $N \times p$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
- Compute and invert
  \[
  \Phi_2(X)^\top \Phi_2(X)
  \]
  \[X: N \times p\]
  \[\Phi_2: N \times p^2\]
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert
    \[ \Phi_2(X)^\top \Phi_2(X) \]
    - $X$: $N \times p$
    - $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
- Compute and invert
  \[ \Phi_2(X)^T \Phi_2(X) \]

$X$: $N \times p$
$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
- Compute and invert $\Phi_2(X)^\top \Phi_2(X)$

$X$: $N \times p$
$\Phi_2$: $N \times p^2$

$N \quad x \quad p^2$
$p^2 \quad N \quad X$

=
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
- Compute and invert $\Phi_2(X)^\top \Phi_2(X)$
  - $X$: $N \times p$
  - $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

• MCMC option 2: use conditional conjugacy for $\theta$
  • Compute and invert
    $\Phi_2(X)^\top \Phi_2(X)$
    $X$: $N\times p$
    $\Phi_2$: $N\times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert $\Phi_2(X)^{\top} \Phi_2(X)$
    - $X$: $N \times p$
    - $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert $\Phi_2(X)^\top \Phi_2(X)$
    - $X$: $N \times p$
    - $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
- Compute and invert
  \[ \Phi_2(X)^\top \Phi_2(X) \]
  \[ X: N \times p \]
  \[ \Phi_2: N \times p^2 \]
- Naive time cost: $O(p^4N + p^6)$
Kernel Interaction Sampler vs. Naive MCMC

• MCMC option 2: use conditional conjugacy for $\theta$
  • Compute and invert
  $\Phi_2(X)^\top \Phi_2(X)$
  $X$: $N \times p$
  $\Phi_2$: $N \times p^2$

• Naive time cost: $O(p^4N + p^6)$
• Woodbury time cost: $O(p^2N^2 + N^3)$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
- Compute and invert
  $$\Phi_2(X)^\top \Phi_2(X)$$
  $X$: $N \times p$
  $\Phi_2$: $N \times p^2$

- Naive time cost: $O(p^4N + p^6)$
- Woodbury time cost: $O(p^2N^2 + N^3)$
Kernel Interaction Sampler vs. Naive MCMC

- MCMC option 2: use conditional conjugacy for $\theta$
  - Compute and invert
    \[ \Phi_2(X)^\top \Phi_2(X) \]
    \[ X: N \times p \]
    \[ \Phi_2: N \times p^2 \]
  - Naive time cost: $O(p^4N + p^6)$
  - Woodbury time cost: $O(p^2N^2 + N^3)$

- Naive MCMC:
  - $\mathbf{X}$: $N \times p$
  - $\Phi_2$: $N \times p^2$

**Graphs:**
- Runtime vs. Dimension (p)
- Memory vs. Dimension (p)
Kernel Interaction Sampler vs. Naive MCMC

- Compute and invert
  \[ \Phi_2(X)^\top \Phi_2(X) \]
  
  \( X: N \times p \)
  
  \( \Phi_2: N \times p^2 \)
Kernel Interaction Sampler vs. Naive MCMC

• Compute and invert

\[ \Phi_2(X)^\top \Phi_2(X) \]

\( X: N \times p \)
\( \Phi_2: N \times p^2 \)
Kernel Interaction Sampler vs. Naive MCMC

use conditional conjugacy for $\theta^T \Phi_2(X)$

- Compute and invert $\Phi_2(X)^T \Phi_2(X)$

$X$: $N \times p$
$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
- Compute and invert

\[ \Phi_2(X)^\top \Phi_2(X) \]

$X$: $N \times p$
$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
- Compute and invert
  \[ \Phi_2(X)^\top \Phi_2(X) \]
  
  - $X$: $N \times p$
  - $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  - Compute and invert
    $$\Phi_2(X) \Phi_2(X)^\top \Phi_2(X) \Phi_2(X)^\top$$

  $X$: $N \times p$
  $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  - Compute and invert
  \[
  \Phi_2(X) \Phi_2(X)^\top \Phi_2(X) \Phi_2(X)^\top
  \]
  - $X$: $N \times p$
  - $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

• Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  • Compute and invert

\[
\Phi_2(X) \Phi_2(X)^\top \Phi_2(X) \Phi_2(X)^\top
\]

$X$: $N \times p$

$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  - Compute and invert
    $$\Phi_2(X) \Phi_2(X)^T$$

$X$: $N \times p$
\(\Phi_2\): $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
- Compute and invert
  $$\Phi_2(X) \Phi_2(X)^T$$

$X$: $N \times p$
$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

• Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  • Compute and invert $\Phi_2(X)\Phi_2(X)^T$

$X$: $N \times p$
$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

• Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  
• Compute and invert
  
  $\Phi_2(X) \Phi_2(X)^T$

  $X$: $N \times p$
  
  $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

• Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  • Compute and invert
    $\Phi_2(X)\Phi_2(X)^T$
    $X$: $N \times p$
    $\Phi_2$: $N \times p^2$

N x N = N p^2 x p^2
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
- Compute and invert $\Phi_2(X)\Phi_2(X)^T$

$X$: $N \times p$
$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

• Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  • Compute and invert

\[
\Phi_2(X) \Phi_2(X)^T
\]

$X$: $N \times p$

$\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
- Compute and invert $\Phi_2(X)\Phi_2(X)^T$
  - $X$: $N \times p$
  - $\Phi_2$: $N \times p^2$
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
- Compute and invert $\Phi_2(X) \Phi_2(X)^T$
  - $X$: $N \times p$
  - $\Phi_2$: $N \times p^2$
- Kernel trick: $O(p)$ cost
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
- Compute and invert $\Phi_2(X)\Phi_2(X)^T$
  - $X$: $N \times p$
  - $\Phi_2$: $N \times p^2$
- Kernel trick: $O(p)$ cost
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  - Compute and invert $\Phi_2(X) \Phi_2(X)^T$
    - $X$: $N \times p$
    - $\Phi_2$: $N \times p^2$
  - Kernel trick: $O(p)$ cost
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for \( \theta^T \Phi_2(X) \)
  - Compute and invert \( \Phi_2(X) \Phi_2(X)^T \)
    - \( X: N \times p \)
    - \( \Phi_2: N \times p^2 \)
  - Kernel trick: \( O(p) \) cost
Kernel Interaction Sampler vs. Naive MCMC

- Our approach: use conditional conjugacy for $\theta^T \Phi_2(X)$
  - Compute and invert $\Phi_2(X)\Phi_2(X)^T$
    - $X$: $N \times p$
    - $\Phi_2$: $N \times p^2$
  - Kernel trick: $O(p)$ cost
  - Our time cost: $O(pN^2 + N^3)$
Reporting: Kernel Interaction Trick
Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
Reporting: Kernel Interaction Trick

• Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
• But our goal is to find main and interaction effects
• Step A: Report posterior of $\theta_{x_i}$ or $\theta_{x_i x_j}$ in $O(1)$ time
Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- But our goal is to find main and interaction effects
- Step A: Report posterior of $\theta_{x_i}$ or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0]$$

$i^{th}$ position
Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- *But* our goal is to find main and interaction effects
- Step A: Report posterior of $\theta_{x_i}$ or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0] \quad g(e_i) = \theta_{x_i} + \theta_{x_i^2}$$

$i^{th}$ position
Reporting: Kernel Interaction Trick

- Can access posterior of \( g = \theta^T \Phi_2 \) in \( O(p) \) time per iteration
- But our goal is to find main and interaction effects
- Step A: Report posterior of \( \theta_{x_i} \) or \( \theta_{x_i x_j} \) in \( O(1) \) time

\[
e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0] \quad \text{at } i^{\text{th}} \text{ position}
\]

\[
g(e_i) = \theta_{x_i} + \theta x_i^2
\]

\[
g(-e_i) = -\theta_{x_i} + \theta x_i^2
\]
Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- But our goal is to find main and interaction effects
- Step A: Report posterior of $\theta x_i$ or $\theta x_i x_j$ in $O(1)$ time

\[ e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0] \]

\[ g(e_i) = \theta x_i + \theta x_i^2 \]

\[ g(-e_i) = -\theta x_i + \theta x_i^2 \]

\[ \frac{g(e_i) - g(-e_i)}{2} = \theta x_i \]
Reporting: Kernel Interaction Trick

• Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration

• But our goal is to find main and interaction effects

• Step A: Report posterior of $\theta_{x_i}$ or $\theta_{x_i x_j}$ in $O(1)$ time

$$e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0]$$

$i^{th}$ position

$$g(e_i) = \theta_{x_i} + \theta_{x_i^2}$$

$$g(-e_i) = -\theta_{x_i} + \theta_{x_i^2}$$

$$\frac{g(e_i) - g(-e_i)}{2} = \theta_{x_i}$$

• Step B: Find $k \ll p$ sparse main effects: takes $O(p)$ time
Reporting: Kernel Interaction Trick

- Can access posterior of $g = \theta^T \Phi_2$ in $O(p)$ time per iteration
- But our goal is to find main and interaction effects
- Step A: Report posterior of $\theta_{x_i}$ or $\theta_{x_i x_j}$ in $O(1)$ time

\[
e_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0] \quad g(e_i) = \theta_{x_i} + \theta_{x_i^2}\]

\[
g(-e_i) = -\theta_{x_i} + \theta_{x_i^2}\]

\[
\frac{g(e_i) - g(-e_i)}{2} = \theta_{x_i}\]

- Step B: Find $k \ll p$ sparse main effects: takes $O(p)$ time
- Step C: Report just the $k^2$ strong-hierarchy interaction effects: takes $O(k^2)$ time
Roadmap

• Setup: Discovering main and interaction effects
• Our method
  • A Bayesian generative model
  • Fast inference
  • Fast reporting of results
• Experiments on simulated and real data
Roadmap

• Setup: Discovering main and interaction effects
• Our method
  • A Bayesian generative model
  • Fast inference
  • Fast reporting of results
• Experiments on simulated and real data
Timing vs. LASSO-based methods
Timing vs. LASSO-based methods

- LASSO (pairs, hierarchical): $O(p^2)$ per iteration
Timing vs. LASSO-based methods

- LASSO (pairs, hierarchical): $O(p^2)$ per iteration [Lim, Hastie 2015]
Timing vs. LASSO-based methods

- LASSO (pairs, hierarchical): $O(p^2)$ per iteration [Lim, Hastie 2015]
- Our method: $O(p)$ per iteration
Timing vs. LASSO-based methods

- LASSO (pairs, hierarchical): $O(p^2)$ per iteration [Lim, Hastie 2015]
- Our method: $O(p)$ per iteration
- Competitive empirically for moderate $p$:
Experiments: Simulated
Experiments: Simulated

• 36 different simulated data sets (so know true effects)
Experiments: Simulated

- 36 different simulated data sets (so know true effects)
- Up to $p = 500 \implies \approx 125,000$ total parameters
Experiments: Simulated, Selection

- 36 different simulated data sets (so know true effects)
  - Up to $p = 500 \Rightarrow \approx 125,000$ total parameters
Experiments: Simulated, Selection

- 36 different simulated data sets (so know true effects)
- Up to $p = 500 \Rightarrow \approx 125,000$ total parameters
- False discovery rate (FDR): proportion incorrect
Experiments: Simulated, Selection

- 36 different simulated data sets (so know true effects)
- Up to $p = 500 \rightarrow \approx 125,000$ total parameters
- False discovery rate (FDR): proportion incorrect

### Main effects

![Graph showing main effects]

- Graph title: Main effects
- X-axis: # of Correct Effects
- Y-axis: # of Incorrect Effects
- Legend:
  - FDR=0.95
  - FDR=0.91
  - FDR=0.83
  - FDR=0.67
  - FDR=0.5
  - FDR=0.2
  - Our Method
  - HLASSO
  - PLASSO

Graph shows the relationship between the number of correct effects and the number of incorrect effects for different FDR values and methods.
Experiments: Simulated, Selection

- 36 different simulated data sets (so know true effects)
- Up to \( p = 500 \) ➔ \( \approx 125,000 \) total parameters
- False discovery rate (FDR): proportion incorrect

![Graphs showing main effects and pairwise effects with different FDR values.](image-url)
Experiments: Real covariates
Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
  - Highly correlated: 20 of 105 capture 99% of variance
Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
  - Highly correlated: 20 of 105 capture 99% of variance
- Key: (# correct effects): (# of incorrect effects)
  - **Higher** green is better: **lower** red is better
Experiments: Real covariates

• Simulated effects: 5 main, 10 interaction
• Covariates: Residential Building Data Set
  • Highly correlated: 20 of 105 capture 99% of variance
• Key: (# correct effects): (# of incorrect effects)
  • **Higher** green is better: **lower** red is better

<table>
<thead>
<tr>
<th><strong>Method</strong></th>
<th><strong>#Main</strong></th>
<th><strong>#Pair</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>PLASSO</td>
<td>2 : 5</td>
<td>3 : 21</td>
</tr>
</tbody>
</table>

13
Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
  - Highly correlated: 20 of 105 capture 99% of variance
- Key: (# correct effects): (# of incorrect effects)
- Higher green is better: lower red is better

<table>
<thead>
<tr>
<th>METHOD</th>
<th>#MAIN</th>
<th>#PAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLASSO</td>
<td>2 : 5</td>
<td>3 : 21</td>
</tr>
<tr>
<td>HLASSO</td>
<td>3 : 19</td>
<td>3 : 18</td>
</tr>
</tbody>
</table>
Experiments: Real covariates

- Simulated effects: 5 main, 10 interaction
- Covariates: Residential Building Data Set
  - Highly correlated: 20 of 105 capture 99% of variance
- Key: (# correct effects): (# of incorrect effects)
- **Higher** green is better: **lower** red is better

<table>
<thead>
<tr>
<th>Method</th>
<th>#Main</th>
<th>#Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>3 : 0</td>
<td>3 : 0</td>
</tr>
<tr>
<td>PLASSO</td>
<td>2 : 5</td>
<td>3 : 21</td>
</tr>
<tr>
<td>HLASSO</td>
<td>3 : 19</td>
<td>3 : 18</td>
</tr>
</tbody>
</table>
Experiments: Real data
Experiments: Real data

- Covariates and response: Auto MPG
Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398, \ p = 6$ (real-valued), but...
Experiments: Real data

• Covariates and response: Auto MPG
• $N = 398$, $p = 6$ (real-valued), but…
• Augment $p$ with 200 fake (noise) covariates
  • 21,321 total parameters
Experiments: Real data

• Covariates and response: Auto MPG
• $N = 398, \ p = 6$ (real-valued), but…
• Augment $p$ with 200 fake (noise) covariates
  • 21,321 total parameters
• Key: (# original effects): (# of fake effects)
• No order to blue: lower red is better
**Experiments: Real data**

- Covariates and response: Auto MPG
- \( N = 398, \ p = 6 \) (real-valued), but…
- Augment \( p \) with 200 fake (noise) covariates
  - 21,321 total parameters
- Key: (# original effects): (# of fake effects)
- **No order** to blue: lower red is better

<table>
<thead>
<tr>
<th>Method</th>
<th>#Main</th>
<th>#Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLASSO</td>
<td>4 : 0</td>
<td>2 : 78</td>
</tr>
</tbody>
</table>
Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398, \ p = 6$ (real-valued), but…
- Augment $p$ with 200 fake (noise) covariates
  - 21,321 total parameters
- Key: (# original effects): (# of fake effects)
- **No order** to blue: *lower* red is better

<table>
<thead>
<tr>
<th>Method</th>
<th>#Main</th>
<th>#Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLASSO</td>
<td>4 : 0</td>
<td>2 : 78</td>
</tr>
<tr>
<td>HLASSO</td>
<td>6 : 46</td>
<td>4 : 38</td>
</tr>
</tbody>
</table>
Experiments: Real data

- Covariates and response: Auto MPG
- $N = 398$, $p = 6$ (real-valued), but…
- Augment $p$ with 200 fake (noise) covariates
  - 21,321 total parameters
- Key: (# original effects): (# of fake effects)
- **No order** to blue: lower red is better

<table>
<thead>
<tr>
<th>Method</th>
<th>#Main</th>
<th>#Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>3 : 0</td>
<td>1 : 0</td>
</tr>
<tr>
<td>PLASSO</td>
<td>4 : 0</td>
<td>2 : 78</td>
</tr>
<tr>
<td>HLASSO</td>
<td>6 : 46</td>
<td>4 : 38</td>
</tr>
</tbody>
</table>
Conclusions

**We provide**: fast, accurate detection of pairwise interactions

Conclusions

**We provide**: fast, accurate detection of pairwise interactions

*Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan*

Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions


Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

• In Pyro: http://pyro.ai/numpyro/sparse_regression.html
Conclusions

**We provide**: fast, accurate detection of pairwise (and higher-order) interactions

Up next:


Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

• In Pyro: [http://pyro.ai/numpyro/sparse_regression.html](http://pyro.ai/numpyro/sparse_regression.html)
Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:
• Response types (binary, count, etc) & nonlinearity

Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan
• In Pyro: http://pyro.ai/numpyro/sparse_regression.html
Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:
• Response types (binary, count, etc) & nonlinearity
• Improve scaling in $N$


Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

• In Pyro: http://pyro.ai/numpyro/sparse_regression.html
Conclusions

We provide: fast, accurate detection of pairwise (and higher-order) interactions

Up next:
• Response types (binary, count, etc) & nonlinearity
• Improve scaling in $N$


Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan

• In Pyro: http://pyro.ai/numpyro/sparse_regression.html


Conclusions

**We provide:** fast, accurate detection of pairwise (and higher-order) interactions

Up next:
- Response types (binary, count, etc) & nonlinearity
- Improve scaling in $N$
- Applications!

---


Thanks to Pyro contributors! Martin Jankowiak, Du Phan, Neeraj Pradhan


- For fixed budget, there is trade-off in sequencing more genomes and sequencing at greater depth
- We provide new method for prediction of # new variants and optimal allocation of more genomes vs. depth
  - Lowest error when using pilot TCGA dataset to predict the number of new variants to be observed in the follow-up MSK-impact dataset ($N=9593$) across 197 highly variable, cancerous genes
  - (Only) our prediction can handle when sequencing depth changes between pilot and follow-up study
  - (Only) our method optimizes under fixed budget