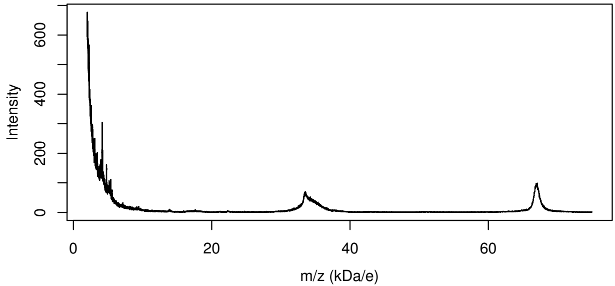
Bayesian Adpative Regression Kernels

Merlise Clyde

Department of Statistical Science
Duke University

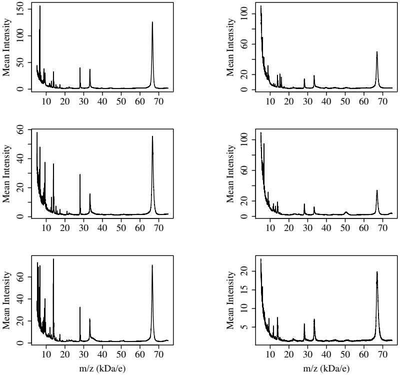
April 11, 2017

MALDI-TOF Mass Spectroscopy



Location of peaks/proteins in spectra in the presence of noise
Non-Gaussian Noise

Multiple Spectra



Learning "features" that are common versus those that separate groups of spectra but do not know a prior the number of ions

Problem Setting

Regression problem

$$\mathrm{E}[Y \mid \boldsymbol{x}] = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X}$$

with unknown function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$

Nonparametric Bayesian would place prior distribution on functions

Problem Setting

Regression problem

$$\mathrm{E}[Y \mid \boldsymbol{x}] = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X}$$

with unknown function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$

Nonparametric Bayesian would place prior distribution on functions

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \omega_j)$ Dictionary elements from a "generator function" g
  - cubic splines
    $$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$
  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)
    $$\psi(\boldsymbol{x}_i, \omega_j) = g(\Lambda_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \Lambda_j (\boldsymbol{x} - \chi_j)\right\}$$
  - translation and scaling wavelet families
  - Need not be symmetric!

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \omega_j)$ Dictionary elements from a "generator function" g
  - cubic splines

    $$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$

  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)

    $$\psi(\boldsymbol{x}_i, \omega_j) = g(\Lambda_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \Lambda_j (\boldsymbol{x} - \chi_j)\right\}$$

  - translation and scaling wavelet families
  - Need not be symmetric!

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \omega_j)$ Dictionary elements from a "generator function" g
  - cubic splines

$$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$

  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)

$$\psi(\boldsymbol{x}_i, \omega_j) = g(\Lambda_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \Lambda_j (\boldsymbol{x} - \chi_j)\right\}$$

  - translation and scaling wavelet families
  - Need not be symmetric!

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \boldsymbol{\omega}_j)$ Dictionary elements from a "generator function" g
  - cubic splines
    $$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$
  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)
    $$\psi(\boldsymbol{x}_i, \omega_j) = g(\Lambda_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \Lambda_j (\boldsymbol{x} - \chi_j)\right\}$$
  - translation and scaling wavelet families
  - Need not be symmetric!

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \boldsymbol{\omega}_j)$ Dictionary elements from a "generator function" g
  - cubic splines

$$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$

  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)

$$\psi(\boldsymbol{x}_i, \omega_j) = g(\Lambda_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \Lambda_j (\boldsymbol{x} - \chi_j)\right\}$$

  - translation and scaling wavelet families
  - Need not be symmetric!

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \boldsymbol{\omega}_j)$ Dictionary elements from a "generator function" g
  - cubic splines

    $$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$

  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)

    $$\psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j) = g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \boldsymbol{\Lambda}_j(\boldsymbol{x} - \chi_j)\right\}$$

  - translation and scaling wavelet families
  - Need not be symmetric!

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \boldsymbol{\omega}_j)$ Dictionary elements from a "generator function" g
  - cubic splines

    $$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$

  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)

    $$\psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j) = g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \boldsymbol{\Lambda}_j(\boldsymbol{x} - \chi_j)\right\}$$

  - translation and scaling wavelet families
  - Need not be symmetric!

Expansions

Write function as

$$f(\boldsymbol{x}_i) = \sum_{j=0}^{J} \psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j)\beta_j$$

in terms of an (over-complete) dictionary where

- $\{\beta_j\}$: unknown coefficients
- J: number of terms in expansion (finite or infinite)
- $\psi(\boldsymbol{x}, \boldsymbol{\omega}_j)$ Dictionary elements from a "generator function" g
  - cubic splines

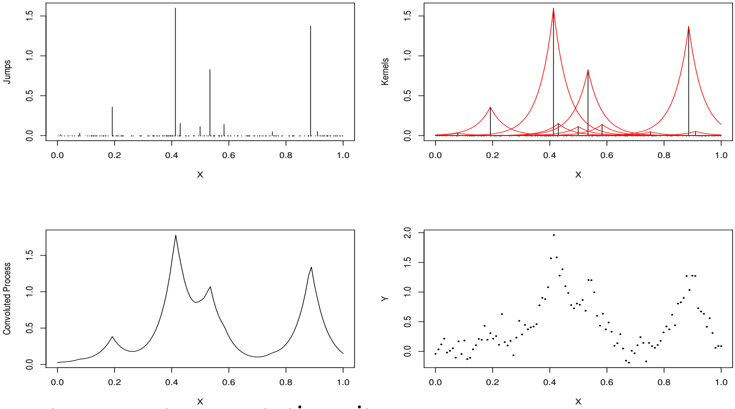
    $$\psi(x_i, \omega_j) = (x_i - \omega_j)_+^3$$

  - multivariate kernels (Gaussian, Cauchy, Exponential, e.g.)

    $$\psi(\boldsymbol{x}_i, \boldsymbol{\omega}_j) = g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \chi_j)) = \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \chi_j)^T \boldsymbol{\Lambda}_j(\boldsymbol{x} - \chi_j)\right\}$$

  - translation and scaling wavelet families
  - Need not be symmetric!

Kernel Convolution



Easy to generate non-stationarity processes

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$

Introduce Lévy Measure $\nu(\beta, \omega)$ (measure)

$\Rightarrow \; J \sim \mathrm{P}(\nu_+), \qquad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \omega) d\beta \, d\omega$

$\Rightarrow \; \beta_j, \omega_j \mid J \overset{iid}{\sim} \pi(\beta, \omega) \propto \nu(\beta, \omega).$

- Finite number of "big" coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R}\times\Omega} \min(1, |\beta|)\nu(\beta, \omega) d\beta \, d\omega < \infty \tag{1}$$

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$
Introduce Lévy Measure $\nu(\beta, \boldsymbol{\omega})$ (measure)

$\Rightarrow \; J \sim \mathsf{P}(\nu_+), \qquad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \omega) d\beta \, d\omega$

$\Rightarrow \; \beta_j, \omega_j \mid J \overset{iid}{\sim} \pi(\beta, \omega) \propto \nu(\beta, \omega).$

- Finite number of "big" coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R}\times\Omega} \min(1, |\beta|)\nu(\beta, \omega) d\beta \, d\omega < \infty \tag{1}$$

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$
Introduce Lévy Measure $\nu(\beta, \boldsymbol{\omega})$ (measure)

$$\Rightarrow \; J \sim \mathsf{P}(\nu_+), \qquad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega}$$

$$\Rightarrow \; \beta_j, \omega_j \mid J \overset{iid}{\sim} \pi(\beta, \omega) \propto \nu(\beta, \omega).$$

- Finite number of "big" coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R}\times\Omega} \min(1, |\beta|)\nu(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega} < \infty \tag{1}$$

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$
Introduce Lévy Measure $\nu(\beta, \boldsymbol{\omega})$ (measure)

$\Rightarrow$ $J \sim \mathsf{P}(\nu_+)$, $\quad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega}$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \mid J \overset{iid}{\sim} \pi(\beta, \boldsymbol{\omega}) \propto \nu(\beta, \boldsymbol{\omega})$.

- Finite number of “big” coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R} \times \Omega} \min(1, |\beta|) \nu(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega} < \infty \tag{1}$$

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$
Introduce Lévy Measure $\nu(\beta, \boldsymbol{\omega})$ (measure)

$\Rightarrow$ $J \sim \mathsf{P}(\nu_+)$, $\quad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega}$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \mid J \overset{iid}{\sim} \pi(\beta, \boldsymbol{\omega}) \propto \nu(\beta, \boldsymbol{\omega})$.

- Finite number of “big” coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R}\times\Omega} \min(1, |\beta|)\nu(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega} < \infty \tag{1}$$

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$
Introduce Lévy Measure $\nu(\beta, \boldsymbol{\omega})$ (measure)

$\Rightarrow$ $J \sim \mathsf{P}(\nu_+)$, $\qquad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega}$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \mid J \overset{iid}{\sim} \pi(\beta, \boldsymbol{\omega}) \propto \nu(\beta, \boldsymbol{\omega})$.

- Finite number of "big" coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R}\times\Omega} \min(1, |\beta|)\nu(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega} < \infty \qquad (1)$$

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$
Introduce Lévy Measure $\nu(\beta, \boldsymbol{\omega})$ (measure)

$\Rightarrow$ $J \sim \mathsf{P}(\nu_+)$, $\qquad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega}$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \mid J \overset{iid}{\sim} \pi(\beta, \boldsymbol{\omega}) \propto \nu(\beta, \boldsymbol{\omega})$.

- Finite number of “big” coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R} \times \Omega} \min(1, |\beta|) \nu(\beta, \boldsymbol{\omega}) d\beta \, d\boldsymbol{\omega} < \infty \tag{1}$$

Lévy- Khinchine Poisson Representation

Goal: $f(x) = \sum_{j<J} \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j = \sum_{j<J} g(\boldsymbol{\Lambda}_j(\boldsymbol{x} - \boldsymbol{\chi}_j))\beta_j$
Introduce Lévy Measure $\nu(\beta, \boldsymbol{\omega})$ (measure)

$$\Rightarrow\ J \sim \mathsf{P}(\nu_+), \qquad \nu_+ \equiv \nu(\mathbb{R} \times \boldsymbol{\Omega}) = \iint v(\beta, \boldsymbol{\omega}) d\beta\, d\boldsymbol{\omega}$$

$$\Rightarrow\ \beta_j, \boldsymbol{\omega}_j \mid J \overset{iid}{\sim} \pi(\beta, \boldsymbol{\omega}) \propto \nu(\beta, \boldsymbol{\omega}).$$

- Finite number of "big" coefficients $|\beta_j|$
- Possibly infinite number of $\beta \in [-\epsilon, \epsilon]$
- Coefficients $|\beta_j|$ are absolutely summable

Sufficient condition for bounded g:

$$\int_{\mathbb{R}\times\Omega} \min(1, |\beta|)\nu(\beta, \boldsymbol{\omega}) d\beta\, d\boldsymbol{\omega} < \infty \tag{1}$$

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha |\beta|^{-(\alpha+1)}\, \gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$

For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$

Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

- Finite number of support points $\boldsymbol{\omega}$ with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega})\mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathsf{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \overset{iid}{\sim} \pi(d\beta, d\boldsymbol{\omega}) \equiv \nu_\epsilon(d\beta, d\boldsymbol{\omega})/\nu_\epsilon^+$ (β_j distributed as Pareto)

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha |\beta|^{-(\alpha+1)}\ \gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$
For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$
Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

- Finite number of support points ω with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega})\mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathsf{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \overset{iid}{\sim} \pi(d\beta, d\boldsymbol{\omega}) \equiv \nu_\epsilon(d\beta, d\boldsymbol{\omega})/\nu_\epsilon^+$ (β_j distributed as Pareto)

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha|\beta|^{-(\alpha+1)}\,\gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$
For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$
Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

- Finite number of support points $\boldsymbol{\omega}$ with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega})\mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathrm{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \overset{iid}{\sim} \pi(d\beta, d\boldsymbol{\omega}) \equiv \nu_\epsilon(d\beta, d\boldsymbol{\omega})/\nu_\epsilon^+$ (β_j distributed as Pareto)

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha |\beta|^{-(\alpha+1)}\ \gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$
For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$
Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

- Finite number of support points $\boldsymbol{\omega}$ with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega})\mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathrm{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \overset{iid}{\sim} \pi(d\beta, d\boldsymbol{\omega}) \equiv \nu_\epsilon(d\beta, d\boldsymbol{\omega})/\nu_\epsilon^+$ (β_j distributed as Pareto)

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha |\beta|^{-(\alpha+1)} \, \gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$

For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$

Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

- Finite number of support points $\boldsymbol{\omega}$ with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega})\mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathsf{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \overset{iid}{\sim} \pi(d\beta, d\boldsymbol{\omega}) \equiv \nu_\epsilon(d\beta, d\boldsymbol{\omega})/\nu_\epsilon^+$ (β_j distributed as Pareto)

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha |\beta|^{-(\alpha+1)} \, \gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$
For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$
Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

- Finite number of support points $\boldsymbol{\omega}$ with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega})\mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathrm{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \overset{iid}{\sim} \pi(d\beta, d\boldsymbol{\omega}) \equiv \nu_\epsilon(d\beta, d\boldsymbol{\omega})/\nu_\epsilon^+$ (β_j distributed as Pareto)

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha |\beta|^{-(\alpha+1)}\,\gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$
For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$
Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

- Finite number of support points $\boldsymbol{\omega}$ with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega})\mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathsf{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \omega_j \overset{iid}{\sim} \pi(d\beta, d\omega) \equiv \nu_\epsilon(d\beta, d\omega)/\nu_\epsilon^+$ (β_j distributed as Pareto)

α-Stable Lévy Measures

Lévy measure: $\nu(d\beta, d\boldsymbol{\omega}) = c_\alpha |\beta|^{-(\alpha+1)}\ \gamma(d\boldsymbol{\omega}) \qquad 0 < \alpha < 2$
For α- Stable $\nu^+(\mathbb{R}, \boldsymbol{\Omega}) = \infty$
Fine in theory, but not in practice for MCMC!

Truncate measure to obtain a finite expansion:

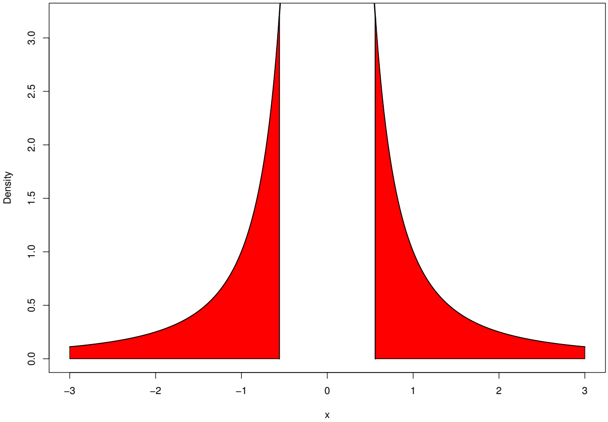
- Finite number of support points $\boldsymbol{\omega}$ with β in $[-\epsilon, \epsilon]^c$
- Fix ϵ (for given prior approximation error)
- Use approximate Lévy measure $\nu_\epsilon(\beta, \boldsymbol{\omega}) \equiv \nu(\beta, \boldsymbol{\omega}) \mathbf{1}(|\beta| > \epsilon)$

$\Rightarrow$ $J \sim \mathsf{P}(\nu_\epsilon^+)$ where $\nu_\epsilon^+ = \nu([-\epsilon, \epsilon]^c, \boldsymbol{\Omega})$

$\Rightarrow$ $\beta_j, \boldsymbol{\omega}_j \overset{iid}{\sim} \pi(d\beta, d\boldsymbol{\omega}) \equiv \nu_\epsilon(d\beta, d\boldsymbol{\omega}) / \nu_\epsilon^+$ (β_j distributed as Pareto)

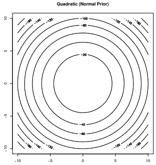
Truncated Cauchy Process

Restriction $|\beta| > \epsilon$

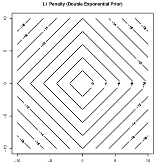


Contours of Log Prior (in $\mathbb{R}^2$) – Penalties

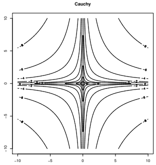
Normal



DE



Cauchy



Penalized Likelihood:

$$-\frac{1}{2\sigma^2}\sum_i (Y_i - f(\boldsymbol{x}_i))^2 - (\alpha+1)\sum_j \log(|\beta_j|) - \nu_\epsilon^+ \ldots$$

Computation - Reversible Jump MCMC

- Birth: generate coefficients β_j near ϵ in absolute value and generate kernel parameters $\boldsymbol{\omega}_j$ given increase in $J \to J+1$
- Death: $\beta_j = 0$ drop dictionary element when J decrements by 1.
- Update: Random-Walk update, but also leads to deaths with coefficients that wander out of bounds and cross ϵ boundary!
- Merge-Split: allow "neighboring points" to merge into one dictionary element (an alternative death) or a split into new dictionary elements

Advantage over fixed dimensional over-complete methods (frames)

Computation - Reversible Jump MCMC

- Birth: generate coefficients β_j near ϵ in absolute value and generate kernel parameters $\boldsymbol{\omega}_j$ given increase in $J \rightarrow J+1$
- Death: $\beta_j = 0$ drop dictionary element when J decrements by 1.
- Update: Random-Walk update, but also leads to deaths with coefficients that wander out of bounds and cross ϵ boundary!
- Merge-Split: allow "neighboring points" to merge into one dictionary element (an alternative death) or a split into new dictionary elements

Advantage over fixed dimensional over-complete methods (frames)

Computation - Reversible Jump MCMC

- Birth: generate coefficients β_j near ϵ in absolute value and generate kernel parameters $\boldsymbol{\omega}_j$ given increase in $J \to J + 1$
- Death: $\beta_j = 0$ drop dictionary element when J decrements by 1.
- Update: Random-Walk update, but also leads to deaths with coefficients that wander out of bounds and cross ϵ boundary!
- Merge-Split: allow "neighboring points" to merge into one dictionary element (an alternative death) or a split into new dictionary elements

Advantage over fixed dimensional over-complete methods (frames)

Computation - Reversible Jump MCMC

- Birth: generate coefficients β_j near ϵ in absolute value and generate kernel parameters $\boldsymbol{\omega}_j$ given increase in $J \to J+1$
- Death: $\beta_j = 0$ drop dictionary element when J decrements by 1.
- Update: Random-Walk update, but also leads to deaths with coefficients that wander out of bounds and cross ϵ boundary!
- Merge-Split: allow "neighboring points" to merge into one dictionary element (an alternative death) or a split into new dictionary elements

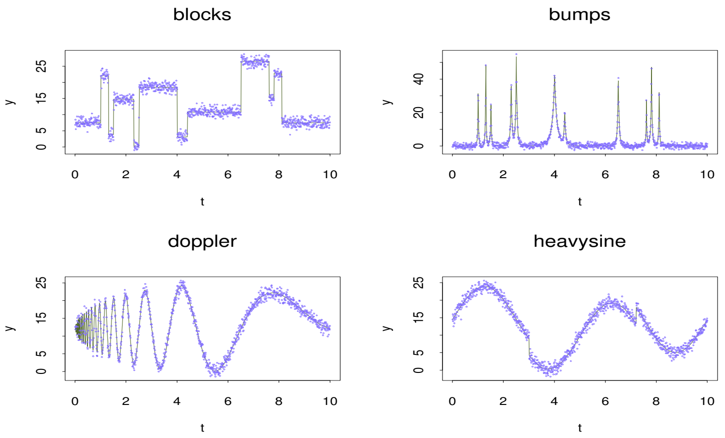
Advantage over fixed dimensional over-complete methods (frames)

Computation - Reversible Jump MCMC

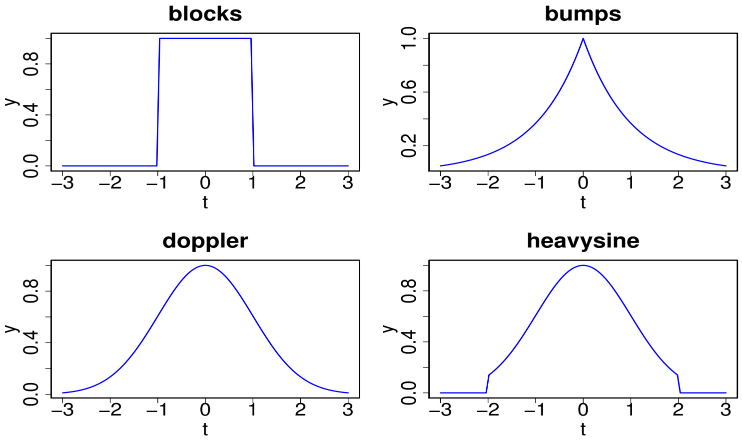
- Birth: generate coefficients β_j near ϵ in absolute value and generate kernel parameters $\boldsymbol{\omega}_j$ given increase in $J \rightarrow J+1$
- Death: $\beta_j = 0$ drop dictionary element when J decrements by 1.
- Update: Random-Walk update, but also leads to deaths with coefficients that wander out of bounds and cross ϵ boundary!
- Merge-Split: allow "neighboring points" to merge into one dictionary element (an alternative death) or a split into new dictionary elements

Advantage over fixed dimensional over-complete methods (frames)

Wavelet Test Functions (SNR = 7)



Kernel Functions



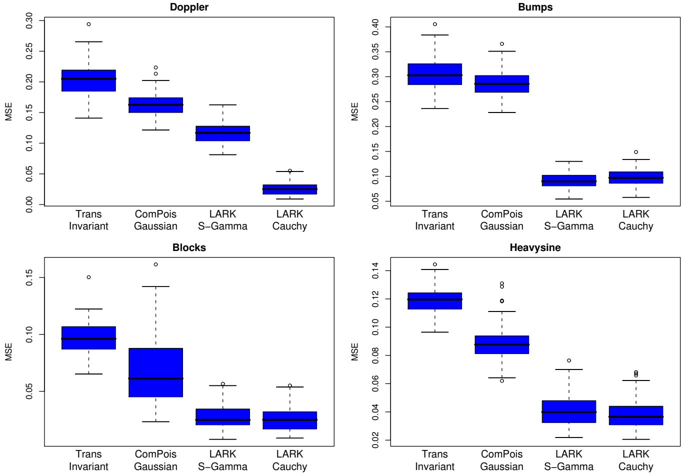
Comparisons of OCD Methods

- Translational Invariant Wavelets – Laplace Priors (Johnstone & Silverman 2005)
- Continuous Wavelet Dictionary – Compound Poisson with Gaussian Priors (Chu, Clyde, Liang 2007)
- LARK Symmetric Gamma
- LARK Cauchy

Range of Over-complete Dictionaries and Priors

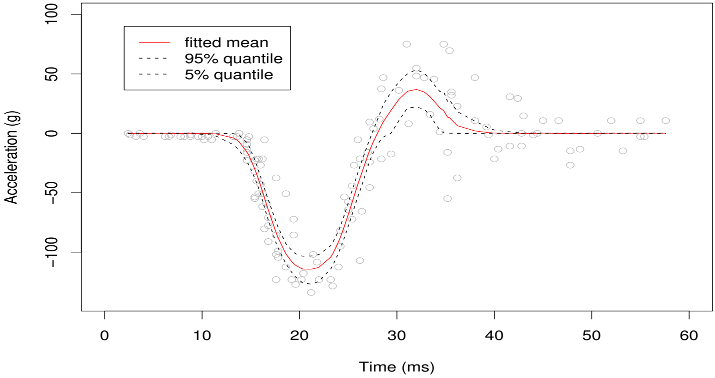
Comparison of Mean Square Error w/ OCDs

100 realizations of each function



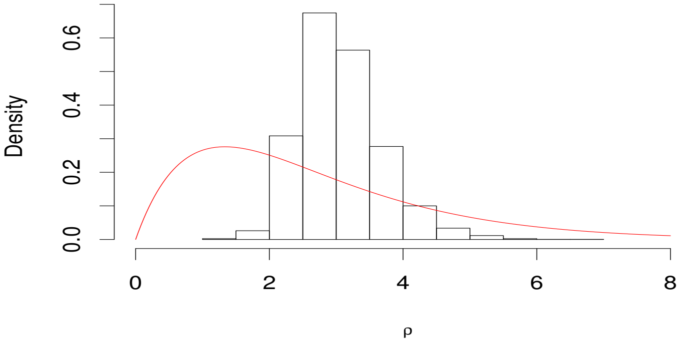
Motorcycle Crash Data: A Real Example

On average, only $\mathrm{E}[J \mid Y] \approx 4$ jumps are needed for fit:



Form of Kernel

$$k(t_i; \tau_j, \lambda_j) = e^{-\lambda_j |t_i - \tau_j|^\rho}$$



Higher Dimensional $\mathcal{X}$

RJ-MCMC is too slow in higher dimensional space to allow

- χ to be completely arbitrary; restrict support to observed $\{\boldsymbol{x}_i\}$
- use diagonal Λ

Kernels take form:

$$\begin{aligned}
\psi(\boldsymbol{x}, \boldsymbol{\omega}_j) &= \prod_d \exp\{-\frac{1}{2}\lambda_d(x_d - \chi_d)^2\} \\
f(\boldsymbol{x}) &= \sum_j \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j
\end{aligned}$$

Higher Dimensional $\mathcal{X}$

RJ-MCMC is too slow in higher dimensional space to allow

- χ to be completely arbitrary; restrict support to observed $\{\boldsymbol{x}_i\}$
- use diagonal Λ

Kernels take form:

$$\begin{aligned}
\psi(\boldsymbol{x}, \boldsymbol{\omega}_j) &= \prod_d \exp\{-\frac{1}{2}\lambda_d(x_d - \chi_d)^2\} \\
f(\boldsymbol{x}) &= \sum_j \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j
\end{aligned}$$

Higher Dimensional $\mathcal{X}$

RJ-MCMC is too slow in higher dimensional space to allow

- $\boldsymbol{\chi}$ to be completely arbitrary; restrict support to observed $\{\boldsymbol{x}_i\}$
- use diagonal $\boldsymbol{\Lambda}$

Kernels take form:

$$
\begin{aligned}
\psi(\boldsymbol{x}, \boldsymbol{\omega}_j) &= \prod_d \exp\{-\frac{1}{2}\lambda_d (x_d - \chi_d)^2\} \\
f(\boldsymbol{x}) &= \sum_j \psi(\boldsymbol{x}, \boldsymbol{\omega}_j)\beta_j
\end{aligned}
$$

Approximate Lévy Prior II

Continuous Approximation Student $t(\alpha, 0, \epsilon)$ approximation:

$$\nu_\epsilon(d\beta, d\boldsymbol{\omega}) = c_\alpha(\beta^2 + \alpha\epsilon^2)^{-(\alpha+1)/2} d\beta \ \gamma(d\boldsymbol{\omega})$$

Based on the following hierarchical prior

$$\begin{aligned} \beta_j \mid \phi_j &\overset{ind}{\sim} \mathsf{N}(0, \varphi_j^{-1}) \\ \phi_j &\overset{ind}{\sim} \mathsf{G}\left(\frac{\alpha}{2}, \frac{\alpha\epsilon^2}{2}\right) \\ J &\sim \mathsf{P}(\nu_\epsilon^+) \end{aligned}$$

where $\nu_\epsilon^+ = \nu_\epsilon(\mathbb{R}, \boldsymbol{\Omega}) = \frac{\alpha^{1-\alpha/2}\Gamma(\alpha)\Gamma(\alpha/2)}{\epsilon^\alpha \pi^{1/2}\Gamma(\frac{\alpha+1}{2})} \sin(\frac{\pi\alpha}{2})\gamma(\boldsymbol{\Omega})$

Key: need to have variance of coefficients decrease as J increases

Approximate Lévy Prior II

Continuous Approximation Student $t(\alpha, 0, \epsilon)$ approximation:

$$\nu_\epsilon(d\beta, d\boldsymbol{\omega}) = c_\alpha(\beta^2 + \alpha\epsilon^2)^{-(\alpha+1)/2} d\beta \; \gamma(d\boldsymbol{\omega})$$

Based on the following hierarchical prior

$$\begin{aligned} \beta_j \mid \phi_j &\overset{ind}{\sim} \mathsf{N}(0, \varphi_j^{-1}) \\ \phi_j &\overset{ind}{\sim} \mathsf{G}\left(\frac{\alpha}{2}, \frac{\alpha\epsilon^2}{2}\right) \\ J &\sim \mathsf{P}(\nu_\epsilon^+) \end{aligned}$$

where $\nu_\epsilon^+ = \nu_\epsilon(\mathbb{R}, \boldsymbol{\Omega}) = \frac{\alpha^{1-\alpha/2}\Gamma(\alpha)\Gamma(\alpha/2)}{\epsilon^\alpha \pi^{1/2}\Gamma(\frac{\alpha+1}{2})} \sin(\frac{\pi\alpha}{2})\gamma(\boldsymbol{\Omega})$

Key: need to have variance of coefficients decrease as J increases

Limiting Case

$$\begin{aligned} \beta_j \mid \varphi_j &\overset{ind}{\sim} \mathrm{N}(0, 1/\varphi_j) \\ \varphi_j &\overset{iid}{\sim} \mathsf{G}(\alpha/2, \text{"}0\text{"}) \end{aligned}$$

Notes:

- Require $0 < \alpha < 2$ Additional restrictions on ω
- Cauchy process corresponds to $\alpha = 1$
- Tipping's "Relevance Vector Machine" corresponds to $\alpha = 0$ (improper posterior!)
- Provides an extension of **Generalized Ridge Priors to infinite dimensional**
- **Infinite dimensional analog of Cauchy priors**

Limiting Case

$$\beta_j \mid \varphi_j \overset{ind}{\sim} \mathrm{N}(0, 1/\varphi_j)$$
$$\varphi_j \overset{iid}{\sim} \mathsf{G}(\alpha/2, \text{"}0\text{"})$$

Notes:

- Require $0 < \alpha < 2$ Additional restrictions on ω
- Cauchy process corresponds to $\alpha = 1$
- Tipping's "Relevance Vector Machine" corresponds to $\alpha = 0$ (improper posterior!)
- Provides an extension of **Generalized Ridge Priors to infinite dimensional**
- **Infinite dimensional analog of Cauchy priors**

Limiting Case

$$\beta_j \mid \varphi_j \overset{ind}{\sim} \mathrm{N}(0, 1/\varphi_j)$$
$$\varphi_j \overset{iid}{\sim} \mathrm{G}(\alpha/2, \text{"}0\text{"})$$

Notes:

- Require $0 < \alpha < 2$ Additional restrictions on ω
- Cauchy process corresponds to $\alpha = 1$
- Tipping's "Relevance Vector Machine" corresponds to $\alpha = 0$ (improper posterior!)
- Provides an extension of **Generalized Ridge Priors to infinite dimensional**
- **Infinite dimensional analog of Cauchy priors**

Limiting Case

$$\begin{aligned} \beta_j \mid \varphi_j &\overset{ind}{\sim} \mathrm{N}(0, 1/\varphi_j) \\ \varphi_j &\overset{iid}{\sim} \mathsf{G}(\alpha/2, "0") \end{aligned}$$

Notes:

- Require $0 < \alpha < 2$ Additional restrictions on ω
- Cauchy process corresponds to $\alpha = 1$
- Tipping's "Relevance Vector Machine" corresponds to $\alpha = 0$ (improper posterior!)
- Provides an extension of **Generalized Ridge Priors to infinite dimensional**
- **Infinite dimensional analog of Cauchy priors**

Limiting Case

$$
\begin{aligned}
\beta_j \mid \varphi_j &\stackrel{ind}{\sim} \mathrm{N}(0, 1/\varphi_j) \\
\varphi_j &\stackrel{iid}{\sim} \mathsf{G}(\alpha/2, \text{"}0\text{"})
\end{aligned}
$$

Notes:

- Require $0 < \alpha < 2$ Additional restrictions on ω
- Cauchy process corresponds to $\alpha = 1$
- Tipping's "Relevance Vector Machine" corresponds to $\alpha = 0$ (improper posterior!)
- Provides an extension of **Generalized Ridge Priors to infinite dimensional**
- **Infinite dimensional analog of Cauchy priors**

Limiting Case

$$\beta_j \mid \varphi_j \overset{ind}{\sim} \mathrm{N}(0, 1/\varphi_j)$$
$$\varphi_j \overset{iid}{\sim} \mathsf{G}(\alpha/2, \text{"}0\text{"})$$

Notes:

- Require $0 < \alpha < 2$ Additional restrictions on ω
- Cauchy process corresponds to $\alpha = 1$
- Tipping's "Relevance Vector Machine" corresponds to $\alpha = 0$ (improper posterior!)
- Provides an extension of **Generalized Ridge Priors to infinite dimensional**
- **Infinite dimensional analog of Cauchy priors**

Further Simplification in Case with $\alpha = 1$

- Poisson number of points $J_\epsilon \sim \mathsf{P}\big(\nu_\epsilon^+(\alpha,\gamma)\big)$ with $\nu_\epsilon^+(\alpha,\gamma) = \frac{\gamma\alpha^{1-\alpha/2}}{2^{1-\alpha}\epsilon^\alpha}\frac{\Gamma(\alpha/2)}{\Gamma(1-\alpha/2)}$
- Given J, $[n_1 : n_n] \sim MN(J, 1/(n+1))$ points supported at each kernel located at $\boldsymbol{x}_j$

The regression mean function can be rewritten as

$$f(\boldsymbol{x}) = \sum_{i=0}^{n} \tilde{\beta}_i \psi(\boldsymbol{x}, \boldsymbol{\omega}_i), \quad \tilde{\beta}_i = \sum_{\{j \,|\, \chi_j = \boldsymbol{x}_i\}} \beta_j.$$

In particular, if $\alpha = 1$, not only the Cauchy process is infinitely divisible, the approximated Cauchy prior distributions on the regression coefficients are also infinitely divisible:

$$\tilde{\beta}_i \overset{ind}{\sim} \mathsf{N}(0, n_i^2 \tilde{\varphi}_i^{-1}), \qquad \tilde{\varphi}_i \overset{iid}{\sim} \mathsf{G}(1/2, \epsilon^2/2)$$

At most n non-zero coefficients! (JAGS is possible now?)

Further Simplification in Case with $\alpha = 1$

- Poisson number of points $J_\epsilon \sim \mathsf{P}\big(\nu_\epsilon^+(\alpha,\gamma)\big)$ with $\nu_\epsilon^+(\alpha,\gamma) = \frac{\gamma\alpha^{1-\alpha/2}}{2^{1-\alpha}\epsilon^\alpha}\frac{\Gamma(\alpha/2)}{\Gamma(1-\alpha/2)}$
- Given J, $[n_1 : n_n] \sim MN(J, 1/(n+1))$ points supported at each kernel located at $\boldsymbol{x}_j$

The regression mean function can be rewritten as

$$f(\boldsymbol{x}) = \sum_{i=0}^{n} \tilde{\beta}_i \psi(\boldsymbol{x}, \boldsymbol{\omega}_i), \quad \tilde{\beta}_i = \sum_{\{j \,|\, \chi_j = \boldsymbol{x}_i\}} \beta_j.$$

In particular, if $\alpha = 1$, not only the Cauchy process is infinitely divisible, the approximated Cauchy prior distributions on the regression coefficients are also infinitely divisible:

$$\tilde{\beta}_i \overset{ind}{\sim} \mathsf{N}(0, n_i^2 \tilde{\varphi}_i^{-1}), \qquad \tilde{\varphi}_i \overset{iid}{\sim} \mathsf{G}(1/2, \epsilon^2/2)$$

At most n non-zero coefficients! (JAGS is possible now?)

Further Simplification in Case with $\alpha = 1$

- Poisson number of points $J_\epsilon \sim \mathsf{P}\big(\nu_\epsilon^+(\alpha,\gamma)\big)$ with $\nu_\epsilon^+(\alpha,\gamma) = \frac{\gamma\alpha^{1-\alpha/2}}{2^{1-\alpha}\epsilon^\alpha}\frac{\Gamma(\alpha/2)}{\Gamma(1-\alpha/2)}$
- Given J, $[n_1 : n_n] \sim MN(J, 1/(n+1))$ points supported at each kernel located at $\boldsymbol{x}_j$

The regression mean function can be rewritten as

$$f(\boldsymbol{x}) = \sum_{i=0}^{n} \tilde{\beta}_i \psi(\boldsymbol{x}, \boldsymbol{\omega}_i), \quad \tilde{\beta}_i = \sum_{\{j \,|\, \chi_j = \boldsymbol{x}_i\}} \beta_j.$$

In particular, if $\alpha = 1$, not only the Cauchy process is infinitely divisible, the approximated Cauchy prior distributions on the regression coefficients are also infinitely divisible:

$$\tilde{\beta}_i \overset{ind}{\sim} \mathsf{N}(0, n_i^2 \tilde{\varphi}_i^{-1}), \qquad \tilde{\varphi}_i \overset{iid}{\sim} \mathsf{G}(1/2, \epsilon^2/2)$$

At most n non-zero coefficients! (JAGS is possible now?)

Further Simplification in Case with $\alpha = 1$

- Poisson number of points $J_\epsilon \sim \mathsf{P}\big(\nu_\epsilon^+(\alpha,\gamma)\big)$ with $\nu_\epsilon^+(\alpha,\gamma) = \frac{\gamma\alpha^{1-\alpha/2}}{2^{1-\alpha}\epsilon^\alpha}\frac{\Gamma(\alpha/2)}{\Gamma(1-\alpha/2)}$
- Given J, $[n_1 : n_n] \sim MN(J, 1/(n+1))$ points supported at each kernel located at $\boldsymbol{x}_j$

The regression mean function can be rewritten as

$$f(\boldsymbol{x}) = \sum_{i=0}^{n} \tilde{\beta}_i \psi(\boldsymbol{x}, \boldsymbol{\omega}_i), \quad \tilde{\beta}_i = \sum_{\{j \,|\, \boldsymbol{\chi}_j = \boldsymbol{x}_i\}} \beta_j.$$

In particular, if $\alpha = 1$, not only the Cauchy process is infinitely divisible, the approximated Cauchy prior distributions on the regression coefficients are also infinitely divisible:

$$\tilde{\beta}_i \overset{ind}{\sim} \mathsf{N}(0, n_i^2 \tilde{\varphi}_i^{-1}), \qquad \tilde{\varphi}_i \overset{iid}{\sim} \mathsf{G}(1/2, \epsilon^2/2)$$

At most n non-zero coefficients! (JAGS is possible now?)

Further Simplification in Case with $\alpha = 1$

- Poisson number of points $J_\epsilon \sim \mathsf{P}\big(\nu_\epsilon^+(\alpha,\gamma)\big)$ with $\nu_\epsilon^+(\alpha,\gamma) = \frac{\gamma\alpha^{1-\alpha/2}}{2^{1-\alpha}\epsilon^\alpha}\frac{\Gamma(\alpha/2)}{\Gamma(1-\alpha/2)}$
- Given J, $[n_1 : n_n] \sim MN(J, 1/(n+1))$ points supported at each kernel located at $\boldsymbol{x}_j$

The regression mean function can be rewritten as

$$f(\boldsymbol{x}) = \sum_{i=0}^{n} \tilde{\beta}_i \psi(\boldsymbol{x}, \boldsymbol{\omega}_i), \quad \tilde{\beta}_i = \sum_{\{j\,|\,\boldsymbol{\chi}_j = \boldsymbol{x}_i\}} \beta_j.$$

In particular, if $\alpha = 1$, not only the Cauchy process is infinitely divisible, the approximated Cauchy prior distributions on the regression coefficients are also infinitely divisible:

$$\tilde{\beta}_i \overset{ind}{\sim} \mathsf{N}(0, n_i^2 \tilde{\varphi}_i^{-1}), \qquad \tilde{\varphi}_i \overset{iid}{\sim} \mathsf{G}(1/2, \epsilon^2/2)$$

At most n non-zero coefficients! (JAGS is possible now?)

Collapsed Sampler

Advantage: Gaussian prior so β can be integrated out for MCMC under Gaussian error model in the n dimensional problem

- Integrate out β vector in Normal regression leaving kernel parameters Λ and φ with multinomial weights n_i
- n_i may be 0 which drops dictionary elements from representation in finite representation for fixed ϵ
- still trans-dimensional in φ but much simpler RJMCMC now

Collapsed Sampler

Advantage: Gaussian prior so β can be integrated out for MCMC under Gaussian error model in the n dimensional problem

- Integrate out β vector in Normal regression leaving kernel parameters $\mathbf{\Lambda}$ and φ with multinomial weights n_i
- n_i may be 0 which drops dictionary elements from representation in finite representation for fixed ϵ
- still trans-dimensional in φ but much simpler RJMCMC now

Collapsed Sampler

Advantage: Gaussian prior so β can be integrated out for MCMC under Gaussian error model in the n dimensional problem

- Integrate out β vector in Normal regression leaving kernel parameters $\boldsymbol{\Lambda}$ and φ with multinomial weights n_i
- n_i may be 0 which drops dictionary elements from representation in finite representation for fixed ϵ
- still trans-dimensional in φ but much simpler RJMCMC now

Collapsed Sampler

Advantage: Gaussian prior so β can be integrated out for MCMC under Gaussian error model in the n dimensional problem

- Integrate out β vector in Normal regression leaving kernel parameters $\mathbf{\Lambda}$ and φ with multinomial weights n_i
- n_i may be 0 which drops dictionary elements from representation in finite representation for fixed ϵ
- still trans-dimensional in φ but much simpler RJMCMC now

Feature Selection in Kernel

- Product structure allows interactions between variables
- Many input variables may be irrelevant
- Feature selection; if $\lambda_d = 0$ variable x_d is removed from all kernels
- Allow point mass on $\lambda_h = 0$ with probability $p_\lambda \sim B(a, b)$ (in practice have used $a = b = 1$

Consider 3 Scenarios

- D Different $\lambda - D$ parameters in each dimension
- $S + D$ Different λ_d parameters + Selection
- $S + E$ Selection + Equal for Remaining $\lambda_d = \lambda$

Feature Selection in Kernel

- Product structure allows interactions between variables
- Many input variables may be irrelevant
- Feature selection; if $\lambda_d = 0$ variable x_d is removed from all kernels
- Allow point mass on $\lambda_h = 0$ with probability $p_\lambda \sim B(a, b)$ (in practice have used $a = b = 1$

Consider 3 Scenarios

- D Different $\lambda - D$ parameters in each dimension
- $S + D$ Different λ_d parameters + Selection
- $S + E$ Selection + Equal for Remaining $\lambda_d = \lambda$

Feature Selection in Kernel

- Product structure allows interactions between variables
- Many input variables may be irrelevant
- Feature selection; if $\lambda_d = 0$ variable x_d is removed from all kernels
- Allow point mass on $\lambda_h = 0$ with probability $p_\lambda \sim B(a, b)$ (in practice have used $a = b = 1$

Consider 3 Scenarios

- D Different $\lambda - D$ parameters in each dimension
- $S + D$ Different λ_d parameters + Selection
- $S + E$ Selection + Equal for Remaining $\lambda_d = \lambda$

Feature Selection in Kernel

- Product structure allows interactions between variables
- Many input variables may be irrelevant
- Feature selection; if $\lambda_d = 0$ variable x_d is removed from all kernels
- Allow point mass on $\lambda_h = 0$ with probability $p_\lambda \sim B(a, b)$ (in practice have used $a = b = 1$

Consider 3 Scenarios

- D Different $\lambda - D$ parameters in each dimension
- $S + D$ Different λ_d parameters + Selection
- $S + E$ Selection + Equal for Remaining $\lambda_d = \lambda$

Feature Selection in Kernel

- Product structure allows interactions between variables
- Many input variables may be irrelevant
- Feature selection; if $\lambda_d = 0$ variable x_d is removed from all kernels
- Allow point mass on $\lambda_h = 0$ with probability $p_\lambda \sim B(a, b)$ (in practice have used $a = b = 1$

Consider 3 Scenarios

- D Different $\lambda - D$ parameters in each dimension
- $S + D$ Different λ_d parameters + Selection
- $S + E$ Selection + Equal for Remaining $\lambda_d = \lambda$

Feature Selection in Kernel

- Product structure allows interactions between variables
- Many input variables may be irrelevant
- Feature selection; if $\lambda_d = 0$ variable x_d is removed from all kernels
- Allow point mass on $\lambda_h = 0$ with probability $p_\lambda \sim B(a, b)$ (in practice have used $a = b = 1$

Consider 3 Scenarios

- D Different $\lambda - D$ parameters in each dimension
- $S + D$ Different λ_d parameters + Selection
- $S + E$ Selection + Equal for Remaining $\lambda_d = \lambda$

Feature Selection in Kernel

- Product structure allows interactions between variables
- Many input variables may be irrelevant
- Feature selection; if $\lambda_d = 0$ variable x_d is removed from all kernels
- Allow point mass on $\lambda_h = 0$ with probability $p_\lambda \sim B(a, b)$ (in practice have used $a = b = 1$

Consider 3 Scenarios

- D Different $\lambda - D$ parameters in each dimension
- $S + D$ Different λ_d parameters + Selection
- $S + E$ Selection + Equal for Remaining $\lambda_d = \lambda$

Regression Out of Sample Prediction

Average Relative MSE to best procedure

Data Sets	BARK			SVM	BART
	D	S + E	S + D		
Friedman1	1.22	2.26	1.93	5.36	1.97
Friedman2	1.07	1.09	1.04	4.36	3.64
Friedman3	1.46	2.30	1.44	2.70	1.00
Boston Housing	1.09	1.23	1.20	1.56	1.01
Body Fat	1.81	1.01	2.19	4.04	1.68
Basketball	1.01	1.01	1.02	1.16	1.10

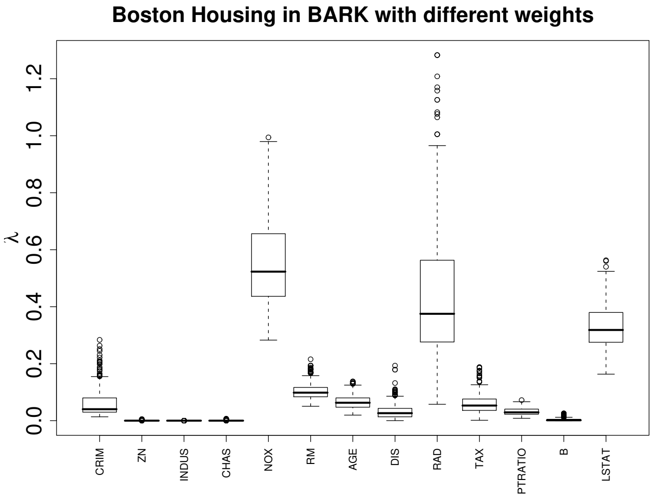
D: dimension specific scale λ_d

E: equal scales $\lambda_d = \lambda \, \forall \, d$

S: selection $\lambda_d = 0$ with probability ρ

Feature Selection in Boston Housing Data

Posterior Distribution of λ_d



Classification Examples

Name	d	data type	n (train/test)
Circle	2	simulation	200/1000
Circle (3 null)	5	simulation	200/1000
Circle (18 null)	20	simulation	200/1000
Swiss Bank Notes	6	real data	200 (5 cv)
Breast Cancer	30	real data	569 (5 cv)
Ionosphere	33	real data	351 (5 cv)

- Add latent Gaussian Z_i for probit regression (as in Albert & Chib)
- Same model as before conditional on $\boldsymbol{Z}$
- Advantage: Draw β in a block from full conditional
- Can extend to Logistic

Predictive Error Rate for Classification

Data Sets	BARK			SVM	BART
	D	S + E	S + D		
Circle 2	4.91%	1.88%	1.93%	5.03%	3.97%
Circle 5	4.70%	1.47%	1.65%	10.99%	6.51%
Circle 20	4.84%	2.09%	3.69%	44.10%	15.10%
Bank	1.25%	0.55%	0.88%	1.12%	0.50%
BC	4.02%	2.49%	6.09%	2.70%	3.36%
Ionosphere	8.59%	5.78%	10.87%	5.17%	7.34%

D: dimension specific scale λ_d

E: equal scales $\lambda_d = \lambda \forall d$

S: selection $\lambda_d = 0$ with probability ρ

Needs & Limitations

- NP Bayes of many flavors often does better than frequentist methods (BARK, BART, Treed GP, more)
- Hyper-parameter specification - theory & computational approximation
- need faster code for BARK that is easier for users (BART & TGP are great!) (`library(bark)` or github
- Can these models be added to JAGS, STAN, etc instead of stand-alone R packages
- With availability of code what are caveats for users?

Needs & Limitations

- NP Bayes of many flavors often does better than frequentist methods (BARK, BART, Treed GP, more)
- Hyper-parameter specification - theory & computational approximation
- need faster code for BARK that is easier for users (BART & TGP are great!) (`library(bark)` or github
- Can these models be added to JAGS, STAN, etc instead of stand-alone R packages
- With availability of code what are caveats for users?

Needs & Limitations

- NP Bayes of many flavors often does better than frequentist methods (BARK, BART, Treed GP, more)
- Hyper-parameter specification - theory & computational approximation
- need faster code for BARK that is easier for users (BART & TGP are great!) (`library(bark)` or github
- Can these models be added to JAGS, STAN, etc instead of stand-alone R packages
- With availability of code what are caveats for users?

Needs & Limitations

- NP Bayes of many flavors often does better than frequentist methods (BARK, BART, Treed GP, more)
- Hyper-parameter specification - theory & computational approximation
- need faster code for BARK that is easier for users (BART & TGP are great!) (`library(bark)`) or github
- Can these models be added to JAGS, STAN, etc instead of stand-alone R packages
- With availability of code what are caveats for users?

Needs & Limitations

- NP Bayes of many flavors often does better than frequentist methods (BARK, BART, Treed GP, more)
- Hyper-parameter specification - theory & computational approximation
- need faster code for BARK that is easier for users (BART & TGP are great!) (`library(bark)` or github
- Can these models be added to JAGS, STAN, etc instead of stand-alone R packages
- With availability of code what are caveats for users?

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)

Summary

Lévy Random Field Priors & LARK models:

- Provide limit of finite dimensional priors (GRP & SVSS) to infinite dimensional setting
- Adaptive bandwidth for kernel regression
- Allow flexible generating functions
- Provide sparser representations compared to SVM & RVM, with coherent Bayesian interpretation
- Incorporation of prior knowledge if available
- Relax assumptions of equally spaced data and Gaussian likelihood
- Hierarchical Extensions
- Formulation allows one to define stochastic processes on arbitrary spaces (spheres, manifolds)