Hierarchical Bayesian Methods

Bhaskar D Rao
University of California, San Diego

1Thanks to David Wipf, Jason Palmer, Zhilin Zhang, R. Prasad and Ritwik Giri
1. MAP Estimation Framework (Type I)
Bayesian Methods

1. MAP Estimation Framework (Type I)

2. Hierarchical Bayesian Framework (Type II)
MAP Estimation Framework (Type I)

Problem Statement

\[ \hat{x} = \arg \max_x p(x|y) = \arg \max_x p(y|x)p(x) \]

Choice of \( p(x) = \frac{a}{2} e^{-a|x|} \) as Laplacian and \( p(y|x) \) as Gaussian will lead to the familiar LASSO framework.
Hierarchical Bayes (Type II): Sparse Bayesian Learning (SBL)

SBL uses posterior information beyond the mode, i.e. posterior distribution $p(x|y)$.

Problem

For all sparse priors it is not possible to compute the normalized posterior $p(x|y)$, hence some approximations are needed.
Hierarchical Bayes (Type II): Sparse Bayesian Learning (SBL)

MAP methods were interested in the mode of the posterior $p(x|y)$ but SBL uses posterior information beyond the mode, i.e. posterior distribution $p(x|y)$. 
Hierarchical Bayes (Type II): Sparse Bayesian Learning (SBL)

MAP methods were interested in the mode of the posterior $p(x|y)$ but SBL uses posterior information beyond the mode, i.e. posterior distribution $p(x|y)$.

**Problem**

For all sparse priors it is not possible to compute the normalized posterior $p(x|y)$, hence some approximations are needed.
Hierarchical Bayesian Framework (Type II)

Problem Statement

\[ \hat{\gamma} = \arg \max_{\gamma} p(\gamma | y) = \arg \max_{\gamma} \int p(y | x) p(x | \gamma) p(\gamma) \, dx \]

Using this estimate of \( \hat{\gamma} \), one can compute the desired posterior \( p(x | y; \hat{\gamma}) \).

Bhaskar D Rao University of California, San Diego
Hierarchical Bayesian Framework (Type II)

Problem Statement

\[ \hat{\gamma} = \arg\max_{\gamma} p(\gamma|y) = \arg\max_{\gamma} \int p(y|x)p(x|\gamma)p(\gamma)dx \]

Using this estimate of \( \gamma \), one can compute the desired posterior \( p(x|y; \hat{\gamma}) \).
Hierarchical Bayesian Framework (Type II)

Potential Advantages

- Averaging over $x$ leads to fewer minima in $p(\gamma|y)$.
- $\gamma$ can tie several parameters, leading to fewer parameters.
Hierarchical Bayesian Framework (Type II)

Potential Advantages

- Averaging over $x$ leads to fewer minima in $p(\gamma|y)$.
- $\gamma$ can tie several parameters, leading to fewer parameters.

Example: Bayesian LASSO

Laplacian prior $p(x)$ can be represented as a Gaussian Scale Mixture in this fashion,

$$p(x) = \int p(x|\gamma)p(\gamma)d\gamma$$

$$= \int \frac{1}{\sqrt{2\pi\gamma}} \exp\left(-\frac{x^2}{2\gamma}\right) \times \frac{a^2}{2} \exp\left(-\frac{a^2}{2\gamma}\right)d\gamma$$

$$= \frac{a}{2} \exp\left(-a|x|\right)$$
Hierarchical Bayes: Sparse Bayesian Learning (SBL)

Instead of solving a MAP problem in $x$, in this framework one estimates the hyperparameters $\gamma$ and then an estimate of the posterior distribution for $x$, i.e. $p(x|y; \hat{\gamma})$. (Sparse Bayesian Learning)
Hierarchical Bayes: Sparse Bayesian Learning (SBL)

Instead of solving a MAP problem in $x$, in this framework one estimates the hyperparameters $\gamma$ and then an estimate of the posterior distribution for $x$, i.e., $p(x|y; \hat{\gamma})$. (Sparse Bayesian Learning)

Bhaskar D Rao University of California, San Diego
Instead of solving a MAP problem in $x$, in this framework one estimates the hyperparameters $\gamma$ and then an estimate of the posterior distribution for $x$, i.e. $p(x|y; \hat{\gamma})$. (Sparse Bayesian Learning)
In order for this framework to be useful, we need tractable representations:

**Gaussian Scaled Mixtures (GSM)**

A model for random variable $X$ is given by:

$$X = \gamma G$$

where, $G \sim N(\mu; 0, 1)$ and $\gamma$ is a positive random variable, which is independent of $G$.

The probability density function $p(x)$ can be written as:

$$p(x) = \int p(x|\gamma) p(\gamma) d\gamma = \int N(x; 0, \gamma) p(\gamma) d\gamma$$
In order for this framework to be useful, we need tractable representations: GaussianScaled Mixtures (GSM)
Useful Representation for Sparse priors

In order for this framework to be useful, we need tractable representations: Gaussian Scaled Mixtures (GSM)

**Gaussian Scale Mixtures** : Model for random variable $X$

$$X = \gamma G \text{ where, } G \sim N(g; 0, 1)$$

$\gamma$ is a positive random variable, which is independent of $G$. 
In order for this framework to be useful, we need tractable representations: Gaussian Scaled Mixtures (GSM)

**Gaussian Scale Mixtures** : Model for random variable $X$

$$X = \gamma G \text{ where, } G \sim N(g; 0, 1)$$

$\gamma$ is a positive random variable, which is independent of $G$.

$$p(x) = \int p(x|\gamma) \ p(\gamma) d\gamma$$

$$= \int N(x; 0, \gamma) \ p(\gamma) d\gamma$$
Gaussian Scale Mixtures

Most of the sparse priors over $x$ (including those with concave $g$) can be represented in this GSM form, and different scale mixing density i.e, $p(\gamma_i)$ will lead to different sparse priors. [Palmer et al., 2006]

Example: Laplacian density $p(x; a) = a^2 \exp(-a|x|)$

Scale mixing density: $p(\gamma) = a^2 \exp(-a^2 \gamma), \gamma \geq 0$.
Most of the sparse priors over $x$ (including those with concave $g$) can be represented in this GSM form, and different scale mixing density i.e, $p(\gamma_i)$ will lead to different sparse priors. [Palmer et al., 2006]
Most of the sparse priors over $x$ (including those with concave $g$) can be represented in this GSM form, and different scale mixing density i.e, $p(\gamma_i)$ will lead to different sparse priors. [Palmer et al., 2006]

Example: Laplacian density

$$p(x; a) = \frac{a}{2} \exp(-a|x|)$$

Scale mixing density: $p(\gamma) = \frac{a^2}{2} \exp(-\frac{a^2}{2} \gamma), \gamma \geq 0.$
Examples of Gaussian Scale Mixtures

**Student-t Distribution**

\[ p(x; a, b) = \frac{b^a \Gamma(a + 1/2)}{(2\pi)^{0.5} \Gamma(a)} \frac{1}{(b + x^2/2)^{a+1/2}} \]

*Scale mixing density*: Inverse Gamma Distribution

\[ p(\gamma) = \frac{1}{\Gamma(a)} b^a \frac{1}{\gamma^{a+1}} e^{-\frac{b}{\gamma}} u(\gamma). \]
### Examples of Gaussian Scale Mixtures

#### Student-t Distribution

\[
p(x; a, b) = \frac{b^a \Gamma(a + 1/2)}{(2\pi)^{0.5} \Gamma(a)} \frac{1}{(b + x^2/2)^{a+1/2}}
\]

**Scale mixing density**: Inverse Gamma Distribution

\[
p(\gamma) = \frac{1}{\Gamma(a)} b^a \frac{1}{\gamma^{a+1}} e^{-\frac{b}{\gamma}} u(\gamma).
\]

#### Generalized Gaussian

\[
p(x; p) = \frac{1}{2\Gamma(1 + \frac{1}{p})} e^{-|x|^p}
\]

**Scale mixing density**: Positive alpha stable density of order \( p/2 \).
Examples of Gaussian Scale Mixtures

**Student-t Distribution**

\[
p(x; a, b) = \frac{b^a \Gamma(a + 1/2)}{(2\pi)^{0.5} \Gamma(a)} \frac{1}{(b + x^2/2)^{a+1/2}}
\]

**Scale mixing density**: Inverse Gamma Distribution

\[
p(\gamma) = \frac{1}{\Gamma(a)} b^a \frac{1}{\gamma^{a+1}} e^{-\frac{b}{\gamma}} u(\gamma).
\]

**Generalized Gaussian**

\[
p(x; p) = \frac{1}{2\Gamma(1 + \frac{1}{p})} e^{-|x|^p}
\]

**Scale mixing density**: Positive alpha stable density of order \(p/2\).

**Generalized logistic density**

\[
p(x; a) = \frac{\Gamma(2\alpha)}{\Gamma(\alpha)^2} \frac{e^{-\alpha x}}{(1 + e^{-x})^{2\alpha}}
\]

**Scale mixing density**: Related to Kolmogorov-Smirnov distance statistics.
Sparse Bayesian Learning (Tipping)

\[ y = Ax + v \]

Solving for MAP estimate of \( \hat{\gamma} \)

\[
\hat{\gamma} = \arg \max_{\gamma} p(\gamma | y) = \arg \max_{\gamma} p(y, \gamma)
\]

\[
= \arg \max_{\gamma} p(y | \gamma) p(\gamma)
\]

What is \( p(y | \gamma) \)

Given \( \gamma \), \( x \) is Gaussian with mean zero and Covariance matrix \( \Gamma \) with \( \Gamma = \text{diag}(\gamma) \), i.e.

\[
p(x | \gamma) = \mathcal{N}(x; 0, \Gamma) = \prod \mathcal{N}(x_i; 0, \gamma_i)
\]

Then \( p(y | \gamma) = \mathcal{N}(y; 0, \Sigma_y) \), where

\[
\Sigma_y = \sigma^2 I + A \Gamma A^T
\]

\[
p(y | \gamma) = \frac{1}{\sqrt{(2\pi)^N \det(\Sigma_y)}} e^{-\frac{1}{2} y^T \Sigma_y^{-1} y}
\]

Bhaskar D Rao  University of California, San Diego
Sparse Bayesian Learning (Tipping)

\[ y = Ax + v \]
$y = Ax + v$

Solving for MAP estimate of $\gamma$

$$\hat{\gamma} = \arg \max_{\gamma} p(\gamma | y) = \arg \max_{\gamma} p(y, \gamma)p(\gamma) = \arg \max_{\gamma} p(y | \gamma)p(\gamma)$$
Sparse Bayesian Learning (Tipping)

\[ y = Ax + v \]

Solving for MAP estimate of \( \gamma \)

\[ \hat{\gamma} = \arg \max_{\gamma} p(\gamma|y) = \arg \max_{\gamma} p(y, \gamma)p(\gamma) = \arg \max_{\gamma} p(y|\gamma)p(\gamma) \]

What is \( p(y|\gamma) \)
Sparse Bayesian Learning (Tipping)

\[ y = Ax + v \]

Solving for MAP estimate of \( \gamma \)

\[
\hat{\gamma} = \arg\max_{\gamma} p(\gamma | y) = \arg\max_{\gamma} p(y, \gamma)p(\gamma) = \arg\max_{\gamma} p(y | \gamma)p(\gamma)
\]

What is \( p(y | \gamma) \)

Given \( \gamma, x \) is Gaussian with mean zero and Covariance matrix \( \Gamma \) with \( \Gamma = \text{diag}(\gamma) \), i.e. \( p(x | \gamma) = N(x; 0, \Gamma) = \prod N(x_i; 0, \gamma_i) \).
Sparse Bayesian Learning (Tipping)

\[ y = Ax + v \]

Solving for MAP estimate of \( \gamma \)

\[
\hat{\gamma} = \arg \max_{\gamma} p(\gamma | y) = \arg \max_{\gamma} p(y, \gamma)p(\gamma) = \arg \max_{\gamma} p(y | \gamma)p(\gamma)
\]

What is \( p(y | \gamma) \)

Given \( \gamma \), \( x \) is Gaussian with mean zero and Covariance matrix \( \Gamma \) with \( \Gamma = \text{diag}(\gamma) \), i.e. \( p(x | \gamma) = N(x; 0, \Gamma) = \prod N(x_i; 0, \gamma_i) \).

Then \( p(y | \gamma) = N(y; 0, \Sigma_y) \), where \( \Sigma_y = \sigma^2 I + A\Gamma A^T \),

\[
p(y | \gamma) = \frac{1}{\sqrt{(2\pi)^N|\Sigma_y|}} e^{-\frac{1}{2}y^T \Sigma_y^{-1} y}
\]
MAP estimate of $\gamma$

$$\hat{\gamma} = \arg \min_{\gamma} \left( \log |\Sigma_y| + y^T \Sigma_y^{-1} y - 2 \sum_i \log p(\gamma_i) \right)$$
\[ \hat{\gamma} = \arg \min_{\gamma} \left( \log |\Sigma_y| + y^T \Sigma_y^{-1} y - 2 \sum_i \log p(\gamma_i) \right) \]

**Computational Methods**

Many options for solving the above optimization problem, e.g. Majorization Minimization, Expectation-Maximization (EM).
Sparse Bayesian Learning

\[ y = Ax + v \]

**Computing Posterior**

Now because of our convenient GSM choice, posterior can be easily computed, i.e, \( p(x|y; \hat{\gamma}) = N(\mu_x, \Sigma_x) \) where,

\[ \mu_x = E[x|y; \hat{\gamma}] = \hat{\Gamma} A^T (\sigma^2 I + A\hat{\Gamma} A^T)^{-1} y \]

\[ \Sigma_x = Cov[x|y; \hat{\gamma}] = \hat{\Gamma} - \hat{\Gamma} A^T (\sigma^2 I + A\hat{\Gamma} A^T)^{-1} A\hat{\Gamma} \]
Sparse Bayesian Learning

\[ y = Ax + v \]

**Computing Posterior**

Now because of our convenient GSM choice, posterior can be easily computed, i.e, \( p(x|y; \hat{\gamma}) = N(\mu_x, \Sigma_x) \) where,

\[
\mu_x = E[x|y; \hat{\gamma}] = \hat{\Gamma} A^T (\sigma^2 I + A\hat{\Gamma}A^T)^{-1} y
\]

\[
\Sigma_x = \text{Cov}[x|y; \hat{\gamma}] = \hat{\Gamma} - \hat{\Gamma} A^T (\sigma^2 I + A\hat{\Gamma}A^T)^{-1} A\hat{\Gamma}
\]

\( \mu_x \) can be used as a point estimate.
Sparse Bayesian Learning

\[ y = Ax + v \]

**Computing Posterior**

Now because of our convenient GSM choice, posterior can be easily computed, i.e.,

\[ p(x|y; \hat{\gamma}) = N(\mu_x, \Sigma_x) \]

where,

\[ \mu_x = E[x|y; \hat{\gamma}] = \hat{\Gamma}A^T(\sigma^2 I + A\hat{\Gamma}A^T)^{-1}y \]

\[ \Sigma_x = Cov[x|y; \hat{\gamma}] = \hat{\Gamma} - \hat{\Gamma}A^T(\sigma^2 I + A\hat{\Gamma}A^T)^{-1}A\hat{\Gamma} \]

\( \mu_x \) can be used as a point estimate.

Sparsity of \( \mu_x \) is achieved through sparsity in \( \gamma \).
**Sparse Bayesian Learning**

\[ y = Ax + v \]

### Computing Posterior

Now because of our convenient GSM choice, posterior can be easily computed, i.e., \( p(x|y; \hat{\gamma}) = N(\mu_x, \Sigma_x) \) where,

\[
\mu_x = E[x|y; \hat{\gamma}] = \hat{\Gamma}A^T(\sigma^2 I + A\hat{\Gamma}A^T)^{-1}y
\]

\[
\Sigma_x = \text{Cov}[x|y; \hat{\gamma}] = \hat{\Gamma} - \hat{\Gamma}A^T(\sigma^2 I + A\hat{\Gamma}A^T)^{-1}A\hat{\Gamma}
\]

\( \mu_x \) can be used as a point estimate.

Sparsity of \( \mu_x \) is achieved through sparsity in \( \gamma \).

Another parameter of interest for the EM algorithm

\[
E(x_i^2|y, \hat{\gamma}) = \mu_x^2(i) + \Sigma_x(i, i)
\]
EM algorithm: Updating $\gamma$

Treating $(y, x)$ as complete data and vector $x$ as hidden variable.

$$\log p(y, x, \gamma) = \log p(y|x) + \log p(x|\gamma) + \log p(\gamma)$$

**E step**

$$Q(\gamma|\gamma_k) = \mathbb{E}_{x|y; \gamma_k} \left[ \log p(y|x) + \log p(x|\gamma) + \log p(\gamma) \right]$$

**M step**

$$\gamma_{k+1} = \arg\max \gamma Q(\gamma|\gamma_k) = \arg\max \gamma \mathbb{E}_{x|y; \gamma_k} \left[ \log p(x|\gamma) + \log p(\gamma) \right]$$

Solving this optimization problem with a non-informative prior $p(\gamma)$,

$$\gamma_{k+1} = \mathbb{E}(x_i^2|y, \gamma_k) = \mu_x(i)^2 + \sum x(i, i)$$
EM algorithm: Updating $\gamma$

Treating $(y, x)$ as complete data and vector $x$ as hidden variable.

$$\log p(y, x, \gamma) = \log p(y|x) + \log p(x|\gamma) + \log p(\gamma)$$
EM algorithm: Updating $\gamma$

Treating $(y, x)$ as complete data and vector $x$ as hidden variable.

$$\log p(y, x, \gamma) = \log p(y|x) + \log p(x|\gamma) + \log p(\gamma)$$

**E step**

$$Q(\gamma|\gamma^k) = \mathbb{E}_{x|y;\gamma^k}[\log p(y|x) + \log p(x|\gamma) + \log p(\gamma)]$$
EM algorithm: Updating $\gamma$

Treating $(y, x)$ as complete data and vector $x$ as hidden variable.

\[
\log p(y, x, \gamma) = \log p(y|x) + \log p(x|\gamma) + \log p(\gamma)
\]

**E step**

\[
Q(\gamma|\gamma^k) = \mathbb{E}_{x|y;\gamma^k} [\log p(y|x) + \log p(x|\gamma) + \log p(\gamma)]
\]

**M step**

\[
\gamma^{k+1} = \arg\max_{\gamma} Q(\gamma|\gamma^k) = \arg\max_{\gamma} \mathbb{E}_{x|y;\gamma^k} [\log p(x|\gamma) + \log p(\gamma)]
\]

\[
= \arg\min_{\gamma} \mathbb{E}_{x|y;\gamma^k} \left[ \sum_{i=1}^{M} \left( \frac{x_i^2}{2\gamma_i} + \frac{1}{2} \log \gamma_i \right) \right] - \log p(\gamma)
\]
EM algorithm: Updating $\gamma$

Treating $(y, x)$ as complete data and vector $x$ as hidden variable.

$$\log p(y, x, \gamma) = \log p(y|x) + \log p(x|\gamma) + \log p(\gamma)$$

**E step**

$$Q(\gamma | \gamma^k) = \mathbb{E}_{x|y;\gamma^k} [ \log p(y|x) + \log p(x|\gamma) + \log p(\gamma) ]$$

**M step**

$$\gamma^{k+1} = \arg \max_\gamma Q(\gamma | \gamma^k) = \arg \max_\gamma \mathbb{E}_{x|y;\gamma^k} [ \log p(x|\gamma) + \log p(\gamma) ]$$

$$= \arg \min_\gamma \mathbb{E}_{x|y;\gamma^k} \left[ \sum_{i=1}^M \left( \frac{x_i^2}{2\gamma_i} + \frac{1}{2} \log \gamma_i \right) \right] - \log p(\gamma)$$

Solving this optimization problem with a non-informative prior $p(\gamma)$,

$$\gamma_i^{k+1} = E(x_i^2|y, \gamma^k) = \mu_x(i)^2 + \Sigma_x(i, i)$$
SBL properties

Local minima are sparse, i.e. have at most $N$ nonzero $\gamma_i$. Cost function $p(\gamma|y)$ is generally much smoother than the associated MAP estimation objective $p(x|y)$. Fewer local minima.

In high signal to noise ratio, the global minima is the sparsest solution. No structural problems.

Attempts to approximate the posterior distribution $p(x|y)$ in the area with significant mass.

Bhaskar D Rao  University of California, San Diego
SBL properties

- Local minima are sparse, i.e. have at most $N$ nonzero $\gamma_i$
SBL properties

- Local minima are sparse, i.e. have at most $N$ nonzero $\gamma_i$
- Cost function $p(\gamma|y)$ is generally much smoother than the associated MAP estimation objective $p(x|y)$. Fewer local minima.

In high signal to noise ratio, the global minima is the sparsest solution. No structural problems.

Attempts to approximate the posterior distribution $p(x|y)$ in the area with significant mass.
Local minima are sparse, i.e. have at most $N$ nonzero $\gamma_i$

Cost function $p(\gamma|y)$ is generally much smoother than the associated MAP estimation objective $p(x|y)$. Fewer local minima.

In high signal to noise ratio, the global minima is the sparsest solution. No structural problems.
SBL properties

- Local minima are sparse, i.e. have at most $N$ nonzero $\gamma_i$
- Cost function $p(\gamma|y)$ is generally much smoother than the associated MAP estimation objective $p(x|y)$. Fewer local minima.
- In high signal to noise ratio, the global minima is the sparsest solution. No structural problems.
- Attempts to approximate the posterior distribution $p(x|y)$ in the area with significant mass.
Algorithmic Variants

- Fixed Point iteration based on setting the derivative of the objective function to zero (Tipping)
- Sequential search for the significant $\gamma$'s (Tipping and Faul)
- Majorization-Minimization based approach (Wipf and Nagarajan)
- Reweighted $\ell_1$ and $\ell_2$ algorithms (Wipf and Nagarajan)
- Approximate Message Passing (AlShoukairi and Bhaskar D. Rao)
Algorithmic Variants

- Fixed Point iteration based on setting the derivative of the objective function to zero (Tipping)
Algorithmic Variants

- Fixed Point iteration based on setting the derivative of the objective function to zero (Tipping)
- Sequential search for the significant $\gamma$’s (Tipping and Faul)
Algorithmic Variants

- Fixed Point iteration based on setting the derivative of the objective function to zero (Tipping)
- Sequential search for the significant $\gamma$’s (Tipping and Faul)
- Majorization-Minimization based approach (Wipf and Nagarajan)

Bhaskar D Rao  University of California, San Diego
Algorithmic Variants

- Fixed Point iteration based on setting the derivative of the objective function to zero (Tipping)
- Sequential search for the significant $\gamma$’s (Tipping and Faul)
- Majorization-Minimization based approach (Wipf and Nagarajan)
- Reweighted $\ell_1$ and $\ell_2$ algorithms (Wipf and Nagarajan)
Algorithmic Variants

- Fixed Point iteration based on setting the derivative of the objective function to zero (Tipping)
- Sequential search for the significant $\gamma$’s (Tipping and Faul)
- Majorization-Minimization based approach (Wipf and Nagarajan)
- Reweighted $\ell_1$ and $\ell_2$ algorithms (Wipf and Nagarajan)
- Approximate Message Passing (AlShoukairi) and Rao)
Empirical Comparison

For each test case

1. Generate a random dictionary $A$ with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.

2. Select the support for the true sparse coefficient vector $x_0$ randomly.

3. Generate the non-zero components of $x_0$ from the normal distribution.

4. Compute signal, $y = Ax_0$ (Noiseless case).

5. Compare SBL with previous methods with regard to estimating $x_0$.

6. Average over 1000 independent trials.

Bhaskar D Rao  University of California, San Diego
Empirical Comparison

For each test case

1. Generate a random dictionary \( A \) with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.
2. Select the support for the true sparse coefficient vector \( x_0 \) randomly.
3. Generate the non-zero components of \( x_0 \) from the normal distribution.
4. Compute signal, \( y = Ax_0 \) (Noiseless case).
5. Compare SBL with previous methods with regard to estimating \( x_0 \).
6. Average over 1000 independent trials.
Empirical Comparison

For each test case

1. Generate a random dictionary $\mathbf{A}$ with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.
Empirical Comparison

For each test case

1. Generate a random dictionary $A$ with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.

2. Select the support for the true sparse coefficient vector $x_0$ randomly.
Empirical Comparison

For each test case

1. Generate a random dictionary $A$ with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.

2. Select the support for the true sparse coefficient vector $x_0$ randomly.

3. Generate the non-zero components of $x_0$ from the normal distribution.

Compute signal, $y = Ax_0$ (Noiseless case).

Compare SBL with previous methods with regard to estimating $x_0$.

Average over 1000 independent trials.
Empirical Comparison

For each test case

1. Generate a random dictionary $\mathbf{A}$ with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.

2. Select the support for the true sparse coefficient vector $\mathbf{x}_0$ randomly.

3. Generate the non-zero components of $\mathbf{x}_0$ from the normal distribution.

4. Compute signal, $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ (Noiseless case).
Empirical Comparison

For each test case

1. Generate a random dictionary $\mathbf{A}$ with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.

2. Select the support for the true sparse coefficient vector $\mathbf{x}_0$ randomly.

3. Generate the non-zero components of $\mathbf{x}_0$ from the normal distribution.

4. Compute signal, $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ (Noiseless case).

5. Compare SBL with previous methods with regard to estimating $\mathbf{x}_0$. 
Empirical Comparison

For each test case

1. Generate a random dictionary $\mathbf{A}$ with 50 rows and 250 columns from the normal distribution and normalize each column to have 2-norm of 1.

2. Select the support for the true sparse coefficient vector $\mathbf{x}_0$ randomly.

3. Generate the non-zero components of $\mathbf{x}_0$ from the normal distribution.

4. Compute signal, $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ (Noiseless case).

5. Compare SBL with previous methods with regard to estimating $\mathbf{x}_0$.

6. Average over 1000 independent trials.
Empirical Comparison: 1000 trials

Figure: Probability of Successful recovery vs Number of non zero coefficients
Useful Extensions

Multiple Measurement Vectors (MMV)
Block Sparsity
Block MMV
MMV with time varying sparsity
Useful Extensions

- Multiple Measurement Vectors (MMV)
Useful Extensions

- Multiple Measurement Vectors (MMV)
- Block Sparsity
Useful Extensions

- Multiple Measurement Vectors (MMV)
- Block Sparsity
- Block MMV
- MMV with time varying sparsity
Multiple Measurement Vectors (MMV)

Multiple measurements:

$L$ measurements

Common Sparsity Profile:

$k$ nonzero rows
Multiple Measurement Vectors (MMV)

Model:

\[ Y_{N \times L} = \Phi_{N \times M} X_{M \times L} + V_{N \times L} \]

- \( Y_{N \times L} \): Multiple measurements
- \( \Phi_{N \times M} \): Common Sparsity Profile
- \( X_{M \times L} \): Matrix with \( k \) nonzero rows, \( k \ll M \)
- \( V_{N \times L} \): Noise term

Bhaskar D Rao  University of California, San Diego
Multiple Measurement Vectors (MMV)

- Model

\[ Y_{N \times L} = \Phi_{N \times M} X_{M \times L} + V_{N \times L} \]

- Multiple measurements: \( L \) measurements
- Common Sparsity Profile: \( k \) nonzero rows
Block Sparsity

Variations include equal blocks, unequal blocks, block boundary known or unknown.
Block Sparsity

\[ y = \Phi_{N \times M} x + v \]

- \( g \) blocks
- few non-zero blocks
Block Sparsity

Variations include equal blocks, unequal blocks, block boundary known or unknown.
Greedy Search Algorithms:
Extend MP, OMP to search for row sparsity.

Regularization methods

\[ \hat{X} = \arg \min_X \left[ \| Y - A X \|_2^2 + \lambda G(X) \right] \]

Choice of \( G(X) \)

- \( G(X) = \sum_M \| X_{i,.} \|_2 \), where \( X_{i,.} \) is the \( i \)th row of matrix \( X \) (Extension of \( \ell_1 \))
- \( G(X) = \sum_M \log(\| X_{i,.} \|_2 + \epsilon) \) (Extension of the Candes, Wakin and Boyd)
- \( G(X) = \sum_M \log(\| X_{i,.} \|_2^2 + \epsilon) \) (Extension of the Chartrand and Yin penalty)

Bhaskar D Rao  University of California, San Diego
Greedy Search Algorithms:

\[ \hat{X} = \arg \min_X \|Y - AX\|_F^2 + \lambda G(X) \]

Choice of $G(X)$:

- $G(X) = \sum M_i \|X_{i,.}\|_2$, where $X_{i,.}$ is the $i$th row of matrix $X$. (Extension of $\ell_1$)

- $G(X) = \sum M_i \log(\|X_{i,.}\|_2 + \epsilon)$ (Extension of the Candes, Wakin and Boyd)

- $G(X) = \sum M_i \log(\|X_{i,.}\|_2^2 + \epsilon)$ (Extension of the Chartrand and Yin penalty)
Greedy Search Algorithms: Extend MP, OMP to search for row sparsity.
MMV solutions

Greedy Search Algorithms: Extend MP, OMP to search for row sparsity.

Regularization methods
Greedy Search Algorithms: Extend MP, OMP to search for row sparsity.

Regularization methods

$$\hat{X} = \arg \min_X [\| Y - AX \|_F^2 + \lambda G(X)]$$
Greedy Search Algorithms: Extend MP, OMP to search for row sparsity.

Regularization methods

$$\hat{X} = \arg\min_{X} [\| Y - AX \|^2_F + \lambda G(X)]$$

Choice of $G(X)$
Greedy Search Algorithms: Extend MP, OMP to search for row sparsity.

Regularization methods

\[ \hat{X} = \arg \min_X \| Y - AX \|_F^2 + \lambda G(X) \]

Choice of \( G(X) \)

- \( G(X) = \sum_i^M \| X_{i,.} \|_2 \), where \( X_{i,.} \) is the \( i \)th row of matrix \( X \)
  (Extension of \( \ell_1 \))
Greedy Search Algorithms: Extend MP, OMP to search for row sparsity.

Regularization methods

\[ \hat{X} = \arg \min_X [\|Y - AX\|_F^2 + \lambda G(X)] \]

Choice of \( G(X) \)

- \( G(X) = \sum_i^M \|X_{i,.}\|_2 \), where \( X_{i,.} \) is the \( i \)th row of matrix \( X \) (Extension of \( \ell_1 \))
- \( G(X) = \sum_i^M \log(\|X_{i,.}\|_2 + \epsilon) \) (Extension of the Candes, Wakin and Boyd)
Greedy Search Algorithms: Extend MP, OMP to search for row sparsity.

Regularization methods

\[
\hat{X} = \arg \min_X [\| Y - AX \|_F^2 + \lambda G(X)]
\]

Choice of \( G(X) \)

- \( G(X) = \sum_i^M \| X_{i,.} \|_2 \), where \( X_{i,.} \) is the \( i \)th row of matrix \( X \)
  (Extension of \( \ell_1 \))

- \( G(X) = \sum_i^M \log(\| X_{i,.} \|_2 + \epsilon) \) (Extension of the Candes, Wakin and Boyd)

- \( G(X) = \sum_i^M \log(\| X_{i,.} \|_2^2 + \epsilon) \) (Extension of the Chartrand and Yin penalty)
Bayesian Methods

\[ X = \gamma G \]

where, \( G \sim N(g; 0, B) \)

\( \gamma \) is a positive random variable, which is independent of \( G \).

\[ p(x) = \int p(x | \gamma) p(\gamma) d\gamma = \int N(x; 0, \gamma B) p(\gamma) d\gamma \]

\( B = I \) if the row entries are assumed independent.
Bayesian Methods

Representation for Random Vectors

\[ \mathbf{X} = \gamma \mathbf{G} \text{ where, } \mathbf{G} \sim N(\mathbf{g}; 0, \mathbf{B}) \]

\( \gamma \) is a positive random variable, which is independent of \( \mathbf{G} \).
Bayesian Methods

Representation for Random Vectors

\[ X = \gamma G \text{ where, } G \sim N(g; 0, B) \]
\( \gamma \) is a positive random variable, which is independent of \( G \).

\[
p(x) = \int p(x|\gamma) \ p(\gamma) \ d\gamma
\]
\[
= \int N(x; 0, \gamma B) \ p(\gamma) \ d\gamma
\]
Representation for Random Vectors

\[ \mathbf{X} = \gamma \mathbf{G} \text{ where, } \mathbf{G} \sim N(\mathbf{g}; 0, \mathbf{B}) \]

\( \gamma \) is a positive random variable, which is independent of \( \mathbf{G} \).

\[
p(x) = \int p(x|\gamma) \, p(\gamma) \, d\gamma
\]

\[ = \int N(x; 0, \gamma \mathbf{B}) \, p(\gamma) \, d\gamma \]

\( \mathbf{B} = \mathbf{I} \) if the row entries are assumed independent.
EM algorithm: Updating $\gamma$

Treating $(Y, X)$ as complete data and vector $X$ as hidden variable.

$$\log p(Y, X, \gamma) = \log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)$$

**E step**

$$Q(\gamma|\gamma_k) = E_{X|Y, \gamma_k}[\log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)]$$

**M step**

$$\gamma^{k+1} = \arg\max_{\gamma} Q(\gamma|\gamma_k) = E_{X|Y; \gamma_k}[\log p(X|\gamma)]$$

$$= \arg\min_{\gamma} E_{x|y; \gamma_k}[\sum_{i=1}^{M} (\|X_i, .\|^2 + 2\log \gamma_i^{k+1}) - \log p(\gamma)]$$

Solving this optimization problem with a non-informative prior $p(\gamma)$,

$$\gamma^{k+1}_i = E(\|X_i, .\|^2|Y, \gamma_k) = L\sum_l \mu x(i, l)^2 + \sum x(i, i, l)$$
**EM algorithm: Updating $\gamma$**

Treating $(Y, X)$ as complete data and vector $X$ as hidden variable.

$$\log p(Y, X, \gamma) = \log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)$$
EM algorithm: Updating $\gamma$

Treating $(Y, X)$ as complete data and vector $X$ as hidden variable.

$$\log p(Y, X, \gamma) = \log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)$$

**E step**

$$Q(\gamma|\gamma^k) = \mathbb{E}_{X|Y, \gamma^k}[\log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)]$$
EM algorithm: Updating $\gamma$

Treating $(Y, X)$ as complete data and vector $X$ as hidden variable.

$$\log p(Y, X, \gamma) = \log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)$$

**E step**

$$Q(\gamma|\gamma^k) = \mathbb{E}_{X|Y;\gamma^k}[\log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)]$$

**M step**

$$\gamma^{k+1} = \arg\max_{\gamma} Q(\gamma|\gamma^k) = \arg\max_{\gamma} \mathbb{E}_{X|Y;\gamma^k}[\log p(X|\gamma) + \log p(\gamma)]$$

$$= \arg\min_{\gamma} \mathbb{E}_{X|Y;\gamma^k} \left[ \sum_{i=1}^{M} \left( \frac{||X_i,.||_2^2}{2\gamma_i} + \frac{1}{2} \log \gamma_i \right) \right] - \log p(\gamma)$$
EM algorithm: Updating $\gamma$

Treating $(Y, X)$ as complete data and vector $X$ as hidden variable.

$$\log p(Y, X, \gamma) = \log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)$$

**E step**

$$Q(\gamma|\gamma^k) = \mathbb{E}_{X|Y, \gamma^k}[\log p(Y|X) + \log p(X|\gamma) + \log p(\gamma)]$$

**M step**

$$\gamma^{k+1} = \arg\max_{\gamma} Q(\gamma|\gamma^k) = \arg\max_{\gamma} \mathbb{E}_{X|Y, \gamma^k}[\log p(X|\gamma) + \log p(\gamma)]$$

$$= \arg\min_{\gamma} \mathbb{E}_{X|Y, \gamma^k} \left[ \sum_{i=1}^{M} \left( \frac{\|X_i,.\|^2}{2\gamma_i} + \frac{1}{2} \log \gamma_i \right) - \log p(\gamma) \right]$$

Solving this optimization problem with a non-informative prior $p(\gamma),$

$$\gamma_i^{k+1} = E(\|X_i,.\|^2|Y, \gamma^k) = \sum_l \mu_x(i, l)^2 + \Sigma_x(i, i, l)$$
Generate data matrix via $Y = \Phi X_0$ (noiseless), where:

1. $X_0$ is 100-by-5 with random non-zero rows.
2. $\Phi$ is 50-by-100 with Gaussian iid entries.
MMV Empirical Comparison: 1000 trials

![Graph showing probability of success against row sparsity for different methods: SBL, Candès et al. (2008), Chartrand and Yin (2008), and L₁ solution.]
Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem

MAP methods (reweighted $\ell_1$ and $\ell_2$ methods)

Hierarchical Bayesian Methods (Sparse Bayesian Learning)

Versatile and can be more easily employed in problems with structure

Algorithms can often be justified by studying the resulting objective functions.

More applications to come enriching the field.

Bhaskar D Rao University of California, San Diego
Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.
Summary

- Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

- Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem

MAP methods (reweighted $\ell_1$ and $\ell_2$ methods)

Hierarchical Bayesian Methods (Sparse Bayesian Learning)

Versatile and can be more easily employed in problems with structure

Algorithms can often be justified by studying the resulting objective functions.

More applications to come enriching the field.
Summary

- Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.
- Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.
- Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.
Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem.
Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem.

- MAP methods (reweighted $\ell_1$ and $\ell_2$ methods)
Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem:
- MAP methods (reweighted $\ell_1$ and $\ell_2$ methods)
- Hierarchical Bayesian Methods (Sparse Bayesian Learning)
Summary

- Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

- Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

- Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

- Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem:
  - MAP methods (reweighted $\ell_1$ and $\ell_2$ methods)
  - Hierarchical Bayesian Methods (Sparse Bayesian Learning)
  - Versatile and can be more easily employed in problems with structure.

Bhaskar D Rao  University of California, San Diego
Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem:
- MAP methods (rewighted $\ell_1$ and $\ell_2$ methods)
- Hierarchical Bayesian Methods (Sparse Bayesian Learning)
- Versatile and can be more easily employed in problems with structure
- Algorithms can often be justified by studying the resulting objective functions.
Sparse Signal Recovery (SSR) and Compressed Sensing (CS) are interesting new signal processing tools with many potential applications.

Many algorithmic options exist to solve the underlying sparse signal recovery problem; Greedy Search Techniques, regularization methods, Bayesian methods, among others.

Nice theoretical results particularly for the greedy search algorithms and $\ell_1$ recovery methods.

Bayesian methods offer interesting algorithmic options to the Sparse Signal Recovery problem

- MAP methods (reweighted $\ell_1$ and $\ell_2$ methods)
- Hierarchical Bayesian Methods (Sparse Bayesian Learning)
- Versatile and can be more easily employed in problems with structure
- Algorithms can often be justified by studying the resulting objective functions.

More applications to come enriching the field.

Bhaskar D Rao  University of California, San Diego