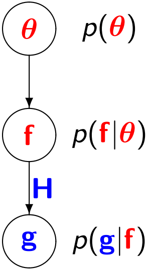
Variational Bayesian Approximation for Learning and Inference in Hierarchical Models

Ali Mohammad-Djafari
Laboratoire des Signaux et Systèmes (L2S)
UMR8506 CNRS-CentraleSupélec-UNIV PARIS SUD
SUPELEC, 91192 Gif-sur-Yvette, France
http://lss.centralesupelec.fr
Email: djafari@lss.supelec.fr
http://djafari.free.fr
http://publicationslist.org/djafari

Seminar given at: AIGM, Grenoble, June 30, 2015.

Hierarchical models (2 layers)



$$p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f})\, p(\mathbf{f}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})$$

Objective: Infer on $\mathbf{f},\boldsymbol{\theta}$

JMAP:

$$(\widehat{\mathbf{f}},\widehat{\boldsymbol{\theta}}) = \arg\max_{(\mathbf{f},\boldsymbol{\theta})} \{p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g})\}$$

Marginalization:

$$p(\mathbf{f}|\mathbf{g}) = \int p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g})\,\mathrm{d}\boldsymbol{\theta}$$

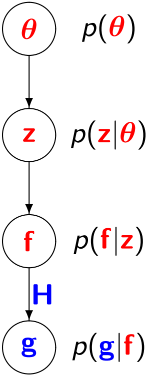
$$p(\boldsymbol{\theta}|\mathbf{g}) = \int p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g})\,\mathrm{d}\mathbf{f}$$

VBA:

Approximate $p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(\boldsymbol{\theta})$

Hierarchical models (3 layers)

$$p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f})\, p(\mathbf{f}|\mathbf{z})\, p(\mathbf{z}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})$$



Objective: Infer on $\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}$

JMAP:

$$(\widehat{\mathbf{f}}, \widehat{\mathbf{z}}, \widehat{\boldsymbol{\theta}}) = \arg\max_{(\mathbf{f},\mathbf{z},\boldsymbol{\theta})} \{p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})\}$$

Marginalization:

$$p(\mathbf{f}|\mathbf{g}) = \int p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g})\, \mathrm{d}\mathbf{z}\, \mathrm{d}\boldsymbol{\theta}$$

$$p(\mathbf{z}|\mathbf{g}) = \int p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g})\, \mathrm{d}\mathbf{f}\, \mathrm{d}\boldsymbol{\theta}$$

$$p(\boldsymbol{\theta}|\mathbf{g}) = \int p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g})\, \mathrm{d}\mathbf{f}\, \mathrm{d}\mathbf{z}$$

VBA:

Approximate $p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(\mathbf{z})\, q_3(\boldsymbol{\theta})$

JMAP

$$(\widehat{\mathbf{f}}, \widehat{\boldsymbol{\theta}}) = \underset{(\mathbf{f},\boldsymbol{\theta})}{\arg\max} \left\{ p(p(\mathbf{f}, \boldsymbol{\theta} | \mathbf{g}) \right\}$$

Alternate optimization:

$$\begin{cases} \widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ p(\widehat{\mathbf{f}}, \boldsymbol{\theta} | \mathbf{g}) \right\} \\ \widehat{\mathbf{f}} = \arg\max_{\mathbf{f}} \left\{ p(\mathbf{f}, \widehat{\boldsymbol{\theta}} | \mathbf{g}) \right\} \end{cases}$$

Main advantages:

- Simple
- Low computational cost

Main drawbacks:

- Convergence issues
- Uncertainties in each step are not accounted for

Marginalization

- Marginal MAP: $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta}|\mathbf{g})\}$ where

$$p(\boldsymbol{\theta}|\mathbf{g}) = \int p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}) \, \mathrm{d}\mathbf{f} \propto p(\mathbf{g}|\boldsymbol{\theta}) \, p(\boldsymbol{\theta})$$

and then:

$$\widehat{\mathbf{f}} = \arg\max_{\mathbf{f}} \left\{ p(\mathbf{f}|\widehat{\boldsymbol{\theta}}, \mathbf{g}) \right\} \text{ or } \widehat{\mathbf{f}} = \int \mathbf{f} \, p(\mathbf{f}|\widehat{\boldsymbol{\theta}}, \mathbf{g}) \, \mathrm{d}\mathbf{f}$$

- Main drawback: Needs the expression of the Likelihood:

$$p(\mathbf{g}|\boldsymbol{\theta}) = \int p(\mathbf{g}|\mathbf{f}, \boldsymbol{\theta}_1) \, p(\mathbf{f}|\boldsymbol{\theta}_2) \, \mathrm{d}\mathbf{f}$$

Not always analytically available $\longrightarrow$ EM, SEM and GEM algorithms

EM and GEM algorithms

- EM and GEM Algorithms: $\mathbf{f}$ as hidden variable,
  $\mathbf{g}$ as incomplete data, $(\mathbf{g}, \mathbf{f})$ as complete data
  $\ln p(\mathbf{g}|\boldsymbol{\theta})$ incomplete data log-likelihood
  $\ln p(\mathbf{g}, \mathbf{f}|\boldsymbol{\theta})$ complete data log-likelihood
- Iterative algorithm:

$$\begin{cases} \text{E-step:} & Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(k)}) = \mathrm{E}_{p(\mathbf{f}|\mathbf{g}, \widehat{\boldsymbol{\theta}}^{(k)})} \{\ln p(\mathbf{g}, \mathbf{f}|\boldsymbol{\theta})\} \\ \text{M-step:} & \widehat{\boldsymbol{\theta}}^{(k)} = \arg\max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(k-1)}) \right\} \end{cases}$$

- GEM (Bayesian) algorithm:

$$\begin{cases} \text{E-step:} & Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(k)}) = \mathrm{E}_{p(\mathbf{f}|\mathbf{g}, \widehat{\boldsymbol{\theta}}^{(k)})} \{\ln p(\mathbf{g}, \mathbf{f}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})\} \\ \text{M-step:} & \widehat{\boldsymbol{\theta}}^{(k)} = \arg\max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}^{(k-1)}) \right\} \end{cases}$$

$$\boxed{p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})} \longrightarrow \boxed{\text{EM, GEM}} \longrightarrow \widehat{\boldsymbol{\theta}} \longrightarrow \boxed{p(\mathbf{f}|\widehat{\boldsymbol{\theta}}, \mathbf{g})} \longrightarrow \widehat{\mathbf{f}}$$

JMAP, Marginalization, VBA

- JMAP:

$$\boxed{\begin{array}{c} p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}) \\ \text{optimization} \end{array}} \begin{array}{l} \longrightarrow \widehat{\mathbf{f}} \\ \longrightarrow \widehat{\boldsymbol{\theta}} \end{array}$$

- Marginalization

$$\boxed{p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})} \longrightarrow \boxed{p(\boldsymbol{\theta}|\mathbf{g})} \longrightarrow \widehat{\boldsymbol{\theta}} \longrightarrow \boxed{p(\mathbf{f}|\widehat{\boldsymbol{\theta}}, \mathbf{g})} \longrightarrow \widehat{\mathbf{f}}$$

Joint Posterior — Marginalize over $\mathbf{f}$

- Variational Bayesian Approximation

$$\boxed{p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})} \longrightarrow \boxed{\begin{array}{c} \text{Variational} \\ \text{Bayesian} \\ \text{Approximation} \end{array}} \begin{array}{l} \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}} \\ \longrightarrow q_2(\boldsymbol{\theta}) \longrightarrow \widehat{\boldsymbol{\theta}} \end{array}$$

Variational Bayesian Approximation

- Approximate $p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g})$ by $q(\mathbf{f},\boldsymbol{\theta}) = q_1(\mathbf{f})\, q_2(\boldsymbol{\theta})$
  and then use them for any inferences on $\mathbf{f}$ and $\boldsymbol{\theta}$ respectively.
- Criterion: $\mathrm{KL}(q(\mathbf{f},\boldsymbol{\theta}|\mathbf{g}) : p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g}))$
  $\mathrm{KL}(q : p) = \int\int q \ln \frac{q}{p} = \int\int q_1 q_2 \ln \frac{q_1 q_2}{p}$
- Iterative algorithm $q_1 \longrightarrow q_2 \longrightarrow q_1 \longrightarrow q_2, \cdots$

$$
\begin{cases}
\widehat{q}_1(\mathbf{f}) & \propto \exp\left[\langle \ln p(\mathbf{g},\mathbf{f},\boldsymbol{\theta};\mathcal{M})\rangle_{\widehat{q}_2(\boldsymbol{\theta})}\right] \\
\widehat{q}_2(\boldsymbol{\theta}) & \propto \exp\left[\langle \ln p(\mathbf{g},\mathbf{f},\boldsymbol{\theta};\mathcal{M})\rangle_{\widehat{q}_1(\mathbf{f})}\right]
\end{cases}
$$

$$
\boxed{p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g})} \longrightarrow \boxed{\text{Variational Bayesian Approximation}} \begin{array}{l} \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}} \\ \longrightarrow q_2(\boldsymbol{\theta}) \longrightarrow \widehat{\boldsymbol{\theta}} \end{array}
$$

VBA, Free Energy, KL and Model selection

$$p(\mathbf{f},\boldsymbol{\theta},\mathbf{g}|\mathcal{M}) = p(\mathbf{g}|\mathbf{f},\boldsymbol{\theta},\mathcal{M})\, p(\mathbf{f}|\boldsymbol{\theta},\mathcal{M})\, p(\boldsymbol{\theta}|\mathcal{M})$$

$$p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g},\mathcal{M}) = \frac{p(\mathbf{f},\boldsymbol{\theta},\mathbf{g}|,\mathcal{M})}{p(\mathbf{g}|\mathcal{M})}$$

$$\mathrm{KL}(q:p) = \iint q(\mathbf{f},\boldsymbol{\theta}) \ln \frac{p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g};\mathcal{M})}{q(\mathbf{f},\boldsymbol{\theta})}\, \mathrm{d}\mathbf{f}\, \mathrm{d}\boldsymbol{\theta}$$

$$\begin{aligned} p(\mathbf{g}|\mathcal{M}) &= \iint q(\mathbf{f},\boldsymbol{\theta}) \frac{p(\mathbf{g},\mathbf{f},\boldsymbol{\theta}|\mathcal{M})}{q(\mathbf{f},\boldsymbol{\theta})}\, \mathrm{d}\mathbf{f}\, \mathrm{d}\boldsymbol{\theta} \\ &\geq \iint q(\mathbf{f},\boldsymbol{\theta}) \ln \frac{p(\mathbf{g},\mathbf{f},\boldsymbol{\theta}|\mathcal{M})}{q(\mathbf{f},\boldsymbol{\theta})}\, \mathrm{d}\mathbf{f}\, \mathrm{d}\boldsymbol{\theta}. \end{aligned}$$

Free energy:

$$\mathcal{F}(q) = \iint q(\mathbf{f},\boldsymbol{\theta}) \ln \frac{p(\mathbf{g},\mathbf{f},\boldsymbol{\theta}|\mathcal{M})}{q(\mathbf{f},\boldsymbol{\theta})}\, \mathrm{d}\mathbf{f}\, \mathrm{d}\boldsymbol{\theta}$$

Evidence of the model $\mathcal{M}$:

$$p(\mathbf{g}|\mathcal{M}) = \mathcal{F}(q) + \mathrm{KL}(q:p)$$

VBA: Separable Approximation

$$p(\mathbf{g}|\mathcal{M}) = \mathcal{F}(q) + \mathrm{KL}(q : p)$$

$$q(\mathbf{f}, \boldsymbol{\theta}) = q_1(\mathbf{f})\; q_2(\boldsymbol{\theta})$$

Minimizing $\mathrm{KL}(q : p)$ = Maximizing $\mathcal{F}(q)$

$$(\widehat{q}_1, \widehat{q}_2) = \arg\min_{(q_1,q_2)} \left\{\mathrm{KL}(q_1 q_2 : p)\right\} = \arg\max_{(q_1,q_2)} \left\{\mathcal{F}(q_1 q_2)\right\}$$

$\mathrm{KL}(q_1 q_2 : p)$ is convexe wrt q_1 when q_2 is fixed and vise versa:

$$\begin{cases} \widehat{q}_1 & = \arg\min_{q_1} \left\{\mathrm{KL}(q_1 \widehat{q}_2 : p)\right\} = \arg\max_{q_1} \left\{\mathcal{F}(q_1 \widehat{q}_2)\right\} \\ \widehat{q}_2 & = \arg\min_{q_2} \left\{\mathrm{KL}(\widehat{q}_1 q_2 : p)\right\} = \arg\max_{q_2} \left\{\mathcal{F}(\widehat{q}_1 q_2)\right\} \end{cases}$$

$$\begin{cases} \widehat{q}_1(\mathbf{f}) & \propto \exp\left[\langle \ln p(\mathbf{g}, \mathbf{f}, \boldsymbol{\theta}; \mathcal{M})\rangle_{\widehat{q}_2(\boldsymbol{\theta})}\right] \\ \widehat{q}_2(\boldsymbol{\theta}) & \propto \exp\left[\langle \ln p(\mathbf{g}, \mathbf{f}, \boldsymbol{\theta}; \mathcal{M})\rangle_{\widehat{q}_1(\mathbf{f})}\right] \end{cases}$$

VBA: Choice of family of laws q_1 and q_2

- Case 1 : $\longrightarrow$ Joint MAP

$$\left\{\begin{array}{ll}\widehat{q}_1(\mathbf{f}|\widetilde{\mathbf{f}}) & =\delta(\mathbf{f}-\widetilde{\mathbf{f}}) \\ \widehat{q}_2(\boldsymbol{\theta}|\widetilde{\boldsymbol{\theta}}) & =\delta(\boldsymbol{\theta}-\widetilde{\boldsymbol{\theta}})\end{array}\right. \longrightarrow \left\{\begin{array}{l}\widetilde{\mathbf{f}}=\arg\max_{\mathbf{f}}\left\{p(\mathbf{f},\widetilde{\boldsymbol{\theta}}|\mathbf{g};\mathcal{M})\right\} \\ \widetilde{\boldsymbol{\theta}}=\arg\max_{\boldsymbol{\theta}}\left\{p(\widetilde{\mathbf{f}},\boldsymbol{\theta}|\mathbf{g};\mathcal{M})\right\}\end{array}\right.$$

- Case 2 : $\longrightarrow$ EM

$$\left\{\begin{array}{ll}\widehat{q}_1(\mathbf{f}) & \propto p(\mathbf{f}|\boldsymbol{\theta},\mathbf{g}) \\ \widehat{q}_2(\boldsymbol{\theta}|\widetilde{\boldsymbol{\theta}}) & =\delta(\boldsymbol{\theta}-\widetilde{\boldsymbol{\theta}})\end{array}\right. \longrightarrow \left\{\begin{array}{ll}Q(\boldsymbol{\theta},\widetilde{\boldsymbol{\theta}}) & = \langle \ln p(\mathbf{f},\boldsymbol{\theta}|\mathbf{g};\mathcal{M})\rangle_{q_1(\mathbf{f}|\widetilde{\boldsymbol{\theta}})} \\ \widetilde{\boldsymbol{\theta}} & =\arg\max_{\boldsymbol{\theta}}\left\{Q(\boldsymbol{\theta},\widetilde{\boldsymbol{\theta}})\right\}\end{array}\right.$$

- Appropriate choice for inverse problems

$$\left\{\begin{array}{ll}\widehat{q}_1(\mathbf{f}) & \propto p(\mathbf{f}|\boldsymbol{\theta},\mathbf{g};\mathcal{M}) \\ \widehat{q}_2(\boldsymbol{\theta}) & \propto p(\boldsymbol{\theta}|\mathbf{f},\mathbf{g};\mathcal{M})\end{array}\right. \longrightarrow \left\{\begin{array}{l}\text{Accounts for the uncertainties of} \\ \widehat{\boldsymbol{\theta}} \text{ for } \widehat{\mathbf{f}} \text{ and vise versa.}\end{array}\right.$$

Exponential families, Conjugate priors

JMAP, EM and VBA

JMAP Alternate optimization Algorithm:

$$\begin{array}{ccc} \boldsymbol{\theta}^{(0)} \longrightarrow \widetilde{\boldsymbol{\theta}} \longrightarrow & \boxed{\widetilde{\mathbf{f}} = \arg\max_{\mathbf{f}} \left\{ p(\mathbf{f}, \widetilde{\boldsymbol{\theta}}|\mathbf{g}) \right\}} & \longrightarrow \widetilde{\mathbf{f}} \longrightarrow \widehat{\mathbf{f}} \\ \uparrow & & \downarrow \\ \widehat{\boldsymbol{\theta}} \longleftarrow \widetilde{\boldsymbol{\theta}} \longleftarrow & \boxed{\widetilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ p(\widetilde{\mathbf{f}}, \boldsymbol{\theta}|\mathbf{g}) \right\}} & \longleftarrow \widetilde{\mathbf{f}} \end{array}$$

EM:

$$\begin{array}{ccc} \boldsymbol{\theta}^{(0)} \longrightarrow \widetilde{\boldsymbol{\theta}} \longrightarrow & \boxed{q_1(\mathbf{f}) = p(\mathbf{f}|\widetilde{\boldsymbol{\theta}}, \mathbf{g})} & \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}} \\ \uparrow & & \downarrow \\ \widehat{\boldsymbol{\theta}} \longleftarrow \widetilde{\boldsymbol{\theta}} \longleftarrow & \boxed{\begin{array}{c} Q(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = \langle \ln p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}) \rangle_{q_1(\mathbf{f})} \\ \widetilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) \right\} \end{array}} & \longleftarrow \quad q_1(\mathbf{f}) \end{array}$$

VBA:

$$\begin{array}{ccc} \boldsymbol{\theta}^{(0)} \longrightarrow q_2(\boldsymbol{\theta}) \longrightarrow & \boxed{q_1(\mathbf{f}) \propto \exp\left[\langle \ln p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}) \rangle_{q_2(\boldsymbol{\theta})} \right]} & \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}} \\ \uparrow & & \downarrow \\ \widehat{\boldsymbol{\theta}} \longleftarrow q_2(\boldsymbol{\theta}) \longleftarrow & \boxed{q_2(\boldsymbol{\theta}) \propto \exp\left[\langle \ln p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}) \rangle_{q_1(\mathbf{f})} \right]} & \longleftarrow q_1(\mathbf{f}) \end{array}$$

Applications in Inverse problems

- Deconvolution

$$g(t) = h(t) * f(t) + \epsilon(t) \rightarrow g_i = \sum_k h_k f_{i-k} + \epsilon_i \rightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

Given **h** and **g** estimate **f**.

- System identification (Supervised Learning)

$$g(t) = h(t) * f(t) + \epsilon(t) \rightarrow g_i = \sum_k h_k f_{i-k} + \epsilon_i \rightarrow \mathbf{g} = \mathbf{F}\mathbf{h} + \boldsymbol{\epsilon}$$

Given **f** and **g** estimate **h**.

- Blind deconvolution (Learning and deconvolution)

$$g(t) = h(t) * f(t) + \epsilon(t) \rightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon} = \mathbf{F}\mathbf{h} + \boldsymbol{\epsilon}$$

Given **g** estimate **h** and **f**.

Applications in Factor Analysis

- PCA, ICA, NNMF, Blind Sources Separation (BSS)

$$g_m(t) = \sum_{n=1}^{N} A_{m,n} f_n(t) + \epsilon_m(t) \longrightarrow \mathbf{g}(t) = \mathbf{A}\mathbf{f}(t) + \boldsymbol{\epsilon}(t)$$

- PCA, ICA: **A** mixing matrix, **B** separating matrix

$$\mathbf{g}(t) = \mathbf{A}\mathbf{f}(t) \longrightarrow \widehat{\mathbf{f}}(t) = \mathbf{B}\mathbf{g}(t)$$

  PCA: find **B** such that components of $\widehat{\mathbf{f}}(t)$ be, as much as possible, uncorrelated.
  ICA: find **B** such that components of $\widehat{\mathbf{f}}(t)$ be, as much as possible, independent.

- Non-Negative Matrix Factorization (NNMF):

$$\mathbf{g}(t) = \mathbf{A}\mathbf{f}(t) \longrightarrow \mathbf{G} = \mathbf{A}\mathbf{F}$$

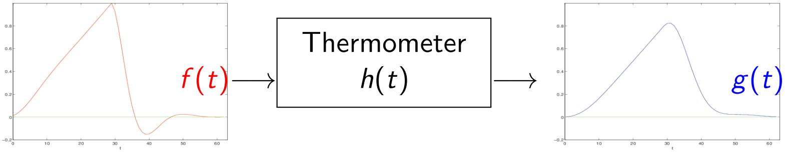
  Given **G** find both **A** and **F** (Factorization).

- BSS: Given $\mathbf{g}(t) = \mathbf{A}\mathbf{f}(t) + \boldsymbol{\epsilon}(t)$ find both **A** and $\mathbf{f}(t)$

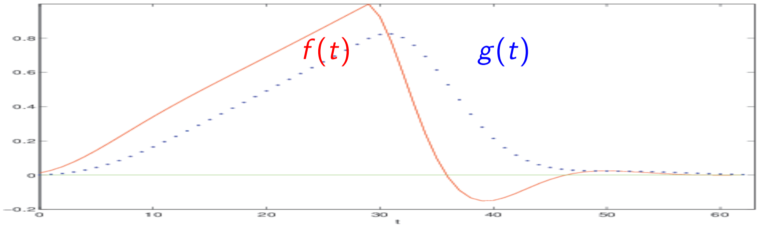
Measuring variation of temperature with a therometer

Forward model: Convolution

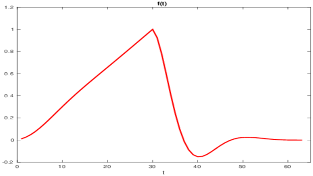
$$g(t) = \int f(t')\, h(t - t')\, \mathrm{d}t' + \epsilon(t)$$



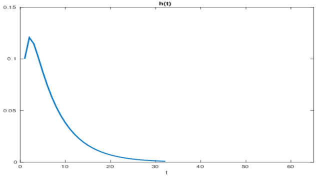
Inversion: Deconvolution



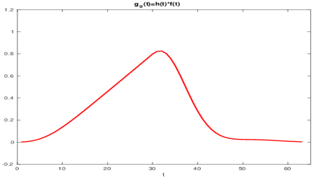
Deconvolution/Identification (1D case)



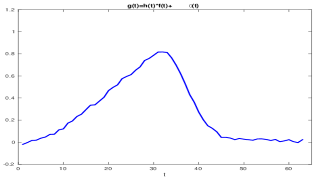
a) $f(t)$



b) $h(t)$



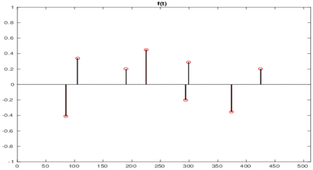
c) $g_0(t) = h(t) * g(t)$



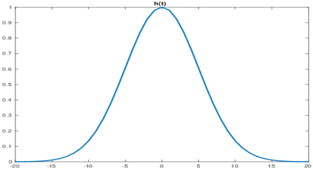
d) $g(t) = g_0(t) + \epsilon(t)$

- Deconvolution: Given $g(t)$ and $h(t)$ estimate $f(t)$.
- Identification: Given $g(t)$ and $f(t)$ estimate $h(t)$.

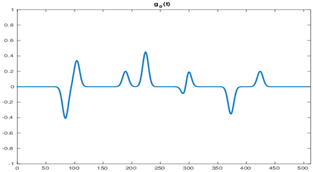
Deconvolution/Identification (1D case)



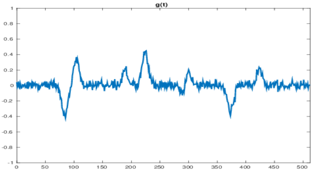
a) $f(t)$



b) $h(t)$



c) $g_0(t) = h(t) * g(t)$



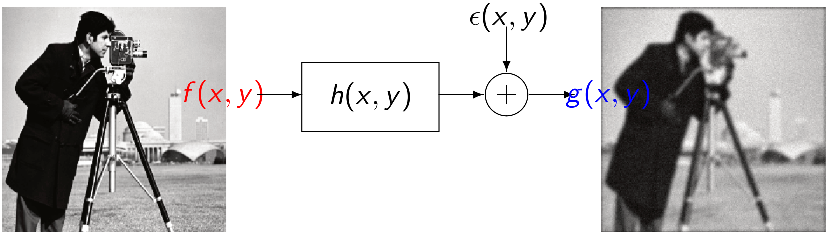
d) $g(t) = g_0(t) + \epsilon(t)$

- Deconvolution: Given $g(t)$ and $h(t)$ estimate $f(t)$.
- Identification: Given $g(t)$ and $f(t)$ estimate $h(t)$.

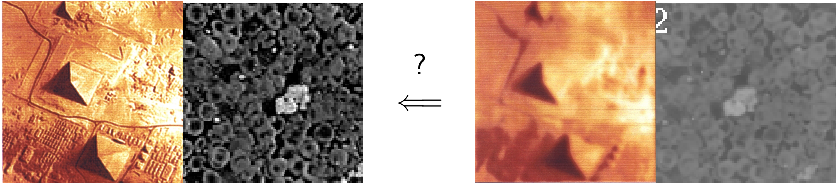
Making an image with an unfocused camera

Forward model: 2D Convolution

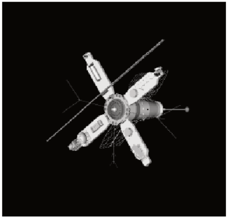
$$g(x, y) = \iint f(x', y')\, h(x - x', y - y')\, \mathrm{d}x'\, \mathrm{d}y' + \epsilon(x, y)$$



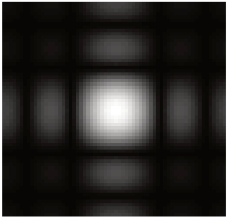
Inversion: Deconvolution



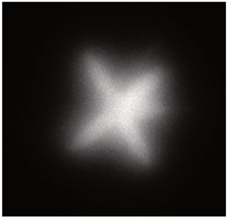
Deconvolution/Identification (2D case)



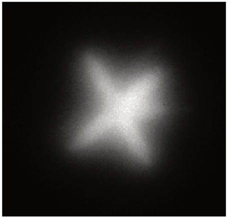
a) $f(x, y)$



b) $h(x, y)$



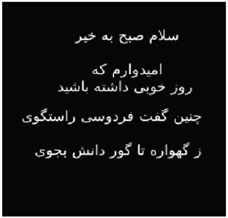
c) $g_0(x, y) = h(x, y) * f(x, y)$



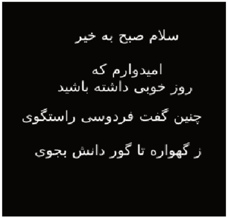
d) $g(x, y) = g_0(x, y) + \epsilon(x, y)$

- Deconvolution: Given $g(x, y)$ and $h(x, y)$ estimate $f(t)$.
- Identification: Given $g(t)$ and $f(x, y)$ estimate $h(x, y)$.

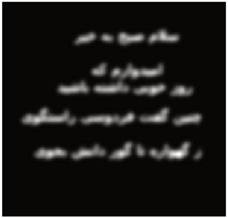
Deconvolution/Identification (2D case)



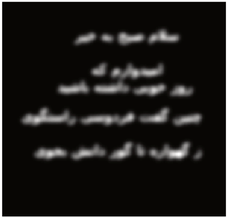
a) $f(x, y)$



b) $h(x, y)$



c) $g_0(x, y) = h(x, y) * f(x, y)$

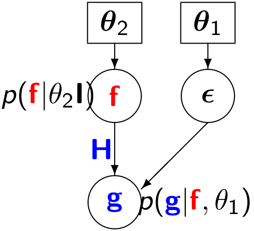


d) $g(x, y) = g_0(x, y) + \epsilon(x, y)$

- Deconvolution: Given $g(x, y)$ and $h(x, y)$ estimate $f(t)$.
- Identification: Given $g(t)$ and $f(x, y)$ estimate $h(x, y)$.

Deconvolution: Simple prior, Supervized case

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$



$$p(\mathbf{f}|\mathbf{g}, \boldsymbol{\theta}) \propto p(\mathbf{g}|\mathbf{f}, \boldsymbol{\theta}_1)\, p(\mathbf{f}|\boldsymbol{\theta}_2)$$

Objective: Infer $\mathbf{f}$

MAP: $\widehat{\mathbf{f}} = \arg\max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{g}, \boldsymbol{\theta})\}$

Posterior Mean (PM): $\widehat{\mathbf{f}} = \int \mathbf{f}\, p(\mathbf{f}|\mathbf{g}, \boldsymbol{\theta})\, \mathrm{d}\mathbf{f}$

Deconvolution: Simple Gaussian prior, Supervized case

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

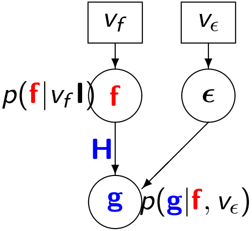
$$p(\mathbf{f}|\mathbf{g}, \boldsymbol{\theta}) \propto p(\mathbf{g}|\mathbf{f}, \boldsymbol{\theta}_1)\, p(\mathbf{f}|\boldsymbol{\theta}_2)$$

Objective: Infer $\mathbf{f}$

Caussian case:

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \propto \exp\left[\frac{-1}{2v_\epsilon}\|\mathbf{g} - \mathbf{h}\mathbf{f}\|^2\right] \\ p(\mathbf{f}|v_f) = \mathcal{N}(\mathbf{f}|0, v_f \mathbf{I}) \qquad \propto \exp\left[\frac{-1}{2v_f}\|\mathbf{f}\|^2\right] \end{cases}$$

$$p(\mathbf{f}|\mathbf{g}, v_\epsilon, v_f) \propto \exp\left[\frac{-1}{2} J(\mathbf{f})\right]$$



MAP: $\quad \widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{J(\mathbf{f})\}$ with

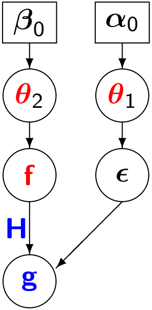
$$J(\mathbf{f}) = \frac{1}{v_\epsilon}\|\mathbf{g} - |\mathbf{H}\mathbf{f}\|^2 + \frac{1}{v_f}\|\mathbf{f}\|^2$$

Posterior Mean (PM)=MAP:

$$\begin{cases} p(\mathbf{f}|\mathbf{g}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\widehat{\mathbf{f}}, \widehat{\boldsymbol{\Sigma}}) \\ \widehat{\mathbf{f}} = (\mathbf{H}^t\mathbf{H} + \frac{v_\epsilon}{v_f}\mathbf{I}]^{-1}\mathbf{H}^t\mathbf{g} \\ \widehat{\boldsymbol{\Sigma}} = (\mathbf{H}^t\mathbf{H} + \frac{v_\epsilon}{v_f}\mathbf{I}]^{-1} \end{cases}$$

Deconvolution: Simple prior, Unsupervized case

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$



$$p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \boldsymbol{\theta}_1)\, p(\mathbf{f}|\boldsymbol{\theta}_2)\, p(\boldsymbol{\theta})$$

Objective: Infer $(\mathbf{f}, \boldsymbol{\theta})$

JMAP: $(\widehat{\mathbf{f}}, \widehat{\boldsymbol{\theta}}) = \arg\max_{(\mathbf{f}, \boldsymbol{\theta})} \{p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})\}$

Marginalization 1:

$$p(\mathbf{f}|\mathbf{g}) = \int p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})\, \mathrm{d}\boldsymbol{\theta}$$

Marginalization 2:

$p(\boldsymbol{\theta}|\mathbf{g}) = \int p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})\, \mathrm{d}\mathbf{f}$ followed by:

$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta}|\mathbf{g})\} \rightarrow$ Simple case

MCMC Gibbs sampling:

$\mathbf{f} \sim p(\mathbf{f}|\boldsymbol{\theta}, \mathbf{g}) \rightarrow \boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathbf{f}, \mathbf{g})$ until convergence

Use samples generated to compute mean and variances

VBA: Approximate $p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(\boldsymbol{\theta})$

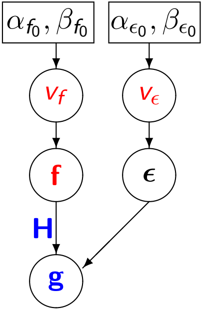
Use $q_1(\mathbf{f})$ to infer $\mathbf{f}$ and $q_2(\boldsymbol{\theta})$ to infer $\boldsymbol{\theta}$

Deconvolution: Simple prior, Unsupervized Gaussian case

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \propto \exp\left[\frac{-1}{2v_\epsilon}\|\mathbf{g} - \mathbf{h}\mathbf{f}\|^2\right] \\ p(\mathbf{f}|v_f) = \mathcal{N}(\mathbf{f}|0, v_f \mathbf{I}) \quad \propto \exp\left[\frac{-1}{2v_f}\|\mathbf{f}\|^2\right] \end{cases}$$

$$\begin{cases} p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(v_f) = \mathcal{IG}(v_f|\alpha_{f_0}, \beta_{f_0}) \end{cases}$$



$$p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, v_\epsilon)\, p(\mathbf{f}|v_f)\, p(v_\epsilon)\, p(v_f)$$

JMAP: $(\widehat{\mathbf{f}}, \hat{v}_\epsilon, \widehat{v}_f) = \arg\max_{(\mathbf{f}, v_\epsilon, v_f)} \{p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})\}$

Alternate optimization:

$$\begin{cases} \widehat{\mathbf{f}} = \left(\mathbf{H}'\mathbf{H} + \frac{\hat{v}_\epsilon}{\widehat{v}_f}\mathbf{I}\right)^{-1} \mathbf{H}'\mathbf{g} \\ \hat{v}_\epsilon = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon},\ \widetilde{\beta}_\epsilon = \beta_{\epsilon_0} + \|\mathbf{g} - \mathbf{H}\widehat{\mathbf{f}}\|^2,\ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon_0} + M/2 \\ \widehat{v}_f = \frac{\widetilde{\beta}_f}{\widetilde{\alpha}_f},\ \widetilde{\beta}_f = \beta_{f_0} + \|\widehat{\mathbf{f}}\|^2,\ \widetilde{\alpha}_f = \alpha_{f_0} + N/2 \end{cases}$$

VBA: Approximate $p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(v_\epsilon)\, q_3(v_f)$

Alternate optimization:

$$\begin{cases} q_1(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}_f) \\ q_2(v_\epsilon) = \mathcal{IG}(v_\epsilon|\widetilde{\alpha}_\epsilon, \widetilde{\beta}_\epsilon) \end{cases}$$

Deconvolution: Simple prior, Unsupervized Gaussian case

JMAP: $(\widehat{\mathbf{f}}, \hat{v}_\epsilon, \widehat{v}_f) = \arg\max_{(\mathbf{f}, v_\epsilon, v_f)} \{p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})\}$

Alternate optimization:

$$\begin{cases} \widehat{\mathbf{f}} = \left(\mathbf{H}'\mathbf{H} + \frac{\hat{v}_\epsilon}{\widehat{v}_f}\mathbf{I}\right)^{-1}\mathbf{H}'\mathbf{g} \\ \hat{v}_\epsilon = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon},\ \widetilde{\beta}_\epsilon = \beta_{\epsilon_0} + \|\mathbf{g} - \mathbf{H}\widehat{\mathbf{f}}\|^2,\ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon_0} + M/2 \\ \widehat{v}_f = \frac{\widetilde{\beta}_f}{\widetilde{\alpha}_f},\ \widetilde{\beta}_f = \beta_{f_0} + \|\widehat{\mathbf{f}}\|^2,\ \widetilde{\alpha}_f = \alpha_{f_0} + N/2 \end{cases}$$

VBA: Approximate $p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(v_\epsilon)\, q_3(v_f)$

Alternate optimization:

$$\begin{cases} q_1(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}_f) \\ q_2(v_\epsilon) = \mathcal{IG}(v_\epsilon|\widetilde{\alpha}_\epsilon, \widetilde{\beta}_\epsilon) \\ q_3(v_f) = \mathcal{IG}(v_\epsilon|\widetilde{\alpha}_f, \widetilde{\beta}_f) \end{cases}$$

$$\begin{cases} \widehat{\mathbf{f}} = \widetilde{\boldsymbol{\mu}} = \left(\mathbf{H}'\mathbf{H} + \frac{\hat{v}_\epsilon}{\widehat{v}_f}\mathbf{I}\right)^{-1}\mathbf{H}'\mathbf{g} \\ \widetilde{\boldsymbol{\Sigma}} = \left(\mathbf{H}'\mathbf{H} + \frac{\hat{v}_\epsilon}{\widehat{v}_f}\mathbf{I}\right)^{-1} \\ \hat{v}_\epsilon = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon},\ \widetilde{\beta}_\epsilon = \beta_{\epsilon_0} + < \|\mathbf{g} - \mathbf{H}\widehat{\mathbf{f}}\|^2 >,\ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon_0} + M/2 \\ \widehat{v}_f = \frac{\widetilde{\beta}_f}{\widetilde{\alpha}_f},\ \widetilde{\beta}_f = \beta_{f_0} + < \|\widehat{\mathbf{f}}\|^2 >,\ \widetilde{\alpha}_f = \alpha_{f_0} + N/2 \end{cases}$$

Sparsity enforcing models

- Generalized Gaussian model (p.c.: Double Exponential)

$$\mathcal{GG}(f|\alpha,\beta) \propto \exp\left[-\alpha|f|^\beta\right]$$

- Student-t model (p.c.: Cauchy, heavy tailed)

$$\mathcal{S}t(f|\nu) \propto \exp\left[-\frac{\nu+1}{2}\log\left(1+f^2/\nu\right)\right]$$

- Infinite Gausian Scaled Mixture (IGSM) equivalence

$$\mathcal{S}t(f|\nu) = \int_0^\infty \mathcal{N}(f|,0,1/z)\,\mathcal{G}(z|\alpha,\beta)\,\mathrm{d}z, \quad \text{with } \alpha=\beta=\nu/2$$

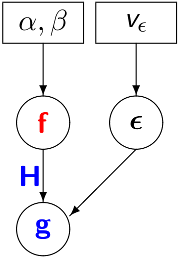
$$\left\{\begin{array}{lll} p(\mathbf{f}|\mathbf{z}) & = \prod_j p(f_j|z_j) = \prod_j \mathcal{N}(f_j|0,1/z_j) \propto \exp\left[-\frac{1}{2}\sum_j z_j f_j^2\right] \\ p(\mathbf{z}|\alpha,\beta) & = \prod_j \mathcal{G}(z_j|\alpha,\beta) \propto \prod_j z_j^{(\alpha-1)} \exp\left[-\beta z_j\right] \\ & \propto \exp\left[\sum_j (\alpha-1)\ln z_j - \beta z_j\right] \\ p(\mathbf{f},\mathbf{z}|\alpha,\beta) & \propto \exp\left[-\frac{1}{2}\sum_j z_j f_j^2 + (\alpha-1)\ln z_j - \beta z_j\right] \end{array}\right.$$

Deconvolution: Generalized Gaussian prior, Supervized

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \propto \exp\left[\frac{-1}{2v_\epsilon}\|\mathbf{g} - \mathbf{h}\mathbf{f}\|^2\right] \\ p(\mathbf{f}|\alpha, \beta) \propto \exp\left[-\alpha\|\mathbf{f}\|_\beta\right] \propto \exp\left[-\alpha\sum_j |f_j|^\beta\right] \end{cases}$$

$$p(\mathbf{f}|\mathbf{g}, v_\epsilon, \alpha, \beta) \propto p(\mathbf{g}|\mathbf{f}, v_\epsilon)\, p(\mathbf{f}|\alpha, \beta)$$



MAP: $\widehat{\mathbf{f}} = \arg\max_{\mathbf{f}} \{p(\mathbf{f}|\mathbf{g}, v_\epsilon, \alpha, \beta)\}$

Optimization:

$\widehat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{J(\mathbf{f})\}$

with

$$\begin{aligned} J(\mathbf{f}) &= \frac{1}{2v_\epsilon}\|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \alpha\|\mathbf{f}\|_\beta \\ &= \frac{1}{2v_\epsilon}\|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \alpha\sum_j |f_j|^\beta \end{aligned}$$

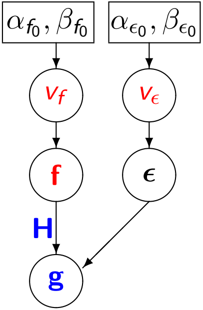
Link with LASSO

Deconvolution: Generalized Gaussian prior, Unsupervized

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \propto \exp\left[\frac{-1}{2v_\epsilon}\|\mathbf{g} - \mathbf{h}\mathbf{f}\|^2\right] \\ p(\mathbf{f}|\alpha, \beta) \propto \exp\left[-\alpha\|\mathbf{f}\|_\beta\right] \propto \exp\left[-\alpha \sum_j |f_j|^\beta\right] \\ p(\mathbf{f}|v_f, \beta) \propto \exp\left[-\frac{1}{2v_f}\|\mathbf{f}\|_\beta\right] \propto \exp\left[-\frac{1}{2v_f}\sum_j |f_j|^\beta\right] \end{cases}$$

$$\begin{cases} p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(v_f) = \mathcal{IG}(v_f|\alpha_{f_0}, \beta_{f_0}) \end{cases}$$



$$p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g}, \beta) \propto p(\mathbf{g}|\mathbf{f}, v_\epsilon)\, p(\mathbf{f}|v_f)\, p(v_\epsilon)\, p(v_f)$$

JMAP: $(\widehat{\mathbf{f}}, \hat{v}_\epsilon, \widehat{v_f}) = \arg\max_{(\mathbf{f}, v_\epsilon, v_f)} \{p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})\}$

Alternate optimization:

$$\begin{cases} J(\mathbf{f}) = \frac{1}{2v_\epsilon}\|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \frac{1}{2v_f}\|\mathbf{f}\|_\beta \\ \quad\quad = \frac{1}{2v_\epsilon}\|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \frac{1}{2v_f}\sum_j |f_j|^\beta \end{cases}$$

$$\begin{cases} \hat{v}_\epsilon = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon}, \ \widetilde{\beta}_\epsilon = \beta_{\epsilon_0} + \|\mathbf{g} - \mathbf{H}\widehat{\mathbf{f}}\|^2, \ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon_0} + M/2 \\ \widehat{v_f} = \frac{\widetilde{\beta}_f}{\widetilde{\alpha}_f}, \ \widetilde{\beta}_f = \beta_{f_0} + \|\widehat{\mathbf{f}}\|_\beta, \ \widetilde{\alpha}_f = \alpha_{f_0} + N/2 \end{cases}$$

VBA: Approximate $p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(v_\epsilon)\, q_3(v_f)$

Alternate optimization:

Deconvolution: Generalized Gaussian prior, Unsupervized

$p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g}, \beta) \propto p(\mathbf{g}|\mathbf{f}, v_\epsilon)\, p(\mathbf{f}|v_f)\, p(v_\epsilon)\, p(v_f)$

JMAP: $\quad (\widehat{\mathbf{f}}, \hat{v}_\epsilon, \widehat{v}_f) = \arg\max_{(\mathbf{f}, v_\epsilon, v_f)} \{p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})\}$

Alternate optimization:

$$\begin{cases} J(\mathbf{f}) & = \frac{1}{2v_\epsilon}\|\mathbf{g}-\mathbf{H}\mathbf{f}\|^2 + \frac{1}{2v_f}\|\mathbf{f}\|_\beta \\ & = \frac{1}{2v_\epsilon}\|\mathbf{g}-\mathbf{H}\mathbf{f}\|^2 + \frac{1}{2v_f}\sum_j |f_j|^\beta \end{cases}$$

$$\begin{cases} \hat{v}_\epsilon = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon},\ \widetilde{\beta}_\epsilon = \beta_{\epsilon 0} + \|\mathbf{g}-\mathbf{H}\widehat{\mathbf{f}}\|^2,\ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon 0} + M/2 \\ \widehat{v}_f = \frac{\widetilde{\beta}_f}{\widetilde{\alpha}_f},\ \widetilde{\beta}_f = \beta_{f_0} + \|\widehat{\mathbf{f}}\|_\beta,\ \widetilde{\alpha}_f = \alpha_{f_0} + N/2 \end{cases}$$

VBA: Approximate $p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(v_\epsilon)\, q_3(v_f)$

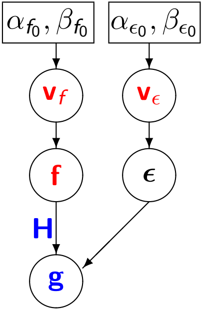
Alternate optimization:

$$\begin{cases} q_1(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}_f) \\ q_2(v_\epsilon) = \mathcal{IG}(v_\epsilon|\widetilde{\alpha}_\epsilon, \widetilde{\beta}_\epsilon) \\ q_3(v_f) = \mathcal{IG}(v_\epsilon|\widetilde{\alpha}_f, \widetilde{\beta}_f) \end{cases}$$

$$\begin{cases} \widehat{\mathbf{f}} = \widetilde{\boldsymbol{\mu}} = \left(\mathbf{H}'\mathbf{H} + \frac{\hat{v}_\epsilon}{\widehat{v}_f}\mathbf{I}\right)^{-1}\mathbf{H}'\mathbf{g} \\ \widetilde{\boldsymbol{\Sigma}} = \left(\mathbf{H}'\mathbf{H} + \frac{\hat{v}_\epsilon}{\widehat{v}_f}\mathbf{I}\right)^{-1} \\ \hat{v}_\epsilon = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon},\ \widetilde{\beta}_\epsilon = \beta_{\epsilon 0} + <\|\mathbf{g}-\mathbf{H}\widehat{\mathbf{f}}\|^2>,\ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon 0} + M/2 \end{cases}$$

Deconvolution: Nonstationnary noise, Student-t prior

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$



$$\begin{cases} p(\mathbf{g}|\mathbf{f},\mathbf{v}_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f},\mathbf{V}_\epsilon), & \mathbf{V}_\epsilon = \text{diag}\,[\mathbf{v}_\epsilon] \\ p(\mathbf{f}|\mathbf{v}_f) = \mathcal{N}(\mathbf{g}|0,\mathbf{V}_f), & \mathbf{V}_f = \text{diag}\,[\mathbf{v}_f] \end{cases}$$

$$\begin{cases} p(\mathbf{v}_\epsilon) = \prod_i \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon 0},\beta_{\epsilon 0}) \\ p(\mathbf{v}_f) = \prod_i \mathcal{IG}(v_{f_j}|\alpha_{f_0},\beta_{f_0}) \end{cases}$$

$$p(\mathbf{f},\mathbf{v}_\epsilon,\mathbf{v}_f|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{v}_\epsilon)\, p(\mathbf{f}|\mathbf{v}_f)\, p(\mathbf{v}_\epsilon)\, p(\mathbf{v}_f)$$

JMAP: $(\widehat{\mathbf{f}},\hat{\mathbf{v}}_\epsilon,\hat{\mathbf{v}}_f) = \arg\max_{(\mathbf{f},\mathbf{v}_\epsilon,\mathbf{v}_f)} \{p(\mathbf{f},\mathbf{v}_\epsilon,\mathbf{v}_f|\mathbf{g})\}$

Alternate optimization:

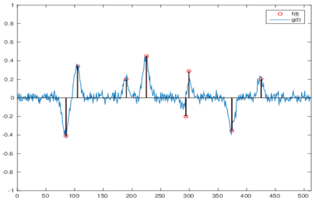
$$\begin{cases} \widehat{\mathbf{f}} = \left(\mathbf{H}'\mathbf{V}_\epsilon{}^{-1}\mathbf{H} + \mathbf{V}_f{}^{-1}\right)^{-1}\mathbf{H}'\mathbf{g} \\ \hat{v}_{\epsilon i} = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon},\ \widetilde{\beta}_\epsilon = \beta_{\epsilon_0} + (g_i - [\mathbf{H}\widehat{\mathbf{f}}]_i)^2,\ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon_0} + 1/2 \\ \widehat{v}_{fj} = \frac{\widetilde{\beta}_f}{\widetilde{\alpha}_f},\ \widetilde{\beta}_f = \beta_{f_0} + (\widehat{f}_j)^2,\ \widetilde{\alpha}_f = \alpha_{f_0} + 1/2 \end{cases}$$

VBA: Approximate $p(\mathbf{f},\mathbf{v}_\epsilon,\mathbf{v}_f|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(\mathbf{v}_\epsilon)\, q_3(\mathbf{v}_f)$

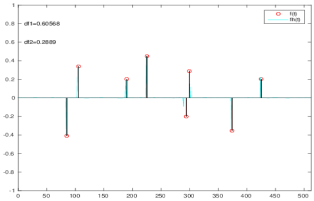
Alternate optimization:

$$\begin{cases} q_1(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\widetilde{\boldsymbol{\mu}},\widetilde{\boldsymbol{\Sigma}}_f) \\ q_2(v_{\epsilon_i}) = \mathcal{IG}(v_{\epsilon_i}|\widetilde{\alpha}_{\epsilon_i},\widetilde{\beta}_{\epsilon_i}) \\ q_3(v_{f_j}) = \mathcal{IG}(v_\epsilon|\widetilde{\alpha}_{f_j},\widetilde{\beta}_{f_j}) \end{cases}$$

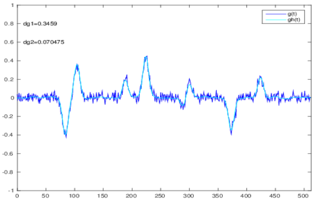
Deconvolution results 1



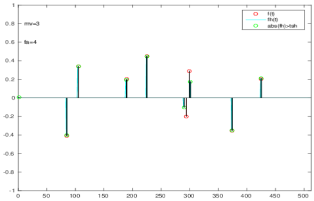
a) $f(t)$ and $g(t)$



b) $f(t)$ and $\widehat{f}(t)$



c) $g(t)$ and $\hat{g}(t)$

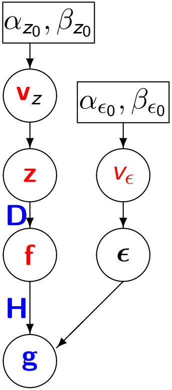


d) $f(t)$ and Thresholded $\widehat{f}(t)$

A typical result obtained with VBA: a) $f(t)$ and $g(t)$, b) $f(t)$ and $\widehat{f}(t)$ c) $g(t)$ and $\hat{g}(t)$ and d) $f(t)$ and thresholded $\widehat{f}(t)$. The relative distances between $f(t)$ and $\widehat{f}(t)$ and between $g(t)$ and

Deconvolution with sparse dictionnary prior

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon, \quad \mathbf{f} = \mathbf{D}\mathbf{z}, \quad \mathbf{z} \text{ sparse}, \quad \mathbf{g} = \mathbf{H}\mathbf{D}\mathbf{z} + \epsilon \rightarrow \widehat{\mathbf{f}} = \mathbf{D}\widehat{\mathbf{z}}.$$



$$\begin{cases} p(\mathbf{g}|\mathbf{z}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{D}\mathbf{z}, v_\epsilon \mathbf{I}) \\ p(\mathbf{z}|\mathbf{v}_z) = \mathcal{N}(\mathbf{z}|0, \mathbf{V}_z), \qquad \mathbf{V}_z = \operatorname{diag}[\mathbf{v}_z] \end{cases}$$

$$\begin{cases} p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(\mathbf{v}_z) = \prod_i \mathcal{IG}(v_{z_j}|\alpha_{z_0}, \beta_{z_0}) \end{cases}$$

$$p(\mathbf{z}, v_\epsilon, \mathbf{v}_z|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{z}, \mathbf{v}_\epsilon)\, p(\mathbf{z}|\mathbf{v}_z)\, p(v_\epsilon)\, p(\mathbf{v}_z)$$

JMAP:

$$(\widehat{\mathbf{z}}, \hat{v}_\epsilon, \widehat{\mathbf{v}}_z) = \underset{(\mathbf{z}, v_\epsilon, \mathbf{v}_z)}{\arg\max} \left\{ p(\mathbf{z}, v_\epsilon, \mathbf{v}_z|\mathbf{g}) \right\}$$

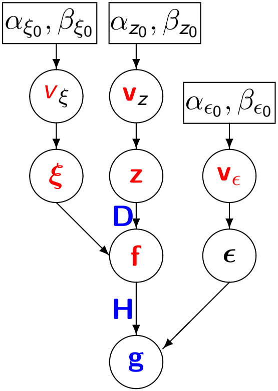
Alternate optimization.

VBA: Approximate

$p(\mathbf{z}, v_\epsilon, \mathbf{v}_z|\mathbf{g})$ by $q_1(\mathbf{z})\, q_2(v_\epsilon)\, q_3(\mathbf{v}_z)$

Alternate optimization.

Deconvolution with sparse dictionnary prior



$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}, \quad \mathbf{f} = \mathbf{D}\mathbf{z} + \boldsymbol{\xi}, \quad \mathbf{z}$ sparse

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, \mathbf{v}_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, \mathbf{V}_\epsilon), & \mathbf{V}_\epsilon = \text{diag}\,[\mathbf{v}_\epsilon] \\ p(\mathbf{f}|\mathbf{z}) = \mathcal{N}(\mathbf{f}|\mathbf{D}\mathbf{z}, v_\xi \mathbf{I}), & \\ p(\mathbf{z}|\mathbf{v}_z) = \mathcal{N}(\mathbf{z}|0, \mathbf{V}_z), & \mathbf{V}_z = \text{diag}\,[\mathbf{v}_z] \end{cases}$$

$$\begin{cases} p(\mathbf{v}_\epsilon) = \prod_i \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(\mathbf{v}_z) = \prod_i \mathcal{IG}(v_{z_j}|\alpha_{z_0}, \beta_{z_0}) \\ p(v_\xi) = \mathcal{IG}(v_\xi|\alpha_{\xi_0}, \beta_{\xi_0}) \end{cases}$$

$$p(\mathbf{f}, \mathbf{z}, \mathbf{v}_\epsilon, \mathbf{v}_z|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{v}_\epsilon)\, p(\mathbf{f}|\mathbf{z}_f)\, p(\mathbf{z}|\mathbf{v}_z)\, p(\mathbf{v}_\epsilon)\, p(\mathbf{v}_z)\, p(v_\xi)$$

JMAP:

$$(\widehat{\mathbf{f}}, \widehat{\mathbf{z}}, \hat{\mathbf{v}}_\epsilon, \widehat{\mathbf{v}}_z, \hat{v}_\xi) = \underset{(\mathbf{f}, \mathbf{z}, \mathbf{v}_\epsilon, \mathbf{v}_z, v_\xi)}{\arg\max} \{p(\mathbf{f}, \mathbf{z}, \mathbf{v}_\epsilon, \mathbf{v}_z, v_\xi|\mathbf{g})\}$$

Alternate optimization.

VBA: Approximate

$p(\mathbf{f}, \mathbf{z}, \mathbf{v}_\epsilon, \mathbf{v}_z, v_\xi|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(\mathbf{z})\, q_3(\mathbf{v}_\epsilon)\, q_4(\mathbf{v}_z)\, q_5(v_\xi)$

Alternate optimization.

Deconvolution/PSF Identification/Blind Deconvolution

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon} = \mathbf{F}\mathbf{h} + \boldsymbol{\epsilon}$$

Deconvolution	Identification
$\mathbf{g} = \mathbf{H}\,\mathbf{f} + \boldsymbol{\epsilon}$	$\mathbf{g} = \mathbf{F}\,\mathbf{h} + \boldsymbol{\epsilon}$
$p(\mathbf{g}\|\mathbf{f}) = \mathcal{N}(\mathbf{g}\|\mathbf{H}\mathbf{f}, \boldsymbol{\Sigma}_\epsilon = v_\epsilon \mathbf{I})$	$p(\mathbf{g}\|\mathbf{h}) = \mathcal{N}(\mathbf{g}\|\mathbf{F}\mathbf{h}, \boldsymbol{\Sigma}_\epsilon = v_\epsilon \mathbf{I})$
$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}\|0, \boldsymbol{\Sigma}_f = v_f[\mathbf{D}_f'\mathbf{D}_f]^{-1})$	$p(\mathbf{h}) = \mathcal{N}(\mathbf{f}\|0, \boldsymbol{\Sigma}_h = v_h[\mathbf{D}_h'\mathbf{D}_h]^{-1})$
$p(\mathbf{f}\|\mathbf{g}) = \mathcal{N}(\mathbf{f}\|\widehat{\mathbf{f}}, \widehat{\boldsymbol{\Sigma}}_f)$	$p(\mathbf{h}\|\mathbf{g}) = \mathcal{N}(\mathbf{f}\|\widehat{\mathbf{h}}, \widehat{\boldsymbol{\Sigma}}_h)$
$\widehat{\boldsymbol{\Sigma}}_f = v_\epsilon[\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}$	$\widehat{\boldsymbol{\Sigma}}_h = v_\epsilon[\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}$
$\widehat{\mathbf{f}} = [\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{D}_f'\mathbf{D}_f]^{-1}\mathbf{H}'\mathbf{g}$	$\widehat{\mathbf{h}} = [\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{D}_h'\mathbf{D}_h]^{-1}\mathbf{F}'\mathbf{g}$

- Blind Deconvolution: Joint posterior law:

$$p(\mathbf{f}, \mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{h})\, p(\mathbf{f})\, p(\mathbf{h})$$

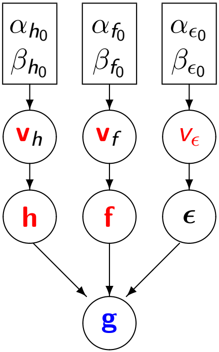
$$p(\mathbf{f}, \mathbf{h}|\mathbf{g}) \propto \exp\left[-J(\mathbf{f}, \mathbf{h})\right]$$

$$\text{with} \quad J(\mathbf{f}, \mathbf{h}) = \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda_f \|\mathbf{D}_f \mathbf{f}\|^2 + \lambda_h \|\mathbf{D}_h \mathbf{h}\|^2$$

- iterative algorithm

Blind Deconvolution: Unsupervized Gaussian case

$$g(t) = h(t) * f(t) + \epsilon(t) \longrightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon} = \mathbf{F}\mathbf{h} + \boldsymbol{\epsilon}$$



$$\begin{cases} p(\mathbf{g}|\mathbf{f}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \\ p(\mathbf{f}|\mathbf{v}_f) = \mathcal{N}(\mathbf{f}|0, \mathbf{V}_f) \\ p(\mathbf{h}|\mathbf{v}_h) = \mathcal{N}(\mathbf{h}|0, \mathbf{V}_h) \end{cases}$$

$$\begin{cases} p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(\mathbf{v}_f) = \prod_j \mathcal{IG}(v_{f\,k}|\alpha_{f_0}, \beta_{f_0}) \\ p(\mathbf{v}_h) = \prod_k \mathcal{IG}(v_{f_j}|\alpha_{f_0}, \beta_{f_0}) \end{cases}$$

$$p(\mathbf{f}, \mathbf{h}, v_\epsilon, \mathbf{v}_f, \mathbf{v}_h|\mathbf{g}) \propto \quad p(\mathbf{g}|\mathbf{f}, v_\epsilon)\, p(\mathbf{f}|\mathbf{v}_f)\, p(\mathbf{h}|\mathbf{v}_h)\, p(v_\epsilon)\, p(\mathbf{v}_f)\, p(\mathbf{v}_h)$$

JMAP: $(\widehat{\mathbf{f}}, \hat{v}_\epsilon, \widehat{v_f}) = \arg\max_{(\mathbf{f}, v_\epsilon, v_f)} \{p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})\}$

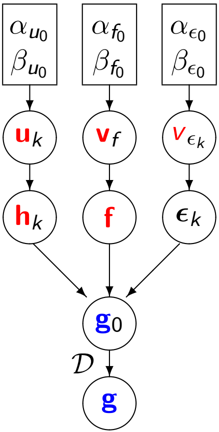
VBA:

Approximate $p(\mathbf{f}, \mathbf{h}, v_\epsilon, \mathbf{v}_f, \mathbf{v}_h|\mathbf{g})$ by

$q_1(\mathbf{f})\, q_2(\mathbf{h})\, q_3(v_\epsilon)\, q_4(\mathbf{v}_f)\, q_5(\mathbf{v}_h)$

Super-resolution: Unsupervized Gaussian case

$$g_k(x,y) = h_k(x,y) * f(x,y) + \epsilon_k(x,y) \longrightarrow \mathbf{g}_k = \mathcal{D}(\mathbf{H}_k \mathbf{f} + \boldsymbol{\epsilon}_k) = \mathcal{D}(\mathbf{F}\mathbf{h}_k + \boldsymbol{\epsilon}$$



$$\begin{cases} p(\mathbf{g}_k|\mathbf{f}, \mathbf{v}_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_{\epsilon_k}\mathbf{I}) \\ p(\mathbf{f}|\mathbf{v}_f) = \mathcal{N}(\mathbf{f}|0, v_f\mathbf{I}) \end{cases}$$

$$\begin{cases} p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(v_f) = \mathcal{IG}(v_f|\alpha_{f_0}, \beta_{f_0}) \end{cases}$$

$$p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, v_\epsilon)\, p(\mathbf{f}|v_f)\, p(v_\epsilon)\, p(v_f)$$

JMAP: $(\widehat{\mathbf{f}}, \hat{v}_\epsilon, \widehat{v}_f) = \arg\max_{(\mathbf{f}, v_\epsilon, v_f)} \{p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})\}$

Alternate optimization:

$$\begin{cases} \widehat{\mathbf{f}} = \left(\mathbf{H}'\mathbf{H} + \frac{\hat{v}_\epsilon}{\widehat{v}_f}\mathbf{I}\right)^{-1} \mathbf{H}'\mathbf{g} \\ \hat{v}_\epsilon = \frac{\widetilde{\beta}_\epsilon}{\widetilde{\alpha}_\epsilon},\ \widetilde{\beta}_\epsilon = \beta_{\epsilon_0} + \|\mathbf{g} - \mathbf{H}\widehat{\mathbf{f}}\|^2,\ \widetilde{\alpha}_\epsilon = \alpha_{\epsilon_0} + M/2 \\ \widehat{v}_f = \frac{\widetilde{\beta}_f}{\widetilde{\alpha}_f},\ \widetilde{\beta}_f = \beta_{f_0} + \|\widehat{\mathbf{f}}\|^2,\ \widetilde{\alpha}_f = \alpha_{f_0} + N/2 \end{cases}$$

VBA: Approximate $p(\mathbf{f}, v_\epsilon, v_f|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(v_\epsilon)\, q_3(v_f$

Alternate optimization:

$$\begin{cases} q_1(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}_f) \\ q_2(v_\epsilon) = \mathcal{IG}(v_\epsilon|\widetilde{\alpha}_\epsilon, \widetilde{\beta}_\epsilon) \\ q_3(v_f) = \mathcal{IG}(v_f|\widetilde{\alpha}_f, \widetilde{\beta}_f) \end{cases}$$

Blind Deconvolution:
Variational Bayesian Approximation algorithm

- Joint posterior law:

$$p(\mathbf{f},\mathbf{h}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f},\mathbf{h})\, p(\mathbf{f})\, p(h\mathbf{h})$$

- Approximation: $p(\mathbf{f},\mathbf{h}|\mathbf{g})$ by $q(\mathbf{f},\mathbf{h}|\mathbf{g}) = q_1(\mathbf{f})\, q_2(\mathbf{h})$
- Criterion of approximation: Kullback-Leiler

$$\mathrm{KL}(q|p) = \int q \ln \frac{q}{p} = \int q_1\, q_2 \ln \frac{q_1\, q_2}{p}$$

$$\begin{aligned}
\mathrm{KL}(q_1\, q_2|p) &= \int q_1 \ln q_1 + \int q_2 \ln q_2 - \int q \ln p \\
&= -\mathcal{H}(q_1) - \mathcal{H}(q_2) + \langle - \ln p((\mathbf{f},\mathbf{h}|\mathbf{g}) \rangle_q
\end{aligned}$$

- When the expression of q_1 and q_2 are obtained, use them.

Joint Estimation of h and f with a Gaussian prior model..

- Joint MAP:

$$
\begin{array}{ccccc}
\mathbf{h}^{(0)} \longrightarrow \mathbf{H} \longrightarrow & \boxed{\widehat{\mathbf{f}} = \left(\mathbf{H}'\mathbf{H} + \lambda_f \mathbf{I}\right)^{-1} \mathbf{H}'\mathbf{g}} & \longrightarrow \widehat{\mathbf{f}} \\
\uparrow & & \downarrow \\
\widehat{\mathbf{h}} \longleftarrow & \boxed{\widehat{\mathbf{h}} = \left(\mathbf{F}'\mathbf{F} + \lambda_h \mathbf{I}\right)^{-1} \mathbf{F}'\mathbf{g}} & \longleftarrow \mathbf{F}
\end{array}
$$

- VBA:

$$
\begin{array}{ccccc}
\begin{array}{c}\mathbf{h}^{(0)} \longrightarrow \mathbf{H} \longrightarrow \\ \boldsymbol{\Sigma}_h^{(0)} \longrightarrow \boldsymbol{\Sigma}_h \longrightarrow\end{array} & \boxed{\begin{array}{c}\widehat{\mathbf{f}} = \left(\mathbf{H}'\widehat{\boldsymbol{\Sigma}}_h^{-1}\mathbf{H} + \lambda_f \boldsymbol{\Sigma}_f\right)^{-1} \mathbf{H}'\mathbf{g} \\ \widehat{\boldsymbol{\Sigma}}_f = \sigma_\epsilon^2 (\mathbf{H}'\widehat{\boldsymbol{\Sigma}}_h\mathbf{H} + \lambda_f \boldsymbol{\Sigma}_f)^{-1}\end{array}} & \begin{array}{c}\longrightarrow \widehat{\mathbf{f}} \\ \longrightarrow \widehat{\boldsymbol{\Sigma}}_f\end{array} \\
\Uparrow & & \Downarrow \\
\begin{array}{c}\widehat{\mathbf{h}} \longleftarrow \\ \widehat{\boldsymbol{\Sigma}}_h \longleftarrow\end{array} & \boxed{\begin{array}{c}\widehat{\mathbf{h}} = \left(\mathbf{F}'\widehat{\boldsymbol{\Sigma}}_f^{-1}\mathbf{F} + \lambda_h \boldsymbol{\Sigma}_h\right)^{-1} \mathbf{F}'\mathbf{g} \\ \widehat{\boldsymbol{\Sigma}}_h = \sigma_\epsilon^2 (\mathbf{F}'\widehat{\boldsymbol{\Sigma}}_f\mathbf{F} + \lambda_h \boldsymbol{\Sigma}_h)^{-1}\end{array}} & \begin{array}{c}\longleftarrow \mathbf{F} \\ \longleftarrow \boldsymbol{\Sigma}_f\end{array}
\end{array}
$$

- Link with Message Passing and Belief Propagation methods

JMAP, EM and VBA for Bayesian BSS

$$\mathbf{g}(t) = \mathbf{A}\mathbf{f}(t) + \epsilon(t)$$

JMAP Alternate optimization Algorithm:

$$\begin{array}{ccc}
\mathbf{A}^{(0)} \longrightarrow \widetilde{\mathbf{A}} \longrightarrow & \boxed{\widetilde{\mathbf{f}} = \arg\max_{\mathbf{f}} \left\{ p(\mathbf{f}, \widetilde{\mathbf{A}}|\mathbf{g}) \right\}} & \longrightarrow \widetilde{\mathbf{f}} \longrightarrow \widehat{\mathbf{f}} \\
\uparrow & & \downarrow \\
\widehat{\mathbf{A}} \longleftarrow \widetilde{\mathbf{A}} \longleftarrow & \boxed{\widetilde{\mathbf{A}} = \arg\max_{\mathbf{A}} \left\{ p(\widetilde{\mathbf{f}}, \mathbf{A}|\mathbf{g}) \right\}} & \longleftarrow \widetilde{\mathbf{f}}
\end{array}$$

EM:

$$\begin{array}{ccc}
\mathbf{A}^{(0)} \longrightarrow \widetilde{\mathbf{A}} \longrightarrow & \boxed{q_1(\mathbf{f}) = p(\mathbf{f}|\widetilde{\mathbf{A}}, \mathbf{g})} & \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}} \\
\uparrow & & \downarrow \\
\widehat{\mathbf{A}} \longleftarrow \widetilde{\mathbf{A}} \longleftarrow & \boxed{\begin{array}{c} Q(\mathbf{A}, \widetilde{\mathbf{A}}) = \langle \ln p(\mathbf{f}, \mathbf{A}|\mathbf{g}) \rangle_{q_1(\mathbf{f})} \\ \widetilde{\mathbf{A}} = \arg\max_{\mathbf{A}} \left\{ Q(\mathbf{A}, \widetilde{\mathbf{A}}) \right\} \end{array}} & \longleftarrow q_1(\mathbf{f})
\end{array}$$

VBA:

$$\begin{array}{ccc}
\mathbf{A}^{(0)} \longrightarrow q_2(\mathbf{A}) \longrightarrow & \boxed{q_1(\mathbf{f}) \propto \exp\left[\langle \ln p(\mathbf{f}, \mathbf{A}|\mathbf{g}) \rangle_{q_2(\mathbf{A})} \right]} & \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}} \\
\uparrow & & \downarrow \\
\widehat{\mathbf{A}} \longleftarrow q_2(\mathbf{A}) \longleftarrow & \boxed{q_2(\mathbf{A}) \propto \exp\left[\langle \ln p(\mathbf{f}, \mathbf{A}|\mathbf{g}) \rangle_{q_1(\mathbf{f})} \right]} & \longleftarrow q_1(\mathbf{f})
\end{array}$$

VBA with Scale Mixture Model of Student-t priors

Scale Mixture Model of Student-t:

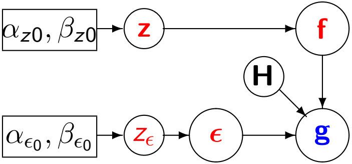
$$\mathcal{S}t(f_j|\nu) = \int_0^\infty \mathcal{N}(f_j|0, 1/z_j)\, \mathcal{G}(z_j|\nu/2, \nu/2)\, \mathrm{d}z_j$$

Hidden variables z_j:

$$\begin{array}{ll} p(\mathbf{f}|\mathbf{z}) & = \prod_j p(f_j|z_j) = \prod_j \mathcal{N}(f_j|0, 1/z_j) \propto \exp\left[-\frac{1}{2}\sum_j z_j f_j^2\right] \\ p(z_j|\alpha,\beta) & = \mathcal{G}(z_j|\alpha,\beta) \propto z_j^{(\alpha-1)} \exp\left[-\beta z_j\right] \text{ with } \alpha = \beta = \nu/2 \end{array}$$

Cauchy model is obtained when $\nu = 1$:

- Graphical model:

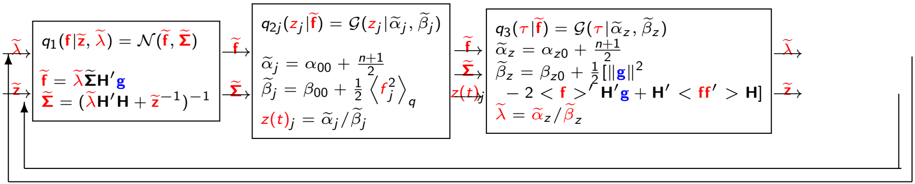


VBA with Student-t priors Algorithm

$$
\begin{cases}
p(\mathbf{g}|\mathbf{f}, z_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, (1/z_\epsilon)\mathbf{I}) \\
p(z_\epsilon|\alpha_{z0}, \beta_{z0}) = \mathcal{G}(z_\epsilon|\alpha_{z0}, \beta_{z0}) \\
p(\mathbf{f}|\mathbf{z}) = \prod_j \mathcal{N}(f_j|0, 1/z_j) \\
p(\mathbf{z}|\alpha_0, \beta_0) = \prod_j \mathcal{G}(z_j|\alpha_0, \beta_0) \\
\\
q_1(\mathbf{f}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}) = \mathcal{N}(\mathbf{f}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}) \\
\widetilde{\boldsymbol{\mu}} = \langle \lambda \rangle_q \widetilde{\boldsymbol{\Sigma}} \mathbf{H}'\mathbf{g} \\
\widetilde{\boldsymbol{\Sigma}} = (\langle \lambda \rangle_q \mathbf{H}'\mathbf{H} + \widetilde{\mathbf{Z}})^{-1}, \\
\text{with} \quad \widetilde{\mathbf{Z}} = \widetilde{\mathbf{z}}^{-1} = \text{diag}\,[\widetilde{\mathbf{z}}]
\end{cases}
$$

$$
\begin{cases}
q_{2j}(z_j) = \mathcal{G}(z_j|\widetilde{\alpha}_j, \widetilde{\beta}_j) \\
\widetilde{\alpha}_j = \alpha_{00} + 1/2 \\
\widetilde{\beta}_j = \beta_{00} + < f_j^2 > /2 \\
\\
q_3(z_\epsilon) = \mathcal{G}(z_\epsilon|\widetilde{\alpha}_{z\epsilon}, \widetilde{\beta}_{z\epsilon}), \\
\widetilde{\alpha}_{z\epsilon} = \alpha_{z0} + (n+1)/2 \\
\widetilde{\beta}_{z\epsilon} = \beta_{z0} + 1/2[\|\mathbf{g}\|^2 \\
-2 \langle \mathbf{f} \rangle_q' \mathbf{H}'\mathbf{g} + \mathbf{H}' \langle \mathbf{f}\mathbf{f}' \rangle_q \mathbf{H}]
\end{cases}
$$

$$
\begin{cases}
< \mathbf{f} > = \widetilde{\boldsymbol{\mu}} \\
< \mathbf{f}\mathbf{f}' > = \widetilde{\boldsymbol{\Sigma}} + \widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{\mu}}' \\
< f_j^2 > = [\widetilde{\boldsymbol{\Sigma}}]_{jj} + \widetilde{\mu}_j^2 \\
\widetilde{\lambda} = \widetilde{\alpha}_z / \widetilde{\beta}_z \\
\\
z(t)_j = \widetilde{\alpha}_j / \widetilde{\beta}_j
\end{cases}
$$



Open problems

- Other factorizations

$$p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g}) \simeq q_1(\mathbf{f})\, q_2(\mathbf{z})\, q_3(\boldsymbol{\theta})$$

or

$$p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g}) \simeq \prod_j q_{1j}(f_j)\, q_{2j}(z_j) \prod_k q_{3k}(\boldsymbol{\theta})$$

- VBA with separable approximation insures the equality of expected values (first moments). How about the second order or higher moments?
- Optimization by other methods than alternate optimization
- Convergence of algorithms
- How to measure the quality of different approximations?
- Application in real problems