Bayesian Blind Deconvolution

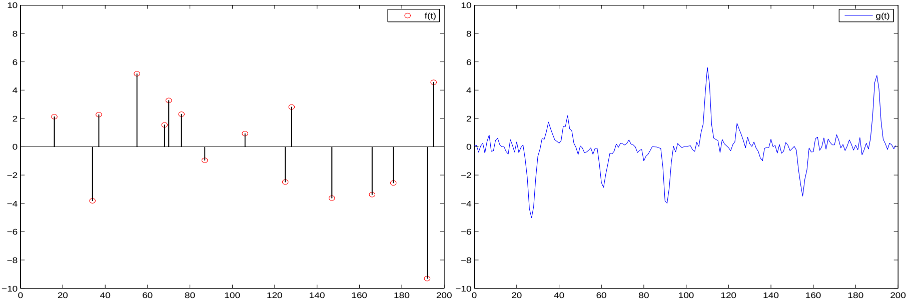
Ali Mohammad-Djafari
Laboratoire des Signaux et Systèmes,
UMR8506 CNRS-SUPELEC-UNIV PARIS SUD 11
SUPELEC, 91192 Gif-sur-Yvette, France
http://lss.supelec.free.fr

Email: djafari@lss.supelec.fr
http://djafari.free.fr

Seminar at Amir Kabir University (Polytechnique de) Tehran,
December 9, 2014

Signal Deconvolution

$$g(t) = \int f(\tau)\, h(t-\tau)\, \mathrm{d}\tau + \epsilon(t)$$



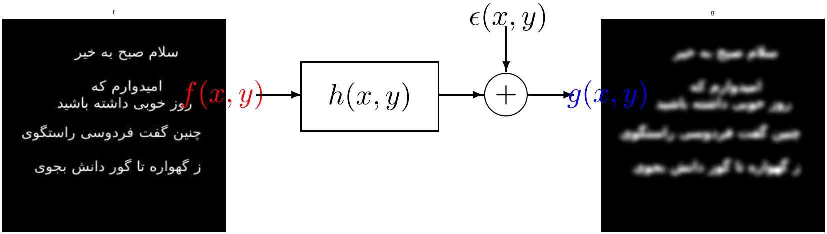
Forward: $f(t) \longrightarrow g(t) = h(t) * f(t) + \epsilon(t)$

Inverse: $\widehat{f}(t) \longleftarrow g(t)$

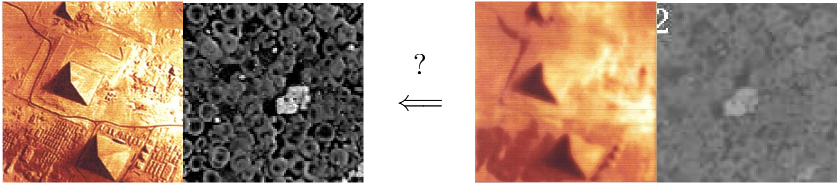
Image Restoration

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



Blind Deconvolution

$$g(t) = h(t) * f(t) + \epsilon(t)$$
$$g(x,y) = h(x,y) * f(x,y) + \epsilon(x,y)$$

- Convolution: Given f and h compute g
- Identification: Given f and g estimate h
- Deconvolution: Given g and h estimate f
- Blind deconvolution: Given g estimate both h and f

Discretization:

- $g = h * f + \epsilon \longrightarrow \boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}$,
  $\boldsymbol{H}$ huge dimensional Toeplitz or TBT matrice
  obtained from the elements of the impulse response $h(t)$ or the Point Spread Function (PSF) $h(x,y)$
- $g = f * h + \epsilon \longrightarrow \boldsymbol{g} = \boldsymbol{F}\boldsymbol{h} + \boldsymbol{\epsilon}$,
  $\boldsymbol{F}$ huge dimensional Hankel or HBT matrice
  obtained from the elements of the input signal $f(t)$ or the image $f(x,y)$

Identification and Deconvolution: Classical methods

Deconvolution	Identification
$g(t) = h(t) * f(t) + \epsilon(t)$	$g(t) = h(t) * f(t) + \epsilon(t)$
Wiener filtering	Wiener filtering
$\widehat{f}(\omega) = \frac{H^*(\omega)}{\|H(\omega)\|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{ff}(\omega)}} g(\omega)$	$\widehat{h}(\omega) = \frac{F^*(\omega)}{\|F(\omega)\|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{hh}(\omega)}} g(\omega)$
Regularization	Regularization
$\boldsymbol{g} = \boldsymbol{H}\,\boldsymbol{f} + \boldsymbol{\epsilon}$	$\boldsymbol{g} = \boldsymbol{F}\,\boldsymbol{h} + \boldsymbol{\epsilon}$
$J(\boldsymbol{f}) = \\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\\|^2 + \lambda_f \\|\boldsymbol{D}_f \boldsymbol{f}\\|^2$	$J(\boldsymbol{h}) = \\|\boldsymbol{g} - \boldsymbol{F}\boldsymbol{h}\\|^2 + \lambda_h \\|\boldsymbol{D}_h \boldsymbol{h}\\|^2$
$\nabla J(\boldsymbol{f}) = -2\boldsymbol{H}'(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}) + 2\lambda_f \boldsymbol{D}_f' \boldsymbol{D}_f \boldsymbol{f}$	$\nabla J(\boldsymbol{h}) = -2\boldsymbol{F}'(\boldsymbol{g} - \boldsymbol{F}\boldsymbol{h}) + 2\lambda_h \boldsymbol{D}_h' \boldsymbol{D}_h \boldsymbol{h}$
$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{D}_f' \boldsymbol{D}_f]^{-1} \boldsymbol{H}'\boldsymbol{g}$	$\widehat{\boldsymbol{h}} = [\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{D}_h' \boldsymbol{D}_h]^{-1} \boldsymbol{F}'\boldsymbol{g}$
Circulante approximation	Circulante approximation
$\widehat{f}(\omega) = \frac{H^*(\omega)}{\|H(\omega)\|^2 + \lambda_f \|D_f(\omega)\|^2} g(\omega)$	$\widehat{h}(\omega) = \frac{\|F^*(\omega)}{\|F(\omega)\|^2 + \lambda_h \|D_h(\omega)\|^2} g(\omega)$

Deconvolution, Identification: Bayesian approach

	Deconvolution	Identification
Forward model	$\boldsymbol{g} = \boldsymbol{H}\,\boldsymbol{f} + \boldsymbol{\epsilon}$	$\boldsymbol{g} = \boldsymbol{F}\,\boldsymbol{h} + \boldsymbol{\epsilon}$
Likelihood	$p(\boldsymbol{g}\|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{g}\|\boldsymbol{H}\,\boldsymbol{f}, \boldsymbol{\Sigma}_\epsilon)$	$p(\boldsymbol{g}\|\boldsymbol{h}) = \mathcal{N}(\boldsymbol{g}\|\boldsymbol{F}\boldsymbol{h}, \boldsymbol{\Sigma}_\epsilon)$
A priori	$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}\|0, \boldsymbol{\Sigma}_f)$	$p(\boldsymbol{h}) = \mathcal{N}(\boldsymbol{h}\|0, \boldsymbol{\Sigma}_h)$
Bayes	$p(\boldsymbol{f}\|\boldsymbol{g}) = \frac{p(\boldsymbol{g}\|\boldsymbol{f})\,p(\boldsymbol{f})}{p(\boldsymbol{g})}$	$p(\boldsymbol{h}\|\boldsymbol{g}) = \frac{p(\boldsymbol{g}\|\boldsymbol{h})\,p(\boldsymbol{h})}{p(\boldsymbol{g})}$

Particular case: $\boldsymbol{\Sigma}_\epsilon = \sigma_\epsilon^2 \boldsymbol{I}$, $\boldsymbol{\Sigma}_f = \sigma_f^2(\boldsymbol{D}_f'\boldsymbol{D}_f)^{-1}$, $\boldsymbol{\Sigma}_h = \sigma_h^2(\boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}$

Deconvolution	Identification
$p(\boldsymbol{f}\|\boldsymbol{g}) \propto p(\boldsymbol{g}\|\boldsymbol{f})\,p(\boldsymbol{f}) \propto \exp\left[-J(\boldsymbol{f})\right]$	$p(\boldsymbol{h}\|\boldsymbol{g}) \propto p(\boldsymbol{g}\|\boldsymbol{h})\,p(\boldsymbol{h}) \propto \exp\left[-J(\boldsymbol{h})\right]$
$J(\boldsymbol{f}) = \\|\boldsymbol{g} - \boldsymbol{H}\,\boldsymbol{f}\\|^2 + \lambda_f \\|\boldsymbol{D}_f\,\boldsymbol{f}\\|^2$	$J(\boldsymbol{h}) = \\|\boldsymbol{g} - \boldsymbol{F}\boldsymbol{h}\\|^2 + \lambda_h \\|\boldsymbol{D}_h\boldsymbol{h}\\|^2$
$\lambda_f = \sigma_\epsilon^2/\sigma_f^2$	$\lambda_h = \sigma_\epsilon^2/\sigma_h^2$
$p(\boldsymbol{f}\|\boldsymbol{g}) = \mathcal{N}(\boldsymbol{f}\|\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\Sigma}}_f)$	$p(\boldsymbol{h}\|\boldsymbol{g}) = \mathcal{N}(\boldsymbol{h}\|\widehat{\boldsymbol{h}}, \widehat{\boldsymbol{\Sigma}}_h)$
$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{D}_f'\boldsymbol{D}_f]^{-1}\boldsymbol{F}'\boldsymbol{g}$	$\widehat{\boldsymbol{h}} = [\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{D}_h'\boldsymbol{D}_h]^{-1}\boldsymbol{F}'\boldsymbol{g}$
$\widehat{\boldsymbol{\Sigma}}_f = [\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{D}_f'\boldsymbol{D}_f]^{-1}$	$\widehat{\boldsymbol{\Sigma}}_h = [\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{D}_h'\boldsymbol{D}_h]^{-1}$

Blind Deconvolution: Bayesian approach

$$\boldsymbol{g} = \boldsymbol{H}\,\boldsymbol{f} + \boldsymbol{\epsilon} = \boldsymbol{F}\,\boldsymbol{h} + \boldsymbol{\epsilon}$$

- Joint posterior law:

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

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

- Joint MAP:
$(\widehat{\boldsymbol{f}},\widehat{\boldsymbol{h}}) = \text{arg}\max_{(\boldsymbol{f},\boldsymbol{h})}\{p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\} = \text{arg}\min_{(\boldsymbol{f},\boldsymbol{h})}\{J(\boldsymbol{f},\boldsymbol{h})\}$
- Alternate optimization:

$$\left\{\begin{array}{ll} \widehat{\boldsymbol{f}}^{(k)} & = \text{arg}\max_{\boldsymbol{f}}\left\{p(\boldsymbol{f},\boldsymbol{h}^{(k)}|\boldsymbol{g})\right\} = \text{arg}\min_{\boldsymbol{f}}\left\{J(\boldsymbol{f},\boldsymbol{h}^{(k)})\right\} \\ \widehat{\boldsymbol{h}}^{(k)} & = \text{arg}\max_{\boldsymbol{h}}\left\{p(\boldsymbol{f}^{(k)},\boldsymbol{h}|\boldsymbol{g})\right\} = \text{arg}\min_{\boldsymbol{h}}\left\{J(\boldsymbol{f}^{(k)},\boldsymbol{h})\right\} \end{array}\right.$$

Blind Deconvolution: Variational Bayesian Approximation algorithm

- Joint posterior law:

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

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

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

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

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

Variational Bayesian Approximation algorithm

- Kullback-Leibler criterion

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

- Free energy

$$\mathcal{F}(q_1\, q_2) = -\left\langle \ln p((\boldsymbol{f}, \boldsymbol{h}|\boldsymbol{g}) \right\rangle_{q_1 q_2}$$

- Equivalence between optimization of $\mathrm{KL}(q_1\, q_2|p)$ and $\mathcal{F}(q_1\, q_2)$
- Alternate optimization:

$$\begin{aligned} \widehat{q}_1 &= \arg\min_{q_1} \{\mathrm{KL}(q_1\, q_2|p)\} = \arg\min_{q_1} \{\mathcal{F}(q_1\, q_2)\} \\ \widehat{q}_2 &= \arg\min_{q_2} \{\mathrm{KL}(q_1\, q_2|p)\} = \arg\min_{q_2} \{\mathcal{F}(q_1\, q_2)\} \end{aligned}$$

Summary of Bayesian methods for Blind Deconvolution

$$p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g}) = \frac{p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{h})\, p(\boldsymbol{f})\, p(\boldsymbol{h})}{p(\boldsymbol{g})}$$

- JMAP: $(\widehat{\boldsymbol{f}},\widehat{\boldsymbol{h}}) = \text{arg}\max_{(\boldsymbol{f},\boldsymbol{h})}\left\{p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\right\}$

$$\left\{\begin{array}{l} \widehat{\boldsymbol{f}}^{(k+1)} = \text{arg}\max_{\boldsymbol{f}}\left\{p(\boldsymbol{f},\boldsymbol{h}^{(k)}|\boldsymbol{g})\right\} \\ \widehat{\boldsymbol{h}}^{(k+1)} = \text{arg}\max_{\boldsymbol{h}}\left\{p(\boldsymbol{f}^{(k)},\boldsymbol{h}|\boldsymbol{g})\right\} \end{array}\right.$$

- VBA: Approximation: $p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})$ by $q(\boldsymbol{f},\boldsymbol{h}) = q_1(\boldsymbol{f})\, q_2(\boldsymbol{h})$

$$\left\{\begin{array}{lcl} \widehat{q}_1(\boldsymbol{f}) & = & \text{arg}\min_{q_1}\left\{\text{KL}(q_1\, q_2|p)\right\} = \text{arg}\min_{q_1}\left\{\mathcal{F}(q_1\, q_2)\right\} \\ \widehat{q}_2(\boldsymbol{h}) & = & \text{arg}\min_{q_2}\left\{\text{KL}(q_1\, q_2|p)\right\} = \text{arg}\min_{q_2}\left\{\mathcal{F}(q_1\, q_2)\right\} \end{array}\right.$$

Summary of Bayesian methods for Blind Deconvolution

Alternate optimization for JMAP

$$\widehat{\boldsymbol{h}}^{(0)} \rightarrow \widehat{\boldsymbol{h}} \rightarrow \boxed{\widehat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} \left\{ p(\boldsymbol{f}, \boldsymbol{h}^{(k)} | \boldsymbol{g}) \right\}} \rightarrow \widehat{\boldsymbol{f}}$$

$$\uparrow \qquad\qquad\qquad\qquad \downarrow$$

$$\widehat{\boldsymbol{h}} \leftarrow \boxed{\widehat{\boldsymbol{h}} = \arg\max_{\boldsymbol{h}} \left\{ p(\boldsymbol{f}^{(k)}, \boldsymbol{h} | \boldsymbol{g}) \right\}} \leftarrow \widehat{\boldsymbol{f}}$$

Alternate optimization for VBA

$$q_2^{(0)}(\boldsymbol{h}) \rightarrow q_2(\boldsymbol{h}) \rightarrow \boxed{\widehat{q}_1 = \arg\min_{q_2} \left\{ \mathrm{KL}(q_1\, \widehat{q}_2 | p) \right\}} \rightarrow \widehat{q}_1(\boldsymbol{f}) \rightarrow \widehat{\boldsymbol{h}}$$

$$\uparrow \qquad\qquad\qquad\qquad \downarrow$$

$$\widehat{\boldsymbol{h}} \leftarrow \widehat{q}_2(\boldsymbol{h}) \leftarrow \boxed{\widehat{q}_2 = \arg\min_{q_2} \left\{ \mathrm{KL}(\widehat{q}_1\, q_2 | p) \right\}} \leftarrow \widehat{q}_1(\boldsymbol{f})$$

JMAP and VBA with Gaussian priors

JMAP:

Initialization: $\boldsymbol{h}^{(0)} = \boldsymbol{h}_0, \quad \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(0)})$

Iterations:

$\boldsymbol{f}^{(k)} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g}$
$\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}^{(k-1)})$
$\boldsymbol{h}^{(k)} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{C}_h'\boldsymbol{C}_h)^{-1}\boldsymbol{F}'\boldsymbol{g}$
$\boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(k-1)})$

VBA:

Initialization: $\boldsymbol{h}^{(0)} = \boldsymbol{h}_0, \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(0)}), \boldsymbol{D}_f = \boldsymbol{I}, \boldsymbol{D}_h = \boldsymbol{I}$

Iterations:

$\boldsymbol{f}^{(k)} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{I} + v_\epsilon \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1}\boldsymbol{H}'\boldsymbol{g}$
$\boldsymbol{\Sigma}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{I} + v_\epsilon \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}$
$\boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}^{(k-1)}) \quad \mathrm{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_h'\boldsymbol{f}\|_2^2$

$\boldsymbol{h}^{(k)} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{C}_h'\boldsymbol{C}_h + v_\epsilon \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}\boldsymbol{F}'\boldsymbol{g}$
$\boldsymbol{\Sigma}_h = v_\epsilon(\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}$
$\boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}^{(k-1)}) \quad \mathrm{Tr}\left\{\boldsymbol{F}\boldsymbol{\Sigma}_h\boldsymbol{F}'\right\} = \|\boldsymbol{D}_f'\boldsymbol{f}\|_2^2$

JMAP and VBA with Gaussian priors

JMAP: $\widehat{\boldsymbol{h}}^{(0)} \rightarrow \widehat{\boldsymbol{h}} \rightarrow$

$$\boxed{\begin{array}{l} \boldsymbol{H} = \texttt{Convmtx}(\boldsymbol{h}) \\ \widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g} \end{array}}$$

$\rightarrow \widehat{\boldsymbol{f}}$

$\uparrow$ $\downarrow$

$\widehat{\boldsymbol{h}} \leftarrow$

$$\boxed{\begin{array}{l} \boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}) \\ \widehat{\boldsymbol{h}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{C}_h'\boldsymbol{C}_h)^{-1}\boldsymbol{F}'\boldsymbol{g} \end{array}}$$

$\leftarrow \widehat{\boldsymbol{f}}$

VBA: $\widehat{\boldsymbol{h}}^{(0)}, \widehat{\boldsymbol{\Sigma}}_h^{(0)} \Rightarrow \widehat{\boldsymbol{h}}, \widehat{\boldsymbol{\Sigma}}_h \Rightarrow$

$$\boxed{\begin{array}{l} \boldsymbol{H} = \texttt{Convmtx}(\widehat{\boldsymbol{h}}) \\ \text{Tr}\left\{\boldsymbol{F}\boldsymbol{\Sigma}_h\boldsymbol{F}'\right\} = \|\boldsymbol{D}_f\boldsymbol{f}\|_2^2 \\ \widehat{\boldsymbol{\Sigma}}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I} + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1} \\ \widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{I} + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1}\boldsymbol{H}'\boldsymbol{g} \end{array}}$$

$\Rightarrow \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\Sigma}}_f$

$\Uparrow$ $\Downarrow$

$\widehat{\boldsymbol{h}}, \widehat{\boldsymbol{\Sigma}}_h \Leftarrow$

$$\boxed{\begin{array}{l} \boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}) \\ \text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_h\boldsymbol{h}\|_2^2 \\ \widehat{\boldsymbol{h}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}\boldsymbol{F}'\boldsymbol{g} \\ \widehat{\boldsymbol{\Sigma}}_h = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1} \end{array}}$$

$\Leftarrow \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\Sigma}}_f$

Student-t prior

- Sparsity enforcing: Heavy tailed priors: Cauchy and Student-t:

$$p(\boldsymbol{f}) = \prod_j \mathcal{T}(f_j|\nu, v_f)$$

Infinite scaled mixture representation:

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

- Hierarchical model: $p(\boldsymbol{f}|\boldsymbol{z})\, p(\boldsymbol{z})$

$$p(\boldsymbol{f}|\boldsymbol{z}) = \prod_j \mathcal{N}(f_j|0, z_j^{-1} v_f), \quad p(\boldsymbol{z}) = \prod_j \mathcal{G}(z_j|\alpha, \beta)$$

- Joint posterior: $p(\boldsymbol{h}, \boldsymbol{f}, \boldsymbol{z}|\boldsymbol{g}) \propto p(\boldsymbol{g}|\boldsymbol{h}, \boldsymbol{f})\, p(\boldsymbol{f}|\boldsymbol{z})\, p(\boldsymbol{z})$
  - JMAP: $(\widehat{\boldsymbol{h}}, \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{z}}) = \arg\max_{(\boldsymbol{h}, \boldsymbol{f}, \boldsymbol{z})} \{p(\boldsymbol{h}, \boldsymbol{f}, \boldsymbol{z}|\boldsymbol{g})\}$
  - VBA: Approximate $p(\boldsymbol{h}, \boldsymbol{f}, \boldsymbol{z}|\boldsymbol{g})$ by $q_1(\boldsymbol{h})\, q_2(\boldsymbol{f})\, q_3(\boldsymbol{z})$

JMAP and VBA with a Student-t prior

- Because we only changed $p(\boldsymbol{f})$ by $p(\boldsymbol{f}|\boldsymbol{z})\,p(\boldsymbol{z})$ where both are separable, the only changes are:
  - replace
$$\boldsymbol{f}^{(k)} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{I})^{-1}\boldsymbol{H}'\boldsymbol{g}$$
  - by:

$$\boldsymbol{f}^{(k)} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f \widehat{\boldsymbol{Z}}^{(k)})^{-1}\boldsymbol{H}'\boldsymbol{g}$$
$$\widehat{\boldsymbol{Z}}^{(k)} = \text{diag}\left[\widehat{z}_j^{(k)}\right], \quad \widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j}$$

with

$$JMAP \qquad\qquad VBA$$
$$\begin{cases} \widehat{\alpha}_j = \alpha + \frac{1}{2}\|\boldsymbol{f}\|^2 \\ \text{and} \\ \widehat{\beta}_j = \beta + \frac{1}{2} + \|\boldsymbol{g} - \boldsymbol{h} * \boldsymbol{f}\|^2 \end{cases} \qquad \begin{cases} \widehat{\alpha}_j = \alpha + \frac{1}{2} < \|\boldsymbol{f}\|^2 > \\ \text{and} \\ \widehat{\beta}_j = \beta + \frac{1}{2} + < \|\boldsymbol{g} - \boldsymbol{h} * \boldsymbol{f}\|^2 > \end{cases}$$

Comparison between JMAP and VBA

Alternate optimization for JMAP

$$\widehat{h}^{(0)}, \widehat{z}^{(0)} \Rightarrow \widehat{h}, \widehat{z} \Rightarrow \boxed{\begin{array}{l} Z = \text{diag}\,[\widehat{z}], \quad H = \texttt{Convmtx}(\widehat{h}) \\ \widehat{f} = (H'H + \lambda_f Z)^{-1} H'g \end{array}} \rightarrow \widehat{f}$$

$$\Uparrow \qquad \qquad \downarrow$$

$$\widehat{h}, z \Leftarrow \boxed{\begin{array}{l} \widehat{\alpha}_j = \alpha + \frac{1}{2}\|f\|^2 \\ \widehat{\beta}_j = \beta + \frac{1}{2} + \|g - h * f\|^2 \\ \widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j} \\ F = \texttt{Convmtx}(f) \\ \widehat{h} = (F'F + \lambda_h C_h' C_h)^{-1} F'g \end{array}} \leftarrow \widehat{f}$$

Comparison between JMAP and VBA

VBA

$$\widehat{\boldsymbol{h}}^{(0)} \quad \boldsymbol{\Sigma}_h^{(0)} \quad \widehat{\boldsymbol{z}}^{(0)} \Rightarrow \widehat{\boldsymbol{h}} \quad \boldsymbol{\Sigma}_h \quad \widehat{\boldsymbol{z}} \Rightarrow$$

$$\begin{array}{l} \boldsymbol{Z} = \text{diag}\left[\widehat{\boldsymbol{z}}\right], \quad \boldsymbol{H} = \texttt{Convmtx}(\widehat{\boldsymbol{h}}) \\ \text{Tr}\left\{\boldsymbol{F}\boldsymbol{\Sigma}_h\boldsymbol{F}'\right\} = \|\boldsymbol{D}_h\boldsymbol{f}\|^2 \\ \widehat{\boldsymbol{\Sigma}}_f = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{Z} + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1} \\ \widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H} + \lambda_f\boldsymbol{Z} + \boldsymbol{D}_f'\boldsymbol{D}_f)^{-1}\boldsymbol{H}'\boldsymbol{g} \end{array}$$

$$\Rightarrow \widehat{\boldsymbol{f}} \quad \widehat{\boldsymbol{\Sigma}}_f$$

$$\Uparrow \qquad \Downarrow$$

$$\widehat{\boldsymbol{h}} \quad \widehat{\boldsymbol{\Sigma}}_h \quad \widehat{\boldsymbol{z}} \Leftarrow$$

$$\begin{array}{l} \widehat{\alpha}_j = \alpha + \frac{1}{2} < \|\boldsymbol{f}\|^2 > \\ \widehat{\beta}_j = \beta + \frac{1}{2} + < \|\boldsymbol{g} - \boldsymbol{h} * \boldsymbol{f}\|^2 > \\ \widehat{z}_j^{(k)} = \frac{\widehat{\beta}_j}{\widehat{\alpha}_j} \\ \boldsymbol{F} = \texttt{Convmtx}(\boldsymbol{f}) \\ \text{Tr}\left\{\boldsymbol{H}\boldsymbol{\Sigma}_f\boldsymbol{H}'\right\} = \|\boldsymbol{D}_h\boldsymbol{h}\|^2 \\ \widehat{\boldsymbol{h}} = (\boldsymbol{F}'\boldsymbol{F} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h + \boldsymbol{D}_h'\boldsymbol{D}_h)^{-1}\boldsymbol{F}'\boldsymbol{g} \\ \widehat{\boldsymbol{\Sigma}}_h = v_\epsilon(\boldsymbol{H}'\boldsymbol{H} + \lambda_h\boldsymbol{C}_h'\boldsymbol{C}_h)^{-1} \end{array}$$

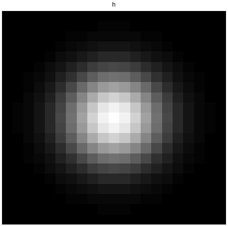
$$\Leftarrow \widehat{\boldsymbol{f}} \quad \widehat{\boldsymbol{\Sigma}}_f$$

Conclusions

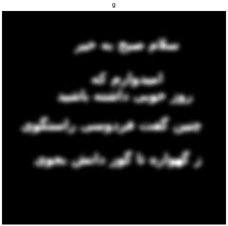
- In this paper, we considered the Blind Image Deconvolution problem in a Bayesian framework.
- We compared two main algorithms: JMAP and VBA giving some detailed insight for each of them for two cases:
  - Gaussian prior for both IRF $\boldsymbol{h}$ and the input signal $\boldsymbol{f}$ and
  - Gaussian prior for the IRF but a Student-t prior for the input signals or images to enhance or to account for possible sparsity structure of the input.
- JMAP: esay and not very costly but uncertainties are not accounted for.
- VBA: is more costly but both uncertainties of $\widehat{\boldsymbol{f}}$ and $\widehat{\boldsymbol{h}}$ are accounted for, in each iteration, for computing respectively, $\widehat{\boldsymbol{h}}$ and $\widehat{\boldsymbol{f}}$.

Simulated results

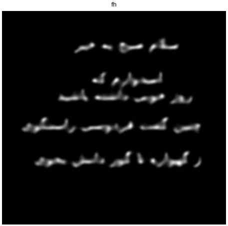
original f



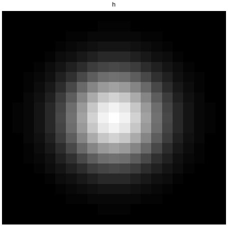
PSF h



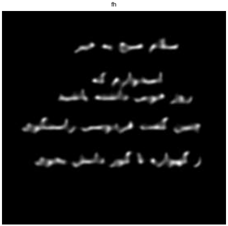
Blurred & noisy g



simple deconv $\widehat{f}$



Estimated PSF $\widehat{h}$



Estimated $\widehat{f}$

THANKS

Questions and Remarks