Sensors, Measurement systems and Inverse problems

DECONVOLUTION

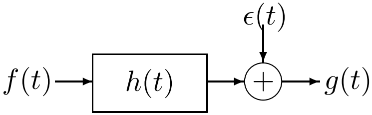
Ali Mohammad-Djafari
Laboratoire des Signaux et Systèmes,
UMR8506 CNRS-SUPELEC-UNIV PARIS SUD 11
SUPELEC, 91192 Gif-sur-Yvette, France
http://lss.supelec.free.fr

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

Case study: Signal deconvolution

- Convolution, Identification and Deconvolution
- Forward and Inverse problems: Well-posedness and Ill-posedness
- Naîve methods of Deconvolution
- Classical methods: Wiener filtering
- Bayesian approach to deconvolution
- Simple and Blind Deconvolution
- Deterministic and probabilistic methods
- Joint source and canal estimation

Convolution, Identification and deconvolution



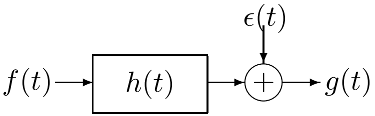
$$g(t) = \int f(t')\, h(t-t')\, \mathrm{d}t' + \epsilon(t) = \int h(t')\, f(t-t')\, \mathrm{d}t' + \epsilon(t)$$

- 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

Convolution: Given f and h compute g

- Direct computation: `g=conv(h,f)`
- Fourier domain: $g(t) = h(t) * f(t) \longrightarrow G(\omega) = H(\omega)F(\omega)$
  - Compute $H(\omega)$, $F(\omega)$ and $G(\omega) = H(\omega)F(\omega)$
  - Compute $g(t)$ by inverse FT of $G(\omega)$
- Take care of dimensions and boarder effects.

Convolution: Discretization



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

- The signals $f(t)$, $g(t)$, $h(t)$ are discretized with the same sampling period $\Delta T = 1$,
- The impulse response is finite (FIR) : $h(t) = 0$, for t such that $t < -q\Delta T$ or $\forall t > p\Delta T$.

$$g(m) = \sum_{k=-q}^{p} h(k)\, f(m-k) + \epsilon(m), \quad m = 0, \cdots, M$$

Convolution: Discretized matrix vector forms

$$
\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix}
=
\begin{bmatrix}
h(p) & \cdots & h(0) & \cdots & h(-q) & 0 & \cdots & \cdots & \cdots & 0 \\
0 & \ddots & & \ddots & & \ddots & & & & \vdots \\
\vdots & & & & & & & & & \vdots \\
\vdots & & h(p) & \cdots & h(0) & \cdots & h(-q) & & & \vdots \\
\vdots & & & & & & & & & \vdots \\
\vdots & & & \ddots & & & \ddots & & & \vdots \\
\vdots & & & & \ddots & & & & & 0 \\
0 & \cdots & \cdots & & 0 & h(p) & \cdots & h(0) & \cdots & h(-q)
\end{bmatrix}
\begin{bmatrix} f(-p) \\ \vdots \\ f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \\ f(M+1) \\ \vdots \\ f(M+q) \end{bmatrix}
$$

$$
\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}
$$

- $\boldsymbol{g}$ is a $(M+1)$-dimensional vector,
- $\boldsymbol{f}$ has dimension $M+p+q+1$,
- $\boldsymbol{h} = [h(p), \cdots, h(0), \cdots, h(-q)]$ has dimension $(p+q+1)$
- $\boldsymbol{H}$ has dimensions $(M+1) \times (M+p+q+1)$.

Convolution: Discretized matrix vector form

- If system is causal $(q=0)$ we obtain

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(p) & \cdots & h(0) & 0 & \cdots & \cdots & 0 \\ 0 & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ \vdots & & h(p) & \cdots & h(0) & & \vdots \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & 0 \\ 0 & \cdots & \cdots & 0 & h(p) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} f(-p) \\ \vdots \\ f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \end{bmatrix}$$

- $\boldsymbol{g}$ is a $(M+1)$-dimensional vector,
- $\boldsymbol{f}$ has dimension $M+p+1$,
- $\boldsymbol{h} = [h(p), \cdots, h(0)]$ has dimension $(p+1)$
- $\boldsymbol{H}$ has dimensions $(M+1) \times (M+p+1)$.

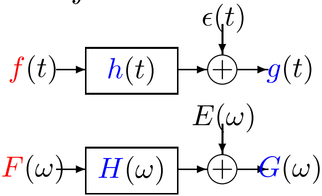
Convolution: Causal systems and causal input

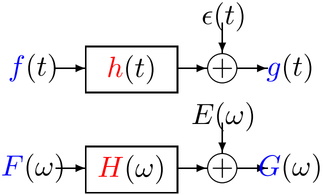
$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} h(0) & & & & & \\ h(1) & \ddots & & & & \\ \vdots & & & & & \\ h(p) & \cdots & & h(0) & & \\ 0 & \ddots & & & \ddots & \\ \vdots & & & & & \\ 0 & \cdots & 0 & h(p) & \cdots & h(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(M) \end{bmatrix}$$

- $\boldsymbol{g}$ is a $(M+1)$-dimensional vector,
- $\boldsymbol{f}$ has dimension $M+1$,
- $\boldsymbol{h} = [h(p), \cdots, h(0)]$ has dimension $(p+1)$
- $\boldsymbol{H}$ has dimensions $(M+1) \times (M+1)$.

Convolution, Identification, Deconvolution and Blind deconvolution problems

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





$$G(\omega) = H(\omega)\, F(\omega) + E(\omega) \qquad G(\omega) = H(\omega)\, F(\omega) + E(\omega)$$

$$F(\omega) = \frac{G(\omega)}{H(\omega)} + \frac{E(\omega)}{H(\omega)} \qquad H(\omega) = \frac{G(\omega)}{F(\omega)} + \frac{E(\omega)}{F(\omega)}$$

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

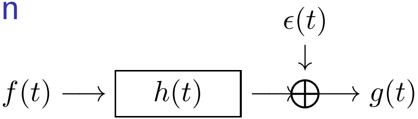
Deconvolution: Given g and h estimate f

- Direct computation: `f=deconv(g,h)`
- Fourier domain: Inverse Filtering $F(\omega) = \frac{G(\omega)}{H(\omega)}$
  - Compute $H(\omega)$, $G(\omega)$ and $F(\omega) = \frac{G(\omega)}{H(\omega)}$
  - Compute $g(t)$ by inverse FT of $F(\omega)$
- Main difficulties: Divide by zero and noise amplification

Identification: Given g and f estimate h

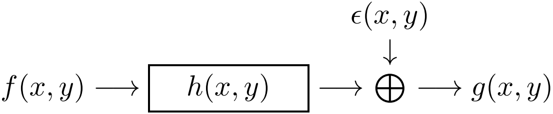
- Direct computation:
  - $f(t) = \delta(t) \longrightarrow g(t) = h(t) \longrightarrow h(t) = g(t)$
  - $f(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases} \longrightarrow g(t) = \int_0^t h(t)\, \mathrm{d}t \longrightarrow h(t) = \frac{\partial g(t)}{\partial t}$
- Fourier domain: Inverse Filtering $H(\omega) = \frac{G(\omega)}{F(\omega)}$
  - Compute $F(\omega)$, $G(\omega)$ and $H(\omega) = \frac{G(\omega)}{F(\omega)}$
  - Compute $h(t)$ by inverse FT of $H(\omega)$
- Main difficulties: Divide by zero and noise amplification

Convolution in 1D and 2D: Signal deconvolution and Image restoration



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

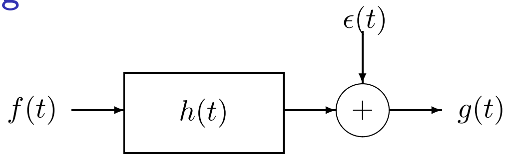
- $f(t)$, $g(t)$ and $\epsilon(t)$ are modelled as Gaussian random signal



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

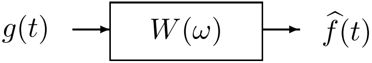
- $f(x,y)$, $g(x,y)$ and $\epsilon(x,y)$ are modelled as homogeneous and Gaussian random fields

Wiener Filtering



$$
\begin{aligned}
\mathsf{E}\{g(t)\} &= h(t) * \mathsf{E}\{f(t)\} + \mathsf{E}\{\epsilon(t)\} \\
R_{gg}(\tau) &= h(t) * h(t) * R_{ff}(\tau) + R_{\epsilon\epsilon}(\tau) \\
R_{gf}(\tau) &= h(t) * R_{ff}(\tau)
\end{aligned}
$$

$$
\begin{aligned}
S_{gg}(\omega) &= |H(\omega)|^2 S_{ff}(\omega) + R_{\epsilon\epsilon}(\omega) \\
S_{gf}(\omega) &= H(\omega) S_{ff}(\omega) \\
S_{fg}(\omega) &= H^*(\omega) S_{ff}(\omega)
\end{aligned}
$$



$$\widehat{f}(t) = w(t) * g(t)$$

Wiener Filtering

$$EQM = \mathsf{E}\left\{[f(t) - \widehat{f}(t)]^2\right\} = \mathsf{E}\left\{[f(t) - w(t) * g(t)]^2\right\}$$

$$\frac{\partial EQM}{\partial f} = -2\mathsf{E}\left\{[f(t) - w(t) * g(t)] * g(t+\tau)\right\} = 0$$

$$\mathsf{E}\left\{[f(t) - w(t) * g(t)]\, g(t+\tau)\right\} = 0 \quad \forall t, \tau \longrightarrow$$

$$R_{fg}(\tau) = w(t) * R_{gg}(\tau)$$

$$W(\omega) = \frac{S_{fg}(\omega)}{S_{gg}(\omega)} = \frac{H^*(\omega)\, S_{ff}(\omega)}{|H(\omega)|^2\, S_{ff}(\omega) + S_{\epsilon\epsilon}(\omega)}$$

$$\boxed{W(\omega) = \frac{H^*(\omega) S_{ff}(\omega)}{|H(\omega)|^2 S_{ff}(\omega) + S_{\epsilon\epsilon}(\omega)} = \frac{1}{H(\omega)} \frac{|H(\omega)|^2}{|H(\omega)|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{ff}(\omega)}}}$$

Wiener Filtering

- Linear Estimation: $\widehat{f}(x,y)$ is such that:
  - $\widehat{f}(x,y)$ depends on $g(x,y)$ in a linear way:

$$\widehat{f}(x,y) = \iint g(x',y')\, w(x-x',y-y')\, \mathrm{d}x'\, \mathrm{d}y'$$

    $w(x,y)$ is the impulse response of the Wiener filtre
  - minimizes MSE: $\mathrm{E}\left\{|f(x,y) - \widehat{f}(x,y)|^2\right\}$
- Orthogonality condition:

$$(f(x,y)-\widehat{f}(x,y)) \perp g(x',y') \quad \longrightarrow \quad \mathrm{E}\left\{(f(x,y) - \widehat{f}(x,y))\, g(x',y')\right\} = 0$$

$$\widehat{f} = g * w \;\longrightarrow\; \mathrm{E}\left\{(f(x,y) - g(x,y) * w(x,y))\, g(x+\alpha_1, y+\alpha_2)\right\} = 0$$

$$R_{fg}(\alpha_1,\alpha_2) = (R_{gg} * w)(\alpha_1,\alpha_2) \longrightarrow \mathrm{TF} \longrightarrow S_{fg}(u,v) = S_{gg}(u,v) W(u,v)$$

$$\Downarrow$$

$$\boxed{W(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(u,v)}}$$

Wiener filtering

Signal	Image
$W(\omega) = \frac{S_{fg}(\omega)}{S_{gg}(\omega)}$	$W(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(u,v)}$

Particular Case:
$f(x,y)$ and $b(x,y)$ are assumed to be centered and non correlated

$$S_{fg}(u,v) = H'(u,v)\, S_{ff}(u,v)$$

$$S_{gg}(u,v) = |H(u,v)|^2\, S_{ff}(u,v) + S_{\epsilon\epsilon}(u,v)$$

$$W(u,v) = \frac{H'(u,v)S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + S_{\epsilon\epsilon}(u,v)}$$

Signal	Image
$W(\omega) = \frac{1}{H(\omega)} \frac{\|H(\omega)\|^2}{\|H(\omega)\|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{ff}(\omega)}}$	$W(u,v) = \frac{1}{H(u,v)} \frac{\|H(u,v)\|^2}{\|H(u,v)\|^2 + \frac{S_{\epsilon\epsilon}(u,v)}{S_{ff}(u,v)}}$

Convolution: Discretization for Identification

Causal systems and causal input

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g(M) \end{bmatrix} = \begin{bmatrix} 0 & & . & 0 & f(0) \\ . & & . & f(0) & f(1) \\ . & & f(0) & f(1) & \vdots \\ . & . & . & . & \vdots \\ f(0) & f(1) & & & f(M-p) \\ f(1) & . & & . & . \\ . & & & . & . \\ . & & . & & . \\ . & . & & & . \\ f(M-p) & . & . & . & f(M) \end{bmatrix} \begin{bmatrix} h(p) \\ h(p-1) \\ \vdots \\ \vdots \\ h(1) \\ h(0) \end{bmatrix}$$

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

- $\boldsymbol{g}$ is a $(M+1)$-dimensional vector,
- $\boldsymbol{F}$ has dimension $(M+1) \times (p+1)$,

Algebraic Approches

Signal	Image
$f(t) \longrightarrow \boxed{h(t)} \longrightarrow g(t)$	$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$

Discretization

$$\Downarrow$$

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f}$$

- **Ideal case:** $\boldsymbol{H}$ invertible $\longrightarrow \widehat{\boldsymbol{f}} = \boldsymbol{H}^{-1}\boldsymbol{g}$
- $M > N$ **Least Squares**:

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}$$

$$e = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 = [\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}]'[\boldsymbol{g} - \boldsymbol{H}\widehat{\boldsymbol{f}}]$$

$$\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \{e\}$$

$$\nabla e = -2\boldsymbol{H}'[\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}] = 0 \longrightarrow \boldsymbol{H}'\boldsymbol{H}\boldsymbol{f} = \boldsymbol{H}'\boldsymbol{g}$$

  - If $\boldsymbol{H}'\boldsymbol{H}$ is invertible $\widehat{\boldsymbol{f}} = (\boldsymbol{H}'\boldsymbol{H})^{-1}\boldsymbol{H}'\boldsymbol{g}$

Algebraic Approches: Generalized Inversion

General case of $[M, N]$ matrix $\boldsymbol{H}$:

- if $M = N$ and rang $\{\boldsymbol{H}\} = N$ then $\boldsymbol{H}^{+} = \boldsymbol{H}^{-1}$
- if $M > N$ and rang $\{\boldsymbol{H}\} = N$ then $\boldsymbol{H}^{+} = (\boldsymbol{H}'\boldsymbol{H})^{-1}\boldsymbol{H}'$
- if $M < N$ and rang $\{\boldsymbol{H}\} = M$ then $\boldsymbol{H}^{+} = \boldsymbol{H}'(\boldsymbol{H}\boldsymbol{H}')^{-1}$
- if rang $\{\boldsymbol{H}\} = K < \inf M, N$ then
  - Singular Value Decomposition (SVD)
  - Iterative methods
  - Recursive methods

Regularization

$$J_\lambda(\boldsymbol{f}) = [\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g}]'[\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g}]+\lambda[\boldsymbol{D}\boldsymbol{f}]'[\boldsymbol{D}\boldsymbol{f}] = \|\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g}\|^2+\lambda\|\boldsymbol{D}\boldsymbol{f}\|^2$$

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & \cdots & & \cdots & 0 \\ -1 & 1 & \ddots & & & \vdots \\ 0 & -1 & 1 & \ddots & & \vdots \\ & 0 & -1 & 1 & \ddots & \\ & & & & & \\ 0 & & & 0 & -1 & 1 \end{bmatrix} \quad \text{or} \quad \boldsymbol{D} = \begin{bmatrix} 1 & 0 & \cdots & & \cdots & 0 \\ -2 & 1 & \ddots & & & \vdots \\ 1 & -2 & 1 & \ddots & & \vdots \\ & 1 & -2 & 1 & \ddots & \\ & & & & & \\ 0 & & & 1 & -2 & 1 \end{bmatrix}$$

$$\nabla J_\lambda = 2\boldsymbol{H}'[\boldsymbol{H}\boldsymbol{f}-\boldsymbol{g}]' + 2\lambda\boldsymbol{D}'\boldsymbol{D}\boldsymbol{f} = 0$$

$$[\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{D}'\boldsymbol{D}]\widehat{\boldsymbol{f}} = \boldsymbol{H}'\boldsymbol{g} \longrightarrow \widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{D}'\boldsymbol{D}]^{-1}\boldsymbol{H}'\boldsymbol{g}$$

Regularization Algorithmes

$$\text{minimize} \quad J(\boldsymbol{f}) = Q(\boldsymbol{f}) + \lambda \Omega(\boldsymbol{f})$$

$$\text{with} \quad Q(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 = [\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}]'[\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}]$$

$$=$$

minimize $\Omega(\boldsymbol{f})$ subj. to the constraint

$$e = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 = [\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}]'[\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}] < \epsilon$$

A priori Information:

- Smoothnesse

$$\Omega(\boldsymbol{f}) = [\boldsymbol{D}\boldsymbol{f}]'[\boldsymbol{D}\boldsymbol{f}] = \|\boldsymbol{D}\boldsymbol{f}\|^2$$

$$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{D}'\boldsymbol{D}]^{-1}\boldsymbol{H}'\boldsymbol{g}$$

- Positivity:
  $\Omega(\boldsymbol{f}) =$ a nonquadratique function of $\boldsymbol{f}$
  No explicite solution

Regularization Algorithmes: 3 main approaches

$$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{D}'\boldsymbol{D}]^{-1}\boldsymbol{H}'\boldsymbol{g}$$

Computation of $\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{D}'\boldsymbol{D}]^{-1}\boldsymbol{H}'\boldsymbol{g}$

- Circulante matrix approximation:
  when $\boldsymbol{H}$ and $\boldsymbol{D}$ are Toeplitz, they can be approximated by the circulant matrices
- Iterative methods:

$$\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \left\{ \| J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 + \lambda \|\boldsymbol{D}\boldsymbol{f}\|^2 \right\}$$

- Recursive methods:
  $\widehat{\boldsymbol{f}}$ at iteration k is computed as a function of $\widehat{\boldsymbol{f}}$ at previous iteration with one less data.

Regularization algorithms: Circulant approximation

1D Deconvolution:

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

$\boldsymbol{H}$ Toeplitz matrix

$$\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \{\boldsymbol{f}\}\, J(\boldsymbol{f}) = Q(\boldsymbol{f}) + \lambda\Omega(\boldsymbol{f})$$

$Q(\boldsymbol{f}) = \|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|^2 = [\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}]'[\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}]$ and $\Omega(\boldsymbol{f}) = \|\boldsymbol{D}\boldsymbol{f}\|^2 = [\boldsymbol{D}\boldsymbol{f}]'[\boldsymbol{D}$

$\boldsymbol{C}$ a convolution matrix with the following impulse response

$$\boldsymbol{h}_1 = [1, -2, 1] \quad \longrightarrow \quad x(i) = x(i+1) - 2x(i) + x(i-1)$$

$$\Omega(\boldsymbol{f}) = \sum_{j=1}^{N} (x(i+1) - 2x(i) + x(i-1))^2 = \|\boldsymbol{D}\boldsymbol{f}\|^2 = \boldsymbol{f}'\boldsymbol{D}'\boldsymbol{D}\boldsymbol{f}$$

Solution :

$$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{C}'\boldsymbol{C}]^{-1} H'\boldsymbol{g}$$

Regularization algorithms: Circulant approximation

Main Idea : expand the vectors $\boldsymbol{f}$, $\boldsymbol{h}$ and $\boldsymbol{g}$ by the zeros to obtain $\boldsymbol{g}_e = \boldsymbol{H}_e \boldsymbol{f}_e$ with $\boldsymbol{H}_e$ a circulante matrix

$$f_e(i) = \begin{cases} f(i) & i = 1, \ldots, N \\ 0 & i = N+1, \ldots, P \geq N + Nh - 1 \end{cases}$$

$$g_e(i) = \begin{cases} g(i) & i = 1, \ldots, M \\ 0 & i = M+1, \ldots, P \end{cases}$$

$$h_e(i) = \begin{cases} h(i) & i = 1, \ldots, Nh \\ 0 & i = Nh+1, \ldots, P \end{cases}$$

$$g_e(k) = \sum_{i=0}^{Nh-1} f_e(k-i) h_e(i) \quad \longrightarrow \quad \boldsymbol{g}_e = \boldsymbol{H}_e \boldsymbol{f}_e$$

with $\boldsymbol{H}_e$ a circulante matrix whcich can diagonalized by FFT

Regularization algorithms: Circulant approximation

$$\boldsymbol{H}_e = \boldsymbol{F}\Lambda\boldsymbol{F}^{-1} \text{ with } \boldsymbol{F}[k,l] = \exp\left\{j2\pi\frac{kl}{P}\right\} \quad \boldsymbol{F}^{-1}[k,l] = \frac{1}{P}\exp\left\{-j2\pi\frac{kl}{P}\right\}$$

$$\boldsymbol{\Lambda} = \text{diag}[\lambda_1,\ldots,\lambda_P] \text{ and } [\lambda_1,\ldots,\lambda_P] = \text{TFD } [h_1,\ldots,h_{Nh},0,\ldots,0]$$

$$d = [1,-2,1] \quad d_e(i) = \begin{cases} d(i) & i = 1,\ldots,3 \\ 0 & i = 4,\ldots,P \end{cases}$$

$$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{D}'\boldsymbol{D}]^{-1}\boldsymbol{H}'\boldsymbol{g} \longrightarrow \boldsymbol{F}\widehat{\boldsymbol{f}}_e = [\Lambda_h'\Lambda_h + \lambda\Lambda_d'\Lambda_d]^{-1}\Lambda_h'\boldsymbol{F}\boldsymbol{g}$$

$$\text{TFD } \{\boldsymbol{f}_e\} = [\Lambda_h'\Lambda_h + \lambda\Lambda_d'\Lambda_d]^{-1}\Lambda_h' \text{TFD } \{\boldsymbol{g}\}$$

$$\widehat{\boldsymbol{f}}(\omega) = \frac{1}{H(\omega)}\frac{|H(\omega)|^2}{|H(\omega)|^2 + \lambda|D(\omega)|^2}\, y(\omega)$$

Link with Wiener filter: $D(\omega) = E(\omega)/F(\omega)$

Image Restoration

$\boldsymbol{C}$ Convolution matrix with the following impulse response:

$$H_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} \Omega(\boldsymbol{f}) = \sum\sum \quad & (f(i+1,j) + f(i-1,j) \\ & + f(i+1,j+1) + f(i-1,j+1) - 4f(i,j))^2 \end{aligned}$$

$$f_e(k,l) = \begin{cases} f(k,l) & k = 1,\ldots,K \qquad\quad l = 1,\ldots,L \\ 0 & k = K+1,\ldots,P \quad l = L+1,\ldots,P \end{cases}$$

Regularization: Iterative methods: Gradient based

$$\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \{J(\boldsymbol{f}) = Q(\boldsymbol{f}) + \lambda\Omega(\boldsymbol{f})\}$$

Let note : $\boldsymbol{g}^k = \nabla J\left(\boldsymbol{f}^k\right)$ gradient, $\boldsymbol{H}^k = \nabla^2 J\left(\boldsymbol{f}^k\right)$ Hessien.

First order gradient methods

- fixed step:

$$\boldsymbol{f}^{(k+1)} = \boldsymbol{f}^{(k)} + \alpha \boldsymbol{g}^{(k)} \qquad \alpha \quad \text{fixe}$$

- Optimal or steepest descente step:

$$\boldsymbol{f}^{(k+1)} = \boldsymbol{f}^{(k)} + \alpha^{(k)} \boldsymbol{g}^{(k)}$$

$$\alpha^{(k)} = -\frac{\boldsymbol{g}^{(k)t}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)t}\boldsymbol{H}^k\boldsymbol{g}^{(k)}} = \frac{\|\boldsymbol{g}^k\|^2}{\|\boldsymbol{g}^k\|_H^2}$$

Regularization: Iterative methods: Conjugate Gradient

- **Conjugate Gradient (CG)**

$$\boldsymbol{f}^{(k+1)} = \boldsymbol{f}^{(k)} + \alpha^{(k)} \boldsymbol{d}^{(k)} \quad \alpha^{(k)} = -\frac{\boldsymbol{d}^{(k)t} \boldsymbol{g}^{(k)}}{\boldsymbol{d}^{(k)t} \boldsymbol{H}^{k} \boldsymbol{d}^{(k)}}$$

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{d}^{(k)} + \beta^{(k)} \boldsymbol{g}^{(k)} \quad \beta^{(k)} = -\frac{\boldsymbol{g}^{(k)t} \boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k-1)t} \boldsymbol{g}^{(k-1)}}$$

- **Newton method**

$$\boldsymbol{f}^{(k+1)} = \boldsymbol{f}^{(k)} + (\boldsymbol{H}^{(k)})^{-1} \boldsymbol{g}^{(k)}$$

- Advantages : $\Omega(\boldsymbol{f})$ can be any convexe function
- Limitations : Computational cost

Regularization: Recursive algorithms

$$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{D}]^{-1}\boldsymbol{H}'\boldsymbol{g}$$

Main idea: Express $\boldsymbol{f}_{i+1}$ as a function of $\boldsymbol{f}_i$

$$\boldsymbol{f}_{i+1} = (\boldsymbol{H}'_{i+1}\boldsymbol{H}_{i+1} + \alpha \boldsymbol{D})^{-1}\boldsymbol{H}'_{i+1}\boldsymbol{g}_{i+1}$$

$$\boldsymbol{f}_{i} = (\boldsymbol{H}^t_{i}\boldsymbol{H}_{i} + \alpha \boldsymbol{D})^{-1}\boldsymbol{H}^t_{i}\boldsymbol{g}_{i}$$

$$\Downarrow$$

$$\boldsymbol{f}_{i+1} = (\boldsymbol{H}^t_{i}\boldsymbol{H}_{i} + \boldsymbol{h}_{i+1}\boldsymbol{h}'_{i+1} + \alpha \boldsymbol{D})^{-1}(\boldsymbol{H}^t_{i}\boldsymbol{g}_{i} - \boldsymbol{h}_{i+1}\boldsymbol{g}_i + 1)$$

Noting:

$$\boldsymbol{P}_i = (\boldsymbol{H}^t_{i}\boldsymbol{H}_{i} + \alpha \boldsymbol{D})^{-1} \quad \text{and} \quad \boldsymbol{P}^t_{i+1} = \boldsymbol{P}^t_i + \boldsymbol{h}_{i+1}\boldsymbol{h}'_{i+1}$$

$$\Downarrow$$

$$\boldsymbol{f}_{i+1} = \boldsymbol{f}_i + \boldsymbol{P}_{i+1}\boldsymbol{h}_{i+1}(\boldsymbol{g}_{i+1} - \boldsymbol{h}'_{i+1}\boldsymbol{f}_i)$$

$$\boldsymbol{P}_{i+1} = \boldsymbol{P}_i - \boldsymbol{P}_i\boldsymbol{h}_{i+1}(\boldsymbol{h}'_{i+1}\boldsymbol{P}_i\boldsymbol{H}_{i+1} + \alpha)^{-1}\boldsymbol{h}'_{i+1}\boldsymbol{P}_i$$

Identification and Deconvolution

Deconvolution

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

$$J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 + \lambda_f \|\boldsymbol{D}_f \boldsymbol{f}\|^2$$
$$\nabla J(\boldsymbol{f}) = -2\boldsymbol{H}'(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}) + 2\lambda_f \boldsymbol{D}_f' \boldsymbol{D}_f \boldsymbol{f}$$
$$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{D}_f'\boldsymbol{D}_f]^{-1}\boldsymbol{H}'\boldsymbol{g}$$
$$\widehat{f}(\omega) = \frac{H^*(\omega)}{|H(\omega)|^2 + \lambda_f |D_f(\omega)|^2}\, g(\omega)$$

$$\widehat{f}(\omega) = \frac{H^*(\omega)}{|H(\omega)|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{ff}(\omega)}}\, g(\omega)$$

$$p(\boldsymbol{g}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{H}\boldsymbol{f}, \boldsymbol{\Sigma}_\epsilon)$$
$$p(\boldsymbol{f}) = \mathcal{N}(0, \boldsymbol{\Sigma}_f)$$

$$p(\boldsymbol{f}|\boldsymbol{g}) = \mathcal{N}(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\Sigma}}_f)$$
$$\widehat{\boldsymbol{\Sigma}}_f = [\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{D}_f'\boldsymbol{D}_f]^{-1}$$
$$\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{D}_f'\boldsymbol{D}_f]^{-1}\boldsymbol{H}'\boldsymbol{g}$$

Identification

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

$$J(\boldsymbol{h}) = \|\boldsymbol{g} - \boldsymbol{F}\boldsymbol{h}\|^2 + \lambda_h \|\boldsymbol{D}_h \boldsymbol{h}\|^2$$
$$\nabla J(\boldsymbol{h}) = -2\boldsymbol{F}'(\boldsymbol{g} - \boldsymbol{F}\boldsymbol{h}) + 2\lambda_h \boldsymbol{D}_h' \boldsymbol{D}_h \boldsymbol{h}$$
$$\widehat{\boldsymbol{h}} = [\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{D}_h'\boldsymbol{D}_h]^{-1}\boldsymbol{F}'\boldsymbol{g}$$
$$\widehat{h}(\omega) = \frac{F^*(\omega)}{|F(\omega)|^2 + \lambda_h |D_h(\omega)|^2}\, g(\omega)$$

$$\widehat{h}(\omega) = \frac{F^*(\omega)}{|F(\omega)|^2 + \frac{S_{\epsilon\epsilon}(\omega)}{S_{hh}(\omega)}}\, g(\omega)$$

$$p(\boldsymbol{g}|\boldsymbol{h}) = \mathcal{N}(\boldsymbol{F}\boldsymbol{h}, \boldsymbol{\Sigma}_\epsilon)$$
$$p(\boldsymbol{h}) = \mathcal{N}(0, \boldsymbol{\Sigma}_h)$$

$$p(\boldsymbol{h}|\boldsymbol{g}) = \mathcal{N}(\widehat{\boldsymbol{h}}, \widehat{\boldsymbol{\Sigma}}_h)$$
$$\widehat{\boldsymbol{\Sigma}}_h = [\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{D}_h'\boldsymbol{D}_h]^{-1}$$
$$\widehat{\boldsymbol{h}} = [\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{D}_h'\boldsymbol{D}_h]^{-1}\boldsymbol{F}'\boldsymbol{g}$$

Blind Deconvolution: Regularization

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

- Joint Criterion

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

- iterative algorithm

Deconvolution	Identification
$\nabla_f J(\boldsymbol{f},\boldsymbol{h}) = -2\boldsymbol{H}'(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}) + 2\lambda_f \boldsymbol{D}_f' \boldsymbol{D}_f \boldsymbol{f}$ $\widehat{\boldsymbol{f}} = [\boldsymbol{H}'\boldsymbol{H} + \lambda_f \boldsymbol{D}_f' \boldsymbol{D}_f]^{-1} \boldsymbol{H}'\boldsymbol{g}$	$\nabla_h J(\boldsymbol{f},\boldsymbol{h}) = -2\boldsymbol{F}'(\boldsymbol{g} - \boldsymbol{F}\boldsymbol{h}) + 2\lambda_h \boldsymbol{D}$ $\widehat{\boldsymbol{h}} = [\boldsymbol{F}'\boldsymbol{F} + \lambda_h \boldsymbol{D}_h' \boldsymbol{D}_h]^{-1} \boldsymbol{F}'$
$\widehat{f}(\omega) = \frac{1}{H(\omega)} \frac{\|H(\omega)\|^2}{\|H(\omega)\|^2 + \lambda_f \|D_f(\omega)\|^2}\, g(\omega)$	$\widehat{f}(\omega) = \frac{1}{F(\omega)} \frac{\|F(\omega)\|^2}{\|F(\omega)\|^2 + \lambda_h \|D_h(\omega)\|^2}$

Blind Deconvolution: Bayesian approach

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

- Joint posterior law:

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

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

with

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

- iterative algorithm

Blind Deconvolution: Bayesian Joint MAP criterion

- Joint posterior law:

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

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

with

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

- iterative algorithm

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

Blind Deconvolution: Marginalization and EM algorithm

- Joint posterior law:
- Marginalization

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

$$p(\boldsymbol{h}|\boldsymbol{g}) = \int p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\,\mathsf{d}\boldsymbol{f}$$

$$\widehat{\boldsymbol{h}} = \arg\max_{\boldsymbol{h}} \{p(\boldsymbol{h}|\boldsymbol{g})\} \longrightarrow \widehat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}} \left\{p(\boldsymbol{f}|\boldsymbol{g},\widehat{\boldsymbol{h}})\right\}$$

- Expression of $p(\boldsymbol{h}|\boldsymbol{g})$ and its maximization are complexes
- Expectation-Maximization Algorithm

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

  - Iterative algorithm
  - Expectation: Compute

$$Q(\boldsymbol{h},\boldsymbol{h}^{k-1}) = \mathsf{E}_{p(\boldsymbol{f},\boldsymbol{h}^{k-1}|\boldsymbol{g})}\{J(\boldsymbol{f},\boldsymbol{h})\} = \langle \ln p(\boldsymbol{f},\boldsymbol{h}|\boldsymbol{g})\rangle_{p(\boldsymbol{f},\boldsymbol{h}^{k-1}|\boldsymbol{g})}$$

  - Maximization:

$$\boldsymbol{h}^k = \arg\max_{\boldsymbol{h}} \{Q(\boldsymbol{h},\boldsymbol{h}^{k-1})\}$$

Blind Deconvolution: Variational Bayesian Approximation algorithm

- Joint posterior law:

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

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

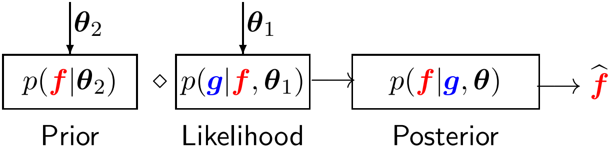
$$\mathcal{F}(q_1\, q_2) = -\langle \ln p((\boldsymbol{f}, \boldsymbol{h}|\boldsymbol{g})\rangle_{q_1 q_2}$$

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

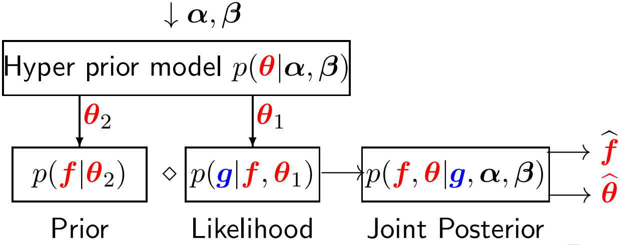
$$\begin{aligned} \widehat{q}_1 &= \arg\min_{q_1} \{\mathrm{KL}(q_1\, q_2|p)\} = \arg\min_{q_1} \{\mathcal{F}(q_1\, q_2)\} \\ \widehat{q}_2 &= \arg\min_{q_2} \{\mathrm{KL}(q_1\, q_2|p)\} = \arg\min_{q_2} \{\mathcal{F}(q_1\, q_2)\} \end{aligned}$$

Summary of Bayesian estimation for Deconvolution

- Simple Bayesian Model and Estimation for Deconvolution

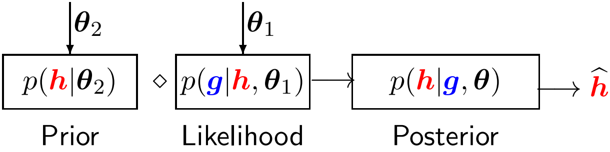


- Full Bayesian Model and Hyperparameter Estimation for Deconvolution

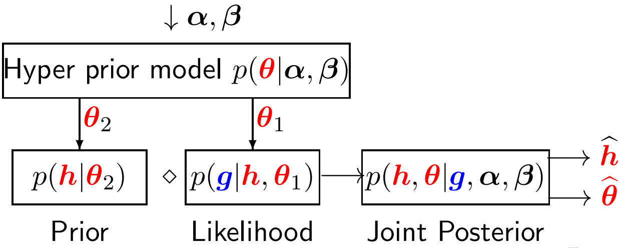


Summary of Bayesian estimation for Identification

- Simple Bayesian Model and Estimation for Identification

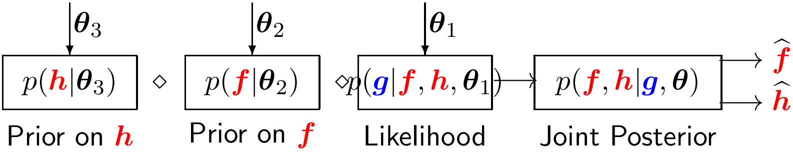


- Full Bayesian Model and Hyperparameter Estimation for Identification



Summary of Bayesian estimation for Blind Deconvolution

Known hyperparameters θ



Unknown hyperparameters θ