# Approximate Bayesian Computation for Big Data

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

Tutorial talk at MaxEnt 2016 workshop, July 10-15, 2016,
Gent, Belgium.

# Contents

# Basic Bayes

- $P(\textit{hypothesis}|\textit{data}) = \frac{P(\textit{data}|\textit{hypothesis})P(\textit{hypothesis})}{P(\textit{data})}$
- Bayes rule tells us how to do inference about hypotheses from data.
- Finite parametric models:

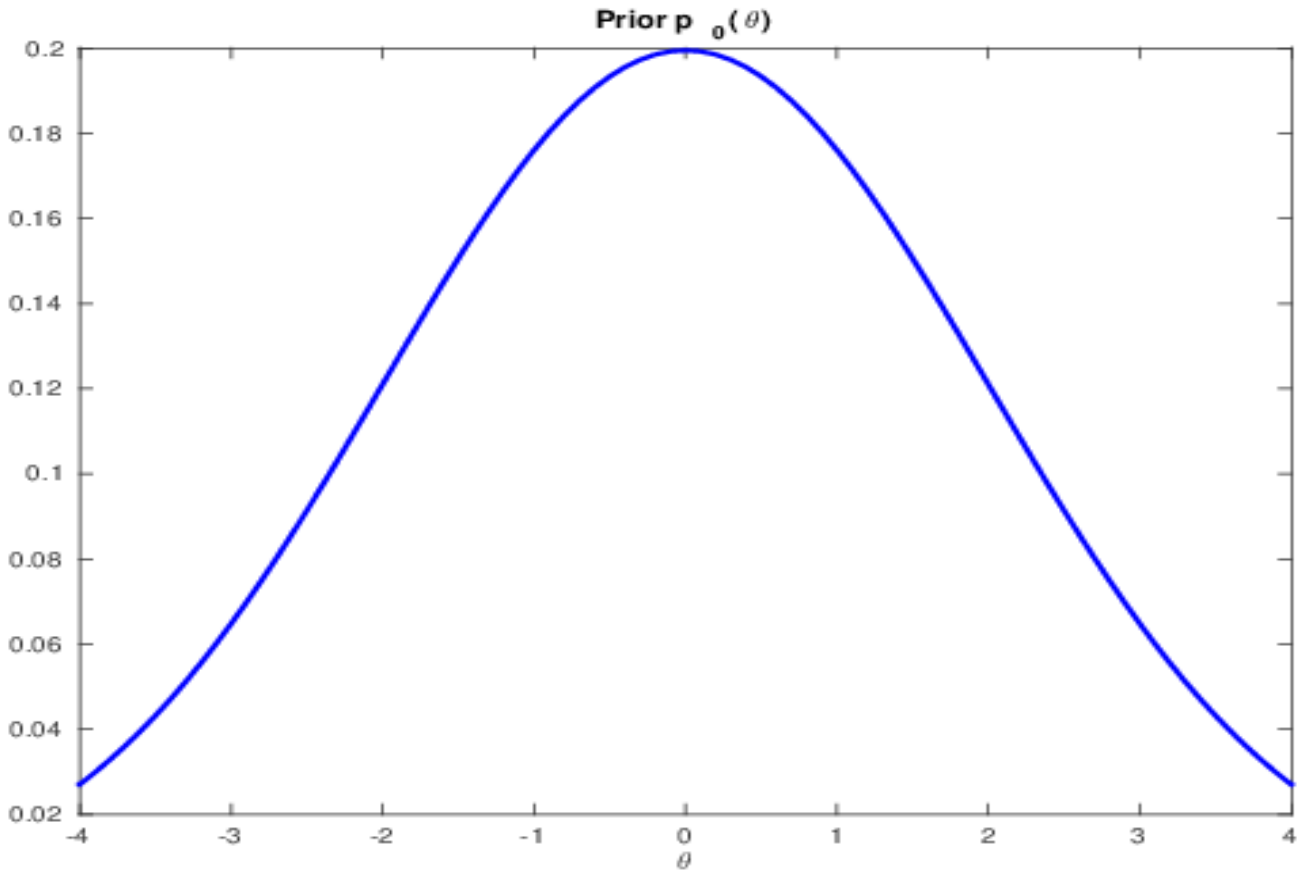
$$p(\boldsymbol{\theta}|\mathbf{d}) = \frac{p(\mathbf{d}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{d})}$$

- Forward model (called also likelihood): $p(\mathbf{d}|\boldsymbol{\theta})$
- Prior knowledge: $p(\boldsymbol{\theta})$
- Posterior knowledge: $p(\boldsymbol{\theta}|\mathbf{d})$

# Bayesian inference: simple one parameter case

$$p(\theta), \mathcal{L}(\theta) = p(\mathbf{d}|\theta) \longrightarrow p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta)\, p(\theta)$$

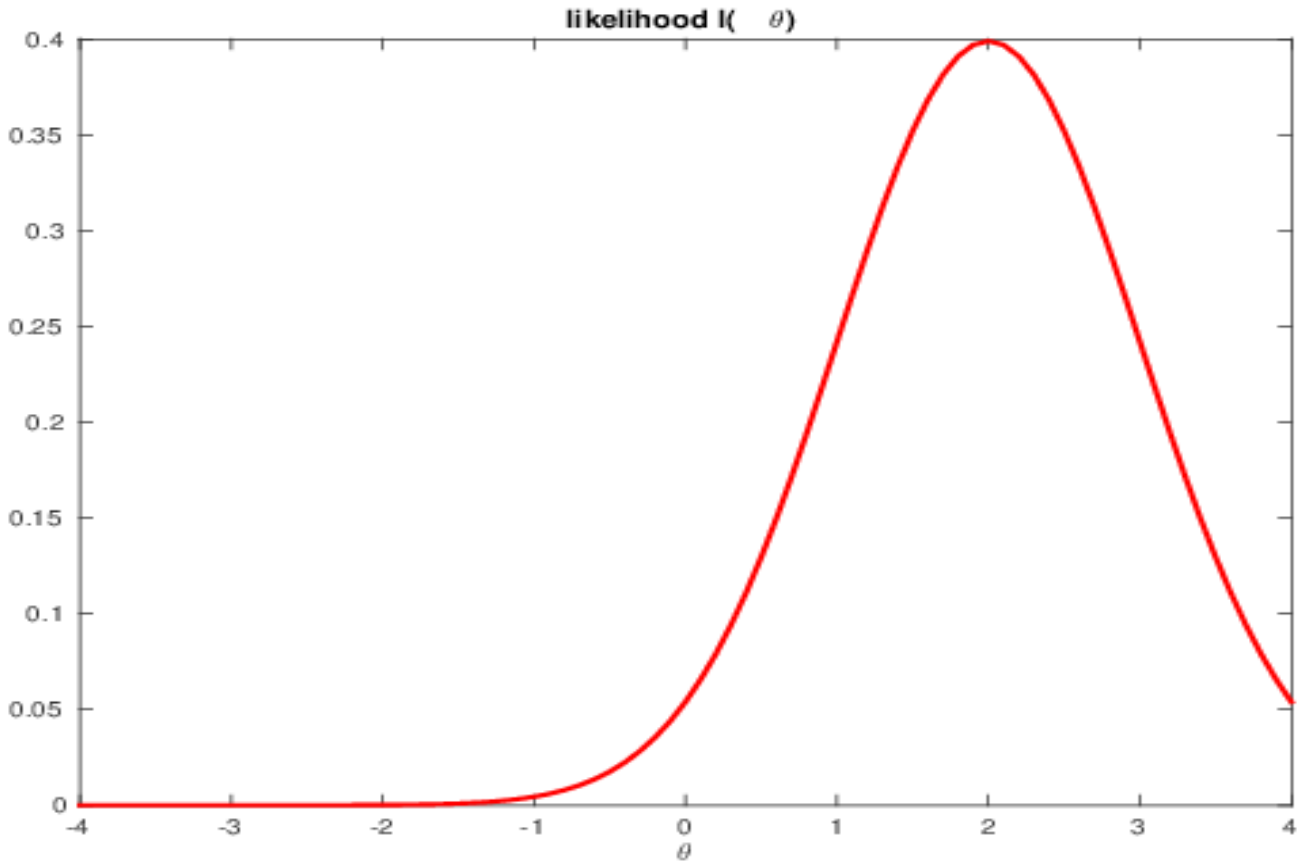
Prior: $p(\theta)$



# Bayesian inference: simple one parameter case

$$p(\theta), \mathcal{L}(\theta) = p(\mathbf{d}|\theta) \longrightarrow p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta)\, p(\theta)$$

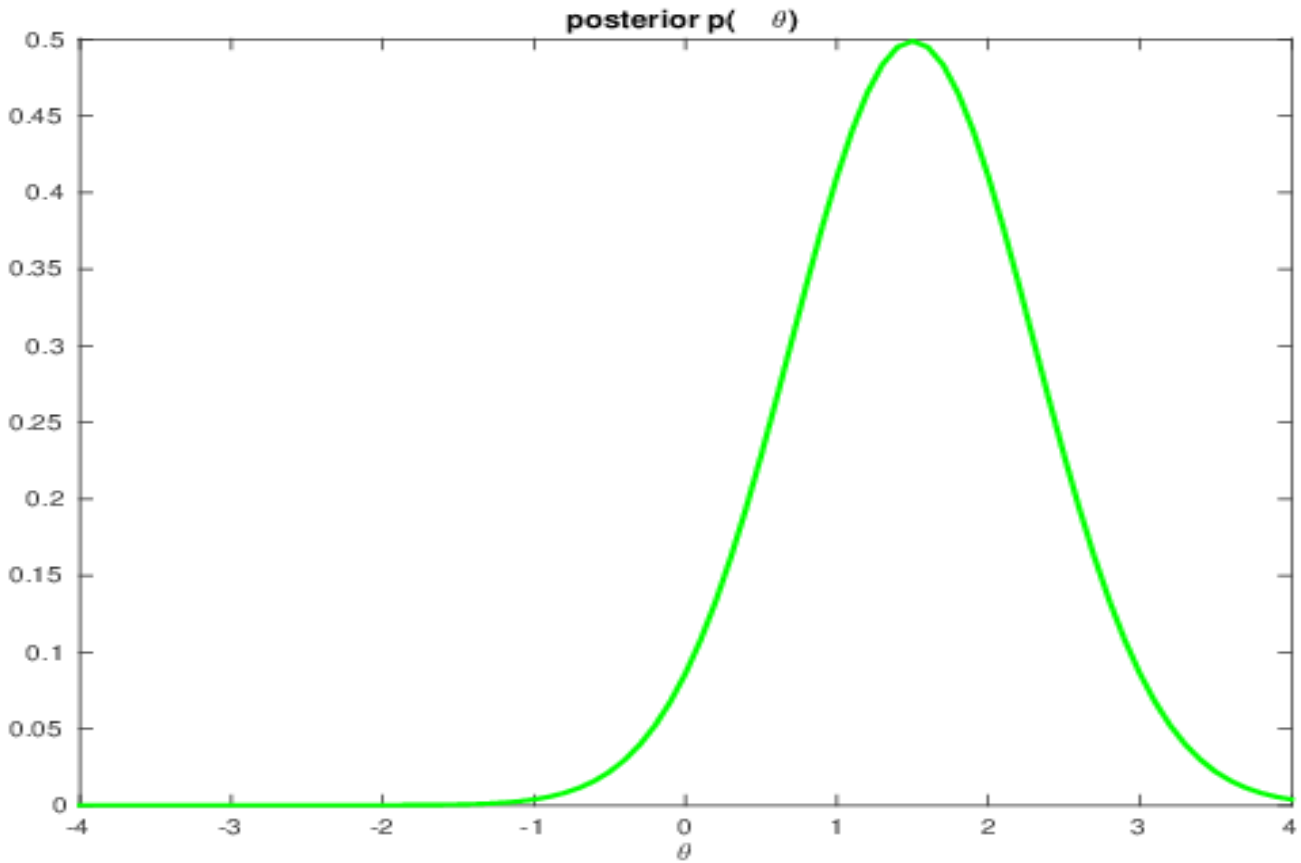
Likelihood: $\mathcal{L}(\theta) = p(\mathbf{d}|\theta)$



# Bayesian inference: simple one parameter case

$$p(\theta), \mathcal{L}(\theta) = p(\mathbf{d}|\theta) \longrightarrow p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta)\, p(\theta)$$

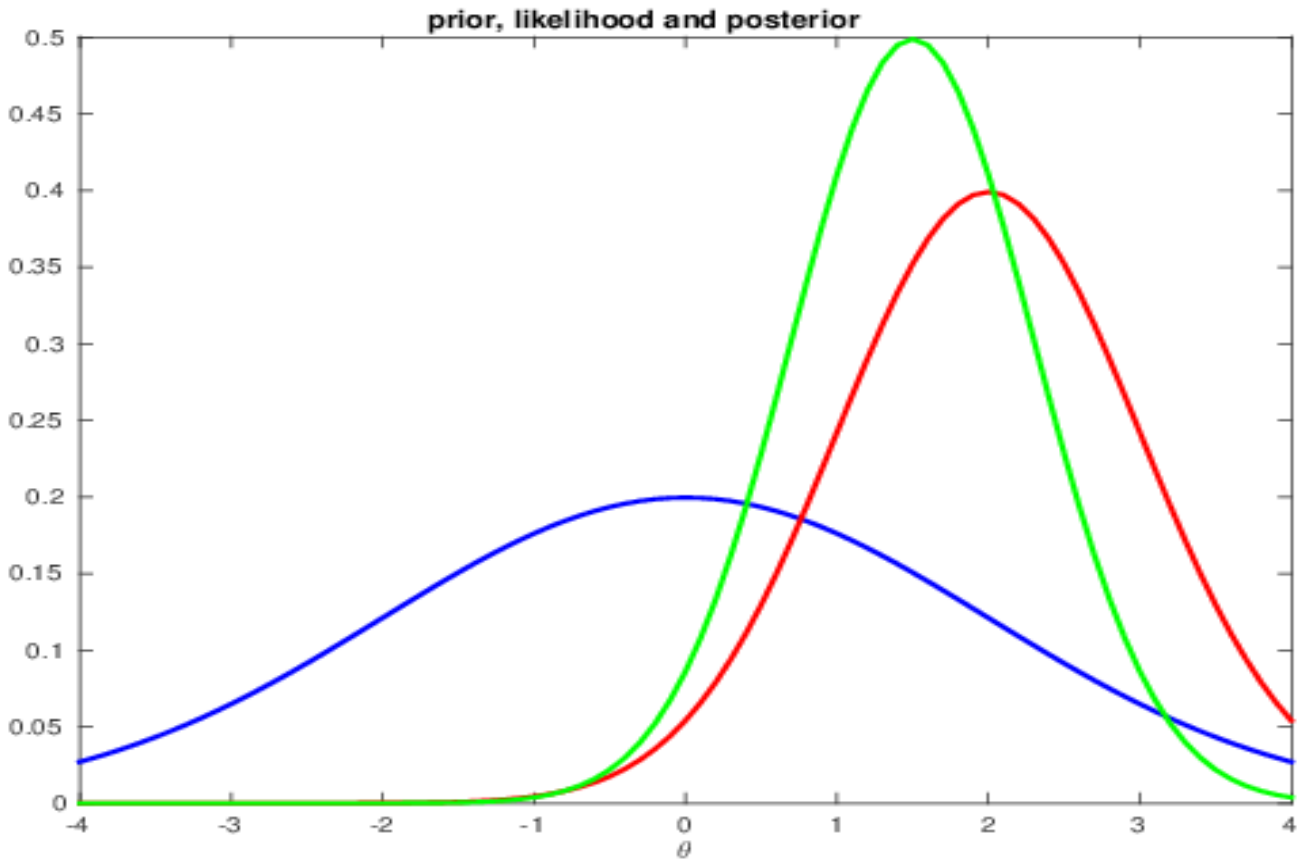
Posterior: $p(\theta|\mathbf{d}) \propto p(\mathbf{d}|\theta)\, p(\theta)$



# Bayesian inference: simple one parameter case

$$p(\theta), \mathcal{L}(\theta) = p(\mathbf{d}|\theta) \longrightarrow p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta)\, p(\theta)$$

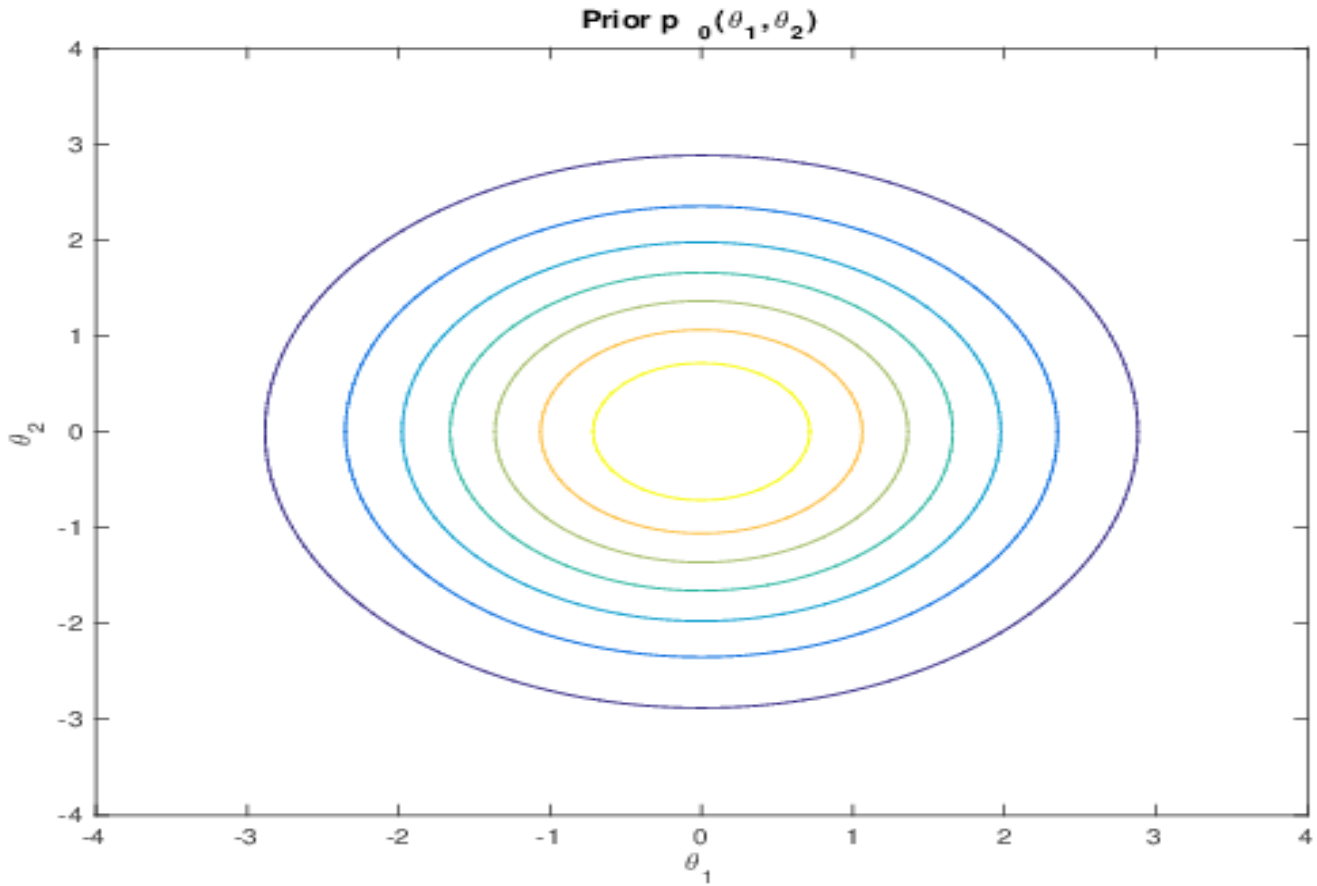
Prior, Likelihood and Posterior:



# Bayesian inference: simple two parameters case

$$p(\theta_1, \theta_2), \mathcal{L}(\theta_1, \theta_2) = p(\mathbf{d}|\theta_1, \theta_2) \longrightarrow p(\theta_1, \theta_2|\mathbf{d}) \propto \mathcal{L}(\theta_1, \theta_2)\, p(\theta_1, \theta_2)$$

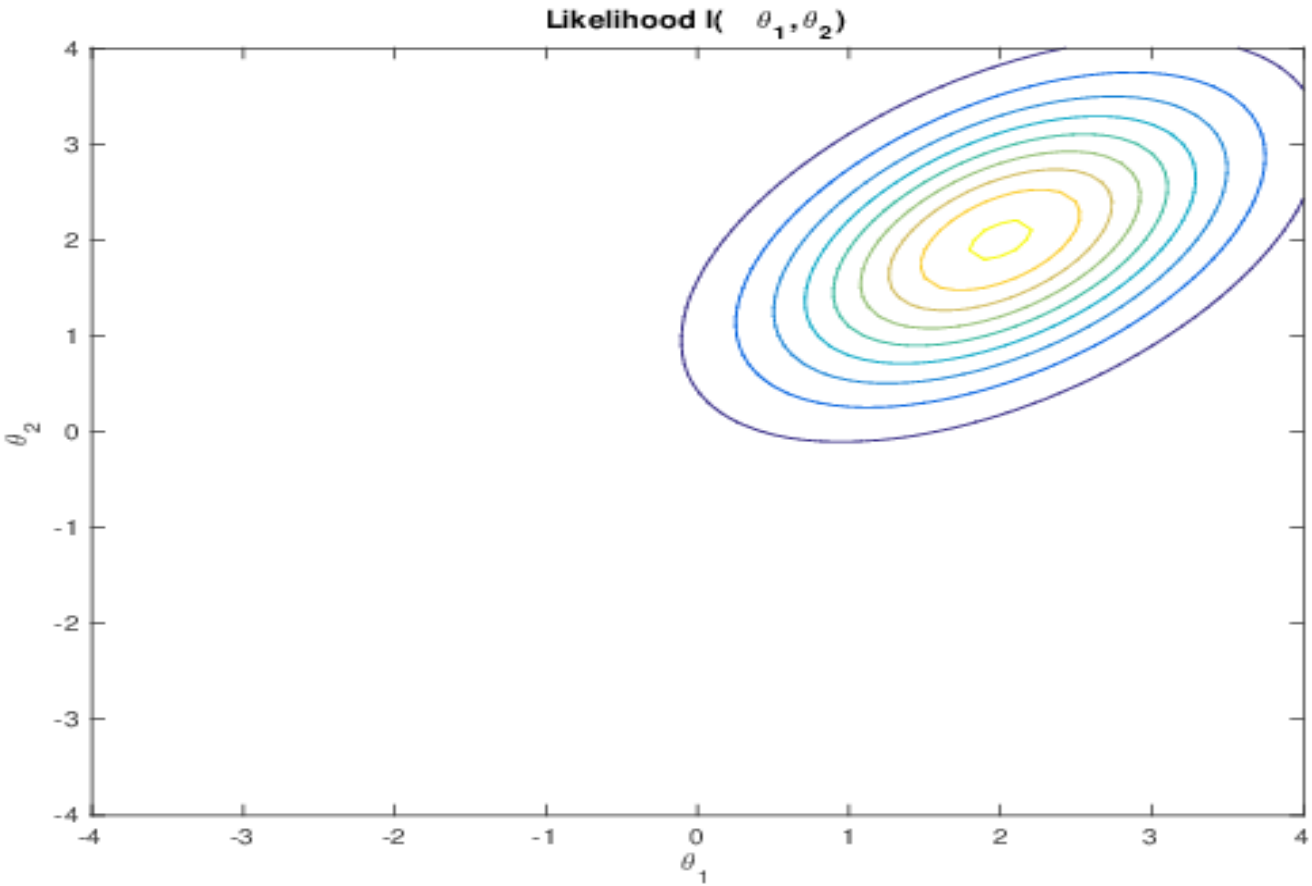
Prior: $p(\theta_1, \theta_2)$



# Bayesian inference: simple one parameter case

$$p(\theta_1, \theta_2), \mathcal{L}(\theta_1, \theta_2) = p(\mathbf{d}|\theta_1, \theta_2) \longrightarrow p(\theta_1, \theta_2|\mathbf{d}) \propto \mathcal{L}(\theta_1, \theta_2)\, p(\theta_1, \theta_2)$$

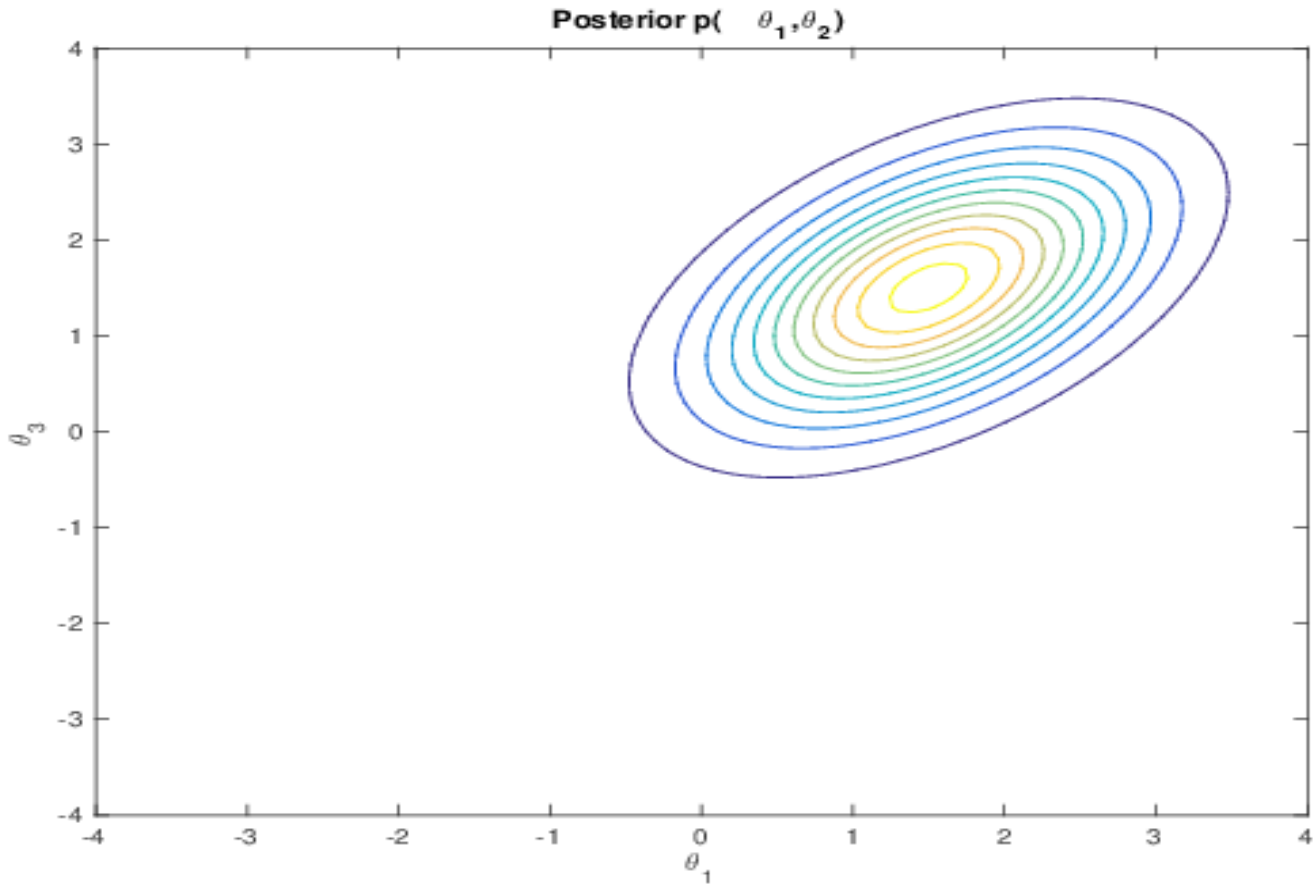
Likelihood: $\mathcal{L}(\theta_1, \theta_2) = p(\mathbf{d}|\theta_1, \theta_2)$



# Bayesian inference: simple one parameter case

$$p(\theta_1, \theta_2), \mathcal{L}(\theta_1, \theta_2) = p(\mathbf{d}|\theta_1, \theta_2) \longrightarrow p(\theta_1, \theta_2|\mathbf{d}) \propto \mathcal{L}(\theta_1, \theta_2)\, p(\theta_1, \theta_2)$$

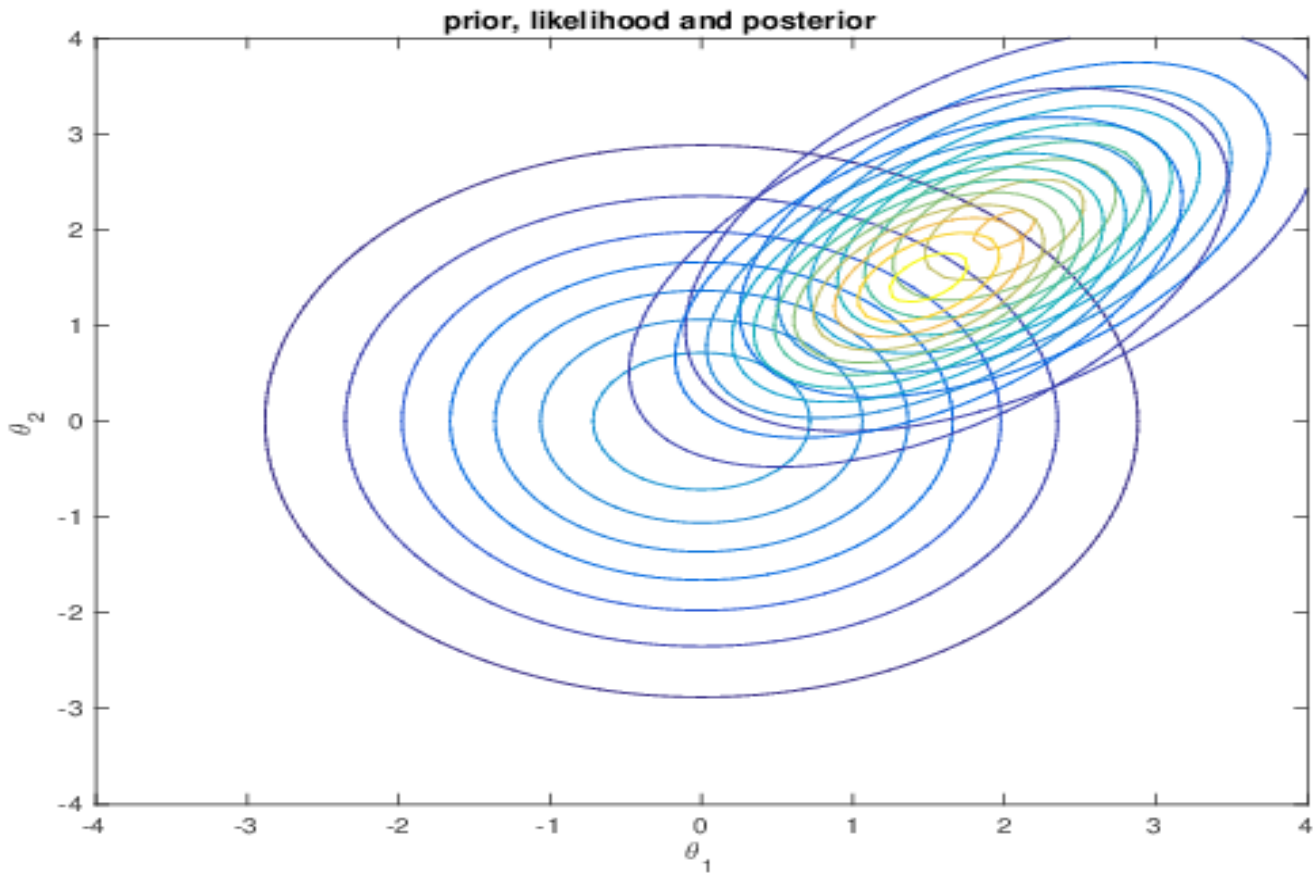
Posterior: $p(\theta_1, \theta_2|\mathbf{d}) \propto p(\mathbf{d}|\theta_1, \theta_2)\, p(\theta_1, \theta_2)$



# Bayesian inference: simple one parameter case

$$p(\theta_1, \theta_2), \mathcal{L}(\theta_1, \theta_2) = p(\mathbf{d}|\theta_1, \theta_2) \longrightarrow p(\theta_1, \theta_2|\mathbf{d}) \propto \mathcal{L}(\theta_1, \theta_2)\, p(\theta_1, \theta_2)$$

Prior, Likelihood and Posterior:



# Bayes: 1D case

$$p(\theta|\mathbf{d}) = \frac{p(\mathbf{d}|\theta)\, p(\theta)}{p(\mathbf{d})} \propto p(\mathbf{d}|\theta)\, p(\theta)$$

- Maximum A Posteriori (MAP)

$$\widehat{\theta} = \arg\max_{\theta} \{p(\theta|\mathbf{d})\} = \arg\max_{\theta} \{p(\mathbf{d}|\theta)\, p(\theta)\}$$

- Posterior Mean

$$\widehat{\theta} = \mathsf{E}_{p(\theta|\mathbf{d})} \{\theta\} = \int \theta p(\theta|\mathbf{d})\, \mathrm{d}\theta$$

- Region of high probabilities

$$[\widehat{\theta}_1, \widehat{\theta}_2] : \int_{\widehat{\theta}_1}^{\widehat{\theta}_2} p(\theta|\mathbf{d})\, \mathrm{d}\theta = 1 - \alpha$$

- Sampling and exploring

$$\theta \sim p(\boldsymbol{\theta}|\mathbf{d})$$

# Bayesian inference: great dimensional case

- Simple Linear case: $\mathbf{d} = \mathbf{H}\boldsymbol{\theta} + \boldsymbol{\epsilon}$
- Gaussian priors:

$$p(\mathbf{d}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{d}|\mathbf{H}\boldsymbol{\theta}, v_\epsilon \mathbf{I})$$
$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|0, v_\theta \mathbf{I})$$

- Gaussian posterior:

$$p(\boldsymbol{\theta}|\mathbf{d}) = \mathcal{N}(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{V}})$$
$$\widehat{\boldsymbol{\theta}} = [\mathbf{H}'\mathbf{H} + \lambda \mathbf{I}]^{-1}\mathbf{H}'\mathbf{d}, \quad \lambda = \frac{v_\epsilon}{v_\theta}$$
$$\widehat{\mathbf{V}} = [\mathbf{H}'\mathbf{H} + \lambda \mathbf{I}]^{-1}$$

- Computation of $\widehat{\boldsymbol{\theta}}$ can be done via optimization of: $J(\boldsymbol{\theta}) = -\ln p(\boldsymbol{\theta}|\mathbf{d}) = \frac{1}{2v_\epsilon}\|\mathbf{d} - \mathbf{H}\boldsymbol{\theta}\|^2 + \frac{1}{2v_\theta}\|\boldsymbol{\theta}\|^2 + c$
- Computation of $\widehat{\mathbf{V}} = [\mathbf{H}'\mathbf{H} + \lambda \mathbf{I}]^{-1}$ needs great dimensional matrix inversion.

# Bayesian inference: great dimensional case

- Gaussian posterior:

$$p(\boldsymbol{\theta}|\mathbf{d}) = \mathcal{N}(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{V}}),$$
$$\widehat{\boldsymbol{\theta}} = [\mathbf{H}'\mathbf{H} + \lambda\mathbf{I}]^{-1}\mathbf{H}'\mathbf{d}, \quad \widehat{\mathbf{V}} = [\mathbf{H}'\mathbf{H} + \lambda\mathbf{I}]^{-1}, \ \lambda = \frac{v_\epsilon}{v_\theta}$$

- Computation of $\widehat{\boldsymbol{\theta}}$ can be done via optimization of:
$$J(\boldsymbol{\theta}) = -\ln p(\boldsymbol{\theta}|\mathbf{d}) = c + \|\mathbf{d} - \mathbf{H}\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\theta}\|^2$$
- Gradient based methods:
$\nabla J(\boldsymbol{\theta}) = -2\mathbf{H}'(\mathbf{d} - \mathbf{H}\boldsymbol{\theta}) + 2\lambda\boldsymbol{\theta}$
- constant step, Steepest descend, ...

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha^{(k)}\nabla J(\boldsymbol{\theta}^{(k)}) = \boldsymbol{\theta}^{(k)} + 2\alpha^{(k)}\left[\mathbf{H}'(\mathbf{d} - \mathbf{H}\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}\right]$$

- Conjugate Gradient, ...
- At each iteration, we need to be able to compute:
  - Forward operation: $\widehat{\mathbf{d}} = \mathbf{H}\boldsymbol{\theta}^{(k)}$
  - Backward (Adjoint) operation: $\mathbf{H}^t(\mathbf{d} - \widehat{\mathbf{d}})$

# Bayesian inference: great dimensional case

- Computation of $\widehat{\mathbf{V}} = [\mathbf{H}'\mathbf{H} + \lambda\mathbf{I}]^{-1}$ needs great dimensional matrix inversion.
- Almost impossible except in particular cases of Toeplitz, Circulante, TBT, CBC,... where we can diagonalize it via Fast Fourier Transform (FFT).
- Recursive use of the data and recursive update of $\widehat{\boldsymbol{\theta}}$ and $\widehat{\mathbf{V}}$ leads to Kalman Filtering which are still computationally demanding for High dimensional data.
- We also need to generate samples from this posterior: There are many special sampling tools.
- Mainly two categories: Using the covariance matrix $\mathbf{V}$ or its inverse (Precision matrix) $\boldsymbol{\Lambda} = \mathbf{V}^{-1}$

# Bayesian inference: non Gaussian priors case

- Linear forward model: $\mathbf{d} = \mathbf{H}\boldsymbol{\theta} + \boldsymbol{\epsilon}$
- Gaussian noise model:

$$p(\mathbf{d}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{d}|\mathbf{H}\boldsymbol{\theta}, v_\epsilon \mathbf{I}) \propto \exp\left[-\frac{1}{2v_\epsilon}\|\mathbf{d} - \mathbf{H}\boldsymbol{\theta}\|_2^2\right]$$

- Sparsity enforcing prior:

$$p(\boldsymbol{\theta}) \propto \exp\left[\alpha\|\boldsymbol{\theta}\|_1\right]$$

- Posterior:

$$p(\boldsymbol{\theta}|\mathbf{d}) \propto \exp\left[-\frac{1}{2v_\epsilon}J(\boldsymbol{\theta})\right] \text{ with } J(\boldsymbol{\theta}) = \|\mathbf{d}-\mathbf{H}\boldsymbol{\theta}\|_2^2+\lambda\|\boldsymbol{\theta}\|_1,\ \lambda = 2v_\epsilon\alpha$$

- Computation of $\widehat{\boldsymbol{\theta}}$ can be done via optimization of $J(\boldsymbol{\theta})$
- Other computations are much more difficult.

# Bayes Rule for Machine Learning (Simple case)

- Inference on the parameters: Learning from data $\mathbf{d}$:

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

- Model Comparison:

$$p(\mathcal{M}_k|\mathbf{d}) = \frac{p(\mathbf{d}|\mathcal{M}_k)\,p(\mathcal{M}_k)}{p(\mathbf{d})}$$

with

$$p(\mathbf{d}|\mathcal{M}_k) = \int p(\mathbf{d}|\boldsymbol{\theta},\mathcal{M}_k)\,p(\boldsymbol{\theta}|\mathcal{M})\,\mathrm{d}\boldsymbol{\theta}$$

- Prediction with selected model:

$$p(z|\mathcal{M}_k) = \int p(z|\boldsymbol{\theta},\mathcal{M}_k)p(\boldsymbol{\theta}|\mathbf{d},\mathcal{M}_k)\,\mathrm{d}\boldsymbol{\theta}$$

# Approximation methods

- Laplace approximation
- Bayesian Information Criterion (BIC)
- Variational Bayesian Approximations (VBA)
- Expectation Propagation (EP)
- Markov chain Monte Carlo methods (MCMC)
- Exact Sampling

# Laplace Approximation

- Data set $\mathbf{d}$, models $\mathcal{M}_1, \cdots, \mathcal{M}_K$, parameters $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K$
- Model Comparison:

$$p(\boldsymbol{\theta}, \mathbf{d}|\mathcal{M}) = p(\mathbf{d}|\boldsymbol{\theta}, \mathcal{M})\, p(\boldsymbol{\theta}|\mathcal{M})$$
$$p(\boldsymbol{\theta}|\mathbf{d}, \mathcal{M}) = p(\boldsymbol{\theta}, \mathbf{d}|\mathcal{M})/p(\mathbf{d}|\mathcal{M})$$
$$p(\mathbf{d}|\mathcal{M}) = \int p(\mathbf{d}|\boldsymbol{\theta}, \mathcal{M})\, p(\boldsymbol{\theta}|\mathcal{M})\, \mathrm{d}\boldsymbol{\theta}$$

- For large amount of data (relative to number of parameters, $m$), $p(\theta|\mathbf{d}, \mathcal{M})$ is approximated by a Gaussian around its maximum (MAP) $\widehat{\boldsymbol{\theta}}$:

$$p(\boldsymbol{\theta}|\mathbf{d}, \mathcal{M}) \approx (2\pi)^{-m/2} |\mathbf{A}|^{1/2} \exp\left[-\frac{1}{2}(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})' \mathbf{A} (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})\right]$$

where $A_{i,j} = \frac{d^2}{\theta_i \theta_j} \ln p(\boldsymbol{\theta}|\mathbf{d}, \mathcal{M})$ is the $m \times m$ Hessian matrix.

- $p(\mathbf{d}|\mathcal{M}) = p(\boldsymbol{\theta}, \mathbf{d}|\mathcal{M})/p(\boldsymbol{\theta}|\mathbf{d}, \mathcal{M})$ and evaluating it at $\widehat{\boldsymbol{\theta}}$:

$$\ln p(\mathbf{d}|\mathcal{M}_k) \approx \ln p(\mathbf{d}|\widehat{\boldsymbol{\theta}}, \mathcal{M}_k) + \ln p(\widehat{\boldsymbol{\theta}}|\mathcal{M}_k) + \frac{m}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|$$

- Needs computation of $\widehat{\boldsymbol{\theta}}$ and $|\mathbf{A}|$.

# Bayesian Information Criteria (BIC)

- BIC is obtained from the Laplace approximation

$$\ln p(\mathbf{d}|\mathcal{M}_k) \approx \ln p(\widehat{\boldsymbol{\theta}}|\mathcal{M}_k) + p(\mathbf{d}|\widehat{\boldsymbol{\theta}}, \mathcal{M}_k) + \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{A}|$$

  by taking the large sample limit ($n \mapsto \infty$) where $n$ is the number of data points:

$$\ln p(\mathbf{d}|\mathcal{M}_k) \approx p(\mathbf{d}|\widehat{\boldsymbol{\theta}}, \mathcal{M}_k) - \frac{d}{2}\ln(n)$$

- Easy to compute
- It does not depend on the prior
- It is equivalent to MDL criterion
- Assumes that as ($n \mapsto \infty$), all the parameters are identifiable.
- Danger: counting parameters can be deceiving (sinusoid, infinite dim models)

# Bayes Rule for Machine Learning with hidden variables

- Data: $\mathbf{d}$, Hidden Variables: $\mathbf{x}$, Parameters: $\boldsymbol{\theta}$, Model: $\mathcal{M}$
- Bayes rule

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

- Model Comparison

$$p(\mathcal{M}_k|\mathbf{d}) = \frac{p(\mathbf{d}|\mathcal{M}_k)\, p(\mathcal{M}_k)}{p(\mathbf{d})}$$

with

$$p(\mathbf{d}|\mathcal{M}_k) = \int\int p(\mathbf{d}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{M}_k)\, p(\mathbf{x}|\boldsymbol{\theta}, \mathcal{M}))p(\boldsymbol{\theta}|\mathcal{M})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\boldsymbol{\theta}$$

- Prediction with a new data $z$

$$p(z|\mathcal{M}) = \int\int p(z|\mathbf{x}, \boldsymbol{\theta}, \mathcal{M})p(\mathbf{x}|\boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta}|\mathcal{M}))\, \mathrm{d}\mathbf{x}\, \mathrm{d}\boldsymbol{\theta}$$

# Lower Bounding the Marginal Likelihood

Jensen's inequality:

$$\begin{aligned}
\ln p(\mathbf{d}|\mathcal{M}_k) &= \ln \int\int p(\mathbf{d},\mathbf{x},\boldsymbol{\theta}|\mathcal{M}_k)\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta} \\
&= \ln \int\int q(\mathbf{x},\boldsymbol{\theta})\frac{p(\mathbf{d},\mathbf{x},\boldsymbol{\theta}|\mathcal{M}_k)}{q(\mathbf{x},\boldsymbol{\theta})}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta} \\
&\geq \int\int q(\mathbf{x},\boldsymbol{\theta})\ln\frac{p(\mathbf{d},\mathbf{x},\boldsymbol{\theta}|\mathcal{M}_k)}{q(\mathbf{x},\boldsymbol{\theta})}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}
\end{aligned}$$

Using a factorised approximation for $q(\mathbf{x},\boldsymbol{\theta}) = q_1(\mathbf{x})q_2(\boldsymbol{\theta})$:

$$\begin{aligned}
\ln p(\mathbf{d}|\mathcal{M}_k) &\geq \int\int q_1(\mathbf{x})q_2(\boldsymbol{\theta})\ln\frac{p(\mathbf{d},\mathbf{x},\boldsymbol{\theta}|\mathcal{M}_k)}{q_1(\mathbf{x})q_2(\boldsymbol{\theta})}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta} \\
&= \mathcal{F}_{\mathcal{M}_k}(q_1(\mathbf{x}), q_2(\boldsymbol{\theta}), \mathbf{d})
\end{aligned}$$

Maximising this free energy leads to VBA.

# Variational Bayesian Learning

$$\begin{aligned}\mathcal{F}_{\mathcal{M}}(q_1(\mathbf{x}), q_2(\boldsymbol{\theta}), \mathbf{d}) &= \int\int q_1(\mathbf{x}) q_2(\boldsymbol{\theta}) \ln \frac{p(\mathbf{d}, \mathbf{x}, \boldsymbol{\theta}|\mathcal{M})}{q_1(\mathbf{x}) q_2(\boldsymbol{\theta})} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\boldsymbol{\theta} \\ &= \mathcal{H}(q_1) + \mathcal{H}(q_2) + \langle \ln p(\mathbf{d}, \mathbf{x}, \boldsymbol{\theta}|\mathcal{M}) \rangle_{q_1 q_2}\end{aligned}$$

Minimising this lower bound with respect to $q_1$ and then $q_2$ leads to EM-like iterative update

$$\begin{aligned} q_1^{(t+1)}(\mathbf{x}) &\propto \exp\left[\langle \ln p(\mathbf{d}, \mathbf{x}, \boldsymbol{\theta}|\mathcal{M}) \rangle_{q_2^{(t)}(\boldsymbol{\theta})}\right] \quad \text{E-like step} \\ q_2^{(t+1)}(\boldsymbol{\theta}) &\propto \exp\left[\langle \ln p(\mathbf{d}, \mathbf{x}, \boldsymbol{\theta}|\mathcal{M}) \rangle_{q_1^{(t+1)}(\mathbf{x})}\right] \quad \text{M-like step}\end{aligned}$$

which can also be written as:

$$\begin{aligned} q_1^{(t+1)}(\mathbf{x}) &\propto \exp\left[\langle \ln p(\mathbf{d}, \mathbf{x}|\boldsymbol{\theta}, \mathcal{M}) \rangle_{q_2^{(t)}(\boldsymbol{\theta})}\right] \quad \text{E-like step} \\ q_2^{(t+1)}(\boldsymbol{\theta}) &\propto p(\boldsymbol{\theta}|\mathcal{M}) \exp\left[\langle \ln p(\mathbf{d}, \mathbf{x}|\boldsymbol{\theta}, \mathcal{M}) \rangle_{q_1^{(t+1)}(\mathbf{x})}\right] \quad \text{M-like step}\end{aligned}$$

# EM and VBEM algorithms

**EM for Marginal MAP estimation**

Goal: maximize $p(\boldsymbol{\theta}|\mathbf{d},\mathcal{M})$ w.r.t. $\boldsymbol{\theta}$

**E Step:** Compute

$q_1^{(t+1)}(\mathbf{x}) = p(\mathbf{x}|\mathbf{d},\boldsymbol{\theta}^{(t)})$ and

$Q(\boldsymbol{\theta}) = \langle \ln p(\mathbf{d},\mathbf{x},\boldsymbol{\theta}|\mathcal{M})\rangle_{q_1^{(t+1)}(\mathbf{x})}$

**M Step:** Maximize

$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \{Q(\boldsymbol{\theta})\}$

**Variational Bayesian EM**

Goal: lower bound $p(\mathbf{d}|\mathcal{M})$

**VB-E Step:** Compute

$q_1^{(t+1)}(\mathbf{x}) = p(\mathbf{x}|\mathbf{d},\boldsymbol{\phi}^{(t)})$ and

$Q(\boldsymbol{\theta}) = \langle \ln p(\mathbf{d},\mathbf{x},\boldsymbol{\theta}|\mathcal{M})\rangle_{q_1^{(t+1)}(\mathbf{x})}$

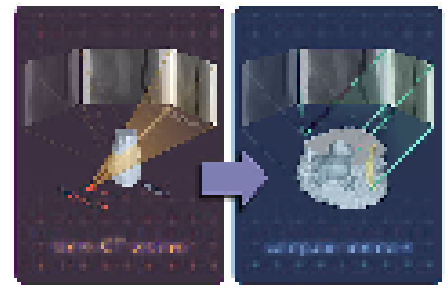
**M Step:** Maximize

$q_2^{(t+1)}(\boldsymbol{\theta}) = \exp\left[Q(\boldsymbol{\theta})\right]$

Properties:

- VB-EM reduces to EM if $q_2(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})$
- VB-EM has the same complexity than EM
- If we choose $q_2(\boldsymbol{\theta})$ in the conjugate family of $p(\mathbf{d},\mathbf{x}|\boldsymbol{\theta})$, then $\boldsymbol{\phi}$ becomes the expected natural parameters
- The main computational part of both methods is in the E-step. We can use belief propagation, Kalman filter, etc. to do it. In VB-EM, $\boldsymbol{\phi}$ replaces $\boldsymbol{\theta}$.

# Computed Tomography: Seeing inside of a body



- $f(x,y)$ a section of a real 3D body $f(x,y,z)$
- $g_\phi(r)$ a line of observed radiography $g_\phi(r,z)$
- Forward model:
  Line integrals or Radon Transform

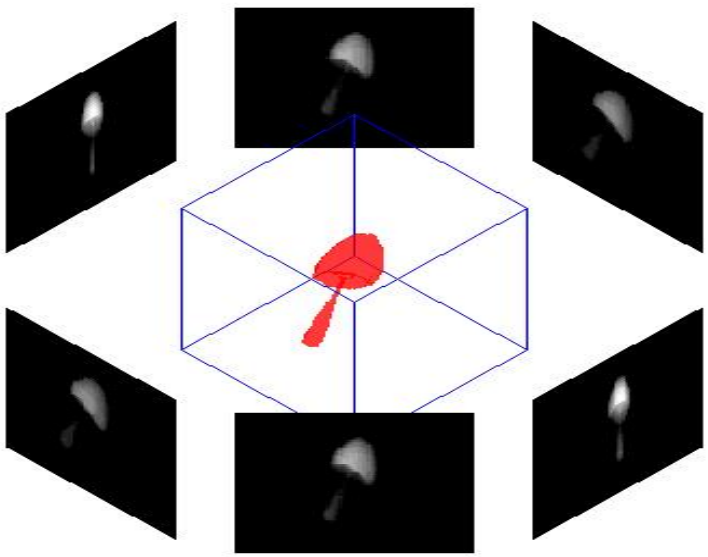
$$
\begin{aligned}
g_\phi(r) &= \int_{L_{r,\phi}} f(x,y)\,\mathrm{d}l + \epsilon_\phi(r) \\
&= \iint f(x,y)\,\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y + \epsilon_\phi(r)
\end{aligned}
$$

- Inverse problem: Image reconstruction

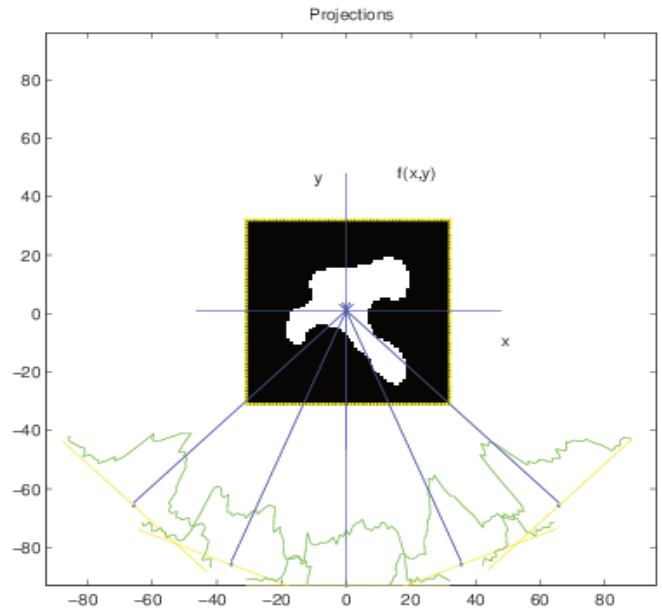
  Given the forward model $\mathcal{H}$ (Radon Transform) and
  a set of data $g_{\phi_i}(r), i = 1, \cdots, M$
  find $f(x,y)$

# 2D and 3D Computed Tomography

3D



2D

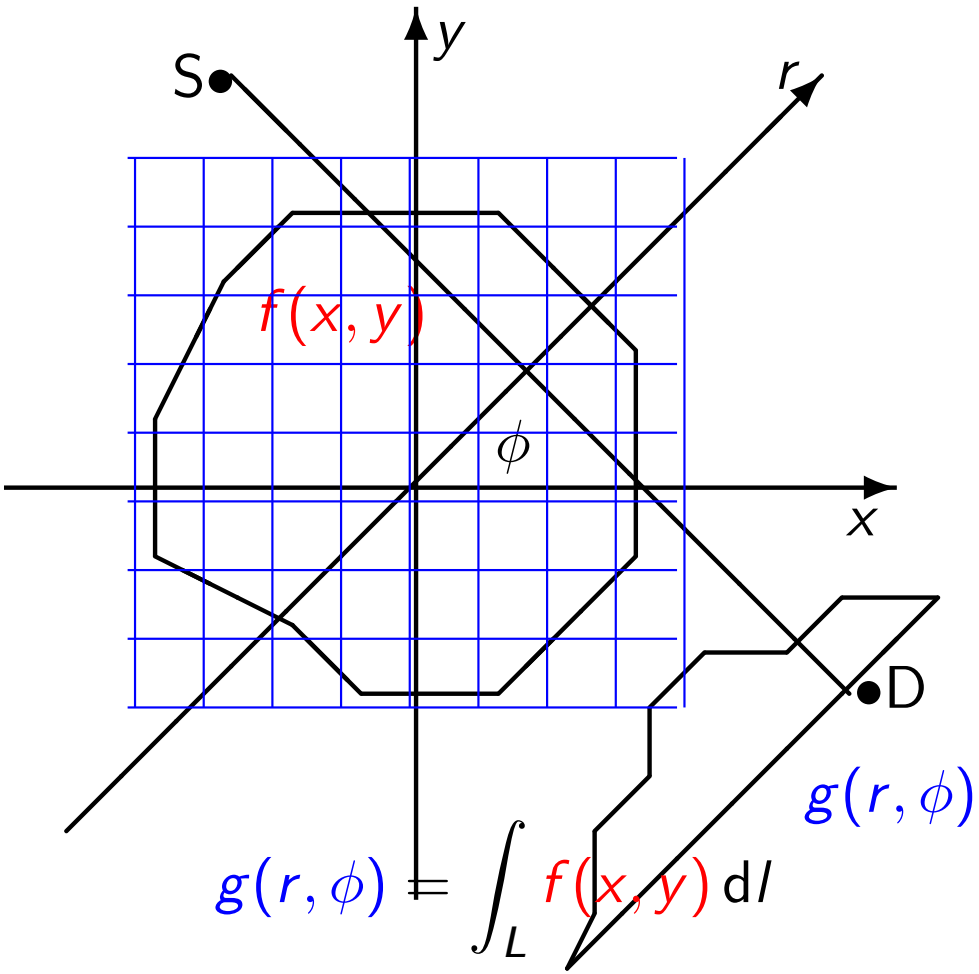



$$g_\phi(r_1, r_2) = \int_{\mathcal{L}_{r_1,r_2,\phi}} f(x, y, z)\, \mathrm{d}l \qquad g_\phi(r) = \int_{\mathcal{L}_{r,\phi}} f(x, y)\, \mathrm{d}l$$

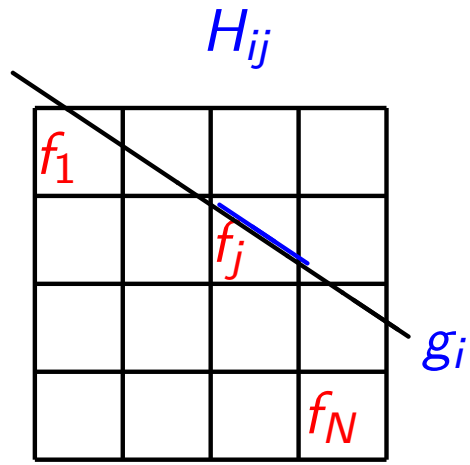
Forward probelm: $f(x, y)$ or $f(x, y, z) \longrightarrow g_\phi(r)$ or $g_\phi(r_1, r_2)$

Inverse problem: $g_\phi(r)$ or $g_\phi(r_1, r_2) \longrightarrow f(x, y)$ or $f(x, y, z)$

# Algebraic methods: Discretization




$$g(r,\phi) = \int_L f(x,y)\, \mathrm{d}l$$




$$f(x,y) = \sum_j f_j\, b_j(x,y)$$

$$b_j(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \text{ pixel } j \\ 0 & \text{else} \end{cases}$$

$$g_i = \sum_{j=1}^{N} H_{ij}\, f_j + \epsilon_i \rightarrow \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

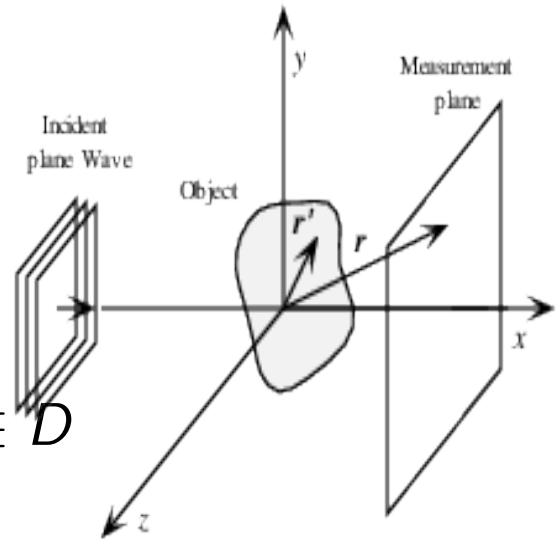
- $\mathbf{H}$ is huge dimensional: 2D: $10^6 \times 10^6$, 3D: $10^9 \times 10^9$.
- $\mathbf{H}\mathbf{f}$ corresponds to forward projection
- $\mathbf{H}^t\mathbf{g}$ corresponds to Back projection (BP)

# Microwave or ultrasound imaging

Measures: diffracted wave by the object $g(\mathbf{r}_i)$

Unknown quantity: $f(\mathbf{r}) = k_0^2(n^2(\mathbf{r}) - 1)$

Intermediate quantity : $\phi(\mathbf{r})$

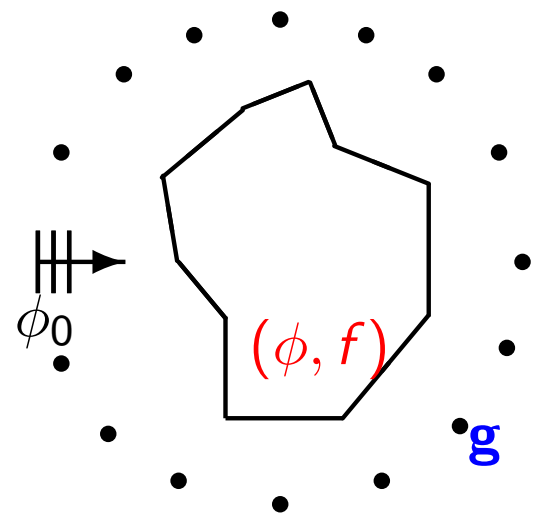



$$g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}')\phi(\mathbf{r}')\, f(\mathbf{r}')\, d\mathbf{r}', \quad \mathbf{r}_i \in S$$

$$\phi(\mathbf{r}) = \phi_0(\mathbf{r}) + \iint_D G_o(\mathbf{r}, \mathbf{r}')\phi(\mathbf{r}')\, f(\mathbf{r}')\, d\mathbf{r}', \quad \mathbf{r} \in D$$

**Born approximation** $(\phi(\mathbf{r}') \simeq \phi_0(\mathbf{r}'))$ **:**

$$g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}')\phi_0(\mathbf{r}')\, f(\mathbf{r}')\, d\mathbf{r}', \quad \mathbf{r}_i \in S$$




**Discretization:**

$$\begin{cases} \mathbf{g} = \mathbf{G}_m\mathbf{F}\phi \\ \phi = \phi_0 + \mathbf{G}_o\mathbf{F}\phi \end{cases} \longrightarrow \begin{cases} \mathbf{g} = \mathbf{H}(\mathbf{f}) \\ \text{with} \quad \mathbf{F} = \text{diag}(\mathbf{f}) \\ \mathbf{H}(\mathbf{f}) = \mathbf{G}_m\mathbf{F}(\mathbf{I} - \mathbf{G}_o\mathbf{F})^{-1}\phi_0 \end{cases}$$

# Microwave or ultrasound imaging: Bilinear model

Nonlinear model:

$$g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}')\phi(\mathbf{r}')\, f(\mathbf{r}')\, \mathrm{d}\mathbf{r}',\ \mathbf{r}_i \in S$$

$$\phi(\mathbf{r}) = \phi_0(\mathbf{r}) + \iint_D G_o(\mathbf{r}, \mathbf{r}')\phi(\mathbf{r}')\, f(\mathbf{r}')\, \mathrm{d}\mathbf{r}',\ \mathbf{r} \in D$$

Bilinear model: $w(\mathbf{r}') = \phi(\mathbf{r}')\, f(\mathbf{r}')$

$$g(\mathbf{r}_i) = \iint_D G_m(\mathbf{r}_i, \mathbf{r}')w(\mathbf{r}')\, \mathrm{d}\mathbf{r}',\ \mathbf{r}_i \in S$$

$$\phi(\mathbf{r}) = \phi_0(\mathbf{r}) + \iint_D G_o(\mathbf{r}, \mathbf{r}')w(\mathbf{r}')\, \mathrm{d}\mathbf{r}',\ \mathbf{r} \in D$$

$$w(\mathbf{r}) = f(\mathbf{r})\phi_0(\mathbf{r}) + \iint_D G_o(\mathbf{r}, \mathbf{r}')w(\mathbf{r}')\, \mathrm{d}\mathbf{r}',\ \mathbf{r} \in D$$

**Discretization:** $\mathbf{g} = \mathbf{G}_m\mathbf{w} + \epsilon, \quad \mathbf{w} = \phi\,.\,\mathbf{f}$

- Constrast $\mathbf{f}$ - Field $\phi$: $\quad \phi = \phi_0 + \mathbf{G}_o\mathbf{w} + \xi$
- Constrast $\mathbf{f}$ - Source $\mathbf{w}$ : $\quad \mathbf{w} = \mathbf{f}\,.\,\phi_0 + \mathbf{G}_o\mathbf{w} + \xi$

# Bayesian approach for linear inverse problems

$$\mathcal{M}: \qquad \mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

- Observation model $\mathcal{M}$ + Information on the noise $\boldsymbol{\epsilon}$:
  $$p(\mathbf{g}|\mathbf{f}, \theta_1; \mathcal{M}) = p_{\boldsymbol{\epsilon}}(\mathbf{g} - \mathbf{H}\mathbf{f}|\theta_1)$$
- A priori information $p(\mathbf{f}|\theta_2; \mathcal{M})$
- Basic Bayes :
  $$p(\mathbf{f}|\mathbf{g}, \theta_1, \theta_2; \mathcal{M}) = \frac{p(\mathbf{g}|\mathbf{f}, \theta_1; \mathcal{M})\, p(\mathbf{f}|\theta_2; \mathcal{M})}{p(\mathbf{g}|\theta_1, \theta_2; \mathcal{M})}$$
- Unsupervised:
  $$p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{g}, \boldsymbol{\alpha}_0) = \frac{p(\mathbf{g}|\mathbf{f}, \theta_1)\, p(\mathbf{f}|\theta_2)\, p(\boldsymbol{\theta}|\boldsymbol{\alpha}_0)}{p(\mathbf{g}|\boldsymbol{\alpha}_0)}, \quad \boldsymbol{\theta} = (\theta_1, \theta_2)$$
- Hierarchical prior models:

$$p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g}, \boldsymbol{\alpha}_0) = \frac{p(\mathbf{g}|\mathbf{f}, \theta_1)\, p(\mathbf{f}|\mathbf{z}, \theta_2)\, p(\mathbf{z}|\theta_3)\, p(\boldsymbol{\theta}|\boldsymbol{\alpha}_0)}{p(\mathbf{g}|\boldsymbol{\alpha}_0)}, \quad \boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$$

# Bayesian approach for bilinear inverse problems

$$\mathcal{M}: \quad \mathbf{g} = \mathbf{G}_m\mathbf{w} + \boldsymbol{\epsilon}, \quad \mathbf{w} = \mathbf{f}.\boldsymbol{\phi}_0 + \mathbf{G}_o\mathbf{w} + \boldsymbol{\xi}, \quad \mathbf{w} = \boldsymbol{\phi}\,.\,\mathbf{f}$$

$$\mathcal{M}: \quad \mathbf{g} = \mathbf{G}_m\mathbf{w} + \boldsymbol{\epsilon}, \quad \mathbf{w} = (\mathbf{I} - \mathbf{G}_o)^{-1}(\boldsymbol{\Phi}_0\mathbf{f} + \boldsymbol{\xi}), \quad \mathbf{w} = \boldsymbol{\phi}\,.\,\mathbf{f}$$

- Basic Bayes:

$$p(\mathbf{f}, \mathbf{w}|\mathbf{g}, \boldsymbol{\theta}) = \frac{p(\mathbf{g}|\mathbf{w}, , \theta_1)\, p(\mathbf{w}|\mathbf{f}, , \theta_2)\, p(\mathbf{f}|, \theta_3)}{p(\mathbf{g}|\boldsymbol{\theta})} \propto p(\mathbf{g}|\mathbf{w}, \theta_1)\, p(\mathbf{w}|\mathbf{f}, \theta_2)\, p(\mathbf{f}|\theta_3)$$

- Unsupervised:

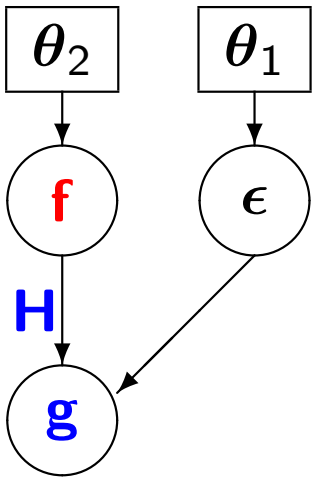
$$p(\mathbf{f}, \mathbf{w}, \boldsymbol{\theta}|\mathbf{g}, \boldsymbol{\alpha}_0) \propto p(\mathbf{g}|\mathbf{w}, \theta_1)\, p(\mathbf{f}|\mathbf{w}, \theta_2) p(\mathbf{f}|\theta_3)\, p(\boldsymbol{\theta}|\alpha_0), \ \boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$$

- Hierarchical prior models:

$$p(\mathbf{f}, \mathbf{w}, \mathbf{z}, \boldsymbol{\theta}|\mathbf{g}, \boldsymbol{\alpha}_0) \propto p(\mathbf{g}|\mathbf{w}, \theta_1)\, p(\mathbf{w}|\mathbf{f}, \theta_2)\, p(\mathbf{f}|\mathbf{z}, \theta_3)\, p(\mathbf{z}|\theta_4)\, p(\boldsymbol{\theta}|\boldsymbol{\alpha}_0)$$

# Bayesian inference for inverse problems

Simple case:



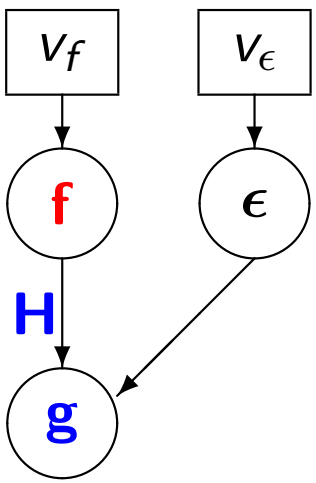

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon$$

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

– Objective: Infer $\mathbf{f}$

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

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




Example: Caussian case:

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \\ p(\mathbf{f}|v_f) = \mathcal{N}(\mathbf{f}|0, v_f \mathbf{I}) \end{cases} \rightarrow p(\mathbf{f}|\mathbf{g}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\widehat{\mathbf{f}}, \widehat{\boldsymbol{\Sigma}})$$

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

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

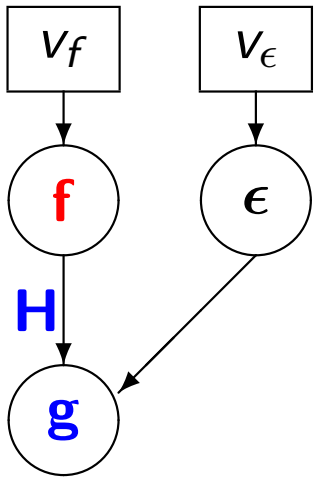
– Posterior Mean (PM)=MAP:

$$\begin{cases} \widehat{\mathbf{f}} = (\mathbf{H}^t\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^t\mathbf{g} \\ \widehat{\boldsymbol{\Sigma}} = (\mathbf{H}^t\mathbf{H} + \lambda\mathbf{I})^{-1} \end{cases} \quad \text{with } \lambda = \frac{v_\epsilon}{v_f}.$$

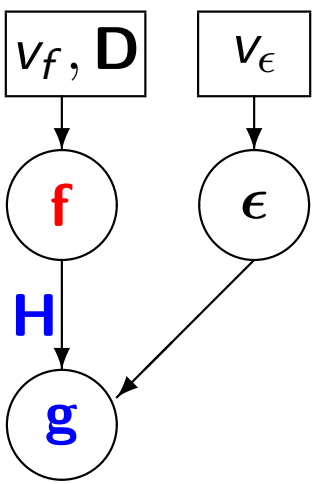
# Gaussian model: Simple separable and Markovian

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

Separable Gaussian




Gauss-Markov




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

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, \theta_1) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \\ p(\mathbf{f}|v_f) = \mathcal{N}(\mathbf{f}|0, v_f \mathbf{I}) \end{cases} \rightarrow p(\mathbf{f}|\mathbf{g}, \theta) = \mathcal{N}(\mathbf{f}|\widehat{\mathbf{f}}, \widehat{\mathbf{\Sigma}})$$

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

$$\begin{cases} \widehat{\mathbf{f}} = (\mathbf{H}^t\mathbf{H} + \lambda\mathbf{I}]^{-1}\mathbf{H}^t\mathbf{g} \\ \widehat{\mathbf{\Sigma}} = v_\epsilon(\mathbf{H}^t\mathbf{H} + \lambda\mathbf{I}]^{-1} \end{cases} \quad \text{with } \lambda = \frac{v_\epsilon}{v_f}.$$

Markovian case:

$p(\mathbf{f}|v_f, \mathbf{D}) = \mathcal{N}(\mathbf{f}|0, v_f(\mathbf{D}\mathbf{D}^t)^{-1})$

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

– Posterior Mean (PM)=MAP:

$$\begin{cases} \widehat{\mathbf{f}} = (\mathbf{H}^t\mathbf{H} + \lambda\mathbf{D}^t\mathbf{D}]^{-1}\mathbf{H}^t\mathbf{g} \\ \widehat{\mathbf{\Sigma}} = v_\epsilon(\mathbf{H}^t\mathbf{H} + \lambda\mathbf{D}^t\mathbf{D}]^{-1} \end{cases} \quad \text{with } \lambda = \frac{v_e}{v_f}.$$

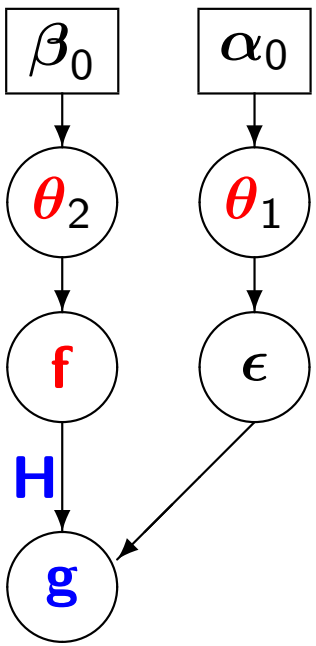
# Bayesian inference (Unsupervised case)

Unsupervised case: Hyper parameter estimation

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

– 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:

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

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

– 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 $\boldsymbol{\theta}$

# 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 $q_1 \longrightarrow q_2 \longrightarrow q_1 \longrightarrow q_2, \cdots$

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

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

# Variational Bayesian Approximation

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

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

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

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

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 $\mathcal{F}(q)$

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

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

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

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

# BVA: Choice of family of laws $q_1$ and $q_2$

- Case 1 : $\longrightarrow$ Joint MAP

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

- Case 2 : $\longrightarrow$ EM

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

- Appropriate choice for inverse problems

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

Exponential families, Conjugate priors

# JMAP, EM and VBA

JMAP Alternate optimization Algorithm:

$$\boxed{\widetilde{\mathbf{f}} = \arg\max_{\mathbf{f}} \left\{ p(\mathbf{f}, \widetilde{\boldsymbol{\theta}} | \mathbf{g}) \right\}} \longrightarrow \widetilde{\mathbf{f}} \longrightarrow \widehat{\mathbf{f}}$$

$$\widetilde{\boldsymbol{\theta}} \uparrow \qquad\qquad \widetilde{\mathbf{f}} \downarrow$$

$$\widehat{\boldsymbol{\theta}} \longleftarrow \widetilde{\boldsymbol{\theta}} \longleftarrow \boxed{\widetilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ p(\widetilde{\mathbf{f}}, \boldsymbol{\theta} | \mathbf{g}) \right\}} \longleftarrow \widetilde{\mathbf{f}}$$

EM:

$$\boldsymbol{\theta}^{(0)} \longrightarrow \widetilde{\boldsymbol{\theta}} \longrightarrow \boxed{q_1(\mathbf{f}) = p(\mathbf{f} | \widetilde{\boldsymbol{\theta}}, \mathbf{g})} \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}}$$

$$\widetilde{\boldsymbol{\theta}} \uparrow \qquad\qquad q_1(\mathbf{f}) \downarrow$$

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

VBA:

$$\boldsymbol{\theta}^{(0)} \longrightarrow q_2(\boldsymbol{\theta}) \longrightarrow \boxed{q_1(\mathbf{f}) \propto \exp\left[ \langle \ln p(\mathbf{f}, \boldsymbol{\theta} | \mathbf{g}) \rangle_{q_2(\boldsymbol{\theta})} \right]} \longrightarrow q_1(\mathbf{f}) \longrightarrow \widehat{\mathbf{f}}$$

$$q_2(\boldsymbol{\theta}) \uparrow \qquad\qquad q_1(\mathbf{f}) \downarrow$$

$$\widehat{\boldsymbol{\theta}} \longleftarrow q_2(\boldsymbol{\theta}) \longleftarrow \boxed{q_2(\boldsymbol{\theta}) \propto \exp\left[ \langle \ln p(\mathbf{f}, \boldsymbol{\theta} | \mathbf{g}) \rangle_{q_1(\mathbf{f})} \right]} \longleftarrow q_1(\mathbf{f})$$

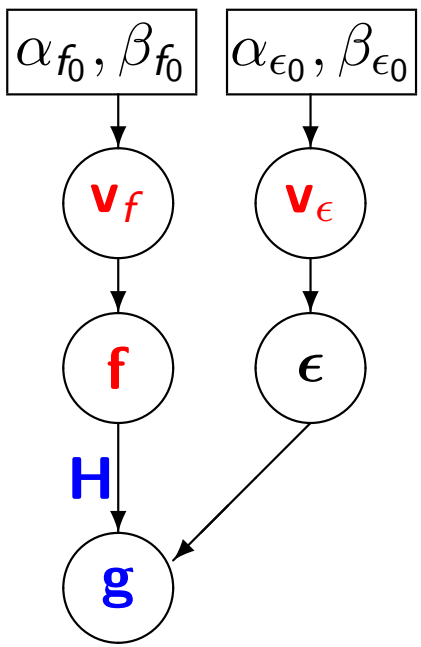
# Non stationary noise and sparsity enforcing model

– Non stationary noise:

$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon},\ \epsilon_i \sim \mathcal{N}(\epsilon_i|0, v_{\epsilon_i}) \rightarrow \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|0, \mathbf{V}_\epsilon = \text{diag}\,[v_{\epsilon 1}, \cdots, v_{\epsilon M}])$

– Student-t prior model and its equivalent IGSM :

$f_j|v_{f_j} \sim \mathcal{N}(f_j|0, v_{f_j})$ and $v_{f_j} \sim \mathcal{IG}(v_{f_j}|\alpha_{f_0}, \beta_{f_0}) \rightarrow f_j \sim \mathcal{S}t(f_j|\alpha_{f_0}, \beta_{f_0})$




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

$$\begin{cases} p(\mathbf{v}_\epsilon) = \prod_i \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(\mathbf{v}_f) = \prod_i \mathcal{IG}(v_{f_j}|\alpha_{f_0}, \beta_{f_0}) \end{cases}$$

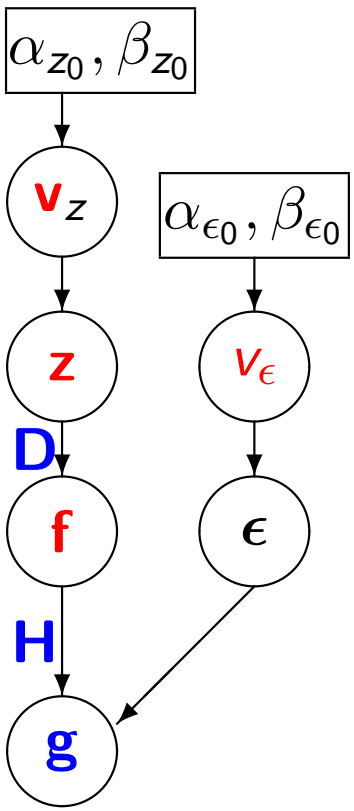
$$p(\mathbf{f}, \mathbf{v}_\epsilon, \mathbf{v}_f|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{v}_\epsilon)\, p(\mathbf{f}|\mathbf{v}_f)\, p(\mathbf{v}_\epsilon)\, p(\mathbf{v}_f)$$

Objective: Infer $(\mathbf{f}, \mathbf{v}_\epsilon, \mathbf{v}_f)$

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

# Sparse model in a Transform domain 1

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




$$\begin{cases} p(\mathbf{g}|\mathbf{z}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{D}\mathbf{f}, v_\epsilon \mathbf{I}) \\ p(\mathbf{z}|\mathbf{v}_z) = \mathcal{N}(\mathbf{z}|0, \mathbf{V}_z), \qquad \mathbf{V}_z = \text{diag}\,[\mathbf{v}_z] \\ p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(\mathbf{v}_z) = \prod_i \mathcal{IG}(v_{z_j}|\alpha_{z_0}, \beta_{z_0}) \end{cases}$$

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

– JMAP:

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

Alternate optimization:

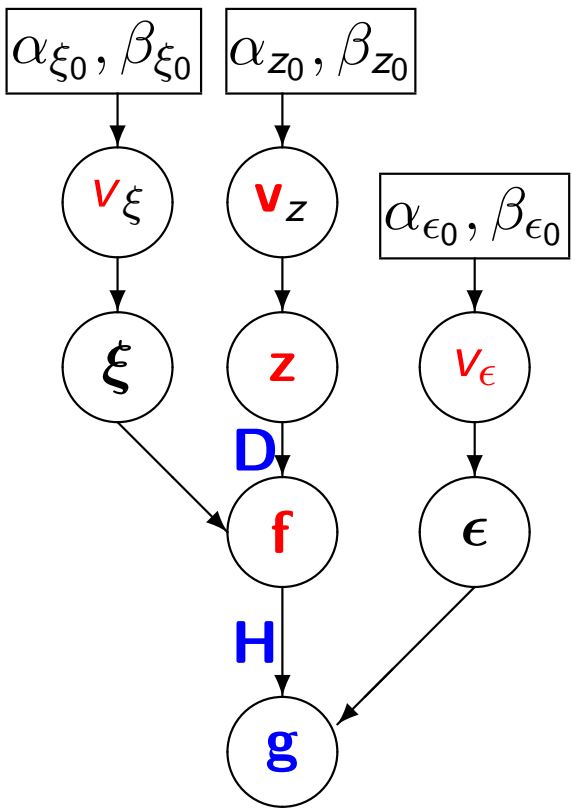
$$\begin{cases} \widehat{\mathbf{z}} = \arg\min_{\mathbf{z}} \{J(\mathbf{z})\} \text{ with:} \\ J(\mathbf{z}) = \frac{1}{2\hat{v}_\epsilon}\|\mathbf{g} - \mathbf{H}\mathbf{D}\mathbf{z}\|^2 + \|\mathbf{V}_z{}^{-1/2}\mathbf{z}\|^2 \\ \widehat{v}_{z_j} = \frac{\beta_{z_0} + \widehat{z}_j^2}{\alpha_{z_0} + 1/2} \\ \hat{v}_\epsilon = \frac{\beta_{\epsilon_0} + \|\mathbf{g} - \mathbf{H}\mathbf{D}\widehat{\mathbf{z}}\|^2}{\alpha_{\epsilon_0} + M/2} \end{cases}$$

– VBA: Approximate

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

Alternate optimization.

# Sparse model in a Transform domain 2




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

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

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

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

– JMAP:

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

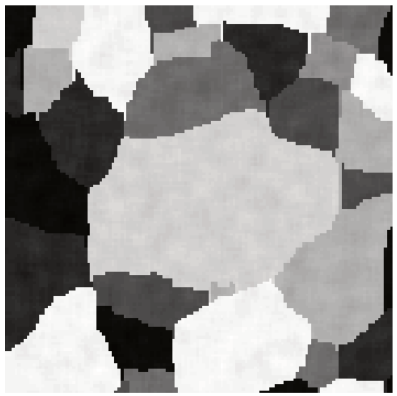
Alternate optimization.

– VBA: Approximate

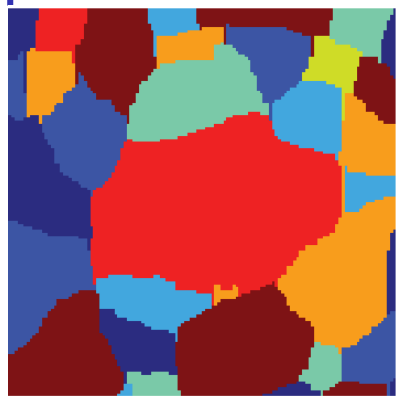
$p(\mathbf{f}, \mathbf{z}, v_\epsilon, \mathbf{v}_z, v_\xi|\mathbf{g})$ by $q_1(\mathbf{f})\, q_2(\mathbf{z})\, q_3(v_\epsilon)\, q_4(\mathbf{v}_z)\, q_5(v_\xi)$

Alternate optimization.

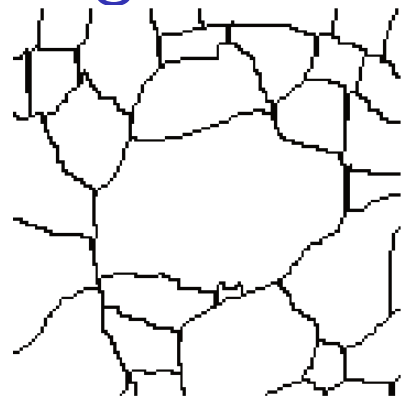
# Gauss-Markov-Potts prior models for images



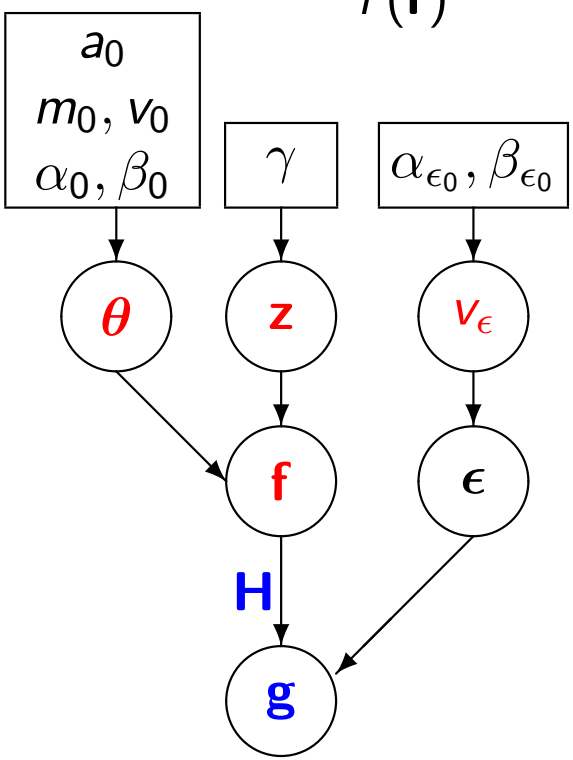
$f(\mathbf{r})$



$z(\mathbf{r})$



$c(\mathbf{r}) = 1 - \delta(z(\mathbf{r}) - z(\mathbf{r}'))$




$$
\begin{cases}
\mathbf{g} = \mathbf{H}\mathbf{f} + \epsilon \\
p(\mathbf{g}|\mathbf{f}, v_\epsilon) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, v_\epsilon \mathbf{I}) \\
p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0})
\end{cases}
$$

$$
\begin{cases}
p(f(\mathbf{r})|z(\mathbf{r}) = k, m_k, v_k) = \mathcal{N}(f(\mathbf{r})|m_k, v_k) \\
p(\mathbf{f}|\mathbf{z}, \boldsymbol{\theta}) = \sum_k \prod_{\mathbf{r} \in \mathcal{R}_k} a_k \mathcal{N}(f(\mathbf{r})|m_k, v_k), \\
\qquad \boldsymbol{\theta} = \{(a_k, m_k, v_k), k = 1, \cdots, K\} \\
p(\boldsymbol{\theta}) = \mathcal{D}(\mathbf{a}|a_0)\mathcal{N}(\mathbf{a}|m_0, v0)\mathcal{IG}(\mathbf{v}|\alpha_0, \beta_0) \\
p(\mathbf{z}|\gamma) \propto \exp\left[\gamma \sum_{\mathbf{r}} \sum_{\mathbf{r}' \in \mathcal{N}(\mathbf{r})} \delta(z(\mathbf{r}) - z(\mathbf{r}'))\right] \quad \text{Potts MRF}
\end{cases}
$$

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

MCMC: Gibbs Sampling

VBA: Alternate optimization.

# Mixture Models

1. Mixture models
2. Different problems related to classification and clustering
   - Training
   - Supervised classification
   - Semi-supervised classification
   - Clustering or unsupervised classification
3. Mixture of Gaussian (MoG)
4. Mixture of Student-t (MoSt)
5. Variational Bayesian Approximation (VBA)
6. VBA for Mixture of Gaussian
7. VBA for Mixture of Student-t
8. Conclusion

# Mixture models

- General mixture model

$$p(\mathbf{x}|\mathbf{a},\mathbf{\Theta},K)=\sum_{k=1}^{K}a_k\,p_k(\mathbf{x}_k|\boldsymbol{\theta}_k),\quad 0<a_k<1,\ \sum_{k=1}^{K}a_k=1$$

- Same family $p_k(\mathbf{x}_k|\boldsymbol{\theta}_k)=p(\mathbf{x}_k|\boldsymbol{\theta}_k),\ \forall k$
- Gaussian $p(\mathbf{x}_k|\boldsymbol{\theta}_k)=\mathcal{N}(\mathbf{x}_k|\boldsymbol{\mu}_k,\mathbf{V}_k)$ with $\boldsymbol{\theta}_k=(\boldsymbol{\mu}_k,\mathbf{V}_k)$
- Data $\mathbf{X}=\{\mathbf{x}_n,n=1,\cdots,N\}$ where each element $\mathbf{x}_n$ can be in one of the $K$ classes $c_n$.
- $a_k=p(c_n=k)$, $\mathbf{a}=\{a_k,k=1,\cdots,K\}$, $\mathbf{\Theta}=\{\boldsymbol{\theta}_k,k=1,\cdots,K\}$, $\mathbf{c}=\{c_n,n=1,\cdots,N\}$

$$p(\mathbf{X},\mathbf{c}|\mathbf{a},\mathbf{\Theta})=\prod_{n=1}^{N}p(\mathbf{x}_n,c_n=k|a_k,\boldsymbol{\theta}_k)$$

# Different problems

- Training:
  Given a set of (training) data $\mathbf{X}$ and classes $\mathbf{c}$, estimate the parameters $\mathbf{a}$ and $\mathbf{\Theta}$.
- Supervised classification:
  Given a sample $\mathbf{x}_m$ and the parameters $K$, $\mathbf{a}$ and $\mathbf{\Theta}$ determine its class

$$k^* = \arg\max_k \{p(c_m = k|\mathbf{x}_m, \mathbf{a}, \mathbf{\Theta}, K)\}.$$

- Semi-supervised classification (Proportions are not known):
  Given sample $\mathbf{x}_m$ and the parameters $K$ and $\mathbf{\Theta}$, determine its class

$$k^* = \arg\max_k \{p(c_m = k|\mathbf{x}_m, \mathbf{\Theta}, K)\}.$$

- Clustering or unsupervised classification (Number of classes K is not known):
  Given a set of data $\mathbf{X}$, determine $K$ and $\mathbf{c}$.

# Training

- Given a set of (training) data $\mathbf{X}$ and classes $\mathbf{c}$, estimate the parameters $\mathbf{a}$ and $\mathbf{\Theta}$.
- Maximum Likelihood (ML):

$$(\widehat{\mathbf{a}}, \widehat{\mathbf{\Theta}}) = \underset{(\mathbf{a},\mathbf{\Theta})}{\arg\max} \{p(\mathbf{X}, \mathbf{c}|\mathbf{a}, \mathbf{\Theta}, K)\}.$$

- Bayesian: Assign priors $p(\mathbf{a}|K)$ and $p(\mathbf{\Theta}|K) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_k)$ and write the expression of the joint posterior laws:

$$p(\mathbf{a}, \mathbf{\Theta}|\mathbf{X}, \mathbf{c}, K) = \frac{p(\mathbf{X}, \mathbf{c}|\mathbf{a}, \mathbf{\Theta}, K)\, p(\mathbf{a}|K)\, p(\mathbf{\Theta}|K)}{p(\mathbf{X}, \mathbf{c}|K)}$$

where

$$p(\mathbf{X}, \mathbf{c}|K) = \iint p(\mathbf{X}, \mathbf{c}|\mathbf{a}, \mathbf{\Theta}|K) p(\mathbf{a}|K)\, p(\mathbf{\Theta}|K)\, \mathrm{d}\mathbf{a}\, \mathrm{d}\mathbf{\Theta}$$

- Infer on $\mathbf{a}$ and $\mathbf{\Theta}$ either as the Maximum A Posteriori (MAP) or Posterior Mean (PM).

# Supervised classification

- Given a sample $\mathbf{x}_m$ and the parameters $K$, $\mathbf{a}$ and $\mathbf{\Theta}$ determine

$$p(c_m = k|\mathbf{x}_m, \mathbf{a}, \mathbf{\Theta}, K) = \frac{p(\mathbf{x}_m, c_m = k|\mathbf{a}, \mathbf{\Theta}, K)}{p(\mathbf{x}_m|\mathbf{a}, \mathbf{\Theta}, K)}$$

where $p(\mathbf{x}_m, c_m = k|\mathbf{a}, \mathbf{\Theta}, K) = a_k p(\mathbf{x}_m|\boldsymbol{\theta}_k)$ and

$$p(\mathbf{x}_m|\mathbf{a}, \mathbf{\Theta}, K) = \sum_{k=1}^{K} a_k \, p(\mathbf{x}_m|\boldsymbol{\theta}_k)$$

- Best class $k^*$:

$$k^* = \arg\max_k \{p(c_m = k|\mathbf{x}_m, \mathbf{a}, \mathbf{\Theta}, K)\}$$

# Semi-supervised classification

- Given sample $\mathbf{x}_m$ and the parameters $K$ and $\boldsymbol{\Theta}$ (not the proportions $\mathbf{a}$), determine the probabilities

$$p(c_m = k|\mathbf{x}_m, \boldsymbol{\Theta}, K) = \frac{p(\mathbf{x}_m, c_m = k|\boldsymbol{\Theta}, K)}{p(\mathbf{x}_m|\boldsymbol{\Theta}, K)}$$

where

$$p(\mathbf{x}_m, c_m = k|\boldsymbol{\Theta}, K) = \int p(\mathbf{x}_m, c_m = k|\mathbf{a}, \boldsymbol{\Theta}, K)p(\mathbf{a}|K)\,\mathrm{d}\mathbf{a}$$

and

$$p(\mathbf{x}_m|\boldsymbol{\Theta}, K) = \sum_{k=1}^{K} p(\mathbf{x}_m, c_m = k|\boldsymbol{\Theta}, K)$$

- Best class $k^*$, for example the MAP solution:

$$k^* = \arg\max_k \{p(c_m = k|\mathbf{x}_m, \boldsymbol{\Theta}, K)\}.$$

# Clustering or non-supervised classification

- Given a set of data $\mathbf{X}$, determine $K$ and $\mathbf{c}$.
- Determination of the number of classes:

$$p(K = L|\mathbf{X}) = \frac{p(\mathbf{X}, K = L)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|K = L)\, p(K = L)}{p(\mathbf{X})}$$

and

$$p(\mathbf{X}) = \sum_{L=1}^{L_0} p(K = L)\, p(\mathbf{X}|K = L),$$

where $L_0$ is the a priori maximum number of classes and

$$p(\mathbf{X}|K = L) = \int \int \prod_{n} \prod_{k=1}^{L} a_k p(\mathbf{x}_n, c_n = k|\theta_k) p(\mathbf{a}|K)\, p(\mathbf{\Theta}|K)\, \mathrm{d}\mathbf{a}\, \mathrm{d}\mathbf{\Theta}.$$

- When $K$ and $\mathbf{c}$ are determined, we can also determine the characteristics of those classes $\mathbf{a}$ and $\mathbf{\Theta}$.

# Mixture of Gaussian and Mixture of Student-t

$$p(\mathbf{x}|\mathbf{a},\boldsymbol{\Theta},K)=\sum_{k=1}^{K}a_k\,p(\mathbf{x}_k|\boldsymbol{\theta}_k),\quad 0<a_k<1,\ \sum_{k=1}^{K}a_k=1$$

- Mixture of Gaussian (MoG)

$$p(\mathbf{x}_k|\boldsymbol{\theta}_k)=\mathcal{N}(\mathbf{x}_k|\boldsymbol{\mu}_k,\mathbf{V}_k),\quad \boldsymbol{\theta}_k=(\boldsymbol{\mu}_k,\mathbf{V}_k)$$

$$\mathcal{N}(\mathbf{x}_k|\boldsymbol{\mu}_k,\mathbf{V}_k)=(2\pi)^{-\frac{p}{2}}\,|\mathbf{V}_k|^{-\frac{1}{2}}\exp\left[\frac{1}{2}(\mathbf{x}_k-\boldsymbol{\mu}_k)'\mathbf{V}_k^{-1}(\mathbf{x}_k-\boldsymbol{\mu}_k)\right]$$

- Mixture of Student-t (MoSt)

$$p(\mathbf{x}_k|\boldsymbol{\theta}_k)=\mathcal{T}(\mathbf{x}_k|\nu_k,\boldsymbol{\mu}_k,\mathbf{V}_k),\quad \boldsymbol{\theta}_k=(\nu_k,\boldsymbol{\mu}_k,\mathbf{V}_k)$$

$$\mathcal{T}(\mathbf{x}_k|\nu,\boldsymbol{\mu}_k,\mathbf{V}_k)=\frac{\Gamma\left[\frac{(\nu_k+p)}{2}\right]}{\Gamma(\frac{\nu_k}{2})\nu^{\frac{p}{2}}\pi^{\frac{p}{2}}}\,|\mathbf{V}_k|^{-\frac{1}{2}}\left[1+\frac{1}{\nu}(\mathbf{x}_k-\boldsymbol{\mu}_k)'\mathbf{V}_k^{-1}(\mathbf{x}_k-\boldsymbol{\mu}_k)\right]^{-\frac{(\nu+p)}{2}}$$

# Mixture of Student-t model

- Student-t and its Infinite Gaussian Scaled Model (IGSM):

$$\mathcal{T}(\mathbf{x}|\nu,\boldsymbol{\mu},\mathbf{V}) = \int_0^\infty \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},u^{-1}\mathbf{V})\,\mathcal{G}(u|\frac{\nu}{2},\frac{\nu}{2})\,\mathrm{d}z$$

where

$$\begin{aligned}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{V}) &= |2\pi\mathbf{V}|^{-\frac{1}{2}}\exp\left[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\mathbf{V}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]\\ &= |2\pi\mathbf{V}|^{-\frac{1}{2}}\exp\left[-\tfrac{1}{2}\mathrm{Tr}\left\{(\mathbf{x}-\boldsymbol{\mu})\mathbf{V}^{-1}(\mathbf{x}-\boldsymbol{\mu})'\right\}\right]\end{aligned}$$

and

$$\mathcal{G}(u|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}u^{\alpha-1}\exp\left[-\beta u\right].$$

- Mixture of generalized Student-t: $\mathcal{T}(\mathbf{x}|\alpha,\beta,\boldsymbol{\mu},\mathbf{V})$

$$p(\mathbf{x}|\{a_k,\boldsymbol{\mu}_k,\mathbf{V}_k,\alpha_k,\beta_k\},K) = \sum_{k=1}^{K} a_k\,\mathcal{T}(\mathbf{x}_n|\alpha_k,\beta_k,\boldsymbol{\mu}_k,\mathbf{V}_k).$$

# Mixture of Gaussian model

- Introducing $z_{nk} \in \{0,1\}$, $\mathbf{z}_k = \{z_{nk}, n = 1, \cdots, N\}$, $\mathbf{Z} = \{z_{nk}\}$ with $P(z_{nk} = 1) = P(c_n = k) = a_k$, $\boldsymbol{\theta}_k = \{a_k, \boldsymbol{\mu}_k, \mathbf{V}_k\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k, k = 1, \cdots, K\}$
- Assigning the priors $p(\boldsymbol{\Theta}) = \prod_k p(\boldsymbol{\theta}_k)$, we can write:

$$p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \boldsymbol{\Theta}|K) = \prod_n \sum_k a_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{V}_k)\,(1-\delta(z_{nk}))\, \prod_k p(\boldsymbol{\theta}_k)$$

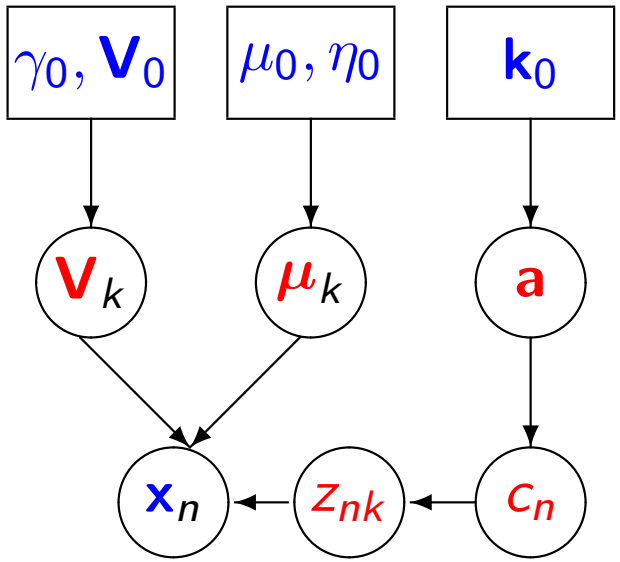
$$p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \boldsymbol{\Theta}|K) = \prod_n \prod_k [a_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{V}_k)]^{z_{nk}}\, p(\boldsymbol{\theta}_k)$$

- Joint posterior law:

$$p(\mathbf{c}, \mathbf{Z}, \boldsymbol{\Theta}|\mathbf{X}, K) = \frac{p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \boldsymbol{\Theta}|K)}{p(\mathbf{X}|K)}.$$

- The main task now is to propose some approximations to it in such a way that we can use it easily in all the above mentioned tasks of classification or clustering.

# Hierarchical graphical model for Mixture of Gaussian




$$\begin{cases} p(\mathbf{a}) = \mathcal{D}(\mathbf{a}|\mathbf{k}_0) \\ p(\mu_k|\mathbf{V}_k) = \mathcal{N}(\mu_k|\mu_0 1, \eta_0{}^{-1}\mathbf{V}_k) \\ p(\mathbf{V}_k) = \mathcal{IW}(\mathbf{V}_k|\gamma_0, \mathbf{V}_0) \\ P(z_{nk} = 1) = P(c_n = k) = a_k \end{cases}$$

$$p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \mathbf{\Theta}|K) = \prod_n \prod_k [a_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{V}_k)]^{z_{nk}}$$

$$p(a_k)p(\mu_k|\mathbf{V}_k)p(\mathbf{V}_k)$$

# Mixture of Student-t model

- Introducing $\mathbf{U} = \{u_{nk}\}$
  $\boldsymbol{\theta}_k = \{\alpha_k, \beta_k, \mathbf{a}_k, \boldsymbol{\mu}_k, \mathbf{V}_k\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k, k = 1, \cdots, K\}$
- Assigning the priors $p(\boldsymbol{\Theta}) = \prod_k p(\boldsymbol{\theta}_k)$, we can write:

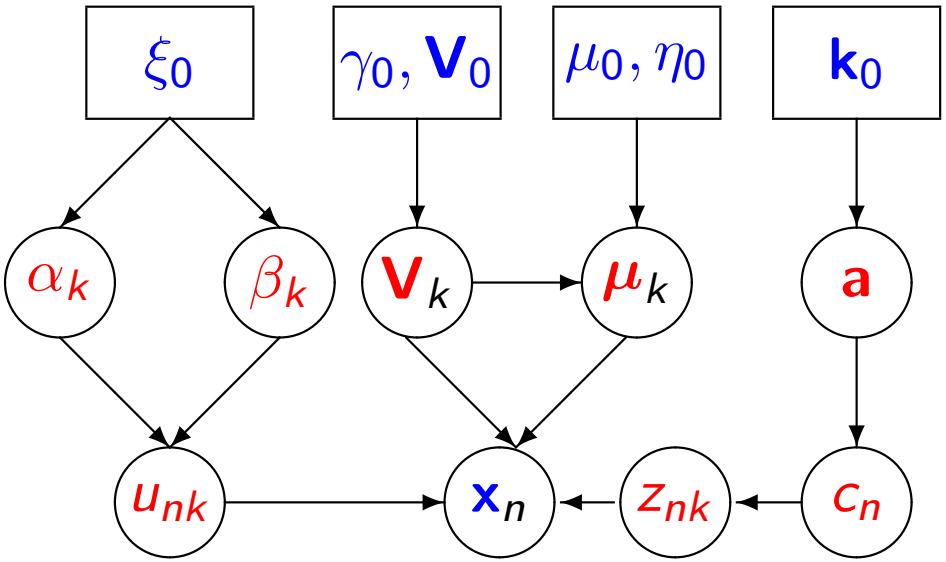
$$p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \mathbf{U}, \boldsymbol{\Theta}|K) = \prod_n \prod_k \left[a_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, u_{n,k}{}^{-1}\mathbf{V}_k)\, \mathcal{G}(u_{nk}|\alpha_k, \beta_k)\right]^{z_{nk}} p(\boldsymbol{\theta}_k)$$

- Joint posterior law:

$$p(\mathbf{c}, \mathbf{Z}, \mathbf{U}, \boldsymbol{\Theta}|\mathbf{X}, K) = \frac{p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \mathbf{U}, \boldsymbol{\Theta}|K)}{p(\mathbf{X}|K)}.$$

- The main task now is to propose some approximations to it in such a way that we can use it easily in all the above mentioned tasks of classification or clustering.

# Hierarchical graphical model for Mixture of Student-t




$$
\begin{cases}
p(\mathbf{a}) = \mathcal{D}(\mathbf{a}|\mathbf{k}_0) \\
p(\boldsymbol{\mu}_k|\mathbf{V}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\mu_0 1, \eta_0{}^{-1}\mathbf{V}_k) \\
p(\mathbf{V}_k) = \mathcal{IW}(\mathbf{V}_k|\gamma_0, \mathbf{V}_0) \\
p(\alpha_k) = \mathcal{E}(\alpha_k|\zeta_0) = \mathcal{G}(\alpha_k|1, \zeta_0) \\
p(\beta_k) = \mathcal{E}(\beta_k|\zeta_0) = \mathcal{G}(\beta_k|1, \zeta_0) \\
P(z_{nk} = 1) = P(c_n = k) = a_k \\
p(u_{nk}) = \mathcal{G}(u_{nk}|\alpha_k, \beta_k)
\end{cases}
$$

$$
p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \mathbf{U}, \boldsymbol{\Theta}|K) = \prod_n \prod_k [a_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{V}_k)\mathcal{G}(u_{nk}|\alpha_k, \beta_k)]^{z_{nk}}
$$

$$
p(a_k)p(\boldsymbol{\mu}_k|\mathbf{V}_k)p(\mathbf{V}_k)p(\alpha_k)p(\beta_k)
$$

# Variational Bayesian Approximation (VBA)

- Main idea: to propose easy computational approximations: $q(\mathbf{c}, \mathbf{Z}, \mathbf{\Theta}) = q(\mathbf{c}, \mathbf{Z})q(\mathbf{\Theta})$ for $p(\mathbf{c}, \mathbf{Z}, \mathbf{\Theta}|\mathbf{X}, K)$ for MoG model, or
  $q(\mathbf{c}, \mathbf{Z}, \mathbf{U}, \mathbf{\Theta}) = q(\mathbf{c}, \mathbf{Z}, \mathbf{U})q(\mathbf{\Theta})$ for $p(\mathbf{c}, \mathbf{Z}, \mathbf{U}, \mathbf{\Theta}|\mathbf{X}, K)$ for MoSt model.
- Criterion:

$$\mathrm{KL}(q : p) = -\mathcal{F}(q) + \ln p(\mathbf{X}|K)$$

  where

$$\mathcal{F}(q) = \langle -\ln p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \mathbf{\Theta}|K)\rangle_q$$

  or

$$\mathcal{F}(q) = \langle -\ln p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \mathbf{U}, \mathbf{\Theta}|K)\rangle_q$$

- Maximizing $\mathcal{F}(q)$ or minimizing $\mathrm{KL}(q : p)$ are equivalent and both give un upper bound to the evidence of the model $\ln p(\mathbf{X}|K)$.
- When the optimum $q^*$ is obtained, $\mathcal{F}(q^*)$ can be used as a criterion for model selection.

# Proposed VBA for Mixture of Student-t priors model

- Dirichlet

$$\mathcal{D}(\mathbf{a}|\mathbf{k}) = \frac{\Gamma(\sum_l k_k)}{\prod_l \Gamma(k_l)} \prod_l a_l^{k_l - 1}$$

- Exponential

$$\mathcal{E}(t|\zeta_0) = \zeta_0 \exp\left[-\zeta_0 t\right]$$

- Gamma

$$\mathcal{G}(t|a,b) = \frac{b^a}{\Gamma(a)} t^{a-1} \exp\left[-bt\right]$$

- Inverse Wishart

$$\mathcal{IW}(\mathbf{V}|\gamma,\gamma\mathbf{\Delta}) = \frac{|\frac{1}{2}\mathbf{\Delta}|^{\gamma/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left\{\mathbf{\Delta}\mathbf{V}^{-1}\right\}\right]}{\Gamma_D(\gamma/2)|\mathbf{V}|^{\frac{\gamma+D+1}{2}}}.$$

# Expressions of $q$

$$
\begin{aligned}
q(\mathbf{c}, \mathbf{Z}, \mathbf{\Theta}) =& q(\mathbf{c}, \mathbf{Z})\, q(\mathbf{\Theta}) \\
&= \prod_n \prod_k [q(c_n = k|z_{nk})\, q(z_{nk})] \\
&\prod_k [q(\alpha_k)\, q(\beta_k)\, q(\boldsymbol{\mu}_k|\mathbf{V}_k)\, q(\mathbf{V}_k)]\, q(\mathbf{a}).
\end{aligned}
$$

with:

$$
\begin{cases}
q(\mathbf{a}) = \mathcal{D}(\mathbf{a}|\tilde{\mathbf{k}}), \quad \tilde{\mathbf{k}} = [\tilde{k}_1, \cdots, \tilde{k}_K] \\
q(\alpha_k) = \mathcal{G}(\alpha_k|\tilde{\xi}_k, \tilde{\eta}_k) \\
q(\beta_k) = \mathcal{G}(\beta_k|\tilde{\xi}_k, \tilde{\eta}_k) \\
q(\boldsymbol{\mu}_k|\mathbf{V}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\tilde{\boldsymbol{\mu}}, \tilde{\eta}^{-1}\mathbf{V}_k) \\
q(\mathbf{V}_k) = \mathcal{IW}(\mathbf{V}_k|\tilde{\gamma}, \tilde{\Sigma})
\end{cases}
$$

With these choices, we have

$$
\mathcal{F}(q(\mathbf{c}, \mathbf{Z}, \mathbf{\Theta})) = \langle \ln p(\mathbf{X}, \mathbf{c}, \mathbf{Z}, \mathbf{\Theta}|K) \rangle_{q(\mathbf{c},\mathbf{Z},\mathbf{\Theta})} = \prod_k \prod_n \mathcal{F}_{1_{kn}} + \prod_k \mathcal{F}_{2_k}
$$

$$
\begin{aligned}
\mathcal{F}_{1_{kn}} &= \langle \ln p(\mathbf{x}_n, c_n, z_{nk}, \boldsymbol{\theta}_k) \rangle_{q(c_n=k|z_{nk})q(z_{nk})} \\
\mathcal{F}_{2_k} &= \langle \ln p(\mathbf{x}_n, c_n, z_{nk}, \boldsymbol{\theta}_k) \rangle_{q(\boldsymbol{\theta})}
\end{aligned}
$$

# VBA Algorithm step

Expressions of the updating expressions of the tilded parameters are obtained by following three steps:

- E step: Optimizing $\mathcal{F}$ with respect to $q(\mathbf{c}, \mathbf{Z})$ when keeping $q(\mathbf{\Theta})$ fixed, we obtain the expression of $q(c_n = k|z_{nk}) = \tilde{a}_k$, $q(z_{nk}) = \mathcal{G}(z_{nk}|\widetilde{\alpha}_k, \widetilde{\beta}_k)$.
- M step: Optimizing $\mathcal{F}$ with respect to $q(\mathbf{\Theta})$ when keeping $q(\mathbf{c}, \mathbf{Z})$ fixed, we obtain the expression of $q(\mathbf{a}) = \mathcal{D}(\mathbf{a}|\tilde{\mathbf{k}})$, $\quad \tilde{\mathbf{k}} = [\tilde{k}_1, \cdots, \tilde{k}_K]$, $q(\alpha_k) = \mathcal{G}(\alpha_k|\tilde{\zeta}_k, \tilde{\eta}_k)$, $q(\beta_k) = \mathcal{G}(\beta_k|\tilde{\zeta}_k, \tilde{\eta}_k)$, $q(\boldsymbol{\mu}_k|\mathbf{V}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\widetilde{\boldsymbol{\mu}}, \tilde{\eta}^{-1}\mathbf{V}_k)$, and $q(\mathbf{V}_k) = \mathcal{IW}(\mathbf{V}_k|\tilde{\gamma}, \tilde{\gamma}\tilde{\Sigma})$, which gives the updating algorithm for the corresponding tilded parameters.
- $\mathcal{F}$ evaluation: After each E step and M step, we can also evaluate the expression of $\mathcal{F}(q)$ which can be used for stopping rule of the iterative algorithm.
- Final value of $\mathcal{F}(q)$ for each value of $K$, noted $\mathcal{F}_k$, can be used as a criterion for model selection, i.e.; the determination of the number of clusters.

# VBA: choosing the good families for $q$

- Main question: We approximate $p(X)$ by $q(X)$. What are the quantities we have conserved?
  - a) Modes values: $\arg\max_x \{p(X)\} = \arg\max_x \{q(X)\}$ ?
  - b) Expected values: $E_p(X) = E_q(X)$ ?
  - c) Variances: $V_p(X) = V_q(X)$ ?
  - d) Entropies: $H_p(X) = H_q(X)$ ?
- Recent works shows some of these under some conditions.
- For example, if $p(x) = \frac{1}{Z}\exp\left[-\phi(x)\right]$ with $\phi(x)$ convex and symetric, properties a) and b) are satisfied.
- Unfortunately, this is not the case for variances or other moments.
- If $p$ is in the exponential family, then choosing appropriate conjugate priors, the structure of $q$ will be the same and we can obtain appropriate fast optimization algorithms.

# Conclusions

- Bayesian approach with Hierarchical prior model with hidden variables are very powerful tools for inverse problems and Machine Learning.
- The computational cost of all the sampling methods (MCMC and many others) are too high to be used in practical high dimensional applications.
- We explored VBA tools for effective approximate Bayesian computation.
- Application in different inverse problems in imaging system (3D X ray CT, Microwaves, PET, Ultrasound, Optical Diffusion Tomography (ODT), Acoustic source localization,...)
- Clustering and classification of a set of data are between the most important tasks in statistical researches for many applications such as data mining in biology.
- Mixture models are classical models for these tasks.
- We proposed to use a mixture of generalised Student-t distribution model for more robustness.
- To obtain fast algorithms and be able to handle large data