

Bayesian inference for Machine Learning, Inverse Problems and Big Data: from Basics to Computational algorithms

Ali Mohammad-Djafari

Laboratoire des Signaux et Systèmes (L2S)
UMR8506 CNRS-CentraleSupélec-UNIV PARIS SUD
SUPELEC, 91192 Gif-sur-Yvette, France

<http://lss.centralesupelec.fr>

Email: djafari@lss.supelec.fr

<http://djafari.free.fr>

<http://publicationslist.org/djafari>

Seminar, Aix-Marseille University, Marseille, Nov. 21-25, 2016.

Contents

1. Basic Bayes
 - ▶ Low dimensional case
 - ▶ High dimensional case
2. Bayes for Machine Learning (model selection and prediction)
3. Approximate Bayesian Computation (ABC)
 - ▶ Laplace approximation
 - ▶ Bayesian Information Criterion (BIC)
 - ▶ Variational Bayesian Approximation
 - ▶ Expectation Propagation (EP), MCMC, Exact Sampling, ...
4. Bayes for inverse problems
 - ▶ Computed Tomography: A Linear problem
 - ▶ Microwave imaging: A Bi-Linear problem
5. Some canonical problems in Machine Learning
 - ▶ Classification, Polynomial Regression, ...
 - ▶ Clustering with Gaussian Mixtures
 - ▶ Clustering with Student-t Mixtures
6. Conclusions

Basic Bayes

- ▶ Two related events A and B with probabilities $P(A, B)$, $P(A|B)$ and $P(B|A)$.
- ▶ Product rule: $P(A, B) = P(A|B)P(B) = P(B|A)P(A)$
- ▶ Sum rule: $P(A) = \sum_B P(A|B)P(B)$
- ▶ Bayes rule: $P(B|A) = \frac{P(A|B)P(B)}{\sum_B P(A|B)P(B)}$

- ▶ Two related variables X and Y with probability distributions: $P(X, Y)$, $P(Y|X)$ and $P(X|Y)$.
- ▶ Bayes rule: $P(X|Y) = \frac{P(Y|X)P(X)}{\sum_x P(Y|X)P(X)}$

- ▶ Two related continuous variables X and Y with probability density functions: $p(x, y)$, $p(y|x)$ and $p(x|y)$.
- ▶ Bayes rule: $p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x) dx}$

Basic Bayes for simple parametric models

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

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

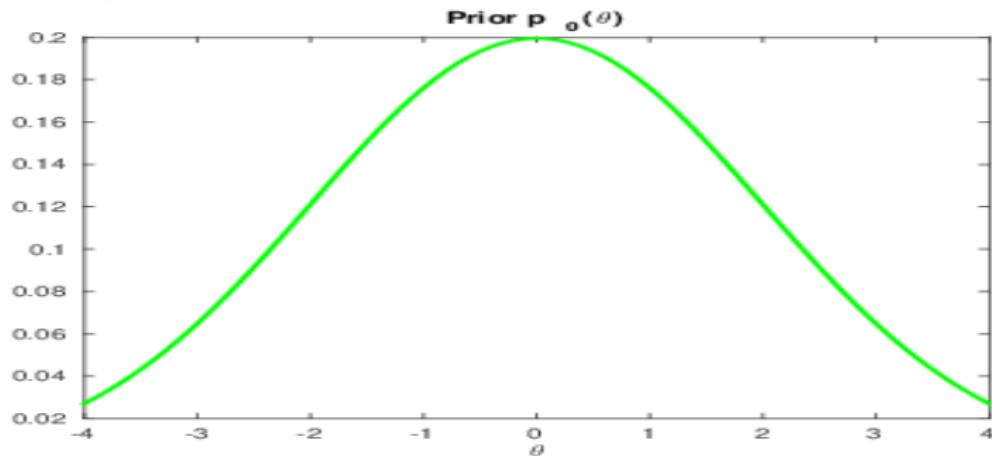
- ▶ Forward model (called also likelihood): $p(\mathbf{d}|\boldsymbol{\theta})$
- ▶ Prior knowledge: $p(\boldsymbol{\theta})$
- ▶ Posterior knowledge: $p(\boldsymbol{\theta}|\mathbf{d})$

Bayesian inference: simple one parameter case

$$d_i \sim p(d_i|\theta) = \mathcal{N}(d_i|\theta, 1), i = 1, \dots, M, \quad \theta \sim p(\theta) = \mathcal{N}(\theta|0, 2)$$

$$\mathcal{L}(\theta) = p(\mathbf{d}|\theta) = \prod_i \mathcal{N}(d_i|\theta, 1) \longrightarrow p(\theta|\mathbf{d}) \propto \mathcal{L}(\theta) p(\theta)$$

Prior: $p(\theta)$

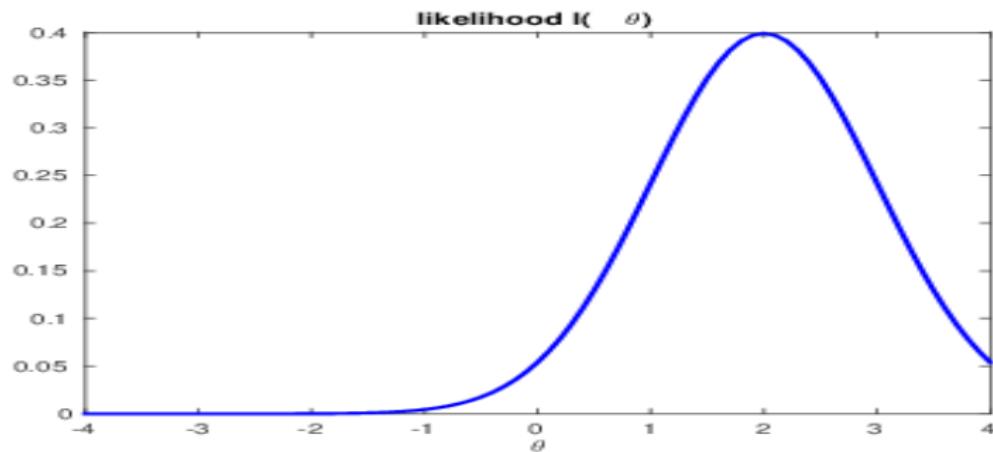


Bayesian inference: simple one parameter case

$$d_i \sim p(d_i|\theta) = \mathcal{N}(d_i|\theta, 1), i = 1, \dots, M, \quad \theta \sim p(\theta) = \mathcal{N}(\theta|0, 2)$$

$$M = 1, d_1 = 2; \mathcal{L}(\theta) = p(d|\theta) = \mathcal{N}(d_1|\theta, 1) = c\mathcal{N}(\theta||d_1, 1)$$

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

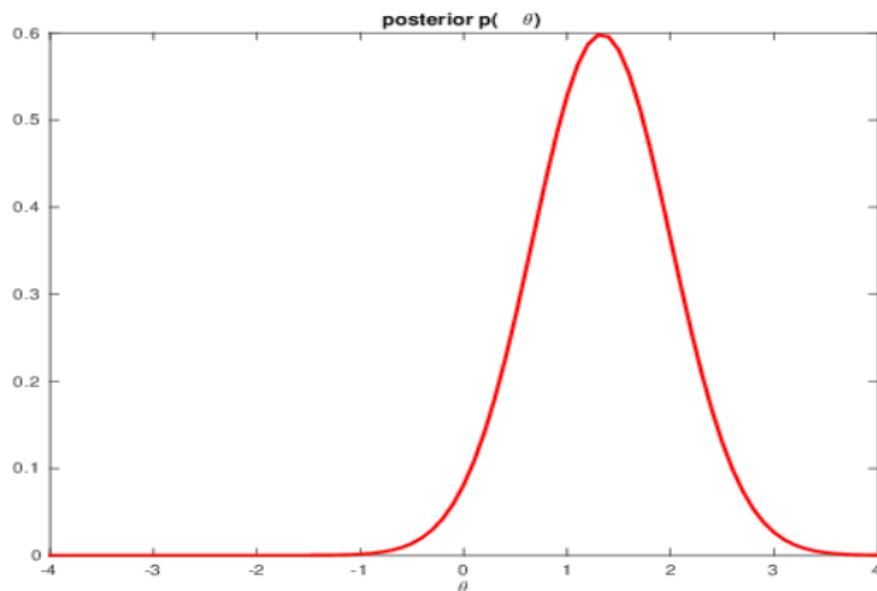


Bayesian inference: simple one parameter case

$$d_i \sim p(d_i|\theta) = \mathcal{N}(d_i|\theta, 1), i = 1, \dots, M, \quad \theta \sim p(\theta) = \mathcal{N}(\theta|0, 2)$$

$$M = 1, d_1 = 2; \mathcal{L}(\theta) = p(\mathbf{d}|\theta) = \mathcal{N}(d_1|\theta, 1) = c\mathcal{N}(\theta|d_1, 1)$$

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

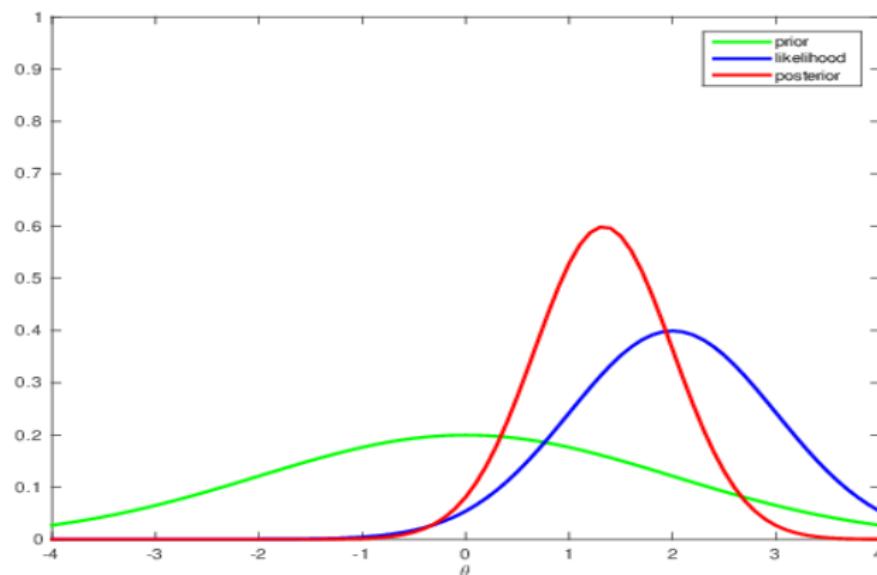


Bayesian inference: simple one parameter case

$$d_i \sim p(d_i|\theta) = \mathcal{N}(d_i|\theta, 1), i = 1, \dots, M, \quad \theta \sim p(\theta) = \mathcal{N}(\theta|0, 2)$$

$$M = 1, d_1 = 2; \mathcal{L}(\theta) = p(\mathbf{d}|\theta) = \mathcal{N}(\mathbf{d}_1|\theta, 1) = c\mathcal{N}(\theta|\mathbf{d}_1, 1)$$

Prior $p(\theta)$, Likelihood $\mathcal{L}(\theta)$ and Posterior $p(\theta|\mathbf{d})$:

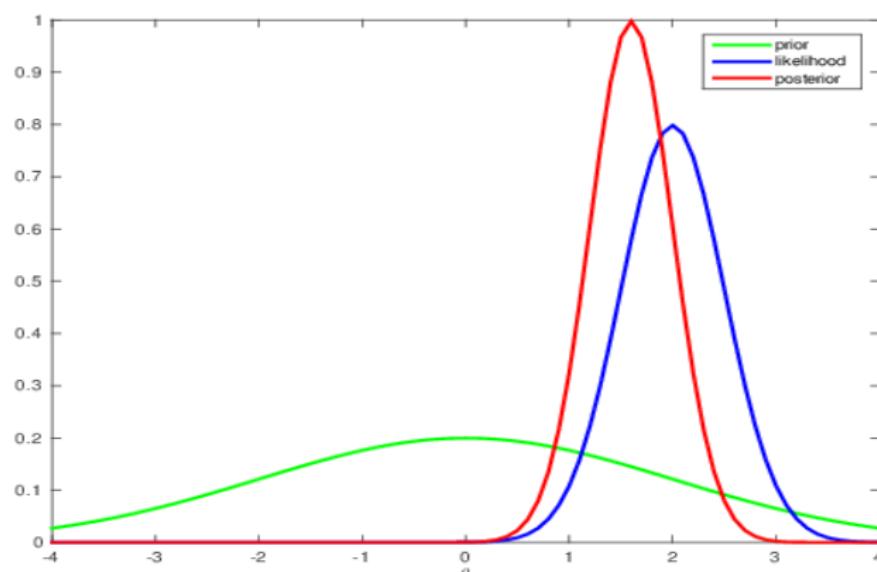


Bayesian inference: simple one parameter case

$$d_i \sim p(d_i|\theta) = \mathcal{N}(d_i|\theta, 1), i = 1, \dots, M, \quad \theta \sim p(\theta) = \mathcal{N}(\theta|0, 2)$$

$$M = 4, \bar{d} = 2; \mathcal{L}(\theta) = p(\mathbf{d}|\theta) = \prod_{i=1}^4 \mathcal{N}(d_i|\theta, 1) = c \mathcal{N}(\theta|\bar{d}, \frac{1}{\sqrt{4}})$$

Prior $p(\theta)$, Likelihood $\mathcal{L}(\theta)$ and Posterior $p(\theta|\mathbf{d})$:

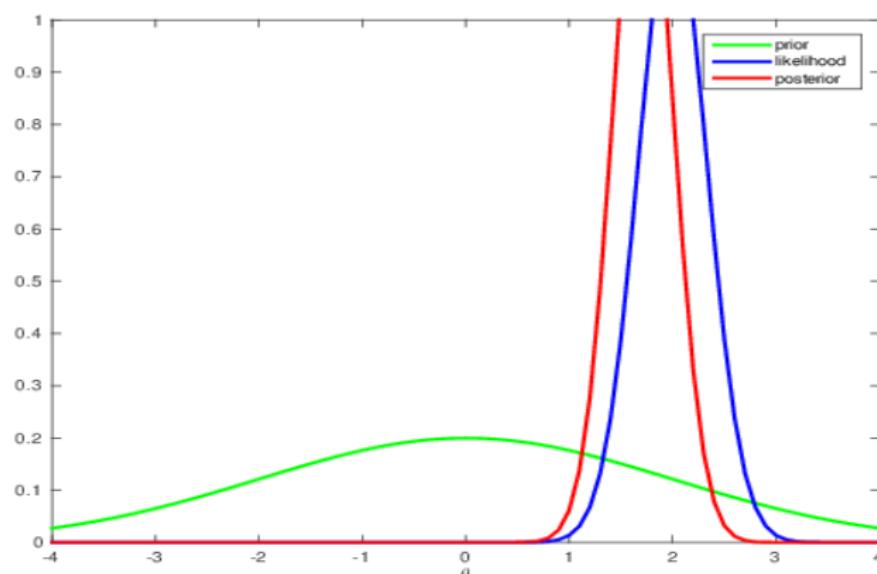


Bayesian inference: simple one parameter case

$$d_i \sim p(d_i|\theta) = \mathcal{N}(d_i|\theta, 1), i = 1, \dots, M, \quad \theta \sim p(\theta) = \mathcal{N}(\theta|0, 2)$$

$$M = 9, \bar{d} = 2; \mathcal{L}(\theta) = p(\mathbf{d}|\theta) = \prod_{i=1}^9 \mathcal{N}(d_i|\theta, 1) = c \mathcal{N}(\theta|\bar{d}, \frac{1}{\sqrt{9}})$$

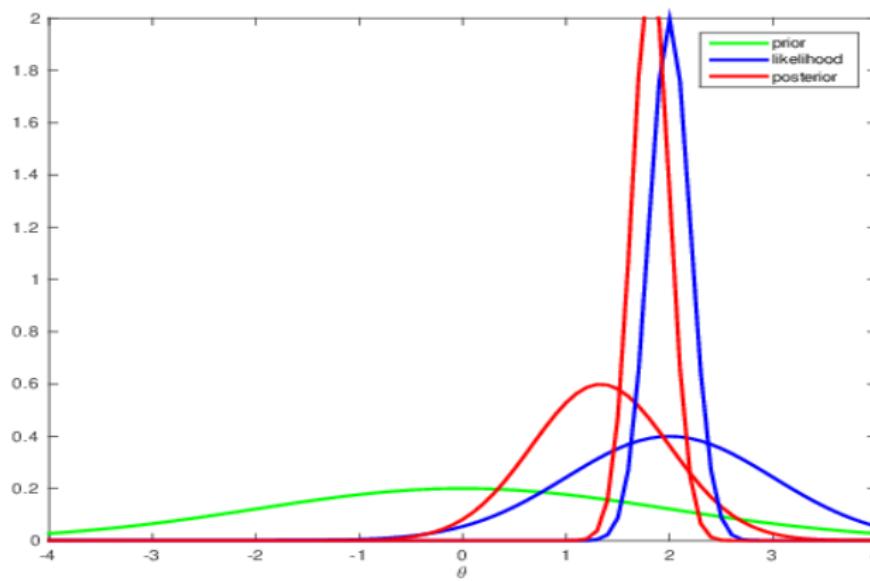
Prior $p(\theta)$, Likelihood $\mathcal{L}(\theta)$ and Posterior $p(\theta|\mathbf{d})$:



Bayesian inference: simple one parameter case

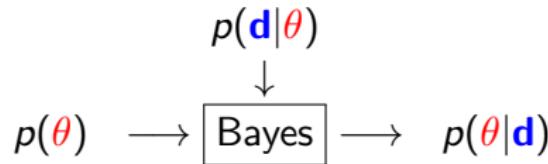
When the number of data increases the likelihood becomes more and more concentrated and, in general, the posterior converges to it.

Prior $p(\theta)$, Likelihood $\mathcal{L}(\theta)$ and Posterior $p(\theta|\mathbf{d})$:



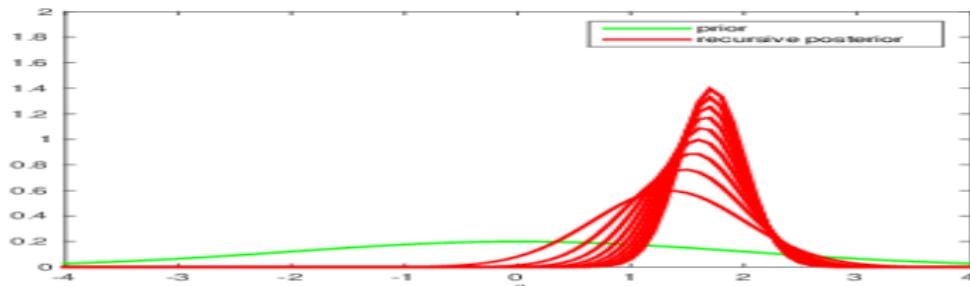
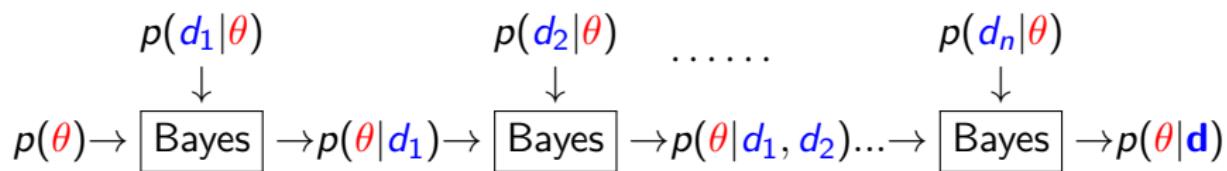
Recursive Bayes

- Direct



- Recursive

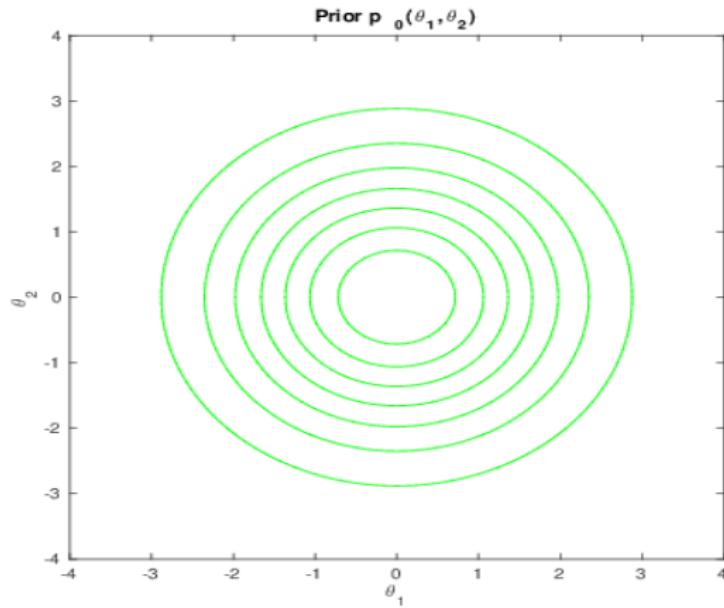
$$p(\theta|\mathbf{d}) \propto \prod_i p(d_i|\theta)p(\theta) \propto \left[[p(\theta)p(d_1|\theta)] p(d_2|\theta) \cdots p(d_n|\theta) \right]$$



Bayesian inference: simple two parameters case

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

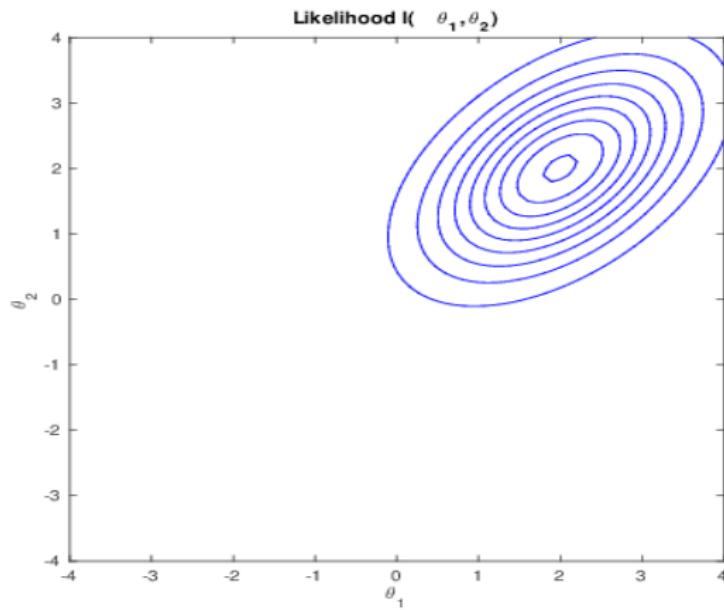
Prior: $p(\theta_1, \theta_2) = \mathcal{N}(\theta_1|0, 1)\mathcal{N}(\theta_2|0, 1)$



Bayesian inference: simple one parameter case

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

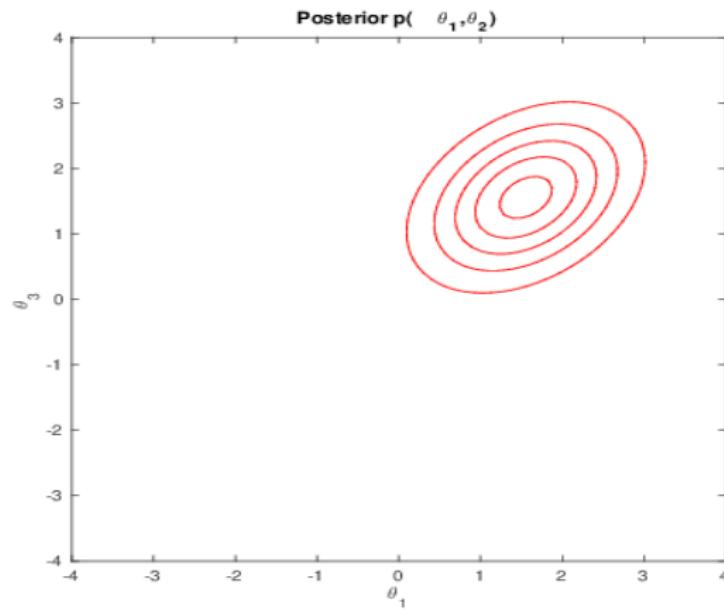
Likelihood: $\mathcal{L}(\theta_1, \theta_2) = p(\mathbf{d}|\theta_1, \theta_2)$, $d_1 = d_2 = 2$



Bayesian inference: simple one parameter case

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

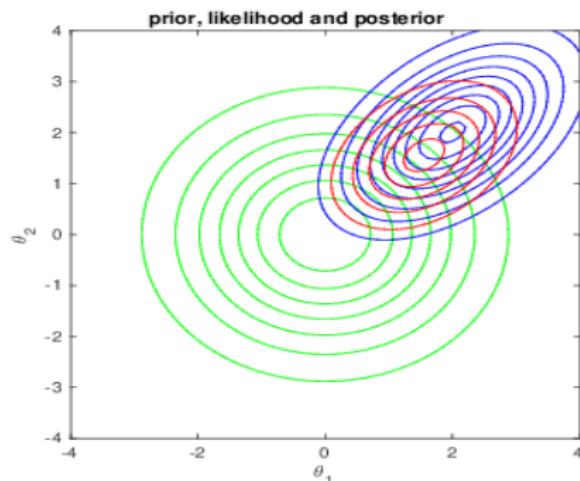
Posterior: $p(\theta_1, \theta_2|\mathbf{d}) \propto p(\mathbf{d}|\theta_1, \theta_2) p(\theta_1, \theta_2)$



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: one parameter (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) [needs optimization algorithms]

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

- ▶ Posterior Mean (PM) [needs integration methods]

$$\widehat{\theta} = E_{p(\theta|\mathbf{d})}\{\theta\} = \int \theta p(\theta|\mathbf{d}) d\theta$$

- ▶ Region of high probabilities: [needs integration methods]

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

- ▶ Sampling and exploring [Mont Carlo methods]

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

Bayesian inference: great dimensional case

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

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

- ▶ Gaussian posterior:

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

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

Bayesian inference: great dimensional case

- ▶ Gaussian posterior:

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

- ▶ Computation of $\hat{\theta}$ can be done via optimization of:

$$J(\theta) = -\ln p(\theta|\mathbf{d}) = c + \|\mathbf{d} - \mathbf{H}\theta\|^2 + \lambda\|\theta\|^2$$

- ▶ Gradient based methods:

$$\nabla J(\theta) = -2\mathbf{H}'(\mathbf{d} - \mathbf{H}\theta) + 2\lambda\theta$$

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

- ▶ At each iteration, we need to be able to compute:

- ▶ Forward operation: $\hat{\mathbf{d}} = \mathbf{H}\theta^{(k)}$

- ▶ Backward (Adjoint) operation: $\mathbf{H}^t(\mathbf{d} - \hat{\mathbf{d}})$

- ▶ Other optimization methods: Conjugate Gradient, ...

Bayesian inference: great dimensional case

- ▶ Gaussian posterior:

$$p(\boldsymbol{\theta}|\mathbf{d}) = \mathcal{N}(\boldsymbol{\theta}|\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{V}}), \quad \widehat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \{J(\boldsymbol{\theta}) = \|\mathbf{d} - \mathbf{H}\boldsymbol{\theta}\|^2 + \lambda \|\boldsymbol{\theta}\|^2\},$$

- ▶ 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: $\mathbf{d} = \mathbf{H}\boldsymbol{\theta} + \epsilon$
- ▶ 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:

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

- ▶ 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 $\hat{\boldsymbol{\theta}}_{MAP}$ can be done via optimization of $J(\boldsymbol{\theta})$
- ▶ Other computations (Posterior Mean or Covariance) are much more difficult: No analytical expressions. Need Approximation methods.

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

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

- Prediction with selected model:

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

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 $\mathcal{M}_1, \dots, \mathcal{M}_K$, parameters $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K$
- ▶ Model Comparison:

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

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

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

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

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

where $A_{i,j} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln p(\boldsymbol{\theta} | \mathbf{d}, \mathcal{M})$ 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 $\hat{\boldsymbol{\theta}}$:

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

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

Bayesian Information Criteria (BIC)

- ▶ BIC is obtained from the Laplace approximation

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

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

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

- ▶ 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: Asymptotic conditions for great 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

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

- ▶ Parameter estimation

Marginalization:

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

Estimation: $\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta} | \mathbf{d}, \mathcal{M})\}$

- ▶ EM algorithm: Complete data: (\mathbf{x}, \mathbf{d})

▶ **E step:** Compute $q_1^{(t+1)}(\mathbf{x}) = p(\mathbf{x} | \mathbf{d}, \boldsymbol{\theta}^{(t)})$

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

▶ **M Step:** $\boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} \{Q(\boldsymbol{\theta})\}$

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

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

- ▶ Prediction with a new data \mathbf{z}

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

Lower Bounding the Marginal Likelihood

Jensen's inequality:

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

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

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

Maximising this free energy leads to VBA.

Variational Bayesian Learning

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

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

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

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

which can also be written as:

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

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

EM and VBEM algorithms

- ▶ EM: Objective: Compute the marginal $p(\theta|\mathbf{d}, \mathcal{M})$ and maximize it with respect to θ to obtain $\hat{\theta}$.

E Step:

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

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

M Step: Maximize

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

- ▶ VBA: Objective: Approximate $p(\mathbf{x}, \theta | \mathbf{d})$ by $q_1(\mathbf{x})q_2(\theta)$

If $q_1(\mathbf{x})$ is choosed to be a conjugate prior to the likelihood $p(\mathbf{d} | \mathbf{x}, \theta^{(t)})$, then $q_1(\mathbf{x} | \mathbf{d})$ will be in the same family:
 $q_1(\mathbf{x} | \mathbf{d}, \phi^{(t)})$ and

E Step:

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

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

M Step: $q_2^{(t+1)}(\theta) = \exp [Q(\theta)]$

EM and VBEM algorithms

EM for Marginal MAP estimation

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

E Step: Compute

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

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

M Step: Maximize

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

Variational Bayesian EM

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

VB-E Step: Compute

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

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

M Step: Maximize

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

Properties:

- ▶ VB-EM reduces to EM if $q_2(\theta) = \delta(\theta - \tilde{\theta})$
- ▶ VB-EM has the same complexity than EM
- ▶ If we choose $q_2(\theta)$ in the conjugate family of $p(\mathbf{d}, \mathbf{x}|\theta)$, then ϕ 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, ϕ replaces θ .

Measuring variation of temperature with a thermometer

- ▶ $f(t)$ variation of temperature over time
- ▶ $g(t)$ variation of length of the liquid in thermometer
- ▶ Forward model: Convolution

$$g(t) = \int f(t') h(t - t') dt' + \epsilon(t)$$

$h(t)$: impulse response of the measurement system

- ▶ Inverse problem: Deconvolution

Given the forward model \mathcal{H} (impulse response $h(t)$)
and a set of data $g(t_i), i = 1, \dots, M$
find $f(t)$

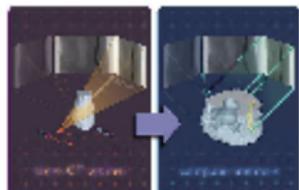


Computed Tomography: Seeing inside of a body

- ▶ $f(x, y)$ 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



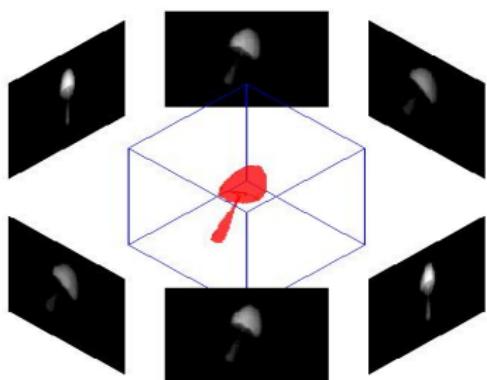
$$\begin{aligned} g_\phi(r) &= \int_{L_{r,\phi}} f(x, y) \, dl + \epsilon_\phi(r) \\ &= \iint f(x, y) \delta(r - x \cos \phi - y \sin \phi) \, dx \, dy + \epsilon_\phi(r) \end{aligned}$$

- ▶ Inverse problem: Image reconstruction

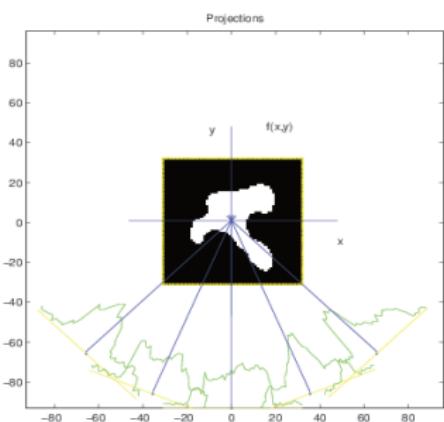
Given the forward model \mathcal{H} (Radon Transform) and
a set of data $g_{\phi_i}(r), i = 1, \dots, M$
find $f(x, y)$

2D and 3D Computed Tomography

3D



2D

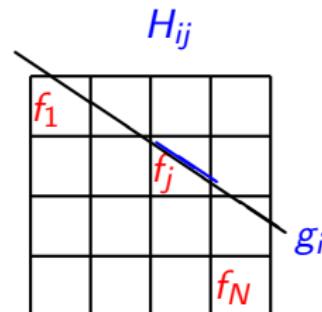
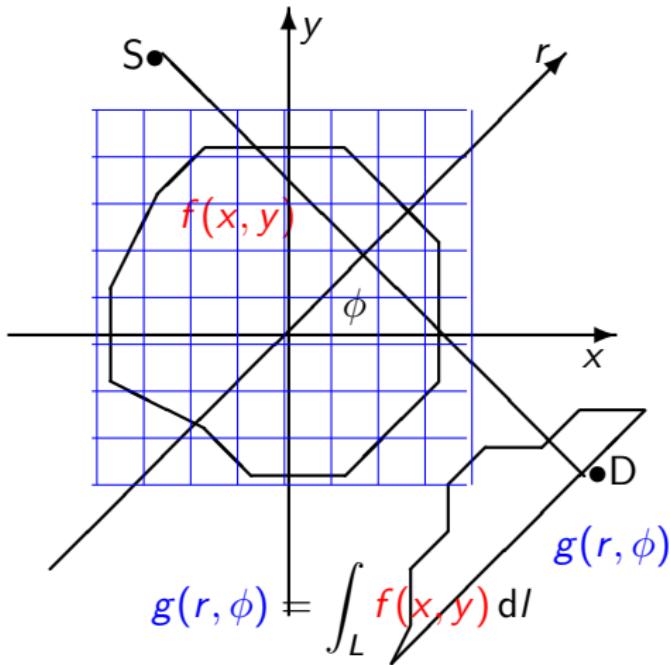


$$g_\phi(r_1, r_2) = \int_{\mathcal{L}_{r_1, r_2, \phi}} f(x, y, z) \, dI \quad g_\phi(r) = \int_{\mathcal{L}_{r, \phi}} f(x, y) \, dI$$

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

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

Algebraic methods: Discretization



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

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

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

- ▶ \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

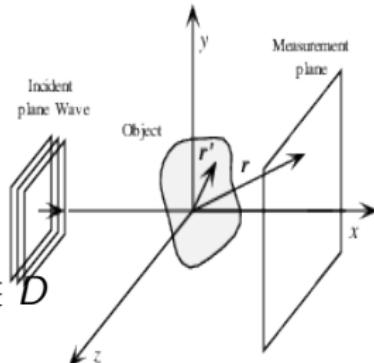
Measures: diffracted wave by the object $g(\mathbf{r}_i)$

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

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

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

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

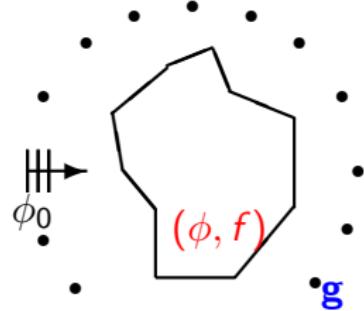


Born approximation ($\phi(\mathbf{r}') \simeq \phi_0(\mathbf{r}')$):

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

Discretization:

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



Microwave or ultrasound imaging: Bilinear model

Nonlinear model:

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

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

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

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

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

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

Discretization: $\mathbf{g} = \mathbf{G}_m \mathbf{w} + \boldsymbol{\epsilon}$, $\mathbf{w} = \phi \cdot \mathbf{f}$

► Contrast \mathbf{f} - Field ϕ : $\phi = \phi_0 + \mathbf{G}_o \mathbf{w} + \boldsymbol{\xi}$

► Contrast \mathbf{f} - Source \mathbf{w} : $\mathbf{w} = \mathbf{f} \cdot \phi_0 + \mathbf{G}_o \mathbf{w} + \boldsymbol{\xi}$

Bayesian approach for linear inverse problems

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

- ▶ 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

$$\mathcal{M} : \quad \mathbf{g} = \mathbf{G}_m \mathbf{w} + \epsilon, \quad \mathbf{w} = \mathbf{f} \cdot \phi_0 + \mathbf{G}_o \mathbf{w} + \xi, \quad \mathbf{w} = \phi \cdot \mathbf{f}$$

$$\mathcal{M} : \quad \mathbf{g} = \mathbf{G}_m \mathbf{w} + \epsilon, \quad \mathbf{w} = (\mathbf{I} - \mathbf{G}_o)^{-1}(\Phi_0 \mathbf{f} + \xi), \quad \mathbf{w} = \phi \cdot \mathbf{f}$$

- ▶ Basic Bayes:

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

- ▶ Unsupervised:

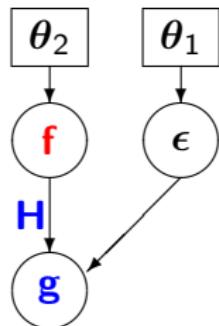
$$p(\mathbf{f}, \mathbf{w}, \theta | \mathbf{g}, \alpha_0) \propto p(\mathbf{g} | \mathbf{w}, \theta_1) p(\mathbf{f} | \mathbf{w}, \theta_2) p(\mathbf{f} | \theta_3) p(\theta | \alpha_0), \quad \theta = (\theta_1, \theta_2, \theta_3)$$

- ▶ Hierarchical prior models:

$$p(\mathbf{f}, \mathbf{w}, \mathbf{z}, \theta | \mathbf{g}, \alpha_0) \propto p(\mathbf{g} | \mathbf{w}, \theta_1) p(\mathbf{w} | \mathbf{f}, \theta_2) p(\mathbf{f} | \mathbf{z}, \theta_3) p(\mathbf{z} | \theta_4) p(\theta | \alpha_0)$$

Bayesian inference for inverse problems

Simple case:



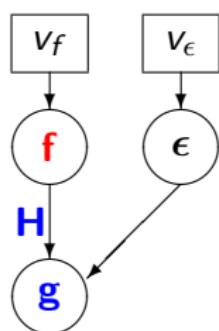
$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$

$$p(\mathbf{f}|\mathbf{g}, \theta) \propto p(\mathbf{g}|\mathbf{f}, \theta_1) p(\mathbf{f}|\theta_2)$$

– Objective: Infer \mathbf{f}

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

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



Example: Gaussian case:

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

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

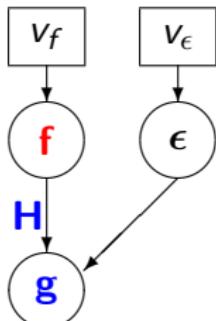
– Posterior Mean (PM)=MAP:

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

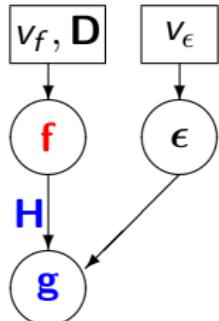
Gaussian model: Simple separable and Markovian

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

Separable Gaussian



Gauss-Markov



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

$$\begin{cases} p(\mathbf{g}|\mathbf{f}, \theta_1) = \mathcal{N}(\mathbf{g}|\mathbf{H}\mathbf{f}, \nu_\epsilon \mathbf{I}) \\ p(\mathbf{f}|v_f) = \mathcal{N}(\mathbf{f}|0, \nu_f \mathbf{I}) \end{cases} \xrightarrow{} p(\mathbf{f}|\mathbf{g}, \theta) = \mathcal{N}(\mathbf{f}|\hat{\mathbf{f}}, \hat{\Sigma})$$

- MAP: $\hat{\mathbf{f}} = \arg \min_{\mathbf{f}} \{J(\mathbf{f})\}$ with

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

- Posterior Mean (PM)=MAP:

$$\begin{cases} \hat{\mathbf{f}} = (\mathbf{H}^t \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^t \mathbf{g} \\ \hat{\Sigma} = \nu_\epsilon (\mathbf{H}^t \mathbf{H} + \lambda \mathbf{I})^{-1} \end{cases} \text{ with } \lambda = \frac{\nu_\epsilon}{\nu_f}.$$

Markovian case:

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

$$- \text{MAP: } J(\mathbf{f}) = \frac{1}{\nu_\epsilon} \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \frac{1}{\nu_f} \|\mathbf{D}\mathbf{f}\|^2$$

- Posterior Mean (PM)=MAP:

$$\begin{cases} \hat{\mathbf{f}} = (\mathbf{H}^t \mathbf{H} + \lambda \mathbf{D}^t \mathbf{D})^{-1} \mathbf{H}^t \mathbf{g} \\ \hat{\Sigma} = \nu_\epsilon (\mathbf{H}^t \mathbf{H} + \lambda \mathbf{D}^t \mathbf{D})^{-1} \end{cases} \text{ with } \lambda = \frac{\nu_\epsilon}{\nu_f}.$$

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

– Marginalization 1:

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

– Marginalization 2:

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

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

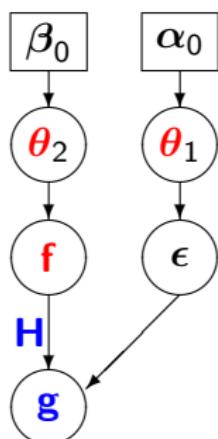
– MCMC Gibbs sampling:

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

Use samples generated to compute mean and variances

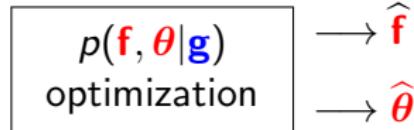
– VBA: Approximate $p(\mathbf{f}, \boldsymbol{\theta} | \mathbf{g})$ by $q_1(\mathbf{f}) q_2(\boldsymbol{\theta})$

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

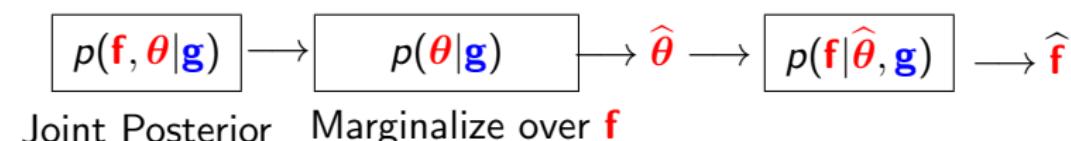


JMAP, Marginalization, VBA

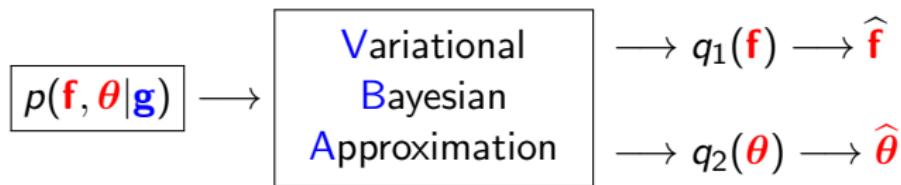
- ▶ JMAP:



- ▶ Marginalization



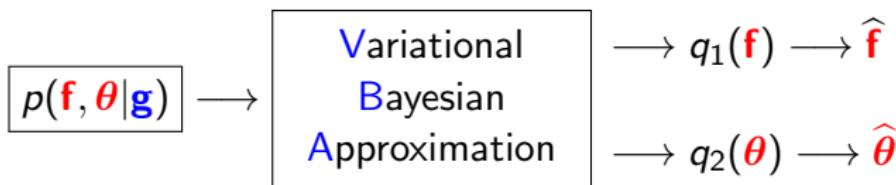
- ▶ Variational Bayesian Approximation



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 $\text{KL}(q(\mathbf{f}, \boldsymbol{\theta} | \mathbf{g}) : p(\mathbf{f}, \boldsymbol{\theta} | \mathbf{g}))$
$$\text{KL}(q : p) = \int \int q \ln \frac{q}{p} = \int \int q_1 q_2 \ln \frac{q_1 q_2}{p}$$
- ▶ Iterative algorithm $q_1 \rightarrow q_2 \rightarrow q_1 \rightarrow q_2, \dots$

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



Variational Bayesian Approximation

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

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

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

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

Free energy:

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

Evidence of the model \mathcal{M} :

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

VBA: Separable Approximation

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

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

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

$$(\hat{q}_1, \hat{q}_2) = \arg \min_{(q_1, q_2)} \{\text{KL}(q_1 q_2 : p)\} = \arg \max_{(q_1, q_2)} \{\mathcal{F}(q_1 q_2)\}$$

$\text{KL}(q_1 q_2 : p)$ is convex wrt q_1 when q_2 is fixed and vice versa:

$$\begin{cases} \hat{q}_1 = \arg \min_{q_1} \{\text{KL}(q_1 \hat{q}_2 : p)\} = \arg \max_{q_1} \{\mathcal{F}(q_1 \hat{q}_2)\} \\ \hat{q}_2 = \arg \min_{q_2} \{\text{KL}(\hat{q}_1 q_2 : p)\} = \arg \max_{q_2} \{\mathcal{F}(\hat{q}_1 q_2)\} \end{cases}$$

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

BVA: Choice of family of laws q_1 and q_2

- Case 1 : \rightarrow Joint MAP

$$\begin{cases} \hat{q}_1(\mathbf{f}|\tilde{\mathbf{f}}) = \delta(\mathbf{f} - \tilde{\mathbf{f}}) \\ \hat{q}_2(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}) = \delta(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) \end{cases} \rightarrow \begin{cases} \tilde{\mathbf{f}} = \arg \max_{\mathbf{f}} \left\{ p(\mathbf{f}, r\tilde{\boldsymbol{\theta}} | \mathbf{g}; \mathcal{M}) \right\} \\ \tilde{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \left\{ p(\tilde{\mathbf{f}}, \boldsymbol{\theta} | \mathbf{g}; \mathcal{M}) \right\} \end{cases}$$

- Case 2 : \rightarrow EM

$$\begin{cases} \hat{q}_1(\mathbf{f}) \propto p(\mathbf{f} | \boldsymbol{\theta}, \mathbf{g}) \\ \hat{q}_2(\boldsymbol{\theta} | \tilde{\boldsymbol{\theta}}) = \delta(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) \end{cases} \rightarrow \begin{cases} Q(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = \langle \ln p(\mathbf{f}, \boldsymbol{\theta} | \mathbf{g}; \mathcal{M}) \rangle_{q_1(\mathbf{f} | \tilde{\boldsymbol{\theta}})} \\ \tilde{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \right\} \end{cases}$$

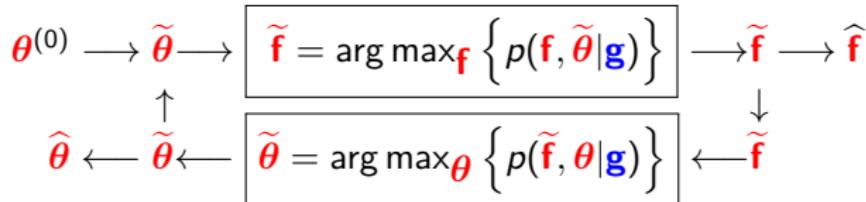
- Appropriate choice for inverse problems

$$\begin{cases} \hat{q}_1(\mathbf{f}) \propto p(\mathbf{f} | \tilde{\boldsymbol{\theta}}, \mathbf{g}; \mathcal{M}) \\ \hat{q}_2(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta} | \mathbf{f}, \mathbf{g}; \mathcal{M}) \end{cases} \rightarrow \begin{cases} \text{Accounts for the uncertainties of} \\ \hat{\boldsymbol{\theta}} \text{ for } \hat{\mathbf{f}} \text{ and vice versa.} \end{cases}$$

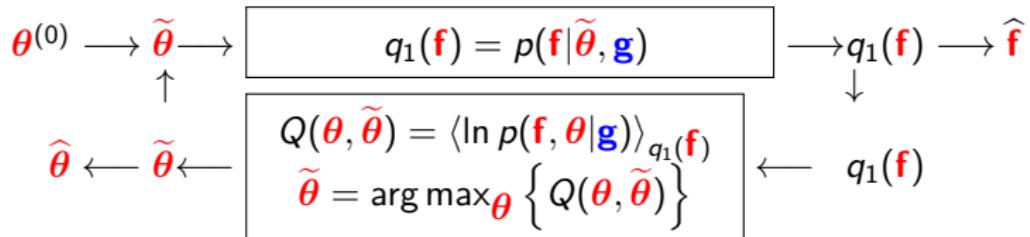
Exponential families, Conjugate priors

JMAP, EM and VBA

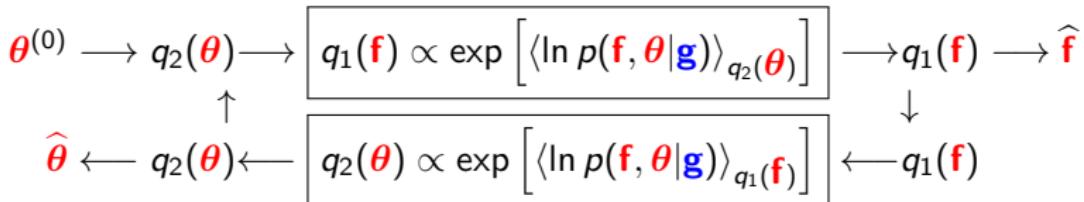
JMAP Alternate optimization Algorithm:



EM:



VBA:



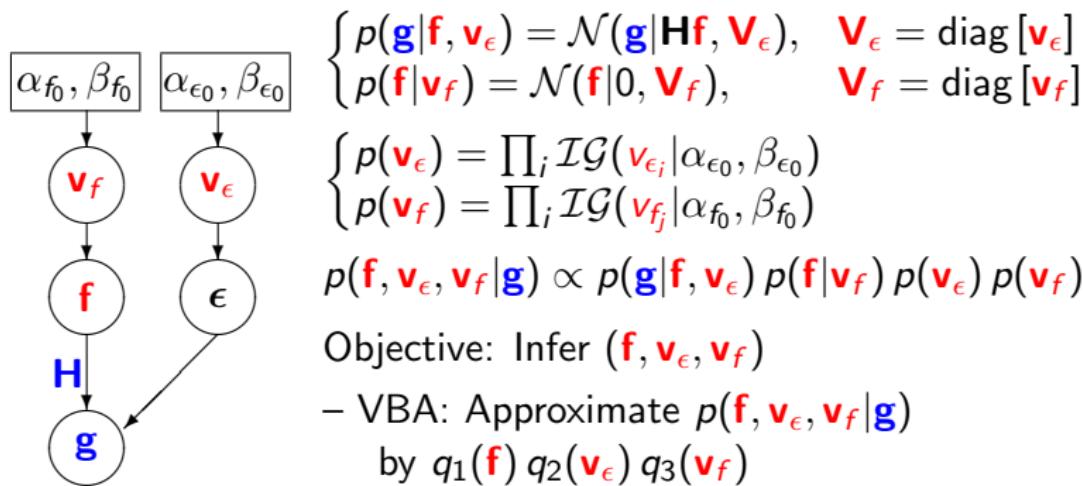
Non stationary noise and sparsity enforcing model

- Non stationary noise:

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

- Student-t prior model and its equivalent IGSM :

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



Sparse model in a Transform domain 1

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

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

$$\begin{cases} p(\textcolor{red}{v}_\epsilon) = \mathcal{IG}(\textcolor{red}{v}_\epsilon | \alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(\textcolor{red}{v}_z) = \prod_i \mathcal{IG}(\textcolor{red}{v}_{zj} | \alpha_{z_0}, \beta_{z_0}) \end{cases}$$

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

- IMPAP:

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

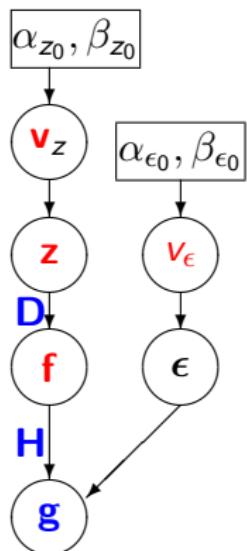
Alternate optimization:

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

– VBA: Approximate

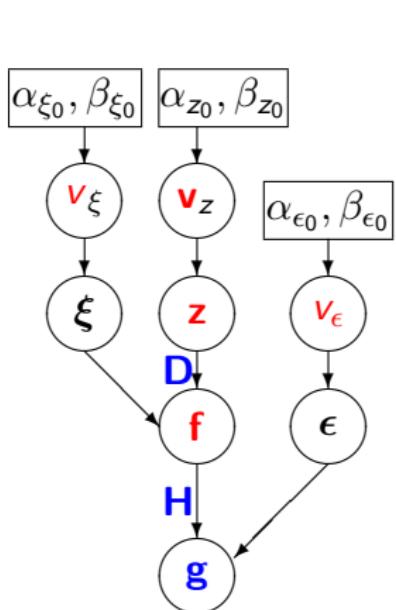
$p(\mathbf{z}, \mathbf{v}_\epsilon, \mathbf{v}_z, \mathbf{v}_\xi | \mathbf{g})$ by $q_1(\mathbf{z}) q_2(\mathbf{v}_\epsilon) q_3(\mathbf{v}_z)$

Alternate optimization.



Sparse model in a Transform domain 2

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



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

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

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

- JMAP:

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

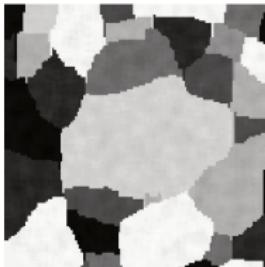
Alternate optimization.

- VBA: Approximate

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

Alternate optimization.

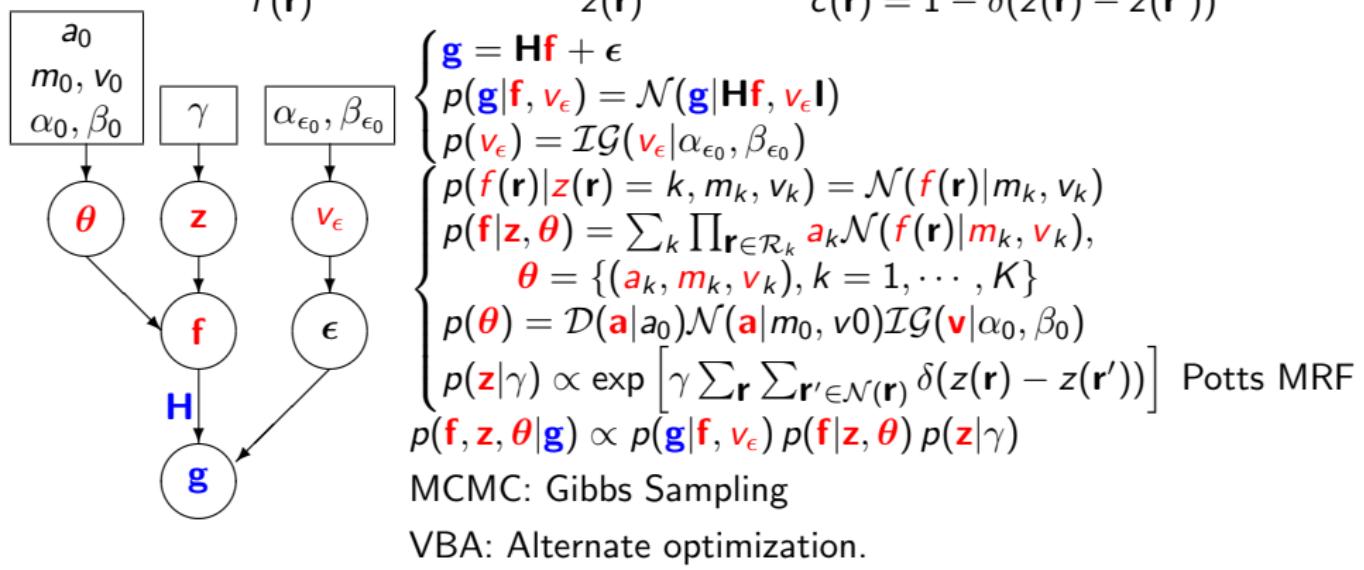
Gauss-Markov-Potts prior models for images



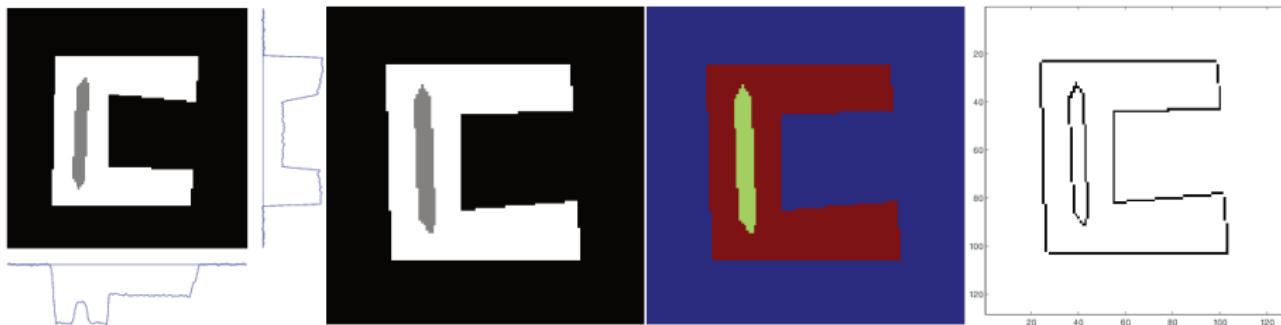
$f(\mathbf{r})$

$z(\mathbf{r})$

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



Application in CT



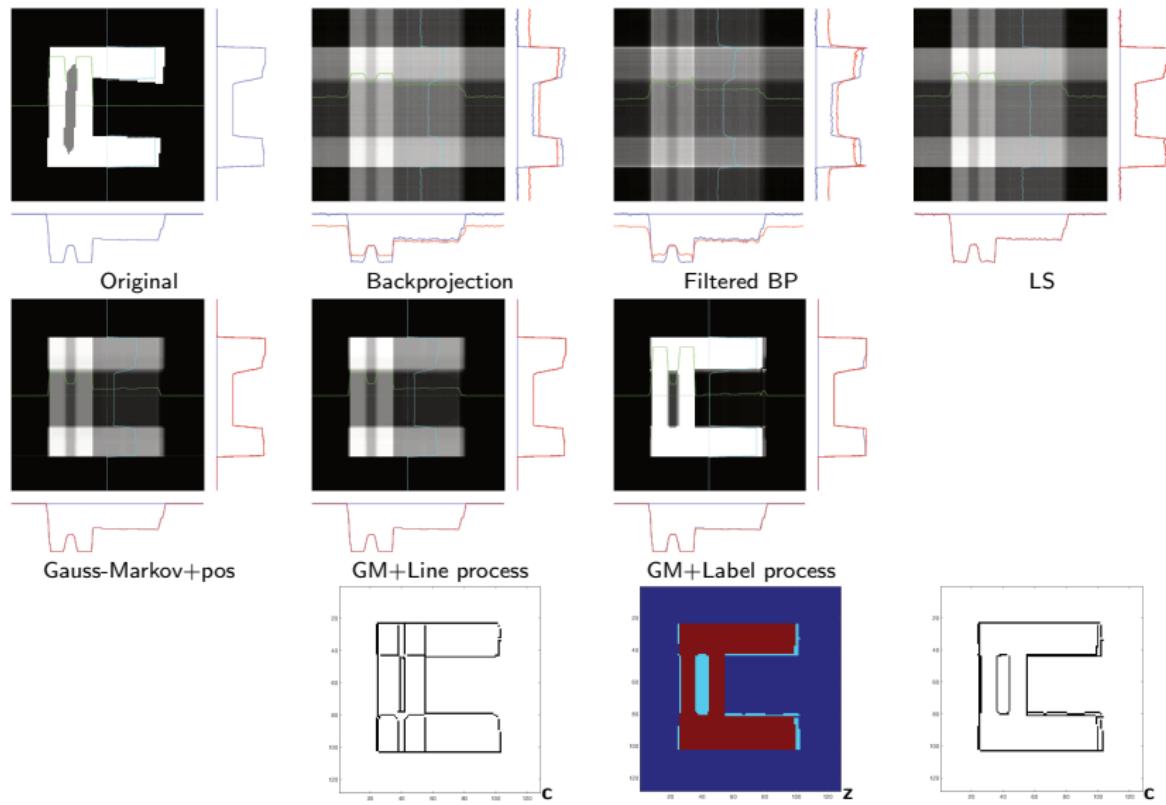
$$\begin{array}{l} \mathbf{g} | \mathbf{f} \\ \mathbf{g} = \mathbf{Hf} + \boldsymbol{\epsilon} \\ \mathbf{g} | \mathbf{f} \sim \mathcal{N}(\mathbf{Hf}, \sigma_\epsilon^2 \mathbf{I}) \\ \text{Gaussian} \end{array}$$

$$\begin{array}{l} \mathbf{f} | \mathbf{z} \\ \text{iid Gaussian} \\ \text{or} \\ \text{Gauss-Markov} \end{array}$$

$$\begin{array}{l} \mathbf{z} \\ \text{iid} \\ \text{or} \\ \text{Potts} \end{array} \quad \begin{array}{l} \mathbf{c} \\ q(\mathbf{r}) \in \{0, 1\} \\ 1 - \delta(z(\mathbf{r}) - z(\mathbf{r}')) \\ \text{binary} \end{array}$$

$$p(\mathbf{f}, \mathbf{z}, \boldsymbol{\theta} | \mathbf{g}) \propto p(\mathbf{g} | \mathbf{f}, \boldsymbol{\theta}_1) p(\mathbf{f} | \mathbf{z}, \boldsymbol{\theta}_2) p(\mathbf{z} | \boldsymbol{\theta}_3) p(\boldsymbol{\theta})$$

Results

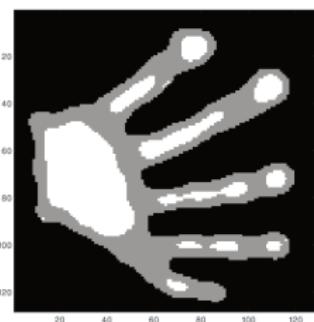
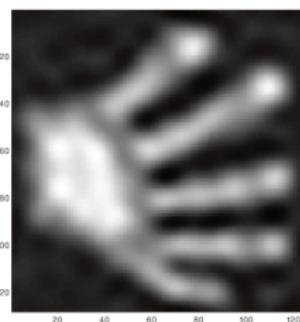
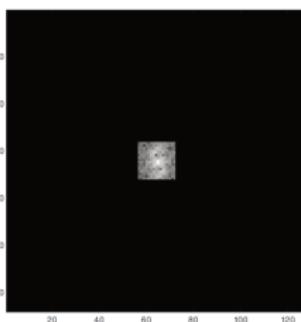
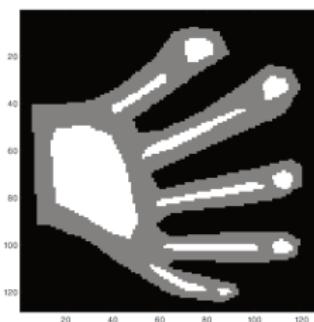


Application in Microwave imaging

$$g(\omega) = \int f(r) \exp[-j(\omega \cdot r)] dr + \epsilon(\omega)$$

$$g(u, v) = \iint f(x, y) \exp[-j(ux + vy)] dx dy + \epsilon(u, v)$$

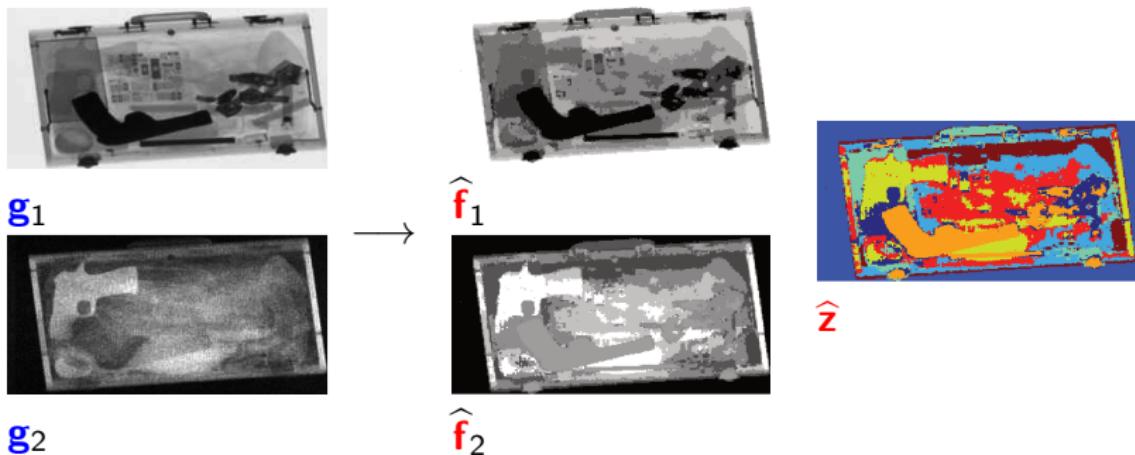
$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\epsilon}$$



Images fusion and joint segmentation

(with O. Féron)

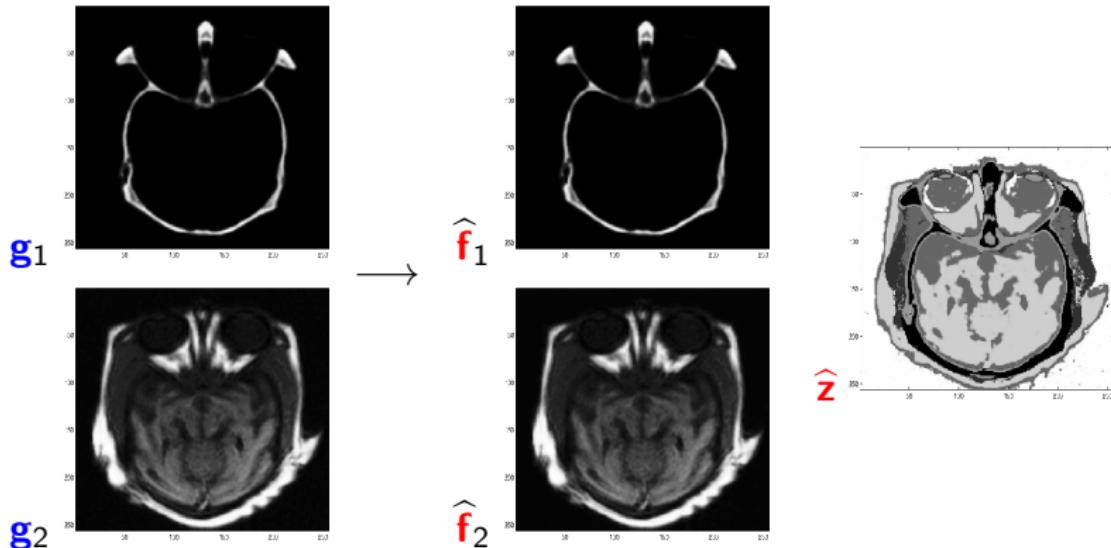
$$\begin{cases} g_i(\mathbf{r}) = f_i(\mathbf{r}) + \epsilon_i(\mathbf{r}) \\ p(f_i(\mathbf{r})|z(\mathbf{r}) = k) = \mathcal{N}(m_{ik}, \sigma_{ik}^2) \\ p(\underline{\mathbf{f}}|\mathbf{z}) = \prod_i p(\mathbf{f}_i|\mathbf{z}) \end{cases}$$



Data fusion in medical imaging

(with O. Féron)

$$\begin{cases} \mathbf{g}_i(\mathbf{r}) = \mathbf{f}_i(\mathbf{r}) + \epsilon_i(\mathbf{r}) \\ p(f_i(\mathbf{r})|z(\mathbf{r}) = k) = \mathcal{N}(m_{ik}, \sigma_{ik}^2) \\ p(\underline{\mathbf{f}}|\underline{\mathbf{z}}) = \prod_i p(\mathbf{f}_i|\mathbf{z}) \end{cases}$$



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

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

- ▶ Same family $p_k(\mathbf{d}_k|\theta_k) = p(\mathbf{d}_k|\theta_k)$, $\forall k$
- ▶ Gaussian $p(\mathbf{d}_k|\theta_k) = \mathcal{N}(\mathbf{d}_k|\mu_k, \mathbf{V}_k)$ with $\theta_k = (\mu_k, \mathbf{V}_k)$
- ▶ Data $\mathbf{D} = \{\mathbf{d}_n, n = 1, \dots, N\}$ where each element \mathbf{d}_n can be in one of the K classes c_n .
- ▶ $a_k = p(c_n = k)$, $\mathbf{a} = \{a_k, k = 1, \dots, K\}$,
 $\Theta = \{\theta_k, k = 1, \dots, K\}$, $\mathbf{c} = \{c_n, n = 1, \dots, N\}$

$$p(\mathbf{D}, \mathbf{c}|\mathbf{a}, \Theta) = \prod_{n=1}^N p(\mathbf{d}_n, c_n = k | a_k, \theta_k)$$

Different problems

- ▶ **Training:**

Given a set of (training) data \mathbf{D} and classes \mathbf{c} , estimate the parameters \mathbf{a} and Θ .

- ▶ **Supervised classification:**

Given a sample \mathbf{x}_m and the parameters K , \mathbf{a} and Θ determine its class

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

- ▶ **Semi-supervised classification (Proportions are not known):**

Given sample \mathbf{x}_m and the parameters K and Θ , determine its class

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

- ▶ **Clustering or unsupervised classification (Number of classes K is not known):**

Given a set of data \mathbf{D} , determine K and \mathbf{c} .

Training

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

$$(\hat{\mathbf{a}}, \hat{\boldsymbol{\Theta}}) = \arg \max_{(\mathbf{a}, \boldsymbol{\Theta})} \{ p(\mathbf{D}, \mathbf{c} | \mathbf{a}, \boldsymbol{\Theta}, K) \}.$$

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

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

where

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

- ▶ Infer on \mathbf{a} and $\boldsymbol{\Theta}$ 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 Θ determine

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

where $p(\mathbf{x}_m, \mathbf{c}_m = k | \mathbf{a}, \Theta, K) = a_k p(\mathbf{d}_m | \theta_k)$ and

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

- Best class k^* :

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

Semi-supervised classification

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

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

where

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

and

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

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

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

Clustering or non-supervised classification

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

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

and

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

where L_0 is the a priori maximum number of classes and

$$p(\mathbf{D}|K = L) = \int \int \prod_n \prod_{k=1}^L a_k p(\mathbf{x}_n, c_n = k | \boldsymbol{\theta}_k) p(\mathbf{a}|K) p(\boldsymbol{\Theta}|K) d\mathbf{a} d\boldsymbol{\Theta}.$$

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

Mixture of Gaussian and Mixture of Student-t

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

- ▶ Mixture of Gaussian (MoG)

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

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

- ▶ Mixture of Student-t (MoSt)

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

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

Mixture of Student-t: Hierarchical model

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

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

where

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

and

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

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

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

Mixture of Gaussian: Introduction of hidden variable

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

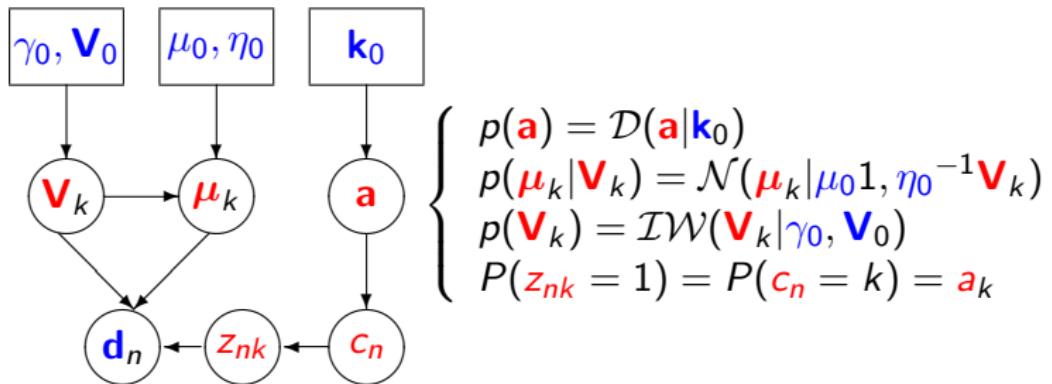
$$p(\mathbf{D}|\mathbf{c}, \mathbf{Z}, \boldsymbol{\Theta}, K) = \prod_n \sum_k a_k \mathcal{N}(\mathbf{d}_n | \boldsymbol{\mu}_k, \mathbf{V}_k) (1 - \delta(z_{nk}))$$
$$p(\mathbf{D}, \mathbf{c}, \mathbf{Z}, \boldsymbol{\Theta}|K) = \prod_n \prod_k [a_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k,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

- ▶ Joint posterior law:

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

- ▶ 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



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

Mixture of Student-t model

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

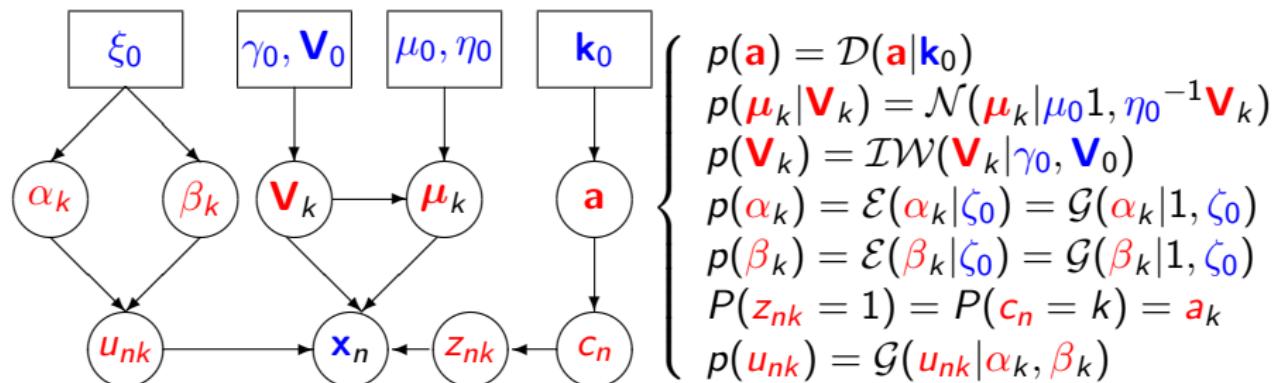
$$p(\mathbf{D}, \mathbf{c}, \mathbf{Z}, \mathbf{U}, \Theta | K) = \prod_n \prod_k \left[a_k \mathcal{N}(\mathbf{d}_n | \boldsymbol{\mu}_k, \mathbf{u}_{n,k}^{-1} \mathbf{V}_k) \mathcal{G}(\mathbf{u}_{nk} | \alpha_k, \beta_k) \right]^{z_{nk}} p(\boldsymbol{\theta}_k)$$

- ▶ Joint posterior law:

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

- ▶ 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



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

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

Expressions of the different prior distributions

- ▶ Dirichlet

$$\mathcal{D}(\mathbf{a}|\mathbf{k}) = \frac{\Gamma(\sum_I k_I)}{\prod_I \Gamma(k_I)} \prod_I a_I^{k_I - 1}$$

- ▶ Exponential

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

- ▶ Gamma

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

- ▶ Inverse Wishart

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

Variational Bayesian Approximation (VBA)

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

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

where

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

or

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

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

Expressions of q

$$\begin{aligned} q(\mathbf{c}, \mathbf{Z}, \boldsymbol{\Theta}) &= q(\mathbf{c}, \mathbf{Z}) q(\boldsymbol{\Theta}) \\ &= \prod_n \prod_k [q(\mathbf{c}_n = k | \mathbf{z}_{nk}) q(\mathbf{z}_{nk})] \\ &\quad \prod_k [q(\boldsymbol{\alpha}_k) q(\boldsymbol{\beta}_k) q(\boldsymbol{\mu}_k | \mathbf{V}_k) q(\mathbf{V}_k)] q(\mathbf{a}). \end{aligned}$$

with:

$$\left\{ \begin{array}{l} q(\mathbf{a}) = \mathcal{D}(\mathbf{a} | \tilde{\mathbf{k}}), \quad \tilde{\mathbf{k}} = [\tilde{k}_1, \dots, \tilde{k}_K] \\ q(\boldsymbol{\alpha}_k) = \mathcal{G}(\boldsymbol{\alpha}_k | \tilde{\zeta}_k, \tilde{\eta}_k) \\ q(\boldsymbol{\beta}_k) = \mathcal{G}(\boldsymbol{\beta}_k | \tilde{\zeta}_k, \tilde{\eta}_k) \\ q(\boldsymbol{\mu}_k | \mathbf{V}_k) = \mathcal{N}(\boldsymbol{\mu}_k | \tilde{\boldsymbol{\mu}}, \tilde{\eta}^{-1} \mathbf{V}_k) \\ q(\mathbf{V}_k) = \mathcal{IW}(\mathbf{V}_k | \tilde{\gamma}, \tilde{\Sigma}) \end{array} \right.$$

With these choices, we have

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

$$\mathcal{F}_{1_{kn}} = \langle \ln p(\mathbf{d}_n, \mathbf{c}_n, \mathbf{z}_{nk}, \boldsymbol{\theta}_k) \rangle_{q(\mathbf{c}_n = k | \mathbf{z}_{nk}) q(\mathbf{z}_{nk})}$$

$$\mathcal{F}_{2_k} = \langle \ln p(\mathbf{d}_n, \mathbf{c}_n, \mathbf{z}_{nk}, \boldsymbol{\theta}_k) \rangle_{q(\boldsymbol{\theta}_k)}$$

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 $q(\boldsymbol{\Theta})$ fixed, we obtain the expression of $q(\mathbf{c}_n = k | \mathbf{z}_{nk}) = \tilde{a}_k$, $q(\mathbf{z}_{nk}) = \mathcal{G}(\mathbf{z}_{nk} | \tilde{\alpha}_k, \tilde{\beta}_k)$.
- ▶ **M step:** Optimizing \mathcal{F} with respect to $q(\boldsymbol{\Theta})$ when keeping $q(\mathbf{c}, \mathbf{Z})$ fixed, we obtain the expression of
 $q(\mathbf{a}) = \mathcal{D}(\mathbf{a} | \tilde{\mathbf{k}})$, $\tilde{\mathbf{k}} = [\tilde{k}_1, \dots, \tilde{k}_K]$, $q(\boldsymbol{\alpha}_k) = \mathcal{G}(\boldsymbol{\alpha}_k | \tilde{\zeta}_k, \tilde{\eta}_k)$,
 $q(\boldsymbol{\beta}_k) = \mathcal{G}(\boldsymbol{\beta}_k | \tilde{\zeta}_k, \tilde{\eta}_k)$, $q(\boldsymbol{\mu}_k | \mathbf{V}_k) = \mathcal{N}(\boldsymbol{\mu}_k | \tilde{\boldsymbol{\mu}}, \tilde{\eta}^{-1} \mathbf{V}_k)$, 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 $p(x) = \frac{1}{Z} \exp [-\phi(x)]$ with $\phi(x)$ convex and symmetric, 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