Chapter

Bayesian Inference for Inverse Problems

Ali Mohammad-Djafari

Abstract

Inverse problems arise everywhere we have indirect measurement. Regularization and Bayesian inference methods are two main approaches to handle inverse problems. Bayesian inference approach is more general and has much more tools for developing efficient methods for difficult problems. In this chapter, first, an overview of the Bayesian parameter estimation is presented, then we see the extension for inverse problems. The main difficulty is the great dimension of unknown quantity and the appropriate choice of the prior law. The second main difficulty is the computational aspects. Different approximate Bayesian computations and in particular the variational Bayesian approximation (VBA) methods are explained in details.

Keywords: inverse problems, hidden variable, hierarchical models, approximate Bayesian computation, variational Bayesian approximation

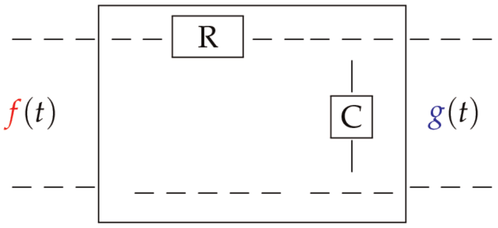
1. Introduction

Inverse problems arise in many scientific and engineering applications. In fact, almost always we want to infer on quantities, variables, distributions and functions which are not directly observable. Inferring on a hidden variable f via the observation of another variable g is the main objective in many scientific area [1–3].

Classically, the Bayesian methods have been developed for direct observation of a quantity, its parametric modeling and the estimation of the parameters of the model. Description and development of the Bayesian inference for the case of inverse problems is the main objective of this chapter. The chapter is organized as follows: First a few inverse problems are mentioned, mainly in two categories: those described by Ordinary Differential Equations (ODE) and Partial Differential Equations (PDE)'s and those described with integral equations. Then, we will see that two main problems arise: parameter estimation and inversion. First the Bayesian parameter estimation is described and then the Bayesian inversion.

2. Inverse problems

To see easily the two categories of inverse problems, first a very simple example is given. Consider the electrical circuit of **Figure 1**.



$$\frac{\partial g(t)}{\partial t} = \frac{1}{C} i(t), \quad i(t) = \frac{(f(t) - g(t))}{R}, \quad \frac{\partial g(t)}{\partial t} = \frac{1}{RC}(f(t) - g(t))$$

$$g(t) + RC\frac{\partial g(t)}{\partial t} = f(t) \longrightarrow g(\omega) + RCj\omega g(\omega) = F(\omega)$$

$$H(\omega) = \frac{g(\omega)}{F(\omega)} = \frac{1}{1+jRC\omega}$$

$$RC = \theta \longrightarrow H(\omega) = \frac{1}{1+j\theta\omega} \rightarrow h(t) = \exp\left[-t/\theta\right] \rightarrow f(t) = h(t) * f(t)$$

Figure 1.
A simple electrical circuit example to show two different expressions of inverse problems modeling: Ordinary differential equation (ODE) or Integral equation (IE).

Using the notations used on the figure, we can easily obtain the following ODE:

$$g(t) + \boldsymbol{\theta}\frac{\partial g(t)}{\partial t} = f(t) \tag{1}$$

Then, using the Fourier transform (FT), we obtain easily the following integral equation:

$$f(t) = \int f(\tau)h(t - \tau)\,\mathrm{d}\tau \tag{2}$$

These two simple equations describe the same linear inverse problem, where we can distinguish the following mathematical problems:

- Forward problem: Given the parameter $\boldsymbol{\theta}$ of the system and the input, $f(t)$ predict the output $g(t)$.
- Parameter estimation: Given the input $f(t)$ and the output $g(t)$, estimate the parameter $\boldsymbol{\theta}$.
- System identification: Given the input $f(t)$ and the output $g(t)$, estimate the impulse response (IR) of the system $h(t)$.
- Inverse problems:
    - Simple: Given the characteristics of the system (either the parameter $\boldsymbol{\theta}$ or equivalently the impulse response $h(t)$) and the output $g(t)$ estimate the input $f(t)$;
    - Blind: Given the output $g(t)$ estimate both the system, parameter $\boldsymbol{\theta}$ or the impulse response $h(t)$, and input $f(t)$.

For general vocabulary and examples, see [2, 4, 5].

2.1 Examples of linear inverse problems

Here, a few examples of classical inverse problems are listed.

2.1.1 Deconvolution

When the forward problem is a convolution operation:

$$g(t) = \int f(\tau)h(t-\tau)\,\mathrm{d}\tau, \tag{3}$$

the inverse problem is called *Deconvolution* (**Figure 2**). **Figure 3** shows an example of deconvolution problem which arise in radio astronomy.

2.1.2 Image restoration

In many imaging systems, such as visual cameras, microscopes, telescopes or Infra Red cameras, due to some limitations such as limited aperture or limited resolution, the forward problem can be approximated by a 2D convolution equation:

$$g(x,y) = \iint f(x',y')h(x-x',y-y')\,\mathrm{d}x\,\mathrm{d}y. \tag{4}$$

The corresponding inverse problem is called image deconvolution or more often *image restoration*. The example given in **Figure 4**, is the case of satellite imaging [6, 7].

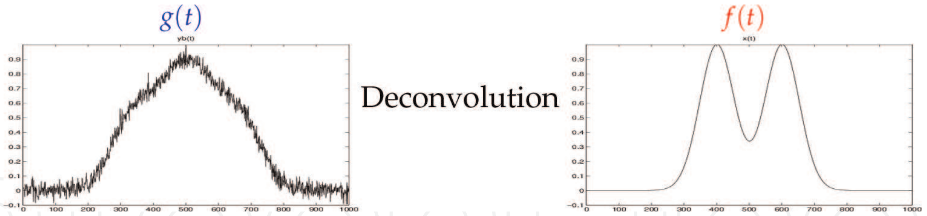


Figure 2.
Signal deconvolution problem.

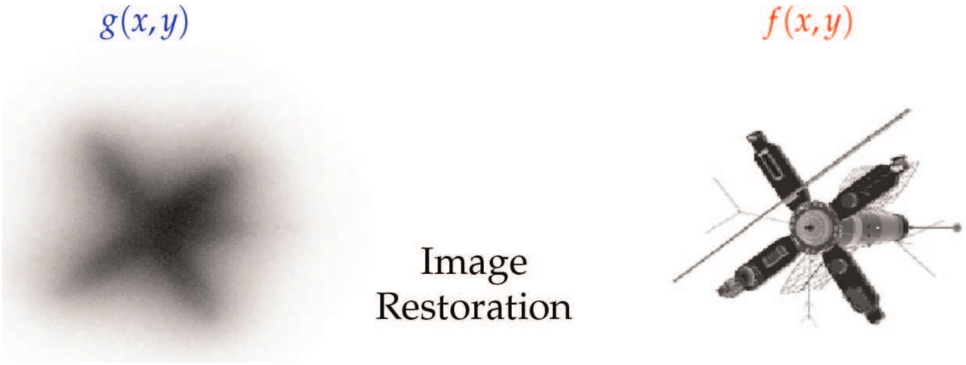


Figure 3.
Image deconvolution or restoration inverse problem in sattelite imaging.

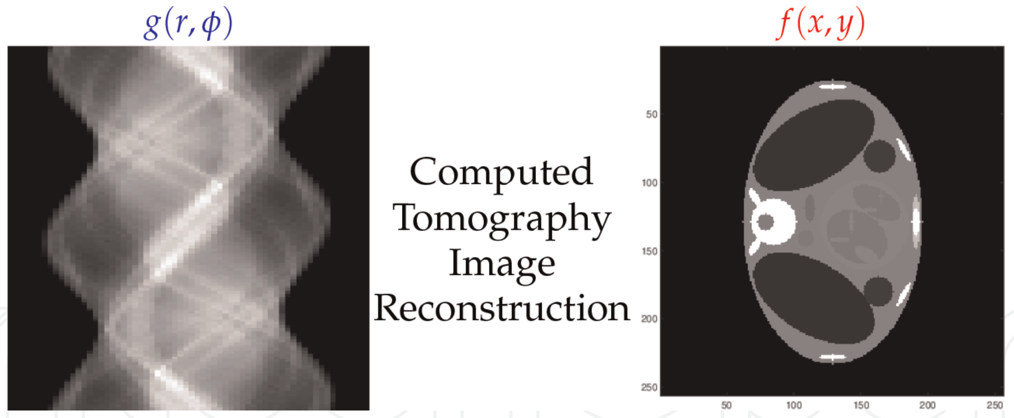


Figure 4.
Image reconstruction in CT. On the left, the projections $g(r, \phi)$ and on the right the object $f(x, y)$.

2.1.3 Image reconstruction in X ray computed tomography (CT)

In X-ray CT, assuming parallel geometry, where a ray is characterized by its angle ϕ and its distance r from the center of the object $f(x, y)$ the relation between the data $g(r, \phi)$, called projections at angle ϕ and the function $f(x, y)$, called object, is given by the Radon transform:

$$g(r,\phi) = \iint f(x,y)\delta(r - x\cos\phi - y\sin\phi)\,\mathrm{d}x\,\mathrm{d}y. \tag{5}$$

The inverse problem here is called *Image reconstruction*. A simulated example is shown in **Figure 4**.

2.1.4 Fourier synthesis

In many imaging systems, when using FT, it is possible to model the inverse problem with the following forward FT relation:

$$g(u,v) = \iint f(x,y)\exp\left[-j(ux+vy)\right]\,\mathrm{d}x\,\mathrm{d}y, \tag{6}$$

where the data, after an appropriate FT, can fill partially the Fourier domain $g(u, v)$ of the unknown interested function $f(x, y)$ [8, 9]. **Figure** 5 shows the case of X-ray CT.

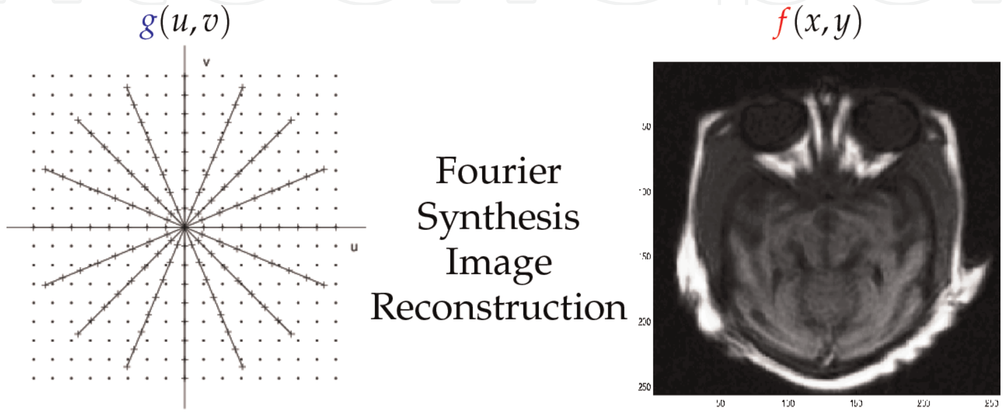


Figure 5.
Fourier synthesis (FS) inverse problems arising in many imaging systems. Here is illustrated the FS problem in X-ray CT.

2.1.5 General linear inverse problems

All the examples of the linear inverse problems listed above, can be summarized in the following general form:

$$g(\boldsymbol{s}) = \int f(\boldsymbol{r})\, h(\boldsymbol{s}, \boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \tag{7}$$

where $\boldsymbol{s}$ can be either t, (x, y), (r, ϕ) or (u, v) and $\boldsymbol{r}$, respectively τ, (x', y'), (x, y) and (x, y).

3. Bayesian parameter estimation

To introduce, in a very simple way, the Bayes rule for parameter estimation, we consider the case where we have a set of data: $\boldsymbol{g} = \{g_1, \cdots, g_n\}$ where we assign them a probability law $p(g_i|\boldsymbol{\theta})$ with a set of unknown parameters $\boldsymbol{\theta}$. The question now is how to infer $\boldsymbol{\theta}$ from those data. We can immediately use the Bayes rule:

$$p(\boldsymbol{\theta}|\boldsymbol{g}) = \frac{p(\boldsymbol{g}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{g})} \propto l(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \tag{8}$$

where:

- $l(\boldsymbol{\theta}) \triangleq p(g|\boldsymbol{\theta}) = \Pi_i p(g_i|\boldsymbol{\theta})$ is called the *likelihood*, representing the uncertainty in the data knowing the parameters;

- $p(\boldsymbol{\theta})$ is called the *prior or* a priori, a probability law assigned to the parameters to represent the prior knowledge (to the observation data) we may have on those parameters;

- the denominator $p(\boldsymbol{g})$

$$p(\boldsymbol{g}) = \int p(g|\boldsymbol{\theta}) p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \tag{9}$$

is called the *evidence*.

So, the process of using the Bayes rule for parameter estimation can be summarized as follows:

- Write the expression of the likelihood $p(\boldsymbol{g}|\boldsymbol{\theta})$

- Assign the prior $p(\boldsymbol{\theta})$ to translate all we know about $\boldsymbol{\theta}$ before observing the data $\boldsymbol{g}$

- Apply the Bayes rule to obtain the expression of the posterior law $p(\boldsymbol{\theta}|\boldsymbol{g})$

- Use the posterior $p(\boldsymbol{\theta}|\boldsymbol{g})$ to do any inference on $\boldsymbol{\theta}$. For example:

    - Compute its expected value, called Expected A Posteriori (EAP) or Posterior Mean (PM):

$$\hat{\boldsymbol{\theta}}_{PM} = \int \theta p(\boldsymbol{\theta}|\mathbf{g})\, \mathrm{d}\boldsymbol{\theta} \tag{10}$$

- Compute the value of $\boldsymbol{\theta}$ for which the $p(\boldsymbol{\theta}|\mathbf{g})$ is maximum; Maximum A Posteriori (MAP):

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta}|\mathbf{g})\} \tag{11}$$

- Sampling and exploring [Monte Carlo methods]

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

which gives the possibility to obtain any statistical information we want to know about $\boldsymbol{\theta}$. For example, if we generate N samples $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N\}$, for large enough N, we have:

$$\mathrm{E}\{\theta\} \simeq \frac{1}{N}\sum_{n=1}^{N} \theta_n. \tag{12}$$

3.1 One parameter case

When θ is a scalar quantity, then, we can also do the following computations:

- Compute the value of θ_{Med} such that:

$$P(\theta > \theta_{Med}) = P(\theta < \theta_{Med}) \tag{13}$$

which is called the *median value*. Its computation needs integration:

$$\int_{-\infty}^{\theta_{Med}} p(\theta|\mathbf{g})\, \mathrm{d}\theta = \int_{\theta_{Med}}^{\infty} p(\theta|\mathbf{g})\, \mathrm{d}\theta \tag{14}$$

- Compute the value θ_α, called α quantile, for which

$$P(\theta > \theta_\alpha) = 1 - P(\theta < \theta_\alpha) = \int_{\hat{\theta}_\alpha}^{\infty} p(\theta|\mathbf{g})\, \mathrm{d}\theta = 1 - \alpha \tag{15}$$

- Region of high probabilities: [needs integration methods]

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

Bayes rule and Bayesian estimation can be illustrated as follows:

$$\begin{tikzcd} & p(\mathbf{g}|\theta) \arrow[d] & & \\ p(\theta) \arrow[r] & \boxed{\text{Bayes}} \arrow[r] & p(\theta|\mathbf{g}) \arrow[r] & \left\{ \begin{array}{l} \widehat{\theta}_{MAP} \\ \widehat{\theta}_{PM} \\ \widehat{\theta}_{Med} \end{array} \right. \end{tikzcd}$$

Two main points are of great importance:

- How to assign the prior $p(\boldsymbol{\theta})$ in the second step; and

- How to do the computations in the last step.

This last problem becomes more serious with multi parameter case.

3.2 Multi-parameter case

If we have more than one parameter, then $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_n]'$. The Bayes rule still holds:

$$p(\boldsymbol{\theta}|g) = \frac{p(\boldsymbol{g}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{g})}{p(\boldsymbol{g})} \tag{16}$$

Now, again, we can compute:

- The Expected A Posteriori (EAP):

$$\hat{\boldsymbol{\theta}}_{PM} = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\boldsymbol{g})\, \mathrm{d}\boldsymbol{\theta}, \tag{17}$$

but this needs efficient integration methods.

- The Maximum A Posteriori (MAP):

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta}|\boldsymbol{g})\} \tag{18}$$

but this needs efficient optimization methods.

- Sampling and exploring [Monte Carlo methods]

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{g})$$

but this needs efficient sampling methods.

- We may also try to localize the region of the highest probability:

$$P(\boldsymbol{\theta} \in \Theta) = \int_{\Theta} p(\boldsymbol{\theta}|g)\, \mathrm{d}\boldsymbol{\theta} = 1 - \alpha \tag{19}$$

for a given small α, but this problem may not have a unique solution.

4. Bayesian inference for inverse problems

As described before, in inverse problems, the unknown f is a function (of time, space, wavelength, ...) and the observable quantity g is also another function which is related to f via an operator $g = \mathcal{H}(f) + \epsilon$. When discretized, they can be represented by the great dimensional vectors $\boldsymbol{f}, \boldsymbol{g}$ and $\boldsymbol{g} = \boldsymbol{H}(\boldsymbol{f}) + \boldsymbol{\epsilon}$, where ϵ represents all the errors (measurement, model and discretization). When the operator is a linear one, we have:

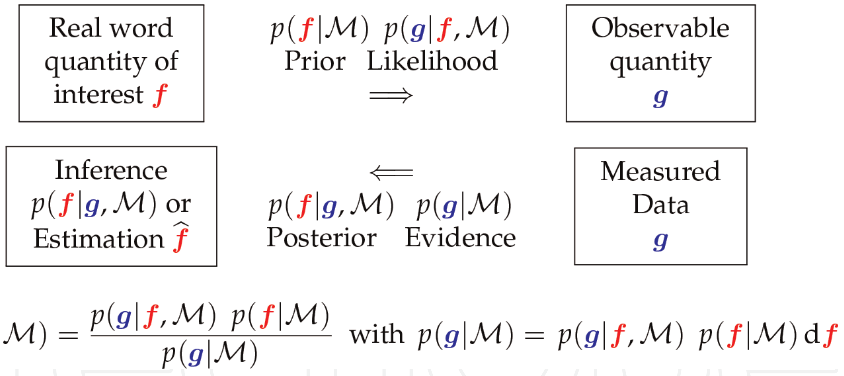

Real word quantity of interest f
$p(f|\mathcal{M})$ $p(g|f,\mathcal{M})$
Prior Likelihood
$\Longrightarrow$
Observable quantity g
Inference $p(f|g,\mathcal{M})$ or Estimation $\widehat{f}$
$\Longleftarrow$
$p(f|g,\mathcal{M})$ $p(g|\mathcal{M})$
Posterior Evidence
Measured Data g
$p(f|g,\mathcal{M}) = \frac{p(g|f,\mathcal{M})\ p(f|\mathcal{M})}{p(g|\mathcal{M})}$ with $p(g|\mathcal{M}) = p(g|f,\mathcal{M})\ p(f|\mathcal{M})\,\mathrm{d}f$

Figure 6.
Illustration of the Bayesian inference for inverse problems.

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}, \tag{20}$$

where $\boldsymbol{f}$ is a vector of length n, $\boldsymbol{H}$ the known forward model matrix of size $m \times n$, and $\boldsymbol{g}$ and $\boldsymbol{\epsilon}$ two vectors of size m.

The Bayes rule for this case is written as:

$$p(\boldsymbol{f}|g,\mathcal{M}) = \frac{p(\boldsymbol{g}|\boldsymbol{f},\mathcal{M})p(\boldsymbol{f}|,\mathcal{M})}{p(\boldsymbol{g},\mathcal{M})} \tag{21}$$

where we introduce $\mathcal{M}$ to represent the model, $p(\boldsymbol{g}|\boldsymbol{f},\mathcal{M})$, called commonly the *likelihood*, is obtained using the forward model (20) and the assigned probability law of the noise $p(\boldsymbol{\epsilon}), p(\boldsymbol{f}|,\mathcal{M})$ is the assigned prior model and $p(\boldsymbol{f}|\boldsymbol{g},\mathcal{M})$ the posterior probability law. **Figure 6** shows in a schematic way the main ingredients of the Bayesian inference for inverse problems.

This even very simple linear model has been used in many areas: linear inverse problems, compressed sensing, curve fitting and linear regression, machine learning, etc.

In inverse problems such as deconvolution, image restoration, $\boldsymbol{f}$ represent the input or original image, $\boldsymbol{g}$ represents the blurred and noisy image and $\boldsymbol{H}$ is the convolution operator matrix. In image reconstruction in Computed Tomography (CT), $\boldsymbol{f}$ represents the distribution of some internal property of an object, for example the density of the material and $\boldsymbol{g}$ represents the radiography data and $\boldsymbol{H}$ is the radiographic projection operator (discretized Radon transform operator).

In Compress Sensing, $\boldsymbol{g}$ is the compressed data, $\boldsymbol{f}$ is the uncompressed image and $\boldsymbol{H}$ the compressing matrix. In machine learning, $\boldsymbol{g}$ are the data, $\boldsymbol{H}$ is a dictionary and $\boldsymbol{f}$ represents the sparse coefficients of the projections of the data on that dictionary.

5. Hyperparameter estimation

When applying the Bayes rule, the main terms which are the likelihood and prior depend on parameters, which cannot be fixed in practical situation. We may thus want to estimate them from the data. In the Bayesian approach, this can be done easily:

$$p(\boldsymbol{f},\boldsymbol{\theta}_1,\boldsymbol{\theta}_2|\boldsymbol{g}) = \frac{p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\theta}_1)p(\boldsymbol{f},\boldsymbol{\theta}_2)p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_2)}{p(\boldsymbol{g})} \tag{22}$$

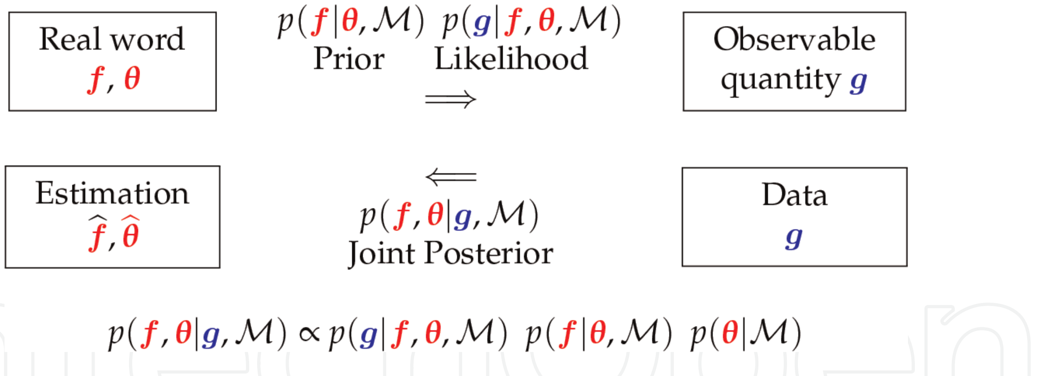

Real word
f, θ
$p(f|\theta,\mathcal{M})$ $p(g|f,\theta,\mathcal{M})$
Prior Likelihood
$\Longrightarrow$
Observable quantity g
Estimation
$\widehat{f}, \widehat{\theta}$
$\Longleftarrow$
$p(f,\theta|g,\mathcal{M})$
Joint Posterior
Data
g
$p(f,\theta|g,\mathcal{M}) \propto p(g|f,\theta,\mathcal{M})\ p(f|\theta,\mathcal{M})\ p(\theta|\mathcal{M})$

Figure 7.
Illustration of the Bayesian approach for inverse problems with unknown hyperparameters.

where $p(\boldsymbol{\theta}_1)$ and $p(\boldsymbol{\theta}_2)$ are the prior probability laws assigned to $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and often $p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_2)$. We can then write more succinctly:

$$p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g},\boldsymbol{\theta}_0) = \frac{p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\theta}_1)p(\boldsymbol{f},\boldsymbol{\theta}_2)p(\boldsymbol{\theta})}{p(\boldsymbol{g})} \tag{23}$$

The scheme of this situation is illustrated in **Figure 7**.

From here, we have different directions for doing estimation:

5.1 Joint maximum a posteriori (JMAP)

Rewriting the expression of the joint posterior law:

$$p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g}) = \frac{p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\theta}_1)p(\boldsymbol{f}|\boldsymbol{\theta}_2)p(\boldsymbol{\theta})}{p(\boldsymbol{g})} \propto p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\theta}_1)p(\boldsymbol{f},\boldsymbol{\theta}_2)p(\boldsymbol{\theta}) \tag{24}$$

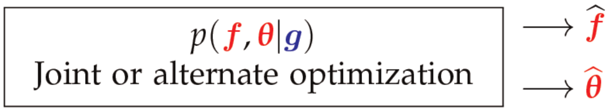
where $\propto$ means equal up to a constant factor which is $1/p(\boldsymbol{g})$. In this case, we can try to optimize it with respect to its two arguments:

$$\left(\hat{\boldsymbol{f}},\hat{\boldsymbol{\theta}}\right) = \underset{(\boldsymbol{f},\boldsymbol{\theta})}{\text{arg max}} \left\{p\left(\boldsymbol{f},\hat{\boldsymbol{\theta}}|\boldsymbol{g}\right)\right\} \tag{25}$$

This can be done, for example, by alternate optimization:

$$\begin{cases} \hat{\boldsymbol{f}}^{(k+1)} = \text{arg max}_{\boldsymbol{f}}\left\{p\left(\boldsymbol{f},\hat{\boldsymbol{\theta}}^{(k)}|\boldsymbol{g}\right)\right\} \\ \hat{\boldsymbol{\theta}}^{(k+1)} = \text{arg max}_{\boldsymbol{\theta}}\left\{p\left(\boldsymbol{f}^{(k)},\hat{\boldsymbol{\theta}}|\boldsymbol{g}\right)\right\} \end{cases} \tag{26}$$

When the optimization algorithm is successful, we have the optimal values of $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{\theta}}$. This method can be summarized as follows:


$p(f,\theta|g)$
Joint or alternate optimization
$\longrightarrow \widehat{f}$
$\longrightarrow \widehat{\theta}$

5.2 Marginalization over *θ*

The main idea here is to consider $\boldsymbol{\theta}$ as a nuisance parameter. Thus, integrating it out, we get

$$p(\boldsymbol{f}|\boldsymbol{g}) = \int p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g})\,\mathrm{d}\boldsymbol{\theta} \tag{27}$$

which can be used to infer on $\boldsymbol{f}$. Also, if we still want to get estimates of $\boldsymbol{\theta}$, we can first obtain an estimate $\hat{\boldsymbol{f}}$ for $\boldsymbol{f}$ and then, if needed, to use it as it is illustrated in the following scheme:

$$\begin{tikzcd} \boxed{p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g})} \arrow[r] & \boxed{p(\boldsymbol{f}|\boldsymbol{g})} \arrow[r] & \hat{\boldsymbol{f}} \arrow[r] & \boxed{p(\boldsymbol{\theta}|\hat{\boldsymbol{f}},\boldsymbol{g})} \arrow[r] & \hat{\boldsymbol{\theta}} \end{tikzcd}$$

Joint Posterior Marginalize over $\boldsymbol{\theta}$

5.3 Marginalization over *f*

The main idea here is first find a good estimate for the parameters $\boldsymbol{\theta}$ and then use it for the inference on $\boldsymbol{f}$. So, first obtain:

$$p(\boldsymbol{\theta}|\boldsymbol{g}) = \int p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g})\,\mathrm{d}\boldsymbol{f} \tag{28}$$

which can be used to first estimate $\boldsymbol{\theta}$ and then use it. For example, the method which is related to the *Second type Maximum likelihood*, first estimate $\hat{\boldsymbol{\theta}}$ by

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ p\left(\hat{\boldsymbol{\theta}}|\boldsymbol{g}\right) \right\} \tag{29}$$

and then use it with $p\left(\boldsymbol{f}|\hat{\boldsymbol{\theta}},\boldsymbol{g}\right)$ to infer on $\hat{\boldsymbol{f}}$. For a flat prior model, $p(\boldsymbol{\theta}|g) \propto p(g|\boldsymbol{\theta})$ which is called the *likelihood* and the estimator

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ p\left(\hat{\boldsymbol{\theta}}|g\right) \right\} = \arg\max_{\boldsymbol{\theta}} \left\{ p\left(g|\hat{\boldsymbol{\theta}}\right) \right\} \tag{30}$$

is called *Maximum Likelihood (ML)* and the whole approach is called *ML of second type*. This method can be summarized as follows:

$$\begin{tikzcd} \boxed{p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g})} \arrow[r] & \boxed{p(\boldsymbol{\theta}|\boldsymbol{g})} \arrow[r] & \hat{\boldsymbol{\theta}} \arrow[r] & \boxed{p(\boldsymbol{f}|\hat{\boldsymbol{\theta}},\boldsymbol{g})} \arrow[r] & \hat{\boldsymbol{f}} \end{tikzcd}$$

Joint Posterior Marginalize over $\boldsymbol{f}$

The main difficulty in this approach is that, rarely we can have an analytical expression for the first marginalization. To overcome this difficulty, many algorithms have been proposed to compute $\boldsymbol{f}$. One of them is called Expectation- Maximization (EM) and its generalization (GEM). The main idea of these algorithms are summarized in the following subsections:

5.3.1 EM and GEM algorithms

To summarize these methods, we use the vocabulary of the main authors of EM method, where $\boldsymbol{f}$ is considered as *hidden variable*, $\boldsymbol{g}$ as *incomplete data*, $(\boldsymbol{g}, \boldsymbol{f})$ as *complete data*, $\ln p(\boldsymbol{g}|\boldsymbol{\theta})$ *incomplete data log-likelihood* and $\ln p(\boldsymbol{g},\boldsymbol{f}|\boldsymbol{\theta})$ as *complete data log-likelihood*. Then, the following iterative algorithms describe the EM and GEM algorithms.:

- EM Iterative algorithm:

$$\begin{cases} \text{E-step :} \quad Q\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(k)}\right) = \mathrm{E}_{p\left(\boldsymbol{f}|\boldsymbol{g},\hat{\boldsymbol{\theta}}^{(k)}\right)}\{\ln p(\boldsymbol{g},\boldsymbol{f}|\boldsymbol{\theta})\} \\ \text{M-step :} \quad \hat{\boldsymbol{\theta}}^{(k)} = \arg\max_{\boldsymbol{\theta}}\left\{Q\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(k-1)}\right)\right\} \end{cases} \tag{31}$$

- GEM (Bayesian) algorithm:

$$\begin{cases} \text{E-step :} \quad Q\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(k)}\right) = \mathrm{E}_{p\left(\boldsymbol{f}|\boldsymbol{g},\hat{\boldsymbol{\theta}}^{(k)}\right)}\{\ln p(\boldsymbol{g},\boldsymbol{f}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})\} \\ \text{M-step :} \quad \hat{\boldsymbol{\theta}}^{(k)} = \arg\max_{\boldsymbol{\theta}}\left\{Q\left(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(k-1)}\right)\right\} \end{cases} \tag{32}$$

These methods can be summarized in the following scheme:

$$\boxed{p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g})} \rightarrow \boxed{\text{EM, GEM}} \rightarrow \hat{\boldsymbol{\theta}} \rightarrow \boxed{p\left(\boldsymbol{f}|\hat{\boldsymbol{\theta}},\boldsymbol{g}\right)} \rightarrow \hat{\boldsymbol{f}}$$

5.4 Variational Bayesian approximation

VBA is a powerful approach to do approximate Bayesian computation. It starts by first obtaining the expression of the joint $p(f, \boldsymbol{\theta}|g)$ and then by approximating it with a simpler probability law $q(\boldsymbol{f}, \boldsymbol{\theta}|g)$ which can be handled much easily for the computations. VBA can be summarized in the following steps:

- Approximate $p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g})$ by $q(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}) = q_1(\boldsymbol{f}|\boldsymbol{g})\, q_2(\boldsymbol{\theta}|\boldsymbol{g})$ and then continue computations.

- To do this approximation, we need a criterion to qualify the approximation. The standard criterion to measure the proximity of two probability laws p and q is the Kullback–Leibler (KL) criterion $\mathrm{KL}\,(q(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}) : p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}))$.

- It is easy to show that:

$$\begin{aligned} \mathrm{KL}(q : p) &= \iint q \ln q/p = \iint q_1 q_2 \ln \frac{q_1 q_2}{p} \\ &= \int q_1 \ln q_1 + \int q_2 \ln q_2 - \iint q \ln p \\ &= -H(q_1) - H(q_2) - < \ln p >_q \end{aligned} \tag{33}$$

- Alternate optimization of $\mathrm{KL}(q_1q_2{:}p)$ with respect to q_1 and q_2 results to:

$$\begin{cases} q_1(\boldsymbol{f}) & \propto \exp\left[\langle \ln p(\boldsymbol{g},\boldsymbol{f},\boldsymbol{\theta};\mathcal{M})\rangle_{q_2(\boldsymbol{\theta})}\right] \\ q_2(\boldsymbol{\theta}) & \propto \exp\left[\langle \ln p(\boldsymbol{g},\boldsymbol{f},\boldsymbol{\theta};\mathcal{M})\rangle_{q_1(\boldsymbol{f})}\right] \end{cases} \tag{34}$$

As $\mathrm{KL}(q_1q_2{:}p)$ is convex as a function of q_1 and q_2, the algorithm converges (locally) to the optimum solution. At the end, we have the expressions of $q_1(\boldsymbol{f})$ and $q_2(\boldsymbol{\theta})$ which can, then, be used to infer on $\boldsymbol{f}$ and $\boldsymbol{\theta}$. VBA is summarized in the following scheme:

$$\begin{tikzcd} \boxed{p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g})} \arrow[r] & \boxed{\begin{array}{c}\text{Variation}\\ \text{Bayesian}\\ \text{Approximation}\end{array}} \arrow[r] \arrow[dr] & q_1(\boldsymbol{f}) \arrow[r] & \hat{\boldsymbol{f}} \\ & & q_2(\boldsymbol{\theta}) \arrow[r] & \hat{\boldsymbol{\theta}} \end{tikzcd}$$

In real applications, we choose parametric probability law for $q_1(\boldsymbol{f})$ $q_2(\boldsymbol{\theta})$, and so, the iterations will be done on the parameters. What is interesting is that, choosing appropriate parametric models for $q_1(\boldsymbol{f})$ and $q_2(\boldsymbol{\theta})$ we obtain either JMAP and GEM as special cases.

- Case 1: Deterministic or degenerate expressions → Joint MAP

$$\begin{cases} \hat{q}_1\left(\boldsymbol{f}|\tilde{\boldsymbol{f}}\right) & = \delta\left(\boldsymbol{f}-\tilde{\boldsymbol{f}}\right) \\ \hat{q}_2(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}) & = \delta(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}) \end{cases} \rightarrow \begin{cases} \tilde{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}}\{p(\boldsymbol{f},\tilde{\boldsymbol{\theta}}|\boldsymbol{g};\mathcal{M})\} \\ \tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}}\left\{p\left(\tilde{\boldsymbol{f}},\boldsymbol{\theta}|\boldsymbol{g};\mathcal{M}\right)\right\} \end{cases} \tag{35}$$

- Case 2: Degenerate expression for $\boldsymbol{\theta}$ and marginal expression for $\boldsymbol{f}$ → EM

$$\begin{cases} \hat{q}_1(\boldsymbol{f}) \propto p(\boldsymbol{f}|\boldsymbol{\theta},\boldsymbol{g}) \\ \hat{q}_2(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}) = \delta(\boldsymbol{\theta}-\tilde{\boldsymbol{\theta}}) \end{cases} \rightarrow \begin{cases} Q(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}) & = \langle \ln p(\boldsymbol{f},\boldsymbol{\theta}|\boldsymbol{g};\mathcal{M})\rangle_{q_1(\boldsymbol{f}|\tilde{\boldsymbol{\theta}})} \\ \tilde{\boldsymbol{\theta}} & = \arg\max_{\boldsymbol{\theta}}\{Q(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}})\} \end{cases} \tag{36}$$

- Case 3: q_1 and q_2 are chosen proportional to the marginals $p(\boldsymbol{f}|\tilde{\boldsymbol{\theta}},\boldsymbol{g};\mathcal{M})$ and $p\left(\boldsymbol{\theta}|\tilde{\boldsymbol{f}},\boldsymbol{g};\mathcal{M}\right)$. This is a very appropriate choice for inverse problems, in particular cases where we use the exponential families and conjugate priors.

$$\begin{cases} \hat{q}_1(\boldsymbol{f}) & \propto p(\boldsymbol{f}|\tilde{\boldsymbol{\theta}},\boldsymbol{g};\mathcal{M}) \\ \hat{q}_2(\boldsymbol{\theta}) & \propto p\left(\boldsymbol{\theta}|\tilde{\boldsymbol{f}},\boldsymbol{g};\mathcal{M}\right) \end{cases} \rightarrow \begin{cases} \text{Accounts for the uncertainties of} \\ \tilde{\boldsymbol{\theta}} \text{ for } \hat{\boldsymbol{f}} \text{ and vise versa.} \end{cases} \tag{37}$$

In the following schemes these three cases are illustrated for comparison.

- JMAP Alternate optimization Algorithm:

$$\begin{tikzcd} \boldsymbol{\theta}^{(0)} \arrow[r] & \tilde{\boldsymbol{\theta}} \arrow[r] & \boxed{\tilde{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}}\{p(\boldsymbol{f},\tilde{\boldsymbol{\theta}}|\boldsymbol{g})\}} \arrow[r] & \tilde{\boldsymbol{f}} \arrow[r] \arrow[d] & \hat{\boldsymbol{f}} \\ \hat{\boldsymbol{\theta}} & \tilde{\boldsymbol{\theta}} \arrow[l] \arrow[u] & \boxed{\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}}\left\{p\left(\tilde{\boldsymbol{f}},\boldsymbol{\theta}|\boldsymbol{g}\right)\right\}} \arrow[l] & \tilde{\boldsymbol{f}} \arrow[l] & \end{tikzcd}$$

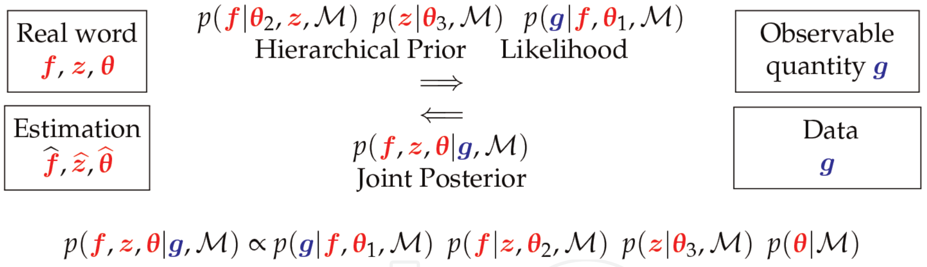

Real word
$\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}$
$p(\boldsymbol{f}|\theta_2, \boldsymbol{z}, \mathcal{M})$ $p(\boldsymbol{z}|\theta_3, \mathcal{M})$
Hierarchical Prior
$p(\boldsymbol{g}|\boldsymbol{f}, \theta_1, \mathcal{M})$
Likelihood
Observable quantity $\boldsymbol{g}$
$\Longrightarrow$
$\Longleftarrow$
Estimation
$\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{\theta}}$
$p(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{g}, \mathcal{M})$
Joint Posterior
Data
$\boldsymbol{g}$
$p(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{g}, \mathcal{M}) \propto p(\boldsymbol{g}|\boldsymbol{f}, \theta_1, \mathcal{M})\ p(\boldsymbol{f}|\boldsymbol{z}, \theta_2, \mathcal{M})\ p(\boldsymbol{z}|\theta_3, \mathcal{M})\ p(\theta|\mathcal{M})$

Figure 8.
Illustration of advanced Bayesian approach with hierarchical prior modeling with hidden variables.

- EM:

$$\begin{tikzcd}
\boldsymbol{\theta}^{(0)} \arrow[r] & \tilde{\boldsymbol{\theta}} \arrow[r] & \boxed{q_1(\boldsymbol{f}) = p(\boldsymbol{f}|\tilde{\boldsymbol{\theta}}, \boldsymbol{g})} \arrow[r] & q_1(\boldsymbol{f}) \arrow[r] \arrow[d] & \hat{\boldsymbol{f}} \\
\hat{\boldsymbol{\theta}} & \tilde{\boldsymbol{\theta}} \arrow[l] \arrow[u] & \boxed{\begin{array}{c} Q(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = \langle \ln p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}) \rangle_{q_1(\boldsymbol{f})} \\ \tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \{ Q(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \} \end{array}} \arrow[l] & q_1(\boldsymbol{f}) \arrow[l] &
\end{tikzcd}$$

- VBA:

$$\begin{tikzcd}
\boldsymbol{\theta}^{(0)} \arrow[r] & q_2(\boldsymbol{\theta}) \arrow[r] & \boxed{q_1(\boldsymbol{f}) \propto \exp\left[\langle \ln p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}) \rangle_{q_2(\boldsymbol{\theta})}\right]} \arrow[r] & q_1(\boldsymbol{f}) \arrow[r] \arrow[d] & \hat{\boldsymbol{f}} \\
\hat{\boldsymbol{\theta}} & q_2(\boldsymbol{\theta}) \arrow[l] \arrow[u] & \boxed{q_2\boldsymbol{\theta} \propto \exp\left[\langle \ln p(\boldsymbol{f}, \boldsymbol{\theta}|\boldsymbol{g}) \rangle_{q_1(\boldsymbol{f})}\right]} \arrow[l] & q_1(\boldsymbol{f}) \arrow[l] &
\end{tikzcd}$$

5.5 Hierarchical priors

One last extension is the case where $\boldsymbol{f}$, itself depends on another hidden variable $\boldsymbol{z}$. So that we have:

$$p(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{g}) \propto p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{\theta}_1) p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{\theta}_2) p(\boldsymbol{z}|\boldsymbol{\theta}_3) p(\boldsymbol{\theta}), \tag{38}$$

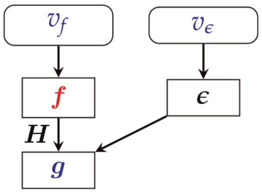
where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$. This situation is shown in **Figure 8**. Again, here, we may only be interested to $\boldsymbol{f}$ or $(\boldsymbol{f}, \boldsymbol{z})$ or to all the three variables $(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta})$. Here too, we can either use methods of JMAP, marginalization or VBA to infer on these unknowns.

6. Linear forward models and Gaussian case

Linear models are of importance. Gaussian prior laws are the most common and the easiest ones to handle. Also, many non-linear problems can be approximated by equivalent linear ones. Linear models with Gaussian prior laws are the easiest and powerful tools for a great number of scientific problems. In this section, an overview and some main properties are given.

6.1 Simple supervised case

Let consider the linear forward model we considered in previous section



$p(f|v_f) = \mathcal{N}(f|0, v_f I)$

$p(g|f, v_\epsilon) = \mathcal{N}(g|H f, v_\epsilon I)$

$p(f|g, v_f, v_\epsilon) = \mathcal{N}(f|\widehat{f}, \widehat{\Sigma}$ with $\widehat{f} = [H'H + \lambda I]^{-1} H'g$ and $\widehat{\Sigma} = v_\epsilon [H'H + \lambda I]^{-1}$

Figure 9.
Supervised linear Gaussian case.

$$g = Hf + \epsilon, \tag{39}$$

and assign Gaussian laws to ϵ and f which leads to:

$$\begin{cases} p(g|f) = \mathcal{N}(g|Hf, v_\epsilon I) \propto \exp\left[-\frac{1}{2v_\epsilon}\|g - Hf\|^2\right] \\ p(f) = \mathcal{N}(f|0, v_f I) \propto \exp\left[-\frac{1}{2v_f}\|f\|^2\right] \end{cases} \tag{40}$$

Using these expressions, we get:

$$\begin{cases} p(f|g) \propto \exp\left[-\frac{1}{2v_\epsilon}\|g - Hf\|^2 - \frac{1}{2v_f}\|f\|^2\right] \\ \qquad \propto \exp\left[-\frac{1}{2v_\epsilon}J(f)\right] \text{with } J(f) = \|g - Hf\|^2 + \lambda\|f\|^2, \quad \lambda = \frac{v_\epsilon}{v_f} \end{cases} \tag{41}$$

which can be summarized as:

$$p(f|g) = \mathcal{N}\left(f|\hat{f}, \hat{\Sigma}\right) \text{ with } \hat{f} = [H'H + \lambda I]^{-1} H'g \text{ and } \hat{\Sigma} = v_\epsilon [H'H + \lambda I]^{-1}, \tag{42}$$

where $\lambda = \frac{v_\epsilon}{v_f}$. This case is summarized in **Figure 9**.

This is the simplest case where we know exactly the expression of the posterior law and all the computations can be done explicitly. However, for great dimensional problems, where the vectors f and g are very great dimensional, we may even not be able to keep in memory the matrix H and surely not be able to compute the inverse of the matrix $[H'H + \lambda I]$. In Section 9 on Bayesian computation, We will see how to do these computations.

6.2 Unsupervised case or hyperparameter estimation

In the previous section, we considered the linear models with Gaussian priors with known parameters v_ϵ and v_f. In many practical situations these parameters are not known, and we want to estimate them too. For this, we can assign them too prior laws. As the variances are positive quantities and using the concept of conjugate priors, we can assign then Inverse Gamma priors:

$$\begin{cases} p(v_\epsilon) = \mathcal{IG}(v_f|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(v_f) = \mathcal{IG}\left(v_f|\alpha_{f_0}, \beta_{f_0}\right) \end{cases} \tag{43}$$

and using the likelihood $p(\boldsymbol{g}|\boldsymbol{f}, v_\epsilon) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{Hf}, v_\epsilon \boldsymbol{I})$ and the prior $p\left(\boldsymbol{f}|v_f\right) = \mathcal{N}\left(\boldsymbol{f}|0, v_f \boldsymbol{I}\right)$, we can easily obtain the expressions of the following conditional posterior laws:

$$\begin{cases} p\left(\boldsymbol{f}|\boldsymbol{g}, \hat{v}_\epsilon, \hat{v}_f\right) = \mathcal{N}\left(\boldsymbol{f}|\hat{\boldsymbol{f}}, \hat{\Sigma}\right) \quad \text{with :} \\ \hat{\boldsymbol{f}} = \left[\boldsymbol{H}^t\boldsymbol{H} + \hat{\lambda}\boldsymbol{I}\right]^{-1}\boldsymbol{H}^t\boldsymbol{g} \\ \hat{\Sigma} = \hat{v}_\epsilon\left[\boldsymbol{H}^t\boldsymbol{H} + \hat{\lambda}\boldsymbol{I}\right]^{-1}, \quad \hat{\lambda} = \hat{v}_\epsilon/\hat{v}_f \end{cases} \tag{44}$$

and

$$\begin{cases} p(v_\epsilon|\boldsymbol{g},\boldsymbol{f}) = \mathcal{IG}(v_\epsilon|\tilde{\alpha}_\epsilon, \tilde{\beta}\epsilon) \\ p(v_f|\boldsymbol{g},\boldsymbol{f}) = \mathcal{IG}\left(v_f|\tilde{\alpha}_f, \tilde{\beta}_f\right) \end{cases} \tag{45}$$

where all the details and in particular the expressions for $\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon, \tilde{\alpha}_f, \tilde{\beta}_f$ can be found in [10].

As we can see, the expressions of $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{\Sigma}}$ are the same as in the previous case, except that the values of $\hat{v}_\epsilon, \hat{v}_f$ and $\hat{\lambda}$ have to be updated. They are obtained from the conditionals $p(v_\epsilon|\boldsymbol{g},\boldsymbol{f})$ and $p\left(v_f|\boldsymbol{g},\boldsymbol{f}\right)$ which depend on $\boldsymbol{f}$. This shows that we can propose an iterative algorithm in two steps: Determine the expression of $p\left(\boldsymbol{f}|\boldsymbol{g}, \hat{v}_\epsilon, \hat{v}_f\right)$ and using the values of in the previous iteration, we can propose an estimate for $\boldsymbol{f}$, and then, using $p(v_\epsilon|\boldsymbol{g},\boldsymbol{f})$ and $p\left(v_f|\boldsymbol{g},\boldsymbol{f}\right)$, we can give estimates for $\hat{v}_\epsilon$ and $\hat{v}_f$ which can again be used in the first step. It is interesting to know that all the three approaches of JMAP, GEM and VBA for this cas follow exactly this same iterative algorithm. The only differences will be in the update values of $\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon, \tilde{\alpha}_f, \tilde{\beta}_f$ and the choice of the estimators (MAP or PM) of $\hat{v}_\epsilon$ and $\hat{v}_f$.

This case is also summarized in **Figure 10**.

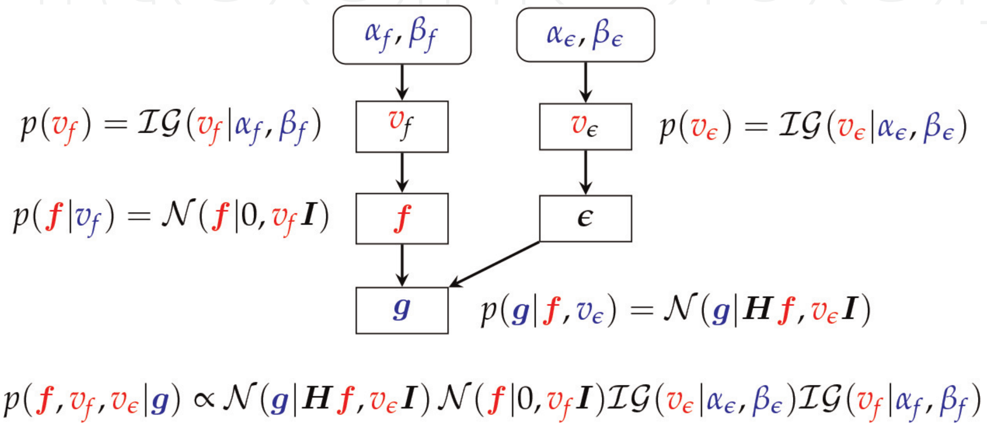

α_f, β_f
$\alpha_\epsilon, \beta_\epsilon$
$p(v_f) = \mathcal{IG}(v_f|\alpha_f, \beta_f)$
v_f
v_ϵ
$p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_\epsilon, \beta_\epsilon)$
$p(\boldsymbol{f}|v_f) = \mathcal{N}(\boldsymbol{f}|0, v_f\boldsymbol{I})$
$\boldsymbol{f}$
$\boldsymbol{\epsilon}$
$\boldsymbol{g}$
$p(\boldsymbol{g}|\boldsymbol{f}, v_\epsilon) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{H}\boldsymbol{f}, v_\epsilon\boldsymbol{I})$
$p(\boldsymbol{f}, v_f, v_\epsilon|\boldsymbol{g}) \propto \mathcal{N}(\boldsymbol{g}|\boldsymbol{H}\boldsymbol{f}, v_\epsilon\boldsymbol{I})\,\mathcal{N}(\boldsymbol{f}|0, v_f\boldsymbol{I})\mathcal{IG}(v_\epsilon|\alpha_\epsilon, \beta_\epsilon)\mathcal{IG}(v_f|\alpha_f, \beta_f)$

Figure 10.
Bayesian inference scheme in linear systems and Gaussian priors. The posterior is also Gaussian, and all the computations can be done analytically.

7. Non-Gaussian priors

Very often, assuming that the noise is Gaussian is valid in many applications, but a Gaussian prior may not be adequate. Thus, the case of Non-Gaussian priors is of great importance. A very well known example is the case of Generalized Gaussian:

$$p(\boldsymbol{f}) \propto \exp\left[-\gamma \sum_j \left|f_j\right|^{\beta}\right]. \tag{46}$$

The case of $\beta = 2$ is the Gaussian case, $\beta > 2$ gives the *Super-Gaussian* and $\beta < 1$ is called *Sub-Gaussian*. Its particular case $\beta = 1$ results to what is called *Double Exponential (DE)* prior law:

$$p(\boldsymbol{f}) \propto \exp\left[-\gamma \sum_j |f_j|\right] \propto \exp\left[-\gamma \|\boldsymbol{f}\|_1\right] \tag{47}$$

which, when using with a Gaussian likelihood, results to:

$$\begin{cases} p(\boldsymbol{f}|\boldsymbol{g}) & \propto \exp\left[-\frac{1}{2v_\epsilon} J(\boldsymbol{f})\right] \quad \text{with} \\ J(\boldsymbol{f}) & = \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H} \quad \boldsymbol{f}\|^2 + \lambda \|\boldsymbol{f}\|_1, \quad \lambda = \gamma v_\epsilon. \end{cases} \tag{48}$$

From this, we can see that the computation of MAP solution needs an appropriate optimization algorithm and the computations of the Posterior Mean (PM) or Posterior Covariance (PCov) or any other expectations become more difficult. However, as we will see later, VBA can be used to do approximate computations.

Another example is the Total Variation (TV) regularization method [11–14] which can be interpreted as choosing the prior

$$p(\boldsymbol{f}) \propto \exp\left[-\gamma \sum_j |f_j - f_{j-1}|\right] \propto \exp\left[-\gamma \|\boldsymbol{D}\boldsymbol{f}\|_1\right] \tag{49}$$

where $\boldsymbol{D}$ is the first order difference matrix.

This prior with a Gaussian model for noise results to:

$$\begin{cases} p(f|\boldsymbol{g}) & \propto \exp\left[-\frac{1}{2v\epsilon} J(f)\right] \quad \text{with} \\ J(f) = \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 + \lambda \|\boldsymbol{D}\boldsymbol{f}\|_1, \quad \lambda = \gamma v_\epsilon. & \end{cases} \tag{50}$$

One last example is using the Cauchy or more generally the Student-t distribution as the prior:

$$p(\boldsymbol{f}) \propto \exp\left[-\gamma \sum_j \ln\left(1 + f_j^2\right)^{\nu/2}\right] \tag{51}$$

which results to:

$$\begin{cases} p(\boldsymbol{f}|\boldsymbol{g}) & \propto \exp\left[-\frac{1}{2v_\epsilon} J(\boldsymbol{f})\right] \quad \text{with} \\ J(\boldsymbol{f}) & = \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 + \lambda \sum_j \ln\left(1 + f_j^2\right)^{\nu/2}, \quad \lambda = \gamma v_\epsilon. \end{cases} \tag{52}$$

These three examples are of great importance. They have been used in the framework of MAP estimation and thus the optimization of the criteria $J(\boldsymbol{f})$ for many linear inverse problems. However, the computation of other Bayesian estimators and uncertainty quantification (UQ) need again specific approximate solutions.

8. Hierarchical prior models

Even if simple Gaussian and non-Gaussian priors used in previous sections are of great importance and use in many applications, still they have, in many cases, limitations. For example, when we know that the signals have impulsive shapes or discontinuous or are piecewise continuous. The same limitations when we know, for example, that the images are composed of homogeneous regions with specified contours, or even, that the object under the test is composed of a limited number of homogeneous materials. Hierarchical models push farther these limitations of simple prior models. In the following, we consider three families of such hierarchical models: Sparsity aware models, Scaled Mixture models and Gauss-Markov-Potts models [10, 15–17].

8.1 Sparsity awarded hierarchical models

An easy way to consider the hierarchical sparsity awarded priors is to introduce a hidden variable, $\boldsymbol{z}$ and so consider the following Forward and prior models:

$$\begin{cases} \boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}, \\ \boldsymbol{f} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{\zeta}, \quad \boldsymbol{z} \text{ sparse modeled by Double Exp (DE)} \end{cases} \tag{53}$$

with

$$\begin{cases} p(\boldsymbol{g}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{H}\boldsymbol{f}, v_\epsilon \boldsymbol{I}) \\ p(\boldsymbol{f}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{D}\boldsymbol{z}, v_\xi \boldsymbol{I}) \rightarrow \\ p(\boldsymbol{z}) = \mathcal{DE}(\boldsymbol{f}|\gamma) \propto \exp\left[-\gamma \|\boldsymbol{z}\|_1\right] \end{cases} \tag{54}$$

Then, we have to find the expression of the joint posterior law $p(\boldsymbol{f}, \boldsymbol{z}|\boldsymbol{g})$:

$$\begin{cases} p(\boldsymbol{f}, \boldsymbol{z}|\boldsymbol{g}) \propto \exp\left[-J(\boldsymbol{f}, \boldsymbol{z})\right] \quad \text{with} \\ J(\boldsymbol{f}, \boldsymbol{z}) = \dfrac{1}{2v_\epsilon} \|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|_2^2 + \dfrac{1}{2v_\xi} \|\boldsymbol{f} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \gamma \|\boldsymbol{z}\|_1 \end{cases} \tag{55}$$

from which we can infer on $\boldsymbol{f}$ and $\boldsymbol{z}$ [10, 16, 18–22].

For the unsupervised case, we can add the appropriate priors:

$$\begin{cases} p(\gamma) = \mathcal{IG}\left(\gamma|\alpha_{\gamma_z}, \beta_{\gamma_z}\right) \\ p(v_\epsilon) = \mathcal{IG}\left(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}\right) \\ p(v_\xi) = \mathcal{IG}\left(v_\xi|\alpha_{\xi_z}, \beta_{\xi_z}\right) \end{cases} \tag{56}$$

and thus obtain:

$$\begin{cases} p(\boldsymbol{f},\boldsymbol{z},\gamma,v_\epsilon,v_\xi|\boldsymbol{g}) \propto \exp\left[-J(\boldsymbol{f},\boldsymbol{z},\gamma,v_\epsilon,v_\xi)\right] \quad \text{with} \\ J(\boldsymbol{f},\boldsymbol{z},v_\epsilon,v_\xi,\gamma) = \dfrac{1}{2v_\epsilon}\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|_2^2 + \dfrac{1}{2v_\xi}\|\boldsymbol{f}-\boldsymbol{D}\boldsymbol{z}\|_2^2 + \gamma\|\boldsymbol{z}\|_1 + \\ \qquad\qquad \left(\alpha_{\gamma_z}+n/2\right)\ln\gamma + \beta_{\gamma_z}/\gamma + \\ \qquad\qquad \left(\alpha_{\epsilon_0}+m/2\right)\ln v_\epsilon + \beta_{\epsilon_0}/v_\epsilon + \\ \qquad\qquad \left(\alpha_{\xi_z}+n/2\right)\ln v_\xi + \beta_{\xi_z}/v_\xi \end{cases} \tag{57}$$

It is interesting to note that the alternate optimization of this criterion gives the ADMM like algorithms [23–25] with the main advantage that here we have direct updates of the hyperparameters.

8.2 Scaled mixture models

Scaled Gaussian Mixture (SGM) models have been used in many applications to model rare events by their heavier tails with respect to Gaussian. They are also used in sparse signals modeling. A general SGM is defined as follows:

$$\mathcal{S}(f) = \int \mathcal{N}(f|0,v)\, p_m(v|\boldsymbol{\theta})\, \mathrm{d}v \tag{58}$$

where the variance of the Gaussian model $\mathcal{N}(f|0,v)$ is assumed to follow the mixing probability law $p_m(v|\boldsymbol{\theta})$. Between many possibilities for this mixing pdf is Inverse-Gamma which results to Student-t:

$$\mathcal{S}(f|\nu) = \int \mathcal{N}(f|0,v)\mathcal{IG}(v|\nu,\nu)\, \mathrm{d}v \tag{59}$$

which have been extended to more general case:

$$\mathcal{S}(f|\alpha,\beta) = \int \mathcal{N}(f|0,v)\mathcal{IG}(v|\alpha,\beta)\, \mathrm{d}v \tag{60}$$

This pdf models have been used with success in many developments in Bayesian approach for inverse problems by:

$$p(\boldsymbol{f}|\alpha,\beta) = \prod_j \int \mathcal{N}\left(\boldsymbol{f}_j|0,v_j\right)\mathcal{IG}\left(v_j|\alpha,\beta\right)\, \mathrm{d}v_j \tag{61}$$

or

$$p(\boldsymbol{f}|\alpha,\beta) = \int \mathcal{N}(\boldsymbol{f}|0,v\Sigma)\mathcal{IG}(v|\alpha,\beta)\, \mathrm{d}v \tag{62}$$

Scaled Gaussian mixture models have been used extensively for modeling sparse signals. However, it happens very often that the signals or images are not sparse directly, but their gradients are, or more generally in a transformed domain such as Fourier or Wavelet domains. We have used these models extensively in hierarchical way:

$$\begin{cases} \boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}, \\ \boldsymbol{f} = \boldsymbol{D}\boldsymbol{z} + \zeta, \quad \boldsymbol{z} \text{ sparse Student} \rightarrow \begin{cases} p\left(z_j|v_{z_j}\right) = \mathcal{N}\left(z_j|0,v_{z_j}\right), \\ p\left(v_{z_j}\right) = \mathcal{IG}\left(v_{z_j}|\alpha_{\boldsymbol{z}_0},\beta_{\boldsymbol{z}_0}\right) \end{cases} \end{cases} \tag{63}$$

where $\boldsymbol{D}$ represents any linear transformations and $\boldsymbol{D}^{-1}$ applied of $\boldsymbol{f}$ transforms it to a sparse vector $\boldsymbol{z}$.

The whole relations of the likelihood and priors are summarized in below:

$$\begin{cases} p(\boldsymbol{g}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{H}\boldsymbol{f}, v_\epsilon \boldsymbol{I}) \\ p(\boldsymbol{f}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{D}\boldsymbol{z}, v_\xi \boldsymbol{I}) \\ p(\boldsymbol{z}|v_{\boldsymbol{z}}) = \mathcal{N}(\boldsymbol{z}|0, V_{\boldsymbol{z}}) \\ p(v_{\boldsymbol{z}}) = \prod_j \mathcal{IG}\left(v_{\boldsymbol{z}_j}|\alpha_{\boldsymbol{z}_0}, \beta_{\boldsymbol{z}_0}\right) \\ p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(v_\xi) = \mathcal{IG}\left(v_\xi|\alpha_{\xi_{\boldsymbol{z}}}, \beta_{\xi_{\boldsymbol{z}}}\right) \end{cases} \tag{64}$$

and the corresponding joint posterior of all the unknowns writes:

$$\begin{cases} p(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{v}_{\boldsymbol{z}}, v_\epsilon, v_\xi|\boldsymbol{g}) \propto \exp\left[-J(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{v}_{\boldsymbol{z}}, v_\epsilon, v_\xi)\right] \\ J(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{v}_{\boldsymbol{z}}, v_\epsilon, v_\xi) = \dfrac{1}{2v_\epsilon}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|_2^2 + \dfrac{1}{2v_\xi}\|\boldsymbol{f} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \left\|\boldsymbol{V}_{\boldsymbol{z}}^{\frac{-1}{2}}\boldsymbol{z}\right\|_2^2 + \\ \qquad \sum_j (\alpha_{\boldsymbol{z}_0} + 1)\ln v_{\boldsymbol{z}_j} + \beta_{\boldsymbol{z}_0}/v_{\boldsymbol{z}_j} + \\ \qquad (\alpha_{\epsilon_0} + m/2)\ln v_\epsilon + \beta_{\epsilon_0}/v_\epsilon + \left(\alpha_{\xi_{\boldsymbol{z}}} + n/2\right)\ln v_\xi + \beta_{\xi_{\boldsymbol{z}}}/v_\xi \end{cases} \tag{65}$$

Looking at this expression, we see that we have:

- Quadratic optimization with respect to $\boldsymbol{f}$ and $\boldsymbol{z}$;
- Direct analytical expressions for the updates of the hyperparameters v_ϵ and v_ξ;
- Possibility to compute posterior mean and quantify uncertainties analytically via VBA.

A final case we consider is the case of Non-stationary noise and sparsity enforcing prior in the same framework.

$$\begin{cases} \boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \text{ non stationary} \rightarrow \begin{cases} p(\epsilon_i|v_{\epsilon_i}) = \mathcal{N}(\epsilon_i|0, v_{\epsilon_i}), \\ p(v_{\epsilon_i}) = \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \end{cases} \\ \boldsymbol{f} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{\zeta}, \quad \boldsymbol{z} \text{ sparse Student} \rightarrow \begin{cases} p\left(z_j|v_{z_j}\right) = \mathcal{N}\left(z_j|0, v_{z_j}\right), \\ p\left(v_{z_j}\right) = \mathcal{IG}\left(v_{z_j}|\alpha_{z_0}, \beta_{z_0}\right) \end{cases} \end{cases} \tag{66}$$

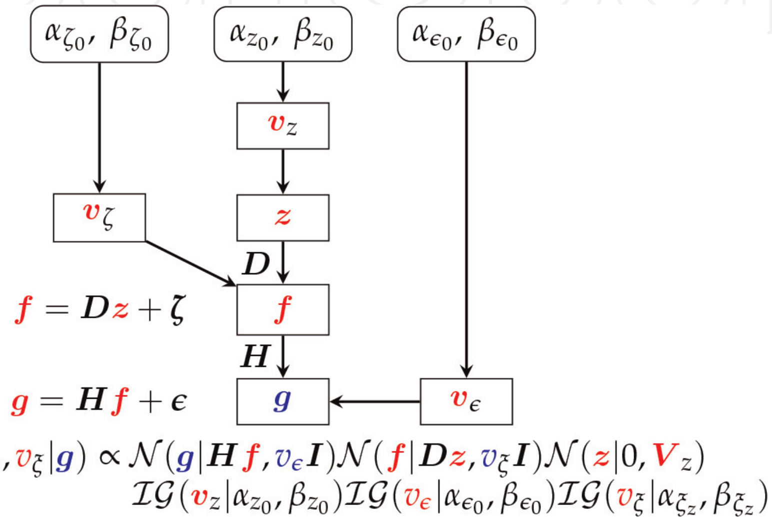
Again here, all the expressions of likelihood and priors can be summarized as follows:

$$\begin{cases} p(\boldsymbol{g}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{H}\boldsymbol{f}, V_\epsilon) \\ p(\boldsymbol{f}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{f}|D\boldsymbol{z}, v_\xi I) \\ p(\boldsymbol{z}|\boldsymbol{v}_{\boldsymbol{z}}) = \mathcal{N}(\boldsymbol{z}|0, V_{\boldsymbol{z}}) \\ p(\boldsymbol{v}_{\boldsymbol{z}}) = \prod_j \mathcal{IG}\left(v_{\boldsymbol{z}_j}|\alpha_{\boldsymbol{z}_0}, \beta_{\boldsymbol{z}_0}\right) \\ p(\boldsymbol{v}_\epsilon) = \prod_i \mathcal{IG}(v_{\epsilon_i}|\alpha_{\epsilon_0}, \beta_{\epsilon_0}) \\ p(v_\xi) = \mathcal{IG}\left(v_\xi|\alpha_{\xi_{\boldsymbol{z}}}, \beta_{\xi_{\boldsymbol{z}}}\right) \end{cases} \tag{67}$$

and the joint posterior of all the unknowns become:

$$\left\{\begin{array}{l} p(\boldsymbol{f},\boldsymbol{z},\boldsymbol{v}_{\boldsymbol{z}},\boldsymbol{v}_\epsilon,v_\xi|\boldsymbol{g}) \propto \exp\left[-J(\boldsymbol{f},\boldsymbol{z},\boldsymbol{v}_{\boldsymbol{z}},\boldsymbol{v}_\epsilon,v_\xi)\right] \\ J(\boldsymbol{f},\boldsymbol{z},\boldsymbol{v}_{\boldsymbol{z}},v_\epsilon,v_\xi) = \left\|\boldsymbol{V}_\epsilon^{\frac{-1}{2}}(\boldsymbol{g}-\boldsymbol{Hf})\right\|_2^2 + \frac{1}{2v_\xi}\|\boldsymbol{f}-D\boldsymbol{z}\|_2^2 + \left\|\boldsymbol{V}_{\boldsymbol{z}}{}^{\frac{-1}{2}}\boldsymbol{z}\right\|_2^2 + \\ \qquad\qquad \sum_j(\alpha_{\boldsymbol{z}_0}+1)\ln v_{\boldsymbol{z}_j} + \beta_{\boldsymbol{z}_0}/v_{\boldsymbol{z}_j} + \\ \qquad\qquad \sum_j(\alpha_{\epsilon_0}+1)\ln v_{\epsilon_i} + \beta_{\epsilon_0}/v_{\epsilon_i} + \\ \qquad\qquad (\alpha_{\xi_{\boldsymbol{z}}}+n/2)\ln v_\xi + \beta_{\xi_{\boldsymbol{z}}}/v_\xi \end{array}\right. \tag{68}$$

The following scheme shows graphically this case.



$\alpha_{\zeta_0}, \beta_{\zeta_0}$
$\alpha_{z_0}, \beta_{z_0}$
$\alpha_{\epsilon_0}, \beta_{\epsilon_0}$
$\boldsymbol{v}_z$
$\boldsymbol{v}_\zeta$
$\boldsymbol{z}$
$\boldsymbol{D}$
$\boldsymbol{f} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{\zeta}$
$\boldsymbol{f}$
$\boldsymbol{H}$
$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}$
$\boldsymbol{g}$
$\boldsymbol{v}_\epsilon$

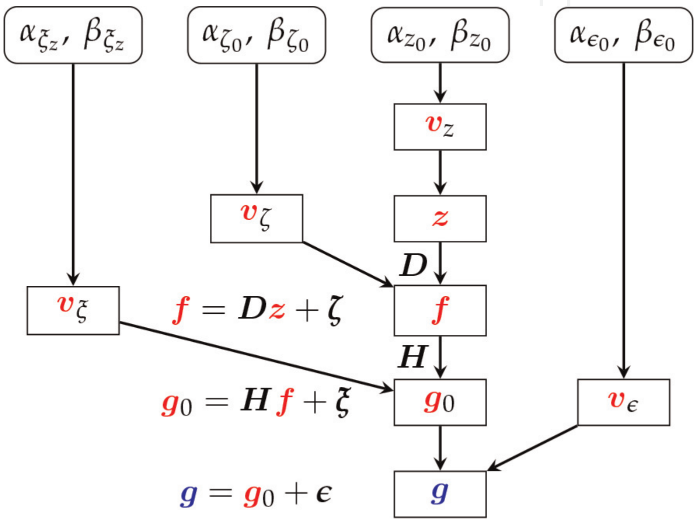
$$p(\boldsymbol{f},\boldsymbol{z},\boldsymbol{v}_z,v_\epsilon,v_\xi|\boldsymbol{g}) \propto \mathcal{N}(\boldsymbol{g}|\boldsymbol{H}\boldsymbol{f},v_\epsilon\boldsymbol{I})\mathcal{N}(\boldsymbol{f}|\boldsymbol{D}\boldsymbol{z},v_\xi\boldsymbol{I})\mathcal{N}(\boldsymbol{z}|0,\boldsymbol{V}_z)$$
$$\mathcal{IG}(\boldsymbol{v}_z|\alpha_{z_0},\beta_{z_0})\mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0},\beta_{\epsilon_0})\mathcal{IG}(v_\xi|\alpha_{\xi_z},\beta_{\xi_z})$$

8.3 A four level hierarchical model

To account separately for the measurement and forward modeling error, a more detailed and four level hierarchical model has been proposed:

$$\left\{\begin{array}{ll} \boldsymbol{g} = \boldsymbol{g}_0 + \boldsymbol{\epsilon}, & \text{measurement error} \\ \boldsymbol{g}_0 = \boldsymbol{Hf} + \xi, & \text{modeling error} \\ \boldsymbol{f} = \boldsymbol{Dz} + \boldsymbol{\zeta}, & \text{Prior model} \end{array}\right. \tag{69}$$

which accounts for two terms of errors (variable splitting) and Sparsity enforcing in Transformed domain prior: $\boldsymbol{f} = \boldsymbol{Dz} + \boldsymbol{\zeta}$ with $\boldsymbol{z}$ sparse, modeled itself by Normal-IG. This model is presented graphically here.



$\alpha_{\xi_z}, \beta_{\xi_z}$
$\alpha_{\zeta_0}, \beta_{\zeta_0}$
$\alpha_{z_0}, \beta_{z_0}$
$\alpha_{\epsilon_0}, \beta_{\epsilon_0}$
$\boldsymbol{v}_z$
$\boldsymbol{v}_\zeta$
$\boldsymbol{z}$
$\boldsymbol{D}$
v_ξ
$\boldsymbol{f} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{\zeta}$
$\boldsymbol{f}$
$\boldsymbol{H}$
$g_0 = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\xi}$
g_0
$\boldsymbol{v}_\epsilon$
$\boldsymbol{g} = g_0 + \boldsymbol{\epsilon}$
$\boldsymbol{g}$

In this model, there are three error terms: $\boldsymbol{\epsilon}$ the observation error, $\boldsymbol{\zeta}$ the forward modeling error and $\boldsymbol{\zeta}$ the transform domain modeling error. These are detailed in the following:

- $\boldsymbol{g} = \boldsymbol{g}_0 + \boldsymbol{\epsilon}, : \boldsymbol{\epsilon}$ is assumed to be Gaussian:

$$p(\boldsymbol{g}|\boldsymbol{g}_0, v_\epsilon) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{g}_0, v_\epsilon I), p(v_\epsilon) = \mathcal{IG}(v_\epsilon|\alpha_{\epsilon_0}, \beta_{\epsilon_0}),$$

- $\boldsymbol{g}_0 = \boldsymbol{H}\boldsymbol{f} + \xi, \xi$ is assumed to be Student-t:

$$\begin{cases} p(\boldsymbol{g}_0|\boldsymbol{f}, v_\xi) = \mathcal{N}(\boldsymbol{g}_0|\boldsymbol{H}\boldsymbol{f}, \boldsymbol{V}_\xi), \boldsymbol{V}_\xi = \text{diag}[v_\xi], \\ p(v_\xi) = \Pi_{i=1}^{M} p(v_{\xi_i}) = \Pi_{i=1}^{M} \mathcal{IG}\left(v_{\xi_i}|\alpha_{\xi_z}, \beta_{\xi_z}\right), \end{cases}$$

- $\boldsymbol{f} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{\zeta}, \boldsymbol{\zeta}$ is assumed to be Gaussian:

$$p(\boldsymbol{f}|\boldsymbol{z}, v_\zeta) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{D}\boldsymbol{z}, v_\zeta I), p(v_\zeta) = \mathcal{IG}\left(v_\zeta|\alpha_{\zeta_0}, \beta_{\zeta_0}\right),$$

- $\boldsymbol{z}$ is assumed to be sparse and thus modeled via Normal-IG:

$$\begin{cases} p(\boldsymbol{z}|\boldsymbol{v}_{\boldsymbol{z}}) = \mathcal{N}(\boldsymbol{z}|0, \boldsymbol{V}_{\boldsymbol{z}}), \boldsymbol{V}_{\boldsymbol{z}} = \text{diag}[\boldsymbol{v}_{\boldsymbol{z}}] \\ p(\boldsymbol{v}_{\boldsymbol{z}}) = \Pi_{j=1}^{N} p\left(\boldsymbol{v}_{\boldsymbol{z}_j}\right) = \Pi_{j=1}^{N} \mathcal{IG}\left(\boldsymbol{v}_{\boldsymbol{z}_j}|\alpha_{\boldsymbol{z}_0}, \beta_{\boldsymbol{z}_0}\right) \end{cases}$$

which results in:

$$p(\boldsymbol{f}, \boldsymbol{g}_0, \boldsymbol{z}, \boldsymbol{v}_\epsilon, \boldsymbol{v}_\xi, \boldsymbol{v}_{\boldsymbol{z}}|\boldsymbol{g}) \propto \exp\left[-J(\boldsymbol{f}, \boldsymbol{g}_0, \boldsymbol{z}, \boldsymbol{v}_\epsilon, \boldsymbol{v}_\xi, \boldsymbol{v}_{\boldsymbol{z}})\right] \tag{70}$$

with

$$\begin{aligned} J(\boldsymbol{f}, \boldsymbol{g}_0, \boldsymbol{z}, \boldsymbol{v}_\epsilon, \boldsymbol{v}_\xi, \boldsymbol{v}_{\boldsymbol{z}}) = & \frac{1}{2\boldsymbol{v}_\epsilon} \|\boldsymbol{g} - \boldsymbol{g}_0\|_2^2 + (\alpha_{\epsilon_0} + 1) \ln \boldsymbol{v}_\epsilon + \frac{\beta_{\epsilon_0}}{\boldsymbol{v}_\epsilon} \\ & + \frac{1}{2} \left\| \boldsymbol{V}_\xi^{-1/2} (\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}) \right\|_2^2 + \sum_{i=1}^{M} \left[(\alpha_{\xi_z} + 1) \ln \boldsymbol{v}_{\xi_i} + \frac{\beta_{\xi_z}}{\boldsymbol{v}_{\xi_i}} \right] \\ & + \frac{1}{2\boldsymbol{v}_\zeta} \|\boldsymbol{f} - \boldsymbol{D}\,\boldsymbol{z}\|_2^2 + (\alpha_{\zeta_0} + 1) \ln \boldsymbol{v}_\xi + \frac{\beta_{\zeta_0}}{\boldsymbol{v}_\xi} \\ & + \frac{1}{2} \|\boldsymbol{V}_{\boldsymbol{z}}{}^{-1/2} \boldsymbol{z}\|_2^2 + \sum_{j=1}^{N} \left[(\alpha_{\boldsymbol{z}_0} + 1) \ln \boldsymbol{v}_{\boldsymbol{z}_j} + \frac{\beta_{\boldsymbol{z}_0}}{\boldsymbol{v}_{\boldsymbol{z}_j}} \right] \end{aligned} \tag{71}$$

Using then the JMAP approach with an alternate optimization strategy needs the following optimization steps:

- with respect to $\boldsymbol{f}$: $J(\boldsymbol{f}) = \frac{1}{2} \left\| \boldsymbol{V}_\xi^{-1/2} (\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}) \right\|_2^2 + \frac{1}{2\boldsymbol{v}_\zeta} \|\boldsymbol{f} - \boldsymbol{D}\boldsymbol{z}\|_2^2$

- with respect to $\boldsymbol{g}_0$: $J(\boldsymbol{g}_0) = \frac{1}{2\boldsymbol{v}_\epsilon} \|\boldsymbol{g} - \boldsymbol{g}_0\|_2^2 + \frac{1}{2} \left\| \boldsymbol{V}_\xi^{-1/2} (\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}) \right\|_2^2$

- with respect to $\boldsymbol{z}$: $J(\boldsymbol{z}) = \frac{1}{2\boldsymbol{v}_\zeta} \|\boldsymbol{f} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \frac{1}{2} \|\boldsymbol{V}_{\boldsymbol{z}}^{-1/2} \boldsymbol{z}\|_2^2$

- with respect to $\boldsymbol{v}_\epsilon : J(\boldsymbol{v}_\epsilon) = \frac{1}{2\boldsymbol{v}_\epsilon}\left\|\boldsymbol{g} - \boldsymbol{g}_0\right\|_2^2 + (\alpha_{\epsilon_0} + 1)\ln \boldsymbol{v}_\epsilon + \frac{\beta_{\epsilon_0}}{\boldsymbol{v}_\epsilon}$
- with respect to $\boldsymbol{v}_{\xi_i} : J\left(\boldsymbol{v}_{\xi_i}\right) = \frac{1}{2}\left\|\boldsymbol{V}_\xi^{-1/2}\left(\boldsymbol{g}_0 - \boldsymbol{H}\boldsymbol{f}\right)\right\|_2^2 + \sum_{i=1}^M\left[\left(\alpha_{\xi_{\boldsymbol{z}}} + 1\right)\ln \boldsymbol{v}_{\xi_i} + \frac{\beta_{\xi_{\boldsymbol{z}}}}{\boldsymbol{v}_{\xi_i}}\right]$
- with respect to $v_\zeta : J\left(v_\zeta\right) = \frac{1}{2v_\zeta}\|\boldsymbol{f} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \left(\alpha_{\zeta_0} + 1\right)\ln v_\xi + \frac{\beta_{\zeta_0}}{v_\xi}$
- to $v_{\boldsymbol{z}_j} : J\left(v_{\boldsymbol{z}_j}\right) = \frac{1}{2}\left\|\boldsymbol{V}_{\boldsymbol{z}}^{-1/2}\boldsymbol{z}\right\|_2^2 + \sum_{j=1}^N\left[\left(\alpha_{\boldsymbol{z}_0} + 1\right)\ln v_{\boldsymbol{z}_j} + \frac{\beta_{\boldsymbol{z}_0}}{v_{\boldsymbol{z}_j}}\right]$

This approach has the following main advantages and limitations.
Advantages:

- All the optimization are either quadratic or explicit
- Quadratic optimizations can be done efficiently
- For great dimensional problems, the needed operators $\boldsymbol{H}$, $\boldsymbol{H}'$, $\boldsymbol{D}$ and $\boldsymbol{D}'$ can be implemented on GPU
- For Computed Tomography, efficient GPU implementation of these operators have been done in our group for 2D and 3D CT.

Limitations:

- Huge amount of memory is needed for $\boldsymbol{f}$, $\boldsymbol{g}_0$, v_ξ and $\boldsymbol{v}_z$
- No easy way to study the global convergence of the algorithm.
- The number of hyper-hyperparameters $(\alpha_{\epsilon_0}, \beta_{\epsilon_0})$, $\left(\alpha_{\xi_z}, \beta_{\xi_z}\right)$, $\left(\alpha_{\zeta_0}, \beta_{\zeta_0}\right)$, $(\alpha_{z_0}, \beta_{z_0})$ to be fixed is important. However, the results are not so sensitive to these parameters.

8.4 Gauss-Markov-Potts models

To introduce the Gauss-Markov-Potts model, let us have a look at the images in **Figure 11**.

The question we want to answer is: which prior model can be more appropriate for these images? One way, to answer to this question is either look at the histogram of the pixels or the pixels of the gradient images. Then, if we take one typical line of the gradient or one typical line of the image itself and draw them as a 1D-signal, we obtain the cases in **Figure 12**.

From these two figures, we see that for some cases, a Gaussian or generalized Gaussian may be very good models. But, for other cases, if we want to explicitly account for the presence of the contours, we can introduce a binary hidden variable to represent it. Finally, for the last example of the image in **Figure 10** and its corresponding typical line in **Figure 11**, we need to introduce a hidden variable $\boldsymbol{z}$ which encodes the following fact that:

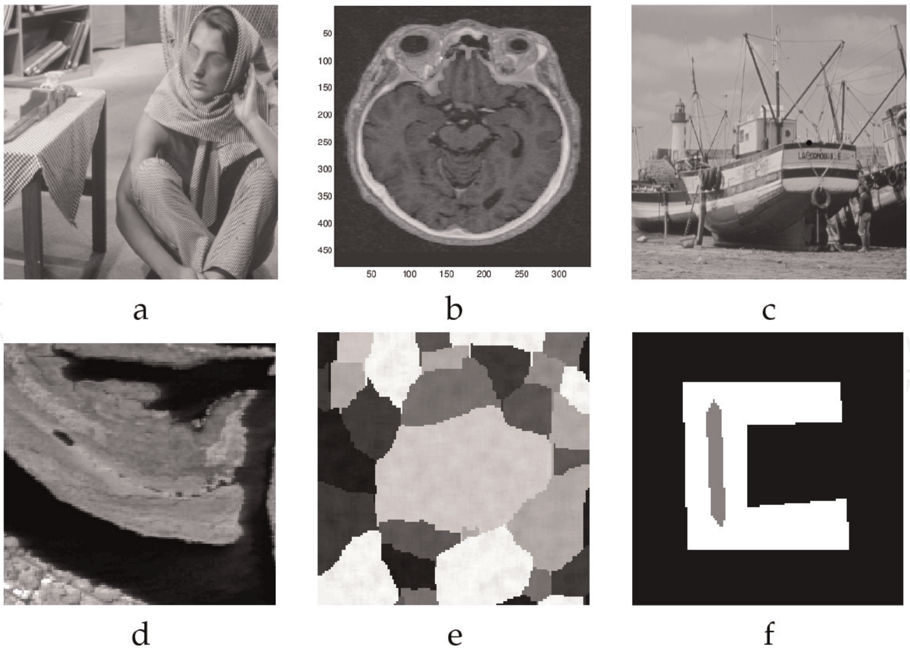


Figure 11.
Different images with different characteristics in different imaging systems.

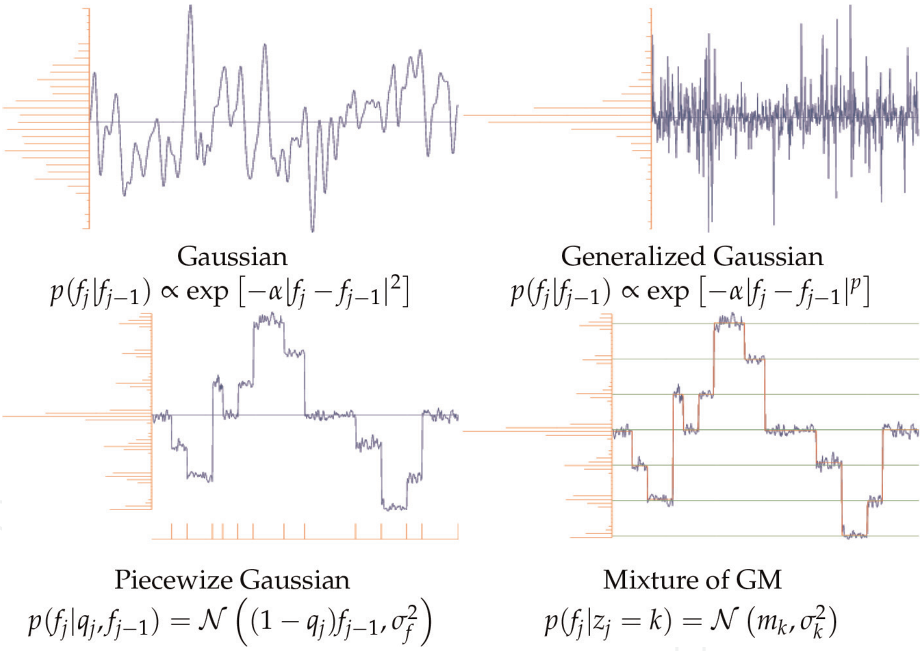


Figure 12.
Different possible prior modeling in relation to the different images of ***Figure 11****.*

In NDT applications of CT, the objects are, in general, composed of a finite number of materials, and the voxels corresponding to each material are grouped in compact regions.

How to model this prior information?

To answer to this question, first consider such an image $f(\boldsymbol{r})$ with its segmentation $z(\boldsymbol{r})$ and contours $q(\boldsymbol{r})$ as shown in **Figure 13**.

As it can be seen, we introduced two hidden variables $z(\boldsymbol{r})$ and $q(\boldsymbol{r})$, the first representing the segmentation and the second the contours of the image. $z(\boldsymbol{r})$ takes the integer values $\{k = 1, \cdots, K\}$, each presented by a different color and $q(\boldsymbol{r})$ a binary

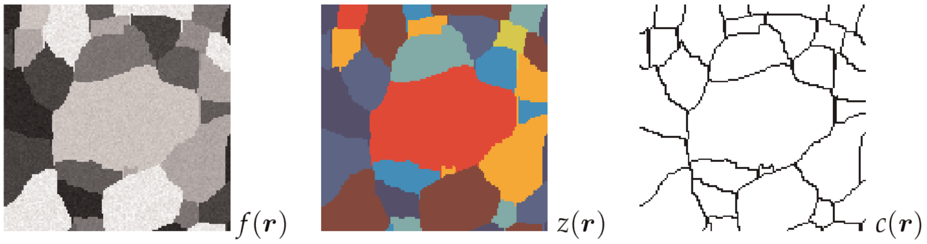


Figure 13.
An image of an object composed of homogeneous compact regions, its segmentation and the contours of thoses regions.

value $\{0, 1\}$. The second can easily be obtained from the first. So, from now, we consider only $z(\boldsymbol{r})$.

As each value of z represents a homogeneous material, we can translate this by:

$$p(\,f(\boldsymbol{r})|z(\boldsymbol{r})=k, m_k, v_k) = \mathcal{N}(m_k, v_k) \tag{72}$$

encoding the fact that inside each homogeneous material, i.e.; all the pixels having $z(r)=k,$ represent a homogenous material characterized by the two parameters $f\ (m_k, v_k)$. This results to:

$$p(f(\boldsymbol{r})) = \sum_k P(z(\boldsymbol{r})=k)\mathcal{N}(m_k, v_k)\ \text{Mixture of Gaussians} \tag{73}$$

which shows the mixture of Gaussian model of the pixel values. See also **Figure 14**.

The next step is to propose a probability distribution for $\boldsymbol{z}$. As we want a compactness of the regions, a Markov modeling is appropriate:

$$p(z(\boldsymbol{r})|z(\boldsymbol{r}'), \boldsymbol{r}' \in \mathcal{V}(\boldsymbol{r})) \propto \exp\left[-\gamma \sum_{\boldsymbol{r}' \in \mathcal{V}(\boldsymbol{r})} \delta(z(\boldsymbol{r}) - z(\boldsymbol{r}'))\right] \tag{74}$$

A Potts Markov model is still more appropriate:

$$p(z(\boldsymbol{r}), \boldsymbol{r} \in \Omega) \propto \exp\left[-\gamma \sum_{\boldsymbol{r} \in \Omega} \sum_{\boldsymbol{r}' \in \mathcal{V}(\boldsymbol{r})} \delta(z(\boldsymbol{r}) - z(\boldsymbol{r}'))\right] \tag{75}$$

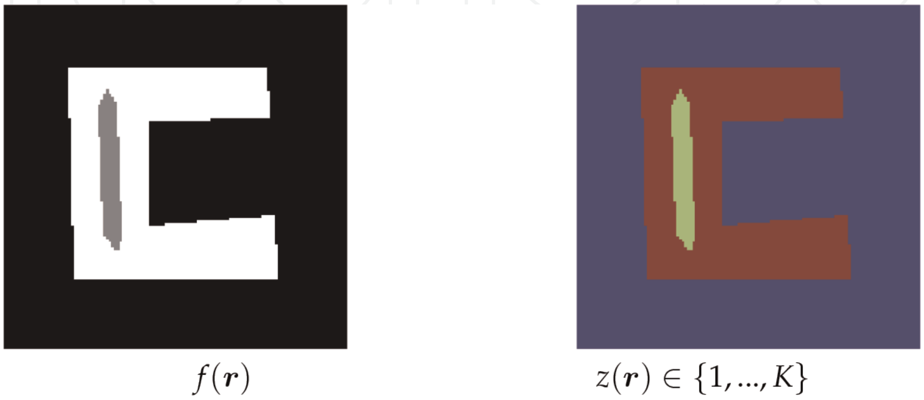


Figure 14.
A metalic object with a default area inside it: Black pixels represent air, white pixels metal and gray pixels the defaults area. On left image these are codes by colors ($z=1$ represents air, $z=2$ represents metal and $z=3$ represents default area.

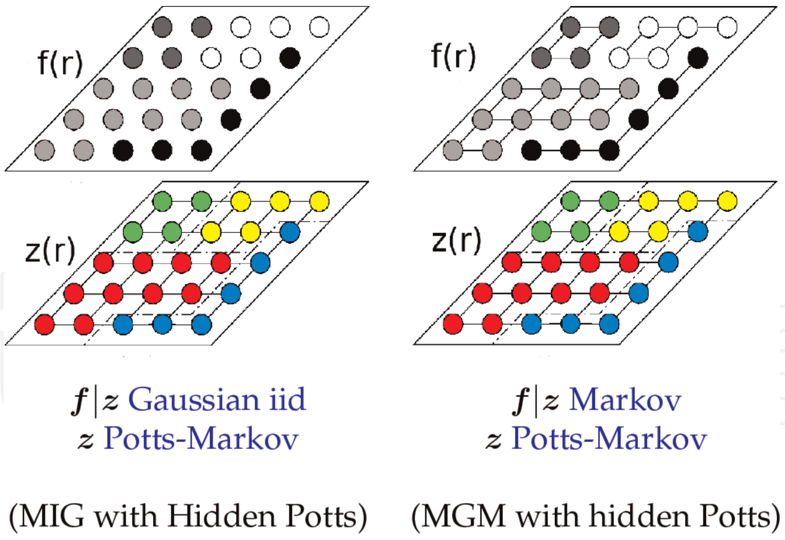


Figure 15.
Two proposed gauss-Markov-Potts models used in many NDT applications.

where Ω represents all pixels of the image.

Thus, to each pixel of the image is associated 2 variables $f(\boldsymbol{r})$ and $z(\boldsymbol{r})$ with the following possible properties:

- $\boldsymbol{f}|\boldsymbol{z}$ Gaussian iid, $\boldsymbol{z}$ iid: Mixture of Gaussians
- $\boldsymbol{f}|\boldsymbol{z}$ Gauss-Markov, $\boldsymbol{z}$ iid: Mixture of Gauss-Markov
- $\boldsymbol{f}|\boldsymbol{z}$ Gaussian iid, $\boldsymbol{z}$ Potts-Markov: Mixture of Independent Gaussians, (MIG with Hidden Potts)
- $\boldsymbol{f}|\boldsymbol{z}$ Markov, $\boldsymbol{z}$ Potts-Markov: Mixture of Gauss-Markov, (MGM with hidden Potts)

From these four different cases, we consider two which are illustrated in **Figure 15**.

Using the notations on this figure, and noting by $\boldsymbol{f}$ all the pixels of the image, by $\boldsymbol{z}$ all the pixels of the segmented image, and by $\boldsymbol{\theta}$ all the parameters $\{v_\epsilon, (\alpha_k, m_k, v_k), k = 1, \cdots, K\}$, we can write:

$$p(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{g}) \propto p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{v}_\epsilon) p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v}) p(\boldsymbol{z}|\gamma, \alpha) p(\boldsymbol{\theta}) \qquad (76)$$

where

$$\boldsymbol{m} = \{m_k, k = 1, \cdot, K\}, v = \{v_k, k = 1, \cdot, K\}, \alpha = \{\alpha_k, k = 1, \cdot, K\}, \boldsymbol{\theta} = \{v_\epsilon, m, v, alphab\}$$

The expressions of $p(\boldsymbol{g}|\boldsymbol{f}, v_\epsilon), p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{v})$ and $p(\boldsymbol{z}|\gamma, \boldsymbol{\alpha})$ have been given before. We need to define $p(\boldsymbol{\theta})$ which can be chosen as the conjugate priors: Dirichlet for α, Gaussian for $\boldsymbol{m}$ and Inverse-Gamma for all the variances.

Direct computation and use of $p(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{g}; \mathcal{M})$ is too complex, because we do not have analytical expression for the proportionality term of the joint probability law:

$$p(\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{g}) \propto p(\boldsymbol{g}|\boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{z}) p(\boldsymbol{\theta}) \qquad (77)$$

As we have three sets of variables $\boldsymbol{f}$, $\boldsymbol{z}$ and $\boldsymbol{\theta}$, we can use different schemes, for example a Gibbs sampling scheme:

$$\hat{\boldsymbol{f}} \sim p\left(\boldsymbol{f}|\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}},\boldsymbol{g}\right) \rightarrow \hat{\boldsymbol{z}} \sim p\left(\boldsymbol{z}|\hat{\boldsymbol{f}},\hat{\boldsymbol{\theta}},\boldsymbol{g}\right) \rightarrow \hat{\boldsymbol{\theta}} \sim \left(\boldsymbol{\theta}|\hat{\boldsymbol{f}},\hat{\boldsymbol{z}},\boldsymbol{g}\right) \tag{78}$$

with:

- Sample $\boldsymbol{f}$ from $p\left(\boldsymbol{f}|\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}},\boldsymbol{g}\right) \propto p(\boldsymbol{g}|\boldsymbol{f},\boldsymbol{\theta})p\left(\boldsymbol{f}|\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}}\right)$

Needs optimisation of a quadratic criterion.

- Sample $\boldsymbol{z}$ from $p\left(\boldsymbol{z}|\hat{\boldsymbol{f}},\hat{\boldsymbol{\theta}},\boldsymbol{g}\right) \propto p\left(\boldsymbol{g}|\hat{\boldsymbol{f}},\hat{\boldsymbol{z}},\hat{\boldsymbol{\theta}}\right)p(\boldsymbol{z})$

Needs sampling of a Potts Markov field.

- Sample $\boldsymbol{\theta}$ from $p\left(\boldsymbol{\theta}|\hat{\boldsymbol{f}},\hat{\boldsymbol{z}},\boldsymbol{g}\right) \propto p\left(\boldsymbol{g}|\hat{\boldsymbol{f}},\sigma_\epsilon^2 I\right)p\left(\hat{\boldsymbol{f}}|\hat{\boldsymbol{z}},(m_k,v_k)\right)p(\boldsymbol{\theta})$

More details and other schemes such as JMAP and VBA can be found in Refs. [26].

To illustrate an example of application, we considered a NDT application, where a metalic object is tested to detect a default inside it. As, the problem was, not only to detect the default, but also to characterize its shape and size, an X-ray computed tomography (CT) with only two projections is proposed and used. This problem is illustrated in **Figure 16**.

The mathematical part of this very ill-posed inverse problem is the following:

Given the functions $g_1(x)$ and $g_2(y)$ find the image $f(x,y)$.

This problem also arise in probability theory and statistics, where $f(x,y)$ is a joint distribution and $g_1(x)$ and $g_2(y)$ its two marginals. We know that this problem has infinite number of solutions: $f(x,y) = g_1(x)g_2(y)\Omega(x,y)$ where $\Omega(x,y)$ is called a Copula:

$$\int \Omega(x,y)\mathrm{d}x = 1 \text{ and } \int \Omega(x,y)\mathrm{d}y = 1$$

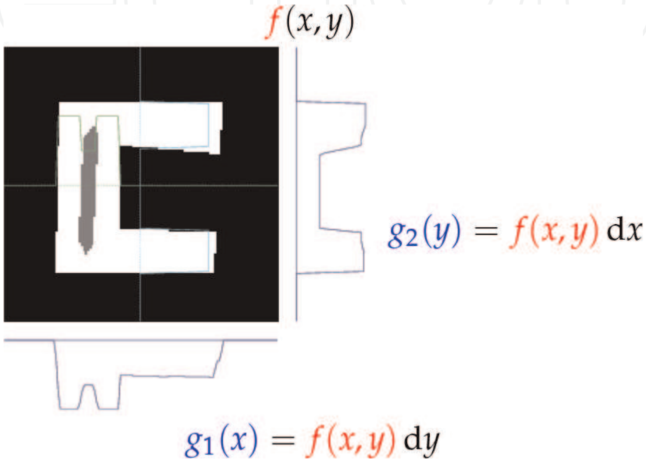


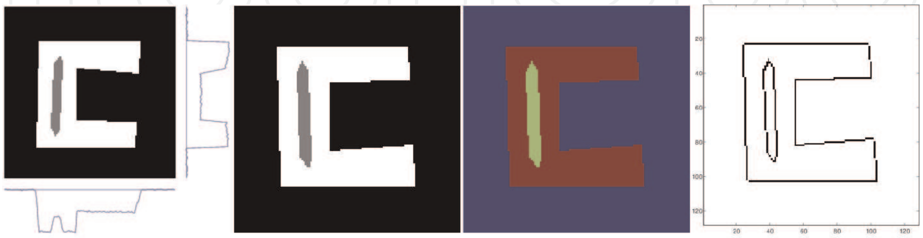
Figure 16.
A non destructive testing (NDT) application where $f(x,y)$ has to be reconstructed from its marginals $g_1(x)$ and $g_2(y)$.

So, any arbitrary copula function defines a solution. The problem is ill-posed and we need to use any possible prior information to try to obtain a unique or acceptable solution. The probabilistic solution we proposed is illustrated in **Figure 17**.

Unsupervised Bayesian estimation:

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

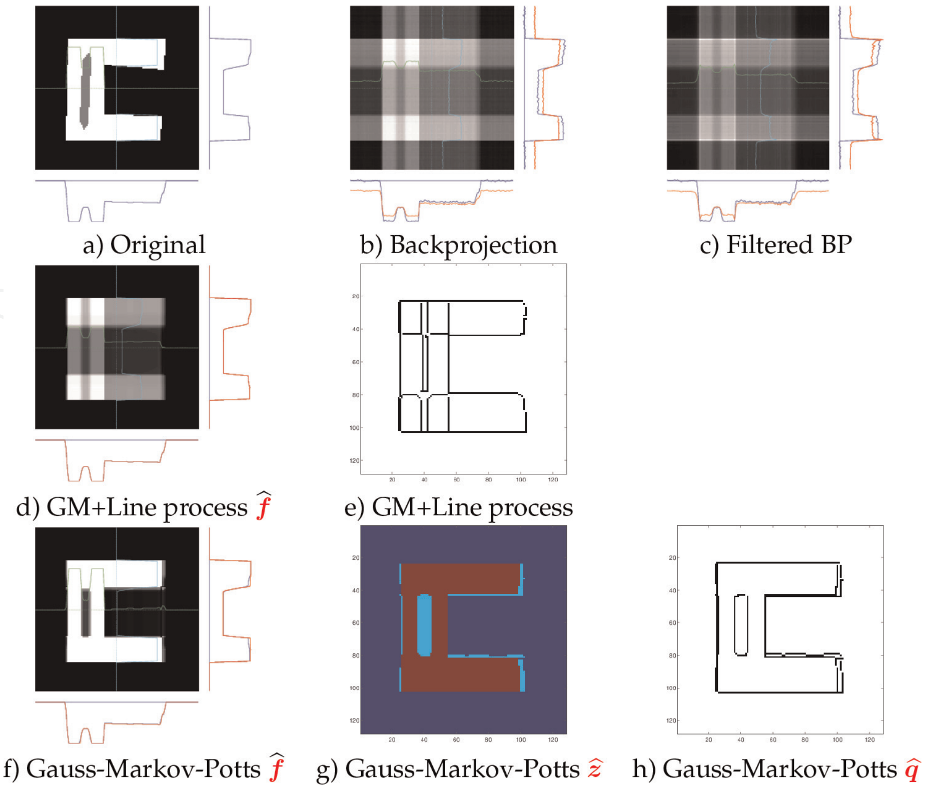
A summary of the results is given in **Figure 18** where the proposed method result.



$\boldsymbol{g}\vert\boldsymbol{f}$	$\boldsymbol{f}\vert\boldsymbol{z}$	$\boldsymbol{z}$	$\boldsymbol{q}$
$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\epsilon}$	iid Gaussian	iid	$q(\boldsymbol{r}) \in \{0,1\}$
$\boldsymbol{g}\vert\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{f},\sigma_\epsilon^2\boldsymbol{I})$	or	or	$1-\delta(z(\boldsymbol{r})-z(\boldsymbol{r}'))$
Gaussian	Gauss-Markov	Potts	binary

Forward model | Gauss-Markov-Potts Prior Model | Auxilary

Figure 17.
Probabilistic Bayesian method for the NDT image resonstruction problem.



a) Original b) Backprojection c) Filtered BP

d) GM+Line process $\widehat{f}$ e) GM+Line process

f) Gauss-Markov-Potts $\widehat{f}$ g) Gauss-Markov-Potts $\widehat{z}$ h) Gauss-Markov-Potts $\widehat{q}$

Figure 18.
Probabilistic Bayesian method for the NDT image resonstruction problem. a) Shows the original image ***f****, b) is the result of Back-projection, c) is the result of filtered Back-projection, d) and e) are the result of a Markov model with hidden line process, and f), g) and h) show the results of the gauss-Markov-Potts method.*

9. Bayesian computation

As we could see, very often, we can find the expression of the posterior law $p(\boldsymbol{f}|g)$, sometimes exactly as is the case of the linear models with Gaussian priors in the previous section, but often up to the normalization constant (the evidence term) $p(\boldsymbol{g})$ in:

$$p(\boldsymbol{f}|\boldsymbol{g}) = \frac{1}{p(\boldsymbol{g})} p(\boldsymbol{g}|\boldsymbol{f}) p(\boldsymbol{f}) = \frac{1}{p(\boldsymbol{g})} p(\boldsymbol{g},\boldsymbol{f}). \tag{79}$$

This term is not necessary for Maximum A Posteriori (MAP) but it is needed for Expected A Posteriori (EAP) and for doing any other expectation computation.

This is the case, almost in all *Non-Gaussian prior models* or *Non-Gaussian noise models* or the *Non-Linear forward models*. In this chapter, a few cases are considered more in detail. Even in the Gaussian and linear case which is the simplest case, and we have analytical expressions for almost everything, the computational cost for large scale problems brings us to search for approximate but fast solutions.

9.1 Large scale linear and Gaussian models

As we could see in previous chapter, the linear forward model $\boldsymbol{g} = \boldsymbol{Hf} + \boldsymbol{\epsilon}$ with Gaussian noise and Gaussian prior is the simplest case where we can do all the computations analytically.

$$\left\{\begin{array}{l} p(\boldsymbol{g}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{Hf}, v_\epsilon \boldsymbol{I}) \\ p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{f}_0, v_f \boldsymbol{I}) \end{array}\right. \rightarrow \left\{\begin{array}{l} p(\boldsymbol{g}) = \mathcal{N}(\boldsymbol{g}|\boldsymbol{H}\boldsymbol{f}_0, v_f \boldsymbol{H}\,\boldsymbol{H}' + v_\epsilon \boldsymbol{I}), \\ p(\boldsymbol{f}|\boldsymbol{g}) = \mathcal{N}\left(\boldsymbol{f}|\hat{\boldsymbol{f}}, \hat{\Sigma}\right) \quad \text{with :} \\ \hat{\boldsymbol{f}} = \boldsymbol{f}_0 + [\boldsymbol{H}'\,\boldsymbol{H} + \lambda I]^{-1} \boldsymbol{H}'(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}_0) \\ \hat{\Sigma} = v_\epsilon [\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{I}]^{-1} \lambda = \dfrac{v_\epsilon}{v_f} \end{array}\right. \tag{80}$$

The trick here is, for example, for computing $\hat{\boldsymbol{f}}$ to use the fact that

$$\left\{\begin{array}{l} p(\boldsymbol{g}|\boldsymbol{f}) \propto \exp\left[-\dfrac{1}{2v_\epsilon}\|\boldsymbol{g} - \boldsymbol{Hf}\|_2^2\right] \\ p(\boldsymbol{f}) \propto \exp\left[-\dfrac{1}{2v_f}\|\boldsymbol{f}\|_2^2\right] \\ p(\boldsymbol{f}|\boldsymbol{g}) \propto \exp\left[-\dfrac{1}{2v_\epsilon}J(\boldsymbol{f})\right] \text{ with } J(\boldsymbol{f}) = \dfrac{1}{2}\|\boldsymbol{g} - \boldsymbol{Hf}\|^2 + \lambda\|\boldsymbol{f}\|_2^2, \quad \lambda = \dfrac{v_\epsilon}{v_f} \end{array}\right. \tag{81}$$

and, as the mean and the mode of a Gaussian probability law are the same, we can use:

$$\hat{\boldsymbol{f}} = \underset{\boldsymbol{f}}{\operatorname{argmax}}\{J(\boldsymbol{f})\} \text{ with } J(\boldsymbol{f}) = \|\boldsymbol{g} - \boldsymbol{Hf}\|^2 + \lambda\|\boldsymbol{f}\|^2 \tag{82}$$

and so the problem can be cast as an *optimization* problem of a quadratic criterion for which there are a great number of algorithms. Let here to show the simplest one which is the gradient based and so needs the expression of the gradient:

$$\nabla J(\boldsymbol{f}) = -2\boldsymbol{H}'(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}) + 2\lambda \boldsymbol{f} \tag{83}$$

which can be summarized as follows:

$$\begin{cases} \boldsymbol{f}^{(0)} = 0 \\ \boldsymbol{f}^{(k+1)} = \boldsymbol{f}^{(k)} + \alpha\left[\boldsymbol{H}'\left(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}^{(k)}\right) + 2\lambda \boldsymbol{f}^{(k)}\right] \end{cases} \tag{84}$$

As we can see, at each iteration, we need to be able to compute the *forward operation* $\boldsymbol{H}\boldsymbol{f}$ and the *backward operation* $\boldsymbol{H}'\delta\boldsymbol{g}$ where $\delta g = \boldsymbol{g}–\boldsymbol{H}\boldsymbol{f}$. This optimization algorithm needs to write two programs:

- Forward operation $\boldsymbol{H}\boldsymbol{f}$

- Adjoint operation $\boldsymbol{H}'\delta\boldsymbol{g}$

These two operations can be implemented using High Performance parallel processors such as Graphical Processor Units (GPU).

The computation of the posterior covariance is much more difficult. There are a few methods: The first category is the methods which use the particular structure of the matrix $\boldsymbol{H}$ and $\boldsymbol{H}'\boldsymbol{H}$ or $\boldsymbol{H}\boldsymbol{H}'$ as we can use the matrix inversion lemma and see that

$$\hat{\Sigma} = v_\epsilon[\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{I}]^{-1} = v_\epsilon \boldsymbol{I} - \boldsymbol{H}'\left[\boldsymbol{H}\boldsymbol{H}' + \lambda^{-1}\boldsymbol{I}\right]^{-1} \tag{85}$$

For example, in a signal deconvolution problem, the matrix $\boldsymbol{H}$ has a Toeplitz structure and so have the matrices $\boldsymbol{H}'\boldsymbol{H}$ and $\boldsymbol{H}\boldsymbol{H}'$ which can be approximated by Circulant matrices and be diagonalized using the Fourier Transform.

The second, more general, is to approximate $\hat{\Sigma}$ by a diagonal matrix, which can also be interpreted as to approximate the posterior law $p(\boldsymbol{f}|\boldsymbol{g})$ by a separable $q(\boldsymbol{f}) = \Pi_j q\left(f_j\right)$. This brings us naturally to the Approximate Bayesian Computation (ABC). But, before going to the details of ABC methods, let consider the case where the hyperparameters of the problem (parameters of the prior laws) are also unknown.

To be able to do the computation, we need mainly to compute the determinant of the matrix $v_f\boldsymbol{H}\boldsymbol{H}' + v_\epsilon \boldsymbol{I}$ for $p(\boldsymbol{g})$ and the inverse of the matrices $[\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{I}]$ or $\left[\boldsymbol{H}\boldsymbol{H}' + \lambda^{-1}\boldsymbol{I}\right]$.

9.2 Large scale computation of the posterior covariance

Computing the determinant of the matrix $v_f\,\boldsymbol{H}\boldsymbol{H}' + v_\epsilon \boldsymbol{I}$ for $p(\boldsymbol{g})$ and the inverse of the matrices $[\boldsymbol{H}'\boldsymbol{H} + \lambda \boldsymbol{I}]$ or, $\left[\boldsymbol{H}\boldsymbol{H}' + \lambda^{-1}\boldsymbol{I}\right]$ which are needed for uncertainty quantification, are between the greatest subjects of open research for *Big Data* problems. Here, we consider a few cases.

9.2.1 Structured matrices

One solution is to use the particular structure of these matrices when possible. This is the case for deconvolution or image restoration, where these matrices have Toeplitz or Bloc-Toeplitz structures which can be well approximated by Circulant or Bloc-Circulant matrices and diagonalized using Fourier Transform (FT) and Fast FT

(FFT). The main idea here is using the properties of the circulant matrices: If $\boldsymbol{H}$ is a circulant matrix, then

$$\boldsymbol{H} = \boldsymbol{F}\Lambda\boldsymbol{F}' \tag{86}$$

where $\boldsymbol{F}$ is the DFT or FFT matrix and $\boldsymbol{F}'$ the IDFT or IFFT and Λ is a diagonal matrix whose elements are the FT of the first line of the circulent matrix. As the first line of that circulent matrix contains the samples of the impulse response, the vector of the diagonal elements represents the spectrum of the impulse response (transfer function). Using this property, we have:

$$[\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{I}]^{-1} = [\boldsymbol{F}'\Lambda\boldsymbol{F}\boldsymbol{F}'\Lambda + \lambda]^{-1} = \left[\boldsymbol{F}'\Lambda^2\boldsymbol{F} + \lambda\boldsymbol{I}\right]^{-1} = \boldsymbol{F}\left[\Lambda^2 + \lambda\boldsymbol{I}\right]^{-1}\boldsymbol{F}' \tag{87}$$

9.2.2 Sampling based methods

Second solution is generating samples from the posterior law and use them to compute the variances and covariances. So, the problem is how to generate a sample from the posterior law

$$\begin{cases} p(\boldsymbol{f}|\boldsymbol{g}) = \mathcal{N}\left(\boldsymbol{f}|\hat{\boldsymbol{f}}, \hat{\Sigma}\right) \quad \text{with :} \\ \hat{\boldsymbol{f}} = \boldsymbol{f}_0 + [\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{I}]^{-1}\boldsymbol{H}'(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}_0) \\ \hat{\Sigma} = v_\epsilon[\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{I}]^{-1}, \quad \lambda = \dfrac{v_\epsilon}{v_f} \end{cases} \tag{88}$$

One solution is to compute the Cholesky decomposition of the covariance matrix $\hat{\Sigma} = \boldsymbol{A}\boldsymbol{A}'$, generate a vector, $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{u}|0, \boldsymbol{I})$ and then generate a sample $\boldsymbol{f} = \boldsymbol{A}\boldsymbol{u} + \hat{\boldsymbol{f}}$ [27]. We can compute $\hat{\boldsymbol{f}}$ by optimizing

$$J(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2 + \lambda\left\|\boldsymbol{f} - \boldsymbol{f}_0\right\|_2^2, \lambda = \frac{v_\epsilon}{v_f}, \tag{89}$$

but the main computational cost is the Cholesky factorization.

Another approach, called Perturbation-Optimization [28, 29] is based on the following property:

If we note $\boldsymbol{x} = \boldsymbol{f} + [\boldsymbol{H}'\boldsymbol{H} + \lambda\boldsymbol{I}]^{-1}\boldsymbol{H}'(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f})$ and look for its expected and covariance matrix, it can be shown that:

$$\begin{cases} \mathrm{E}\{\boldsymbol{x}\} &= \hat{\boldsymbol{f}} \\ \mathrm{Cov}\,[\boldsymbol{x}] &= \hat{\Sigma} \end{cases} \tag{90}$$

So, to generate a sample from the posterior law, we can do the following:

- Generate two random vectors $\boldsymbol{\epsilon}_f \sim \mathcal{N}\left(\boldsymbol{\epsilon}_f|0, v_f I\right)$ and $\boldsymbol{\epsilon}_g \sim \mathcal{N}\left(\boldsymbol{\epsilon}_g|0, v_\epsilon I\right)$;

- Define $\tilde{\boldsymbol{g}} = \boldsymbol{g} + \boldsymbol{\epsilon}_g$ and $\tilde{\boldsymbol{f}} = \boldsymbol{f} + \boldsymbol{\epsilon}_f$ and optimize

$$J\left(\tilde{\boldsymbol{f}}\right) = \frac{1}{2}\|\tilde{\boldsymbol{g}} - \boldsymbol{H}\boldsymbol{f}\|^2 + \lambda\left\|\tilde{\boldsymbol{f}} - \boldsymbol{f}_0\right\|_2^2 \tag{91}$$

- The obtained solution $f^{(n)} = \arg\min_{\tilde{f}}\left\{J\left(\tilde{f}\right)\right\}$ is a sample from the desired posterior law.

By repeating this process for a great number of times, we can use them to obtain good approximations for the posterior mean $\hat{f}$ and the posterior covariance $\hat{\Sigma}$ by computing their empirical mean values. We need however fast and accurate optimization algorithms.

10. References to examples of applications

The above mentioned methods have been used with success in different applications:

- Medical imaging and Computed tomography (CT) [30–36].
- Diffraction tomography and Microwave imaging [37–42].
- 3D Computed Tomography [18, 22, 43]
- Acoustical imaging [44–49]
- Hyperspectral imaging [50]
- Spectrometry [51]
- Eddy current tomography [52]
- Non destructive testing applications [53]
- Emission Tomography [54]
- SAR imaging [55]
- Chronobiological time series [56]

11. Conclusions

Mainly, in this chapter, first we described inverse problems and gave a few classical examples such as deconvolution, image restoration, computed tomography X-ray image reconstruction, Fourier synthesis inversion problem which arise in many imaging systems. Then, we mentioned that there are two classes of methods for inverse problems: deterministic regularization and Bayesian inference methods. Then, we started by describing the Bayesian parameter estimation. The main parts of the chapter is focused on Bayesian inference for inverse problems. We saw that the main difficulty is the great dimension of unknown quantities and the appropriate choice of the prior law. For this, first we described many simple and hierarchical prior models which are used in real applications. For the second main difficulty, which is the computational aspects, we described different approximate Bayesian computations

(ABC) and in particular the variational Bayesian approximation (VBA) methods and showed how to use these methods, for example for hyperparameter estimation or for large scale inverse problems.

BIBLIOGRAPHY

About the author: Ali Mohammad-Djafari is a scientific man, former Research Director in CNRS and Professor of the universities in France and in many international universities. With his more than forty years of scientific research, teaching, and academy-industry cooperation, he has started activities of consulting, technology transfer expertise international cooperation, and training to serve humans and the environment. He has received the B.Sc. in electrical engineering from Polytechnic of Tehran, in 1975, the diploma degree (M.Sc.) from École Supérieure d’Electricité (SUPELEC), Gif-sur-Yvette, France, in 1977, the “Docteur-Ingénieur” (Ph.D.) degree and “Doctorat d’ État” in Physics, from the University of Paris Sud 11 (UPS), Orsay, France, respectively in 1981 and 1987. He supervised more than 22 Ph.D. students and has organized or co-organized more than 10 international workshops and conferences. He has been an expert in a great number of French national and international projects. He has been a member of many scientific societies, e.g., IEEE. He has also participated in and managed many industrial contracts with many French national industries such as EDF, RENAULT, THALES, SAFRAN, and great research institutions such as CEA, INSERM, INRIA as well as the regional (Digiteo), national (ANR), and European projects (ERASYSBIO).

Author details

Ali Mohammad-Djafari[1,2]

1 CNRS, France

2 ISCT, Bures-sur-Yvette, France

*Address all correspondence to: djafari@ieee.org

References

[1] Idier J. Approche bay´esienne pour les probl`emes inverses. Herm`es Science Publications; 2001

[2] Mohammad-Djafari A. Inverse Problems in Vision and 3D Tomography. ISTE-WILEY; 2010

[3] Mohammad-Djafari A. Efficient scalable variational bayesian approximation methods for inverse problems. In: SIAM Uncertainty Quantification UQ16. EPFL; April 2016

[4] Idier J. Bayesian Approach to Inverse Problems. John Wiley & Sons; 2008

[5] Mohammad-Djafari A. Probl`emes inverses en imagerie et en vision en deux volumes ins´eparables. In: Trait´e Signal et Image, IC2. ISTE-WILEY; 2009

[6] Carasso AS. Direct blind deconvolution. SIAM Journal on Applied Mathematics. 2001;**61**(6):1980-2007

[7] Chan T, Wong C-K. Total variation blind deconvolution. IEEE Transactions on Image Processing. 1998;**7**(3):370-375

[8] Kak AC, Slaney M. Principles of Computerized Tomographic Imaging. SIAM; 2001

[9] Jackson JI, Meyer CH, Nishimura DG, Macovski A. Selection of a convolution function for Fourier inversion using gridding [computerized tomography application]. IEEE Transactions on Medical Imaging. 1991;**10**(3):473-478

[10] Chapdelaine C, Mohammad-Djafari A, Gac N, Parra E. A 3D Bayesian computed tomography reconstruction algorithm with Gauss-Markov-Potts prior model and its application to real data. Fundamenta Informaticae. [submitted]

[11] Osher S, Burger M, Goldfarb D, Xu J, Yin W. An iterative regularization method for total variation-based image restoration. Multiscale Modeling & Simulation. 2005;**4**(2):460-489

[12] Wang Y, Yang J, Yin W, Zhang Y. A new alternating minimization algorithm for total variation image reconstruction. SIAM Journal on Imaging Sciences. 2008;**1**(3):248-272

[13] Goldstein T, Osher S. The split Bregman method for L1-regularized problems. SIAM Journal on Imaging Sciences. 2009;**2**(2):323-343

[14] Bertocchi C, Chouzenoux E, Corbineau M-C, Pesquet J-C, Prato M. Deep unfolding of a proximal interior point method for image restoration. Inverse Problems. 2019;**36**

[15] Mohammad-Djafari A. Gauss-Markov-Potts priors for images in computer tomography resulting to joint optimal reconstruction and segmentation. International Journal of Tomography and Statistics (IJTS). 2008; **11**:76-92

[16] Ayasso H, Mohammad-Djafari A. Joint NDT image restoration and segmentation using Gauss–Markov–Potts prior models and variational Bayesian computation. IEEE Transactions on Image Processing. 2010; **19**(9):2265-2277

[17] Feron O, Duchene B, Mohammad-Djafari A. Microwave imaging of inhomogeneous objects made of a finite number of dielectric and conductive materials from experimental data. Inverse Problems. 2005;**21**(6):S95

[18] Wang L, Gac N, Mohammad-Djafari A. Bayesian 3D X-ray computed

tomography image reconstruction with a scaled Gaussian mixture prior model. AIP Conference Proceedings. 2015;**1641**: 556-563

[19] Chapdelaine C, Mohammad-Djafari A, Gac N, Parra E. A joint segmentation and reconstruction algorithm for 3D Bayesian computed tomography using Gauss-Markov-Potts prior model. In: The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017

[20] Chapdelaine C, Gac N, Mohammad-Djafari A, Parra E. New GPU implementation of separable footprint projector and backprojector : First results. In: The 5th International Conference on Image Formation in X-Ray Computed Tomography. 2018

[21] Chapdelaine C. Variational Bayesian approach and Gauss-Markov-Potts prior model. arXiv:1808.09552. 2018

[22] Wang L, Mohammad-Djafari A, Gac N. X-ray computed tomography using a sparsity enforcing prior model based on haar transformation in a Bayesian framework. Fundamenta Informaticae. 2017;**155**(4):449-480

[23] Bioucas-Dias JM, Figueiredo MAT. An iterative algorithm for linear inverse problems with compound regularizers. In: 15th IEEE International Conference on Image Processing, 2008 (ICIP 2008). IEEE; 2008. pp. 685-688

[24] Florea MI, Vorobyov SA. A robust fista-like algorithm. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2017. pp. 4521-4525

[25] Yu S, Wu Z, Xu X, Wohlberg B, Kamilov US. Scalable plug-and-play ADMM with convergence guarantees. IEEE Transactions on Computational Imaging. 2021;**7**:849-863

[26] Chapdelaine C. Reconstruction 3D par rayons X pour le Contrˆole Non Destructif de pi`eces a´eronautique [Th`ese]. Orsay, France: Universit´e de Paris–Sud; 2019

[27] Gilavert C, Moussaoui S, Idier J. Efficient Gaussian sampling for solving large-scale inverse problems using MCMC. IEEE Transactions on Signal Processing. 2015;**63**(1):70-80

[28] Giovannelli J-F. Estimation of the Ising field parameter thanks to the exact partition function. In: ICIP. 2010. pp. 1441-1444

[29] Orieux F, Feron O, Giovannelli J-F. Sampling high dimensional Gaussian distributions for general linear inverse problems. IEEE Signal Processing Letters. 2012;**19**(5):251-254

[30] Mohammad-Djafari A, Demoment G. Maximum entropy image reconstruction in X-ray and diffraction tomography. IEEE Transactions on Medical Imaging. 1988;**7**(4):345-354

[31] Soussen C, Mohammad-Djafari A. Polygonal and polyhedral contour reconstruction in computed tomography. IEEE Transactions on Image Processing. 2004;**13**(11):1507-1523

[32] Wang L, Mohammad-Djafari A, Gac N. Bayesian method with sparsity enforcing prior of dual-tree complex wavelet transform coefficients for X-ray CT image reconstruction. In: 2017 25th European Signal Processing Conference (EUSIPCO). 2017. pp. 478-482

[33] Mohammad-Djafari A. Hierarchical markov modeling for fusion of X ray radiographic data and anatomical data in

computed tomography. In: Proceedings IEEE International Symposium on Biomedical Imaging. July 2002. pp. 401-404

[34] Mohammad-Djafari A, Sauer K, Khagu Y, Cano E. Reconstruction of the shape of a compact object from few projections. In: Proceedings of International Conference on Image Processing. Vol. 1. Oct 1997. pp. 165-168

[35] Wang L, Mohammad-Djafari A, Gac N. X-ray computed tomography simultaneous image reconstruction and contour detection using a hierarchical markovian model. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). March 2017. pp. 6070-6074

[36] Wang L, Mohammad-Djafari A, Gac N, Dumitru M. Computed tomography reconstruction based on a hierarchical model and variational Bayesian method. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; March 2016. pp. 883-887

[37] Carfaatan H, Mohammad-Djafari A, Idier J. A single site update algorithm for nonlinear diffraction tomography. In: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 4. Apr 1997. pp. 2837-2840

[38] Nguyen MK, Mohammad-Djafari A. Bayesian approach with the maximum entropy principle in image reconstruction from microwave scattered field data. IEEE Transactions on Medical Imaging. 1994;**13**(2):254-262

[39] Gharsalli L, Duchˆene B, Mohammad-Djafari A, Ayasso H. Microwave tomography for breast cancer detection within a variational Bayesian approach. In: 21st European Signal Processing Conference (EUSIPCO 2013). Sept 2013. pp. 1-5

[40] Gharsalli L, Duchˆene B, Mohammad-Djafari A, Ayasso H. A gauss markov mixture prior model for a variational bayesian approach to microwave breast imaging. In: 2014 IEEE Conference on Antenna Measurements Applications (CAMA). Nov 2014. pp. 1-4

[41] Gharsalli L, Duchˆene B, Mohammad-Djafari A, Ayasso H. A gradient-like variational Bayesian approach: Application to microwave imaging for breast tumor detection. In: 2014 IEEE International Conference on Image Processing (ICIP). Oct 2014. pp. 1708-1712

[42] Ayasso H, Duchˆene B, Mohammad-Djafari A. A variational Bayesian approach for frequency diverse non-linear microwave imaging. In: 2012 19th IEEE International Conference on Image Processing. Sept 2012. pp. 2069-2072

[43] Gac N, Vabre A, Mohammad-Djafari A, Rabanal A, Buyens F. GPU implementation of a 3D Bayesian CT algorithm and its application on real foam reconstruction. In: The First International Conference on Image Formation in X-Ray Computed Tomography. 2010. pp. 151-155

[44] Chu N, Mohammad-Djafari A, Picheral J. A Bayesian sparse inference approach in near-field wideband aeroacoustic imaging. In: 2012 19th IEEE International Conference on Image Processing. Sept 2012. pp. 2529-2532

[45] Chu N, Mohammad-Djafari A, Gac N, Picheral J. A variational Bayesian approximation approach via a sparsity enforcing prior in acoustic imaging. In: 2014 13th Workshop on Information Optics (WIO). July 2014. pp. 1-4

[46] Chu N, Picheral J, Mohammad-Djafari A. A robust super-resolution approach with sparsity constraint for near-field wideband acoustic imaging. In: 2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). Dec 2011. pp. 310-315

[47] Chu N, Picheral J, Mohammad-Djafari A, Gac N. A robust super-resolution approach with sparsity constraint in acoustic imaging. Applied Acoustics. 2014;**76**(1):197-208

[48] Chu N, Zhao H, Yu L, Huang Q, Ning Y. Fast and high-resolution acoustic beamforming: A convolution accelerated deconvolution implementation. IEEE Transactions on Instrumentation and Measurement. 2020;**99**:1

[49] Chu N, Mohammad-Djafari A, Picheral J. Robust Bayesian superresolution approach via sparsity enforcing a priori for near-field aeroacoustic source imaging. Journal of Sound & Vibration. 2013;**332**(18):4369-4389

[50] Bali N, Mohammad-Djafari A. Bayesian approach with hidden markov modeling and mean field approximation for hyperspectral data analysis. IEEE Transactions on Image Processing. 2008; **17**(2):217-225

[51] Perenon R, Sage E, Mohammad-Djafari A, Duraffourg L, Hentz S, Brenac A, et al. Bayesian inversion of multi-mode NEMS mass spectrometry signal. In: 21st European Signal Processing Conference (EUSIPCO 2013). Sept 2013. pp. 1-5

[52] Premel D, Mohammad-Djafari A. Eddy current tomography in cylindrical geometry. IEEE Transactions on Magnetics. 1995;**31**(3):2000-2003

[53] Mohammad-Djafari A, Robillard L. Hierarchical markovian models for 3D computed tomography in non destructive testing applications. In: 2006 14th European Signal Processing Conference. Sept 2006. pp. 1-5

[54] Fall MD, Barat E, Comtat C, Dautremer T, Montagu T, Mohammad-Djafari A. A discrete-continuous bayesian model for emission tomography. In: 2011 18th IEEE International Conference on Image Processing. Sept 2011. pp. 1373-1376

[55] Zhu S, You P, Wang H, Li X, Mohammad-Djafari A. Recognition-oriented Bayesian sar imaging. In: 2011 3rd International Asia-Pacific Conference on Synthetic Aperture Radar (APSAR). Sept 2011. pp. 1-4

[56] Dumitru M, Mohammad-Djafari A. Periodic components estimation in chronobiological time series via a bayesian approach. In: 2015 23rd European Signal Processing Conference (EUSIPCO). Aug 2015. pp. 2246-2250