An overview of Bayesian parameter estimation and inverse problems

Andrew A. Ganse, Applied Physics Laboratory, University of Washington

Lecture notes, ESS-523, 2015

What we'll cover today

This lecture is an overview, so just warning from beginning that a lot of details will be left out! Hopefully I'll provide enough detail and resources for you to learn more on your own though.

- Underlying mathematical philosophy & formulation
  - Frequentist vs. Bayesian concepts and philosophies
  - Comparing Maximum likelihood vs. Conditional probability vs. Bayes' Rule
- Linear Bayesian problems
- Nonlinear Bayesian problems using linearization (infinitesimally brief!)
- Tracking problems using predictive filters and smoothers (infinitesimally brief!)
- Linear or nonlinear Bayesian problems using importance sampling
  - Show-n-tell of a Bayesian sampling version of the source location problem
  - Inversion vs. parameter estimation: trans-dimensional Bayesian sampling

Underlying mathematical philosophy & formulation

Frequentist vs. Bayesian concepts and philosophies

- Two different paradigms of probability theory – one half of statistics community (frequentists) defines probability in terms of **frequency of events**, the other half (Bayesians) in terms of **degree of belief**. This leads to two separate frameworks for doing parameter estimation and inverse problems.
- Key point is this distinction means frequentists believe probabilities only have meaning **after** experiments, i.e. after events tallied up. Bayesians believe probabilities have meaning **prior** to experiments too by different information that one has in hand, which should not be ignored.
- The frequentist framework estimates deterministic but unknown parameters, while the Bayesian framework estimates random variables that have probabilities both before and after the experiment.
- A few key geophysical inversion people to note on both sides (re seminal books/papers):
  Frequentists
  - Bob Parker
  - Aster, Borchers, Thurber
  - Constable

  Bayesians
  - Albert Tarantola
  - Klaus Mosegaard
  - Stan Dosso, Jan Dettmer
  - Macolm Sambridge

Comparing Maximum likelihood vs. Conditional probability vs. Bayes' Rule

(Note there's a common notational dual-use problem with the "|" operator that causes confusion in comparing these ideas. We'll clarify this below...) Clarify that both frequentist and Bayesian formulations consider the data as a random variable due to noise/errors, but frequentists consider the model vector a deterministic parameter while the Bayesians consider it another random variable.

Maximum likelihood estimation

Here, the goal is point-estimation of the deterministic but unknown model parameter vector $\mathbf{m}$ which produces the best fit in some sense to the observed data. The model vector is not a random variable; it's a parameter, not a probability. For independent probability density functions f_i for each observation given some model parameter vector $\mathbf{m}$, their joint probability density is:

$$f(\mathbf{d}|\mathbf{m}) = f_1(d_1|\mathbf{m}) \cdot f_2(d_2|\mathbf{m}) \cdots f_n(d_n|\mathbf{m}) \tag{1}$$

An important note here is that this $f(\mathbf{d}|\mathbf{m})$ is not a conditional probability. For conditional probability, both $\mathbf{d}$ and $\mathbf{m}$ have to be random variables. Here the observations vector $\mathbf{d}$ is a random variable with a probability, but $\mathbf{m}$ is not; it's a deterministic (although unknown) vector of parameters.

We define the likelihood function like this:

$$L(\mathbf{m}|\mathbf{d}) = f(\mathbf{d}|\mathbf{m}) \tag{2}$$

We say the likelihood of these model parameters $\mathbf{m}$ given the observational random variables $\mathbf{d}$ (because the data have errors on them) is equal to the probability of the observational random variables $\mathbf{d}$ given the model parameters $\mathbf{m}$.

To find the model most likely to have produced the data, maximize L. That's the maximum likelihood principle. For special case of linear problem and independent, normally-distributed data errors, maximizing the likelihood looks like the following. We formulate the data with this form:

$$f_i(d_i|\mathbf{m}) = \frac{1}{(2\pi)^{1/2}\sigma_i} \exp\left[-(d_i - (\mathbf{Gm})_i)^2/2\sigma_i^2\right] \tag{3}$$

The likelihood is:

$$L(\mathbf{m}|\mathbf{d}) = \prod_{i=1}^{n} f_i(d_i|\mathbf{m}) = \frac{1}{C} \prod_{i=1}^{n} \exp\left[-(d_i - (\mathbf{Gm})_i)^2/2\sigma_i^2\right] \tag{4}$$

Finding the maximum likelihood $\max L(\mathbf{m}|\mathbf{d})$ ignores that $\frac{1}{C}$, and for that exponential is equivalent to finding the maximum of the exponent. The product of the exponentials causes a sum of the exponents, and the negative sign in the exponent turns that max into a min, and we have:

$$\max L(\mathbf{m}|\mathbf{d}) = \min \sum_{i=1}^{n} \frac{(d_i - (\mathbf{Gm})_i)^2}{\sigma_i^2} \tag{5}$$

Or an equivalent vector-notation form of this we're used to seeing is:

$$\min \quad (\mathbf{d} - \mathbf{Gm})^T \mathbf{C}_\mathrm{d}^{-1} (\mathbf{d} - \mathbf{Gm}) \tag{6}$$

where $\mathbf{C}_\mathrm{d}$ is the covariance matrix of the data noise, here equal to $\sigma^2\mathbf{I}$. That's least-squares estimation. Remember there are other forms of maximum likelihood estimation, like what if your noise isn't normally distributed, your data aren't independent, or you're using L_1 norms instead of the L_2 norms we used here, and so on. There are tools for those, but we won't get into those here.

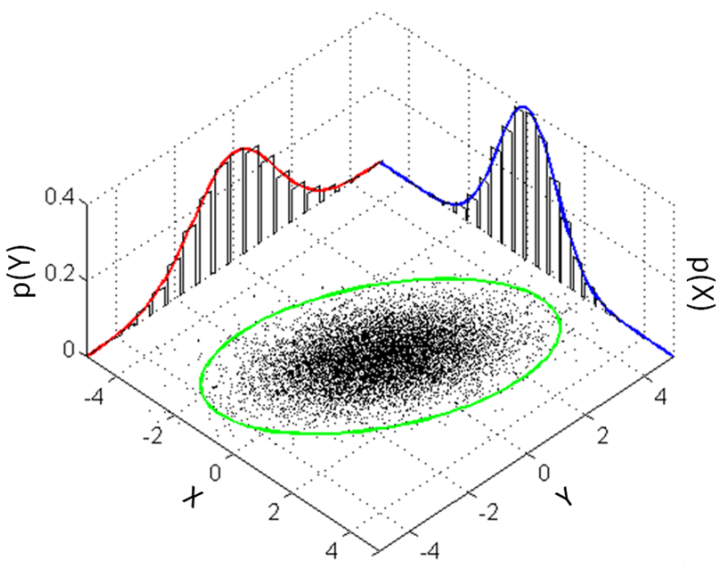


Figure 1: Marginal distributions $P(X)$ and $P(Y)$ are the integrals of the 2D $P(X,Y)$ in the respective directions. This notion is expressed in the "law of total probability" in Equation 8. (Public domain image from Wikipedia's marginal distributions webpage.)

Conditional probability

The above is the framework you learned and focused on all quarter in various forms. But now here's where we shift philosophical paradigms. Now instead of expressing the model vector as a deterministic but unknown parameter, we'll instead express it as a random variable, so it has a probability associated with it. So now both the data and the model vectors will be random variables, unlike the above. This is the underpinning distinction between frequentist (the previous) and Bayesian (this new type) estimation.

Why do this? There are in fact many decades of researchers arguing endlessly about which of those two approaches is more valid, with many in each camp insisting one should always use that approach in all problems, and that use of prior probability/information is either always or never justifiable. But in my opinion it depends on your problem and what data and other information you have available to you. I think if you have additional information it's ideal to set up your problem to use it, and if you don't you shouldn't. And I think it's helpful to be somewhat familiar with both so you can understand other researchers' work in addition to your own.

So – now we're talking about the model parameters as random variables as well as the observations.

We define conditional probability of A given that B has occurred by:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{7}$$

There's a formula called "the law of total probability" that relates conditional probabilities (like $P(A|B)$) and marginal probabilities (like $P(B)$). Marginal probabilities are projections of the conditional probability into a given variable:

$$P(A) = \sum_i P(A \cap B_i) = \sum_i P(A|B)P(B_i) \tag{8}$$

You may have seen marginal probabilities before in the context of when you have say a 2D joint probability density of two variables X and Y, and the marginal distributions of X and Y are the integrals of the 2D distribution in that dimension, as portrayed in Fig. 1.

Bayes' Rule

Bayesian inverse problems consider both the model and the data observations as random variables with PDFs, not as deterministic parameters, so for Bayesian inverse problems we need to have a way to reverse the order of the conditioning in the conditional probabilities. We're interested in the probability density of the model vector $\mathbf{m}$ conditioned on measurements of random variable $\mathbf{d}$, i.e. we want $P(\mathbf{m}|\mathbf{d})$ rather than the likelihood $P(\mathbf{d}|\mathbf{m})$ listed above. This is provided by Bayes' Rule (or Bayes' Theorem):

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{9}$$

Or restated in terms of the model and data vectors:

$$P(\mathbf{m}|\mathbf{d}) = \frac{P(\mathbf{d}|\mathbf{m})P(\mathbf{m})}{P(\mathbf{d})} \tag{10}$$

The $P(\mathbf{d}|\mathbf{m})$ is the "model likelihood", which is the data misfit. $P(\mathbf{m})$ is the "prior probability" of the model vector; this is the regularization. $P(\mathbf{m}|\mathbf{d})$ is the "posterior probability" of the model vector, and $P(\mathbf{d})$ is called the "evidence" or "marginal probability of the measurements". The prior contains previously known information (including uncertainly known) about the model vector, perhaps via previous experimental results, results in a nearby area, or perhaps via supplemental measurements. The evidence is ultimately just a constant; its value using the Equation 8 again is:

$$P(\mathbf{d}) = \sum_i P(\mathbf{d}|\mathbf{m})P(m_i) \tag{11}$$

We'll use these concepts and these definitions and labels below as we review common solution techniques.

Linear Bayesian problems

For the sake of time today we'll need to forego the derivation and skip quickly to the bottom line. If you start from Bayes' Rule:

$$P(\mathbf{m}|\mathbf{d}) = \frac{P(\mathbf{d}|\mathbf{m})P(\mathbf{m})}{P(\mathbf{d})} \tag{12}$$

and you fill in each of those PDFs with Gaussian expressions and define the likelihood (misfit) the same as in the frequentist L_2 case and crank the algebra, you'll find that the expressions for the mean and covariance of the posterior model PDF are almost but not quite the same as for the frequentist case you've learned this quarter. The likelihood $P(\mathbf{m}|\mathbf{d})$ is the same expression as back in Equation 4 (except now the C matters so you need to include that in the algebra). Defining the data PDF as having mean $\mathbf{d}_{\text{mean}}$ and covariance $\mathbf{C}_\text{d} = \sigma^2\mathbf{I}$...

Bayesian posterior model mean $\mathbf{m}_{\text{post}}$ and posterior model covariance $\mathbf{C}_{\text{post}}$:

Defining the model prior PDF as having mean $\mathbf{m}_{\text{prior}}$ and covariance $\mathbf{C}_{\text{prior}}$, we end up with:

$$\mathbf{m}_{\text{post}} = (\mathbf{G}^T\mathbf{C}_\text{d}^{-1}\mathbf{G} + \mathbf{C}_{\text{prior}}^{-1})^{-1}(\mathbf{G}^T\mathbf{C}_\text{d}^{-1}\mathbf{d}_{\text{mean}} + \mathbf{C}_{\text{prior}}^{-1}\mathbf{m}_{\text{ prior}}) \tag{13}$$

$$\mathbf{C}_{\text{post}} = (\mathbf{G}^T\mathbf{C}_\text{d}^{-1}\mathbf{G} + \mathbf{C}_{\text{prior}}^{-1})^{-1} \tag{14}$$

Frequentist model solution estimate $\widehat{\mathbf{m}}$ and model covariance $\mathbf{C}_{\widehat{m}}$:

Defining the Tikhonov regularization as a finite-difference operator $\mathbf{L}$ with tradeoff parameter α and preferred model $\mathbf{m}_\text{o}$ (which recall can be used in iterative solutions to the nonlinear problem as well):

$$\widehat{\mathbf{m}} = (\mathbf{G}^T\mathbf{C}_\text{d}^{-1}\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})^{-1}(\mathbf{G}^T\mathbf{C}_\text{d}^{-1}\mathbf{d}_\text{mean} + \alpha^2\mathbf{L}^T\mathbf{L}\mathbf{m}_\text{o}) \tag{15}$$

$$\mathbf{C}_{\widehat{m}} = (\mathbf{G}^T\mathbf{C}_\text{d}^{-1}\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})^{-1}\mathbf{G}^T\mathbf{G}(\mathbf{G}^T\mathbf{C}_\text{d}^{-1}\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})^{-T} \tag{16}$$

$$= \mathbf{G}^\#\mathbf{C}_\text{d}\mathbf{G}^{\#T} \text{ for Tikhonov-forumulated generalized inverse } \mathbf{G}^\#. \tag{17}$$

Looking back and forth between the two cases, we see a lot of parallels but some important differences. The expressions for $\mathbf{m}_\text{post}$ and $\widehat{\mathbf{m}}$ in Equations 13 and 15 are virtually the same, with the prior covariance matrix $\mathbf{C}_\text{prior}$ equaling the regularization matrix $\alpha^2\mathbf{L}^T\mathbf{L}$. But the frequentist solution is recalculated many times for many values of α sweeping out an L-curve (or there are other methods for choosing an optimal α which also recalculate the problem many times). In contrast that Bayesian solution is only run once – rather than analyzing the statistics to choose a level of smoothness as in the frequentist case, the Bayesian problem determines the level of smoothness from the start using prior information you have.

In fact, the whole idea of "level of smoothness" doesn't fit in the Bayesian framework, because it implies you're smoothing the problem, which you do in the frequentist case but you don't in the Bayesian case. In the frequentist case you don't have enough information to stably solve the problem so you must change the problem and solve for a different quantity than the original model vector $\mathbf{m}$; you solve for some smoothed version $\mathbf{m}_\text{estimated}$, where I use the term "smoothed" loosely for really it can be defined by any linear operation on the original "true" model vector $\mathbf{m}_\text{true}$. As you've learned already, the linear operator is the model resolution matrix $\mathbf{R}$:

$$\mathbf{m}_\text{estimated} = \mathbf{R}\mathbf{m}_\text{true} \tag{18}$$

In the frequentist case, the covariance matrix $\mathbf{C}_{\widehat{m}}$ is the covariance of the smooth estimated model vector, not of the original true model vector. By contrast in the Bayesian case the covariance $\mathbf{C}_\text{post}$ of the posterior model vector is an actual estimate of the covariance of the true model vector. So in the Bayesian case there is no model resolution matrix defining the "level of smoothness" (again loosely speaking there). The resolution of the Bayesian posterior model PDF is inherent in the correlations in its covariance matrix (the off-diagonal elements of $\mathbf{C}_\text{post}$). As long as those correlations are included in there you can still over-parameterize the model just as before, and let the combination of the data and the prior model information determine the structure.

If we interpret the frequentist resolution matrix in terms of "amount of information resolved by the data" (for recall in the frequentist problem the level of smoothness is determined "automatically" by the statistics of the data), we can show a direct analog of a Bayesian resolution matrix in that same sense. If we consider a set of "true" data, i.e. noiseless data that is exactly producable by the forward problem as $\mathbf{d}_\text{true} = \mathbf{G}\mathbf{m}_\text{true}$, the posterior mean model $\mathbf{m}_\text{post}$ and prior mean model $\mathbf{m}_\text{prior}$ satisfy:

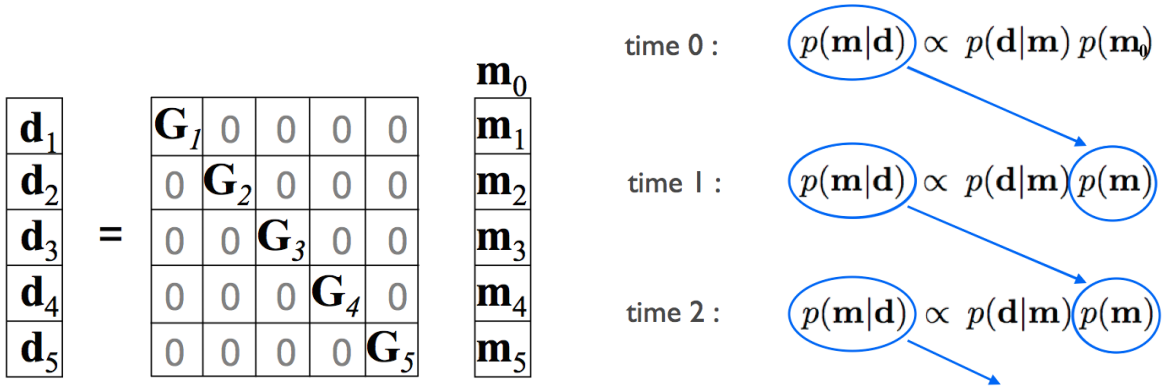
$$\mathbf{m}_\text{post} - \mathbf{m}_\text{prior} = \mathbf{R}(\mathbf{m}_\text{true} - \mathbf{m}_\text{prior}) \tag{19}$$

where

$$\mathbf{R} = \mathbf{I} - \mathbf{C}_\text{post}\mathbf{C}_\text{prior}^{-1} \tag{20}$$

This is interpreted as saying that the solved-for corrections $\mathbf{m}_\text{post} - \mathbf{m}_\text{prior}$ to the model prior are a smoothed version of the exact deviations $\mathbf{m}_\text{true} - \mathbf{m}_\text{prior}$ of the prior model from the true model. If you take the trace of Equation 20, it can be interpreted as saying the number of parameters resolved by the dataset is equal to the total number of model parameters minus the number of parameters resolved by the prior model. Actually the same interpretation can be made of the frequentist resolution matrix (substituting "prior model" with "regularization"); a proof and discussion of that can be found in my PhD thesis, while the rest of this Bayesian resolution material is from Tarantola's book (see reading list).

So overall, we gain a level of simplicity with the Bayesian problem (no L-curve or such α-choosing methods, no confusion over uncertainties from juggling both covariance and resolution matrices), but it's only appropriate



(a) (b)

Figure 2: Tracking problems are a class of Bayesian estimation problem that have sparseness due to time independence of measurements and of dynamics in the regularization (at the time-scale of the experiment). This sparseness is easily depicted in a linear problem (a) where the $\mathbf{m}_i$ represent a set of parameters (maybe an object's position and velocity) that update with each new measurement $\mathbf{d}_i$. The prior model is $\mathbf{m}_0$ and from there on out the model parameters are based on measurements and the dynamics linking only neighboring timepoints. Tracking filters and smoothers take advantage of "conjugate priors", which match the form of the previous posterior (b), to derive efficient expressions that update moments of those priors and posteriors over time.

when we've got the additional information that we can stuff into the model prior. Sometimes you've got it and sometimes you don't. And in this linear formulation where you take matrix inverses, you need the prior to have enough information in it to regularize the problem; in this formulation you can't get away with a super simple prior of 1's and 0's that just bounds the variables... as you can in the sampling approach later.

Nonlinear Bayesian problems using linearization

I'll spend almost no time at all on this section, because the real interest is in the sampling section. But I'll take just enough time to note that just like in the frequentist case, we can use those linear Bayesian expressions in an iterative sense to solve weakly nonlinear problems. It runs a lot faster because you're only doing the problem once, not for many values of some tradeoff parameter α as in the frequentist case. And this iterative linearization certainly runs faster than the generalized Bayesian sampling approach. If you have a weakly nonlinear problem for which you have additional or previous model information that can be put in terms of probabilities, it may be worth using this linearization approach. The full formulation is in Tarantola's book (see reading list), but it's virtually the same layout as the iterative method you learned for the frequestist case, in terms of the parallels shown above. Just remember that when linearizing the problem once more at the solution to estimate the uncertainties, you use the Bayesian covariance matrix not the frequentist one.

Tracking problems using predictive filters and smoothers

This section too will be really brief, just enough to introduce the idea. Standard mathematical tools in tracking problems use Bayesian probability estimation too. Tracking problems are just parameter estimation or inverse problems in the context of repeated measurements over time, and a trajectory or other time-varying quantity that you want to estimate over time. For example in my ocean acoustics field we often need to acoustically track our long vertically-moored cable of hydrophones since it sways around in the tides and currents – repeated pings to and from four acoustic transponders placed around the base of the mooring are

used to estimate cable positions in space over time. Tracking problems also typically have some additional regularization in the form of physical dynamics governing the trajectory, linking one time point to the next.

One *could* just treat the whole estimation using exactly the techniques above and that you've learned this quarter already. For linear problems that means one big matrix covering all times at once. But that's unnecessarily inefficient because the measurements are time-independent and the dynamics are only specified in a way linking neighboring time points ("Markov process"); the big matrix is really sparse, see Fig. 2a. A "predictive smoother" such as the "Kalman smoother" (for linear problems; similar tools exist for nonlinear problems) takes advantage of that sparseness and solves the exact same Bayesian problem much more efficiently. Just like the original big non-sparse problem, you need all the measurements at once for this, solving the tracking after the experiment. But in applications where you don't have the time for that or need real-time updates – for example in more engineering-style applications where say you've a radar to track fast spacecraft – a related tool called a "predictive filter" (such as a "Kalman filter" in linear problems) does a similar estimation but less accurately because it only uses measurements up to the present time, not for all times. In fact it turns out that smoothers can be expressed in terms of a pair of filters, one running forward in time and one running backward.

The trick to these filters and smoothers is that they link a series of Bayesian estimation problems of the same form together. If the posterior model PDF from one time point has the same form as the prior model PDF for the next time point (Gaussian, say), see Fig. 2b, you can write expressions that iterate and update just the moments of those posterior and prior PDFs. When the form of the prior PDF matches that of the posterior, they call it a "conjugate prior" – it doesn't have to be Gaussian but that's the most common. The dynamics regularization is a key component of these tracking problems, and the filters and smoothers are classified by whether the data/model relationship is linear vs. nonlinear, and whether the dynamics linking state at one time and the next is linear vs. nonlinear. The well-known "Kalman filter" and "Kalman smoother" are for everything being linear.

Two classic, very readable texts on tracking filters and smoothers are Gelb and Jazwinski (see reading list). And for discussion and demonstration code applying many common filters and smoothers to a textbook example radar tracking problem in Gelb, see my webapge on that (also in reading list).

Linear or nonlinear Bayesian problems using importance sampling

Now for a big shift, although still within the Bayesian framework. Let's zoom all the way back out to Bayes' Rule:

$$P(\mathbf{m}|\mathbf{d}) = \frac{P(\mathbf{d}|\mathbf{m})P(\mathbf{m})}{P(\mathbf{d})} \tag{21}$$

In all the above discussion of the linear and linearized methods, as well as all the methods you've been learning in class this quarter, there's been an assumption that the data probabilities are expressed as normally distributed, that the model uncertainties can be expressed as at least approximately normally distributed, that the data misfit (the likelihood function) is defined as an L_2 norm of the residuals, and that the regularization or prior had a specific matrix form. That's all by way of conveniences for setting up linearized and maximum likelihood methods, but Bayes' Rule doesn't actually require any of that.

It turns out that if you're willing to spend the computer time for it (which admittedly is a big "if"), there's a very useful and nowadays very widespread numerical approach to solving Bayesian inverse problems via Bayes' Rule, making none of those assumptions. You can specify the probabilities in any form you want, the likelihood function can be any form you wish, and you can specify as much or as little prior information as you want because you don't have to worry about stability of any matrix inverses here (the non-uniqueness effects however still exist and will show up in the results). I'll discuss and demonstrate the technique in a parameter estimation problem of just two parameters, but inverse problems are just as applicable – the model prior does the regularization in either case. Balancing the model parameterization with the practical constraint of computer time limits (each model parameter has its own dimension in the sampling space, so

you can't have a zillion of them) in a way that doesn't artificially impose structure on the model is handled rigorously via techniques mentioned at the end. But first the general technique.

There are several versions of this technique, all within the category of **Markov Chain Monte Carlo (MCMC) sampling** – two that I come across most are the Metropolis-Hastings method (see e.g. Mosegaard paper) and the Fast Gibbs Sampling method (see Dosso paper) – I'll describe Metropolis-Hastings but similar concepts are in the other methods even if details are slightly different. Additional details about MCMC implementation, such as more statistical rigor for how many samples needed for a certain accuracy level of the solution confidence region, are found in the Gilks *et al.* MCMC book. Here's **Metropolis-Hastings**.

The key is that while the denominator of Bayes' Rule is enormously expensive to calculate (integrating all the data over all the model parameters), it is ultimately just a constant and is not necessary. So we sample just the numerator $N(\mathbf{m})$, to obtain a sample set of values that are *proportional* to the posterior probability.

$$N(\mathbf{m}) = P(\mathbf{d}|\mathbf{m})P(\mathbf{m}) \propto P(\mathbf{m}|\mathbf{d}) \tag{22}$$

Note I'm not writing the "$|\mathbf{d}$" in $N(\mathbf{m})$ because it's not strictly a probability (although it's proportional to it). In any case $N(\mathbf{m})$ is used to guide the sampling of the probability space, and then then the *density* (in model space) of those samples is what provides the information when computing the moments of the posterior model PDF at the end.

The process starts at a starting model $\mathbf{m}_o$, and from there on a "modified" random walk (by definition as a Markov process) will decide the choice of next step $\mathbf{m}_{i+1}$ based only on the present step $\mathbf{m}_i$. A **proposal density** $Q(\mathbf{m}_{i+1}|\mathbf{m}_i)$ (often simply an isotropic Gaussian encoding a probabilistic step-size) is used to determine a candidate for the next step. With the candidate for $\mathbf{m}_{i+1}$ chosen, we calculate the **acceptance ratio** α:

$$\alpha = N(\mathbf{m}_{i+1})/N(\mathbf{m}) \tag{23}$$

Since $N(\mathbf{m}) \propto P(\mathbf{m})$, a key operating fact is that the acceptance ratio calculated above is also equal to:

$$\alpha = P(\mathbf{m}_{i+1}|\mathbf{d})/P(\mathbf{m}|\mathbf{d}) \tag{24}$$

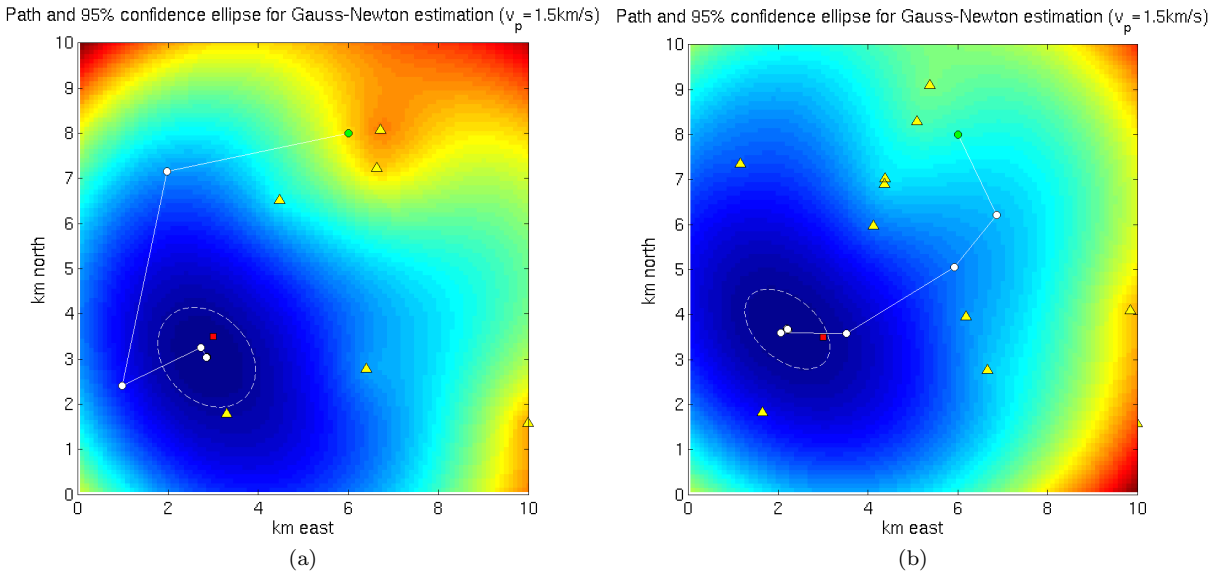
Not all such candidates for $\mathbf{m}_{i+1}$ will be followed (or "accepted"). If the acceptance ratio $\alpha \geq 1$, then that candidate $\mathbf{m}_{i+1}$ is accepted. If $\alpha < 1$, then the candidate $\mathbf{m}_{i+1}$ is only accepted with probability α. If rejected then $\mathbf{m}_i$ remains unchanged for this iteration.

That's it. The power is that this simple process samples the $N(\mathbf{m})$ space with the same density that it would sample $P(\mathbf{m}|\mathbf{d})$. Two remaining details to handle are that 1.) there is a comparitively short "burn-in" period between the starting model $\mathbf{m}_o$ and when the acceptance process first takes hold – these initial burn-in samples are thrown away, and 2.) the samples are not statistically independent since they're consecutive steps in a path. We need to subsample them to obtain independent samples (taking e.g. every Nth sample as determined by the $1/e$ value of the sample autocorrelation) before calculating moments of the posterior model PDF.

Now let's look at a simple example that extends Labs 4 and 5 on source location and objective surfaces. See two examples of results in Labs 4 and 5 in Fig. 3 to refresh your memory on those. Then see Fig. 4 which shows a twist on those labs, where the receivers are bunched together with noise in the travel times of the same order as the travel time between the receivers. This causes the probability region for the source location to have a strong arc since the receiver array can resolve the source range but cannot well resolve the source bearing. Fig. 4a compares the 95% confidence ellipse calculated using linear methods learned this quarter to the actual posterior probability region sampled by MCMC. Fig. 4b compares the five solution iterations in the Gauss-Newton process learned this quarter to the ten thousand give iterations in the MCMC process – they both lead to virtually the same solution point; it's the representation of the solution uncertainty that's the difference. Note there are hybrid methods out there which are appropriate for some problems more than others.

I didn't deal with computing moments from the samples here, but Fig. 5 shows the first 750 samples (of the more than ten thousand, which may not even be enough depending on your desired accuracy of confidence interval). We see the burn-in period, perhaps about 75 samples (where $N(\mathbf{m}_i)$ is so low because the only

samples accepted yet were accepted by random chance when α was already < 1, not because the acceptance ratio had increased), but I was arbitrarily chopping it at 100th sample just to be safe. Comparing back to Fig. 4b we can see those first samples are in a very low probability region indeed. If we re-ran the MCMC process the burn-in period could even be longer; it's all up to the random walk.



(a) (b)

Figure 3: Two example solutions from Labs 4 and 5 on source location and objective functions, where just two parameters are estimated, the east and north coordinates of source location. For a constant velocity field, noisy travel time measurements from a source (red dot) to a set of recording stations (yellow triangles) are used to estimate the source location and its 95% confidence ellipse by iterating from a starting guess (green dot). Five to eight iterations were typical in this problem. When the stations are spread around enough (as here), the objective surface is close to parabolic in the region of the confidence ellipse so it's a reasonable approximation to the uncertainty in the estimated location.

Inversion vs. parameter estimation: trans-dimensional Bayesian sampling

Just lastly, a key issue remains regarding the difference between estimating a few parameters vs. a continuous function, i.e. parameter estimation vs. inversion. A big point may have been made this quarter about trying not to artificially constrain your inverse problem's estimation of a continuous function by just using a few parameters. A common traditional approach is to over-parameterize the model in order to let the data and regularization/prior determine the structure. The only problem with overparameterization in Bayesian sampling is that the already-significant computation time increases exponentially with every additional model parameter. So it's preferable to have a scheme that minimizes the number of model parameters, without minimizing so far as to affect the model structure. In other words the wish is to let the data and regularization/prior decide not only the model structure, but also the parameterization itself. A very efficient method to do this in the sampling paradigm is called **transdimensional Bayesian sampling**. This is a type of **hierarchical Bayesian modeling** that sets up the problem to estimate the number k of parameters as one of the parameters, like this:

$$P(k, \mathbf{m}_k|\mathbf{d}) \propto P(\mathbf{d}|k, \mathbf{m}_k)P(\mathbf{m}_k|k)P(k) \tag{25}$$

Everything else is basically the same as above, except there's additional sampling machinery to add in and subtract out parameters in the appropriate places (using for example Voronoi cells) as the random walk steps around in the number-of-parameters dimension along with the rest. Two of the papers in the reading list nicely lay out all these details.

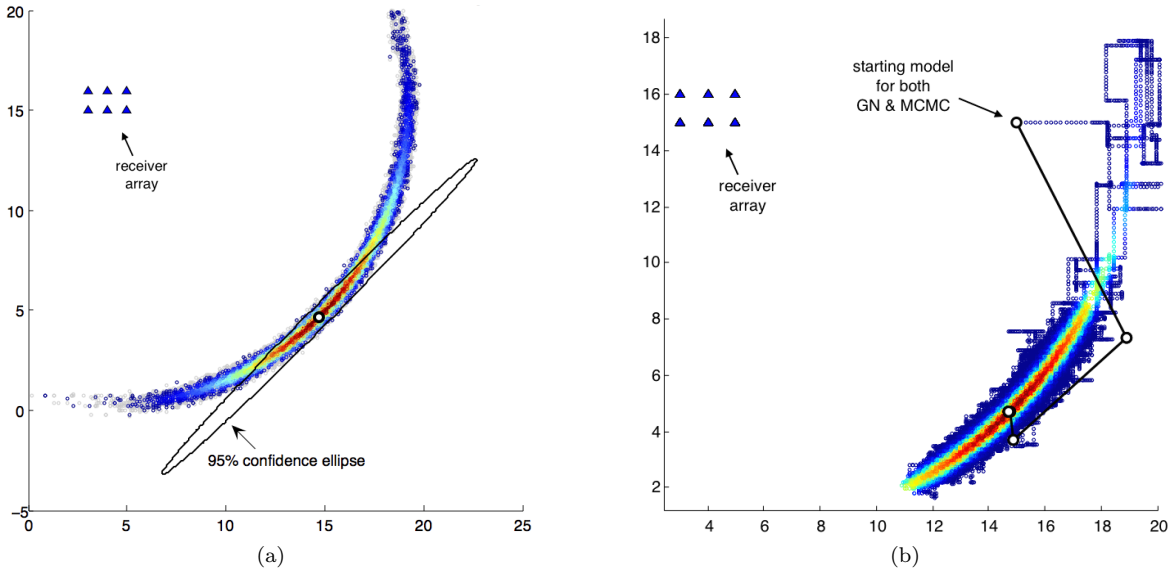


Figure 4: A twist on the source location problem of Fig. 3, from a problem at the end of Tarantola's book. Now the stations are bunched close together and far from the source, and the noise is now the same order of magnitude as the travel times between the stations. So the array can resolve the range to the source but not its bearing very well. This causes the objective function to have a strongly arced shape (a), and the 95% confidence ellipse from the linearized method is no longer a good approximation to the actual 95% confidence region (although note the solution location is still estimated well with linearized methods in this problem). The arc is well resolved by the MCMC sampling technique described in this section. The tradeoff is computation time and number of samples; in (b) both linearized and MCMC methods start from same starting guess of the source location, but while the linearized method took five iterations, the MCMC takes over ten thousand to fully characterize the arc, which the linearized method can't do.

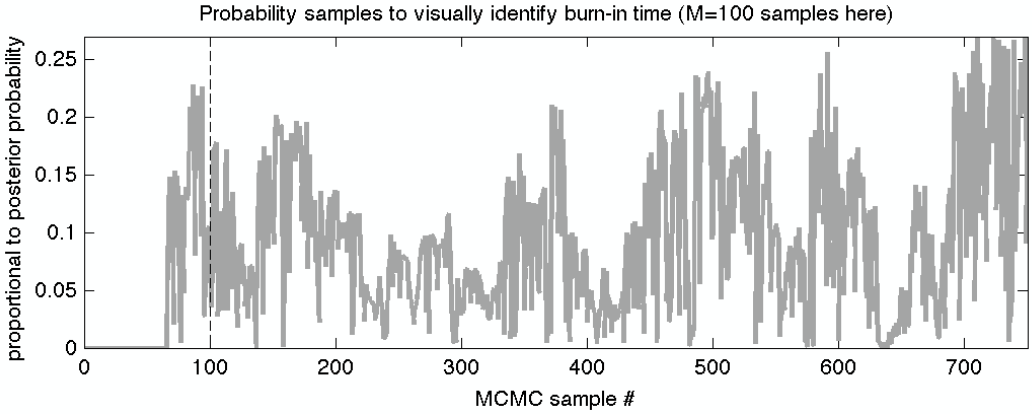


Figure 5: The values proportional to posterior probability that were encountered by the MCMC random walk as a function of iteration number (only first 750 iterations shown here – there were over ten thousand iterations). The model parameter values associated with the points in this series can be used to estimate moments of the model posterior PDF in a given local minimum (in Fig. 4 there was only one minimum). There was a "burn-in" period at the beginning between the starting model and when the random walk finds the region of the minimum and starts filling it out; this burn-in period is removed from the samples used to estimate the moments. Also the ensuing probability samples are not independent since they're consecutive steps in a path. Moment estimation from the samples also requires subsampling them by an amount that obtains independent samples, determined by e.g. autocorrelation.

Brief reading list

A few books related to Bayesian inversion

- Albert Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, 2005. (available free online! http://www.ipgp.jussieu.fr/~tarantola/Files/Professional/Books)
- W.R. Gilks, S. Richardson, & D.J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, 1996.
- Andrew Jazwinski, *Stochastic Processes and Filtering Theory*, 2007.
- The Analytic Sciences Corp., Arthur Gelb ed., *Applied Optimal Estimation*, 1974.

A few recommended papers related to Bayesian inversion

- *Metropolis-Hastings MCMC implementation:*

  Klaus Mosegaard and Albert Tarantola, "Monte Carlo sampling of solutions to inverse problems," *Journal of Geophysical Research*, Vol. 100, No., B7, p.12,43112,447, 1995.

- *Trans-dimensional inversion:*

  Jan Dettmer, Stan Dosso, and Charles Holland, "Trans-dimensional geoacoustic inversion," *J. Acoust. Soc. Am.* 128 (6), December 2010.

  Anandaroop Ray, Kerry Key, Thomas Bodin, David Myer, and Steven Constable, "Bayesian inversion of marine CSEM data from the Scarborough gas field using a transdimensional 2-D parametrization," *Geophys. J. Int.* (2014) 199, 18471860.

Useful websites

- Note there are actually pretty well-written Wikipedia pages on many of the topics in this lecture...
- My geophysical inverse theory resources webpage (http://staff.washington.edu/aganse/invresources)
- My tracking filters/smoothers tutorial + codes (http://staff.washington.edu/aganse/gelbexample)
- Brad Bell's tracking filters/smoothers (and automatic differentiation too): Matlab, Fortran, C++ code (http://www.seanet.com/~bradbell/packages.htm)