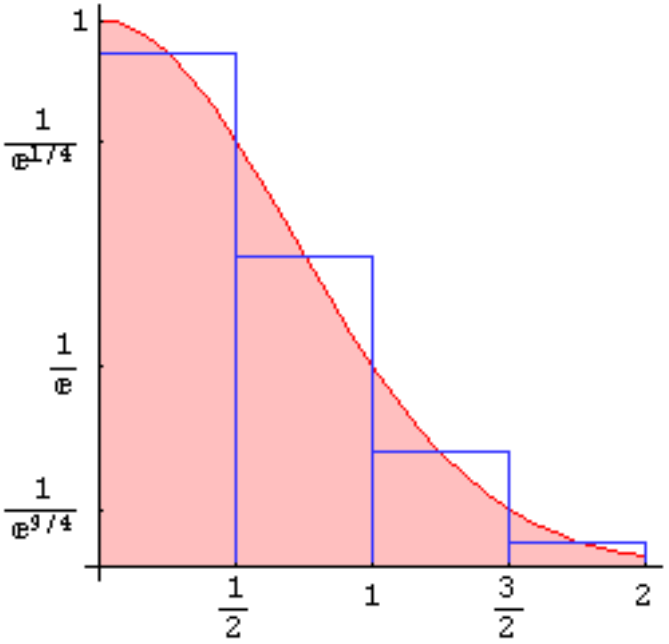
Physics 509: Numerical methods for Bayesian analyses

Scott Oser
Lecture #15



Review of Bayesian problem-solving

Bayesian analysis offers a conceptually simple and easy to remember solution to most statistics problems:

$$P(H|D,I)=\frac{P(H|I)P(D|H,I)}{P(D|I)}$$

Every problem reduces to an application of Bayes' theorem.

You can spend less time thinking about the problem, and more time actually solving it.

Nuisance parameters

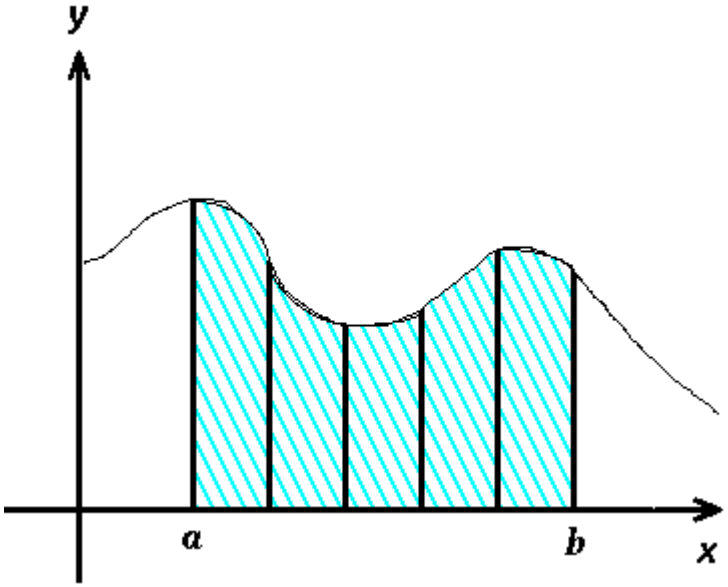
But those damned nuisance parameters are, well, a nuisance. Suppose that there is a set of free parameters α in the model. If what you care about is the probability of the model H_i being true, regardless of the value of the parameters, you have to marginalize by integrating over the unwanted parameters:

$$P(H_i|D,I)=\frac{\int d\alpha\, P(H_i,\alpha|I)\,P(D|H,\alpha,I)}{P(D|I)}$$

For every parameter you don't care about, you generate an integral. These can add up! Very often, the integrals cannot be done analytically, and you have to resort to sometimes messy numerical techniques.

Numerical Integration by Simpson's rule

You've probably had to do integrals numerically before. A very common approach is to divide the interval of integration into N (~100) equally spaced bins. You then evaluate the function at the center of each bin, sum up the result, and multiply by $\Delta x=(b-a)/N$.



This is OK, and maybe even the best solution, if you have to integrate over one or two variables. But number of function evaluations goes like $(100)^m$, which is heinous when the dimensionality m gets large.

Linear Gaussian models

There is a very special case in which the solution is easy. Suppose that your data is described by a linear function of its parameters:

$$f(\vec{x}|\vec{\alpha}) = \sum_j^M g_j(\vec{x})\alpha_j$$

and that all experimental errors are Gaussian. The likelihood function is then:

$$p(D|\vec{\alpha}) = \prod_i^N \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma_i^2}(d_i - f(\vec{x}_i|\vec{\alpha}))^2\right]$$

$$= \prod_i^N \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma_i^2}(d_i - \sum_j^M g_j(\vec{x}_i)\alpha_j)^2\right]$$

If the priors on the α_i are either flat or Gaussian, then the posterior distribution is itself a Gaussian! (Just combine all the terms in the exponentials, and group quadratic, linear, and constant terms separately---nothing is left over!)

Gaussian approximation

The exponential of any multidimensional Gaussian g(θ) can be written as:

$$g(\theta) \propto \exp\left[-\frac{1}{2}(\vec{\theta}-\vec{\theta}_0)^T \cdot \mathbf{I} \cdot (\vec{\theta}-\vec{\theta}_0)\right]$$

where the matrix **I** is an MxM matrix. It is really just the curvature matrix of the function at its peak:

$$\mathbf{I}_{\alpha\beta} = -\frac{\partial^2 \ln g}{\partial\theta_\alpha \partial\theta_\beta} \text{ evaluated at } \vec{\theta}_0$$

If the posterior function is sufficiently Gaussian, then we can expand it around its maximum. In that case g(θ)=p(θ|I)P(D|θ,I), and **I** gets called the Fisher information matrix.

Integrating Gaussians

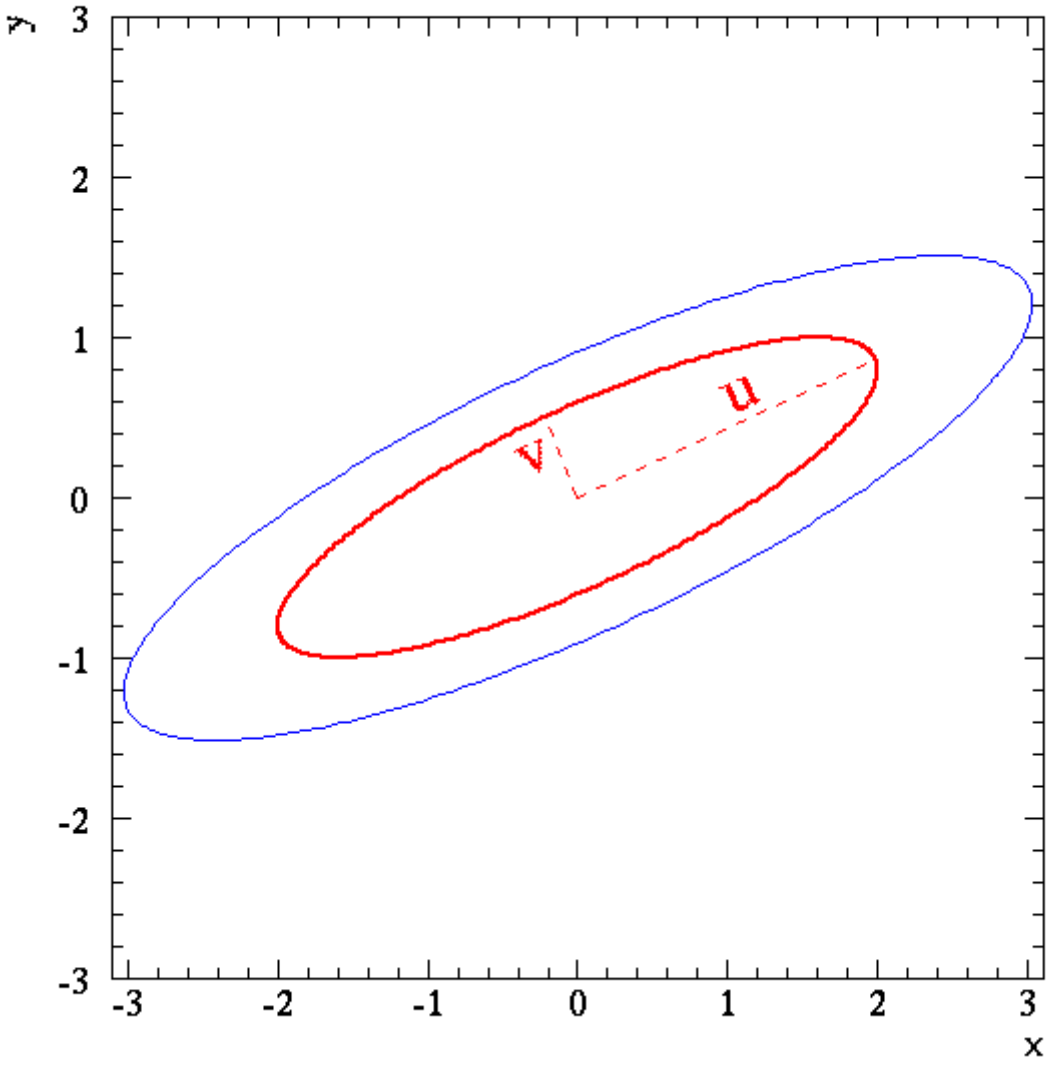
For any multi-dim Gaussian

$$\exp\left[-\frac{1}{2}(\vec{\theta}-\vec{\theta}_0)^T\cdot\mathbf{I}\cdot(\vec{\theta}-\vec{\theta}_0)\right]$$

we can do a change of variables to produce a set of orthogonal axes that make it easy to do the integral.

The integral of the above exponential over all parameters can be shown to equal:

$$(2\pi)^{M/2}(\det\ \mathbf{I})^{-1/2}$$



Laplacian approximation

Almost any sharply peaked function can be approximated as a Gaussian. Even if the likelihood depends non-linearly on the model parameters, the central limit theorem means that the likelihood approaches a Gaussian. First we figure out the parameter values θ_0 that maximize the posterior PDF. Then we approximate:

$$P(\theta|D,M,I) \approx P(\vec{\theta}_0|M,I)\, L(\vec{\theta}_0) \exp\left[-\frac{1}{2}(\vec{\theta}-\vec{\theta}_0)^T \cdot \mathbf{I} \cdot (\vec{\theta}-\vec{\theta}_0)\right]$$

evaluating the Fisher information matrix **I** at the maximum. Finally we can evaluate the integral over all of the parameters as:

$$P(D|M,I) \approx P(\vec{\theta}_0|M,I)\, L(\vec{\theta}_0) (2\pi)^{M/2} (\det \mathbf{I})^{-1/2}$$

More on marginalization

Recall from previous classes that there are two ways to remove a nuisance parameter from a problem:

1) integrate the posterior distribution over the nuisance parameter to yield the marginal distribution for the remaining parameters
2) keeping the other parameters fixed, minimize the -ln(L) over all the nuisance parameters. Scan over the parameters you care about and minimize with respect to the rest. We call the resulting function the "projected" distribution:

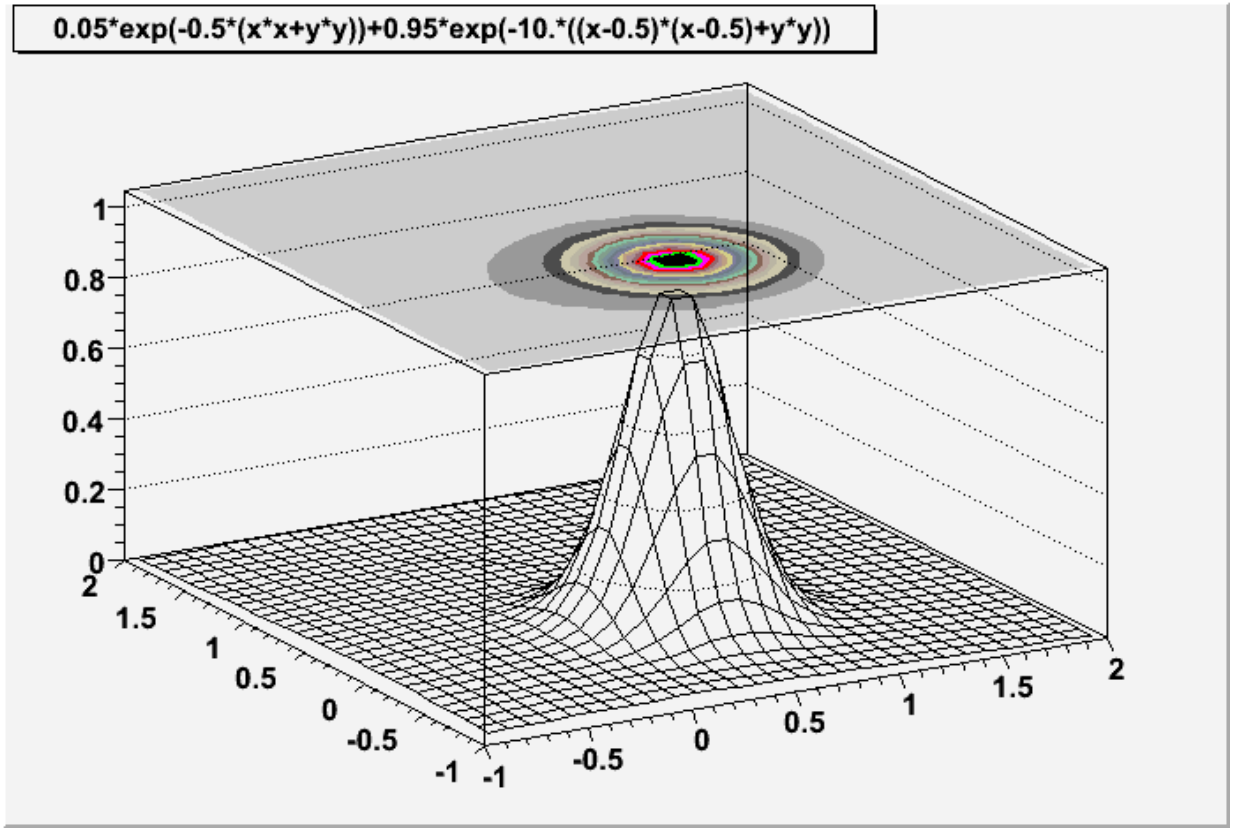
$$-\ln L_{proj}(x) = \min_{y}\left[-\ln L(x, y)\right]$$

For a uniform prior, the posterior distribution is exp[ln(L)].

How are these two marginalizing procedures related?

A comparison of two marginalizations

$$0.05\exp\left[-\frac{1}{2}(x^2+y^2)\right]+0.95\exp\left[-\frac{1}{2(0.05)}((x-0.5)^2+y^2)\right]$$



Has 51.3% of probability content in first term, and 48.7% in the second.

A comparison of two marginalization procedures

$$0.05\exp\left[-\frac{1}{2}(x^2+y^2)\right]+0.95\exp\left[-\frac{1}{2(0.05)}((x-0.5)^2+y^2)\right]$$

We can calculate the marginal and projected distributions analytically:

$$f_{marg}(x)=\int dy\ f(x,y)\propto 0.05\exp\left[-\frac{1}{2}x^2\right]+\frac{0.95}{\sqrt{20}}\exp\left[-\frac{1}{2(0.05)}(x-0.5)^2\right]$$

$$f_{proj}(x)=\max_y f(x,y)\propto 0.05\exp\left[-\frac{1}{2}x^2\right]+0.95\exp\left[-\frac{1}{2(0.05)}(x-0.5)^2\right]$$

Let's use Laplace's approximation instead 1

$$0.05\exp\left[-\frac{1}{2}(x^2+y^2)\right]+0.95\exp\left[-\frac{1}{2(0.05)}((x-0.5)^2+y^2)\right]$$

Laplace's approximation gives us another way of integrating over the nuisance parameter. We approximate the PDF by a Gaussian, then use

$$P(D|M,I)\approx P(\vec{\theta}_0|M,I)\,L(\vec{\theta}_0)(2\pi)^{M/2}(\det\,\mathbf{I})^{-1/2}$$

What this means for marginalization is that the marginal distribution should be approximated by:

$$f_{marg}(x)\propto f_{proj}(x)(\det\,\mathbf{I}(x))^{-1/2}$$

What this means is that for each value of x, we find the value of y that maximizes *f*. That gives us f_{proj}. We then calculate the information matrix, which is the matrix of partial derivatives of -ln *f* with respect to the nuisance parameters, with x considered to be fixed.

Let's use Laplace's approximation instead 2

$$0.05\exp\left[-\frac{1}{2}(x^2+y^2)\right]+0.95\exp\left[-\frac{1}{2(0.05)}((x-0.5)^2+y^2)\right]$$

In this example, we use:

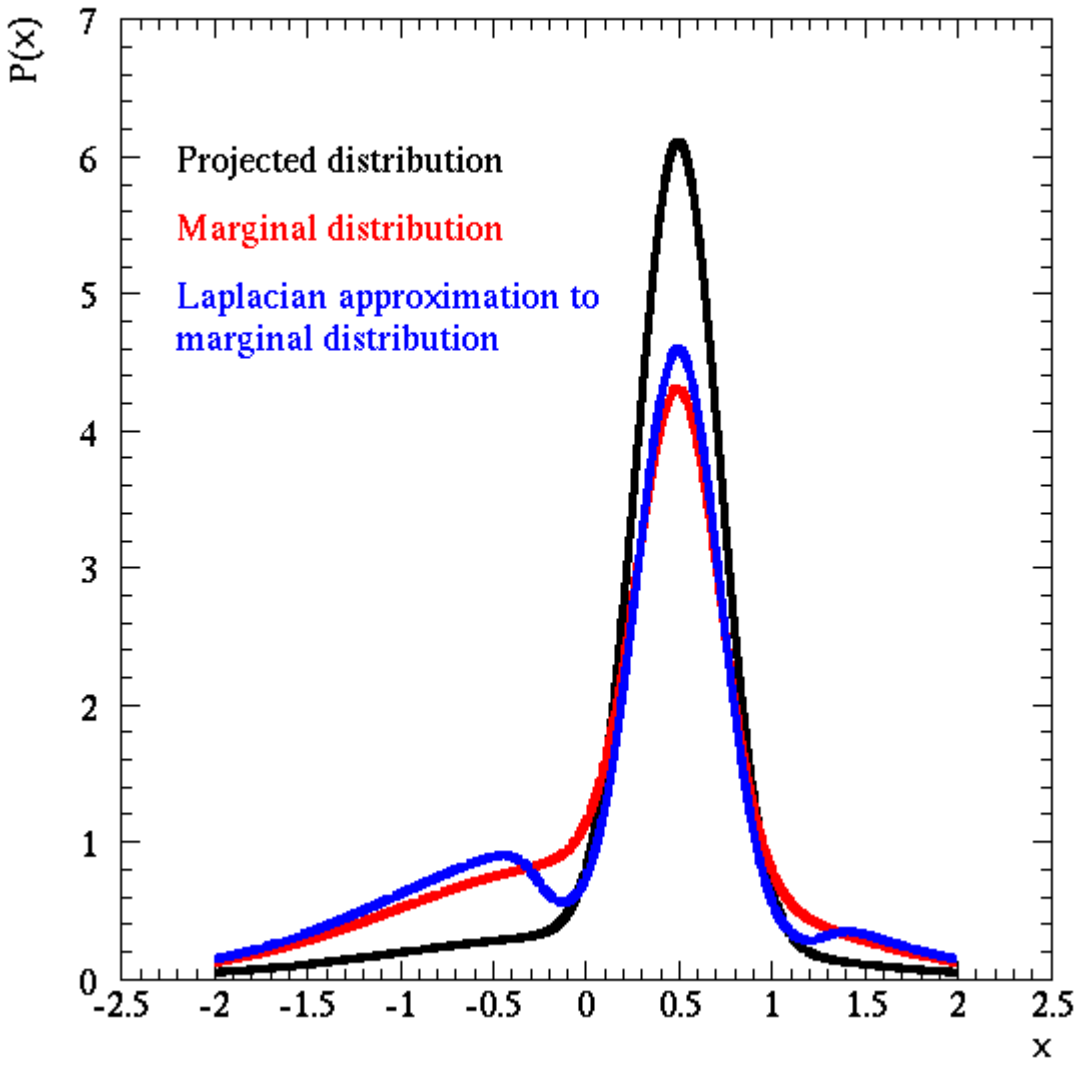
$$f_{marg}(x)\propto f_{proj}(x)(\det\ \mathbf{I}(x))^{-1/2}$$

Here **I**(x) is the matrix of derivatives with respect to the nuisance parameter. Since there's only one, y, **I**(x) is a 1x1 matrix, and is equal to:

$$I(x)=\frac{-\partial^2 \ln f(x,y)}{\partial y^2}\ \text{with}\ x\ \text{held fixed}$$

Results for Laplace's transformation:

The Laplacian approximation is clearly much closer to the marginal distribution than the projected distribution is!



Results for Laplace's transformation:

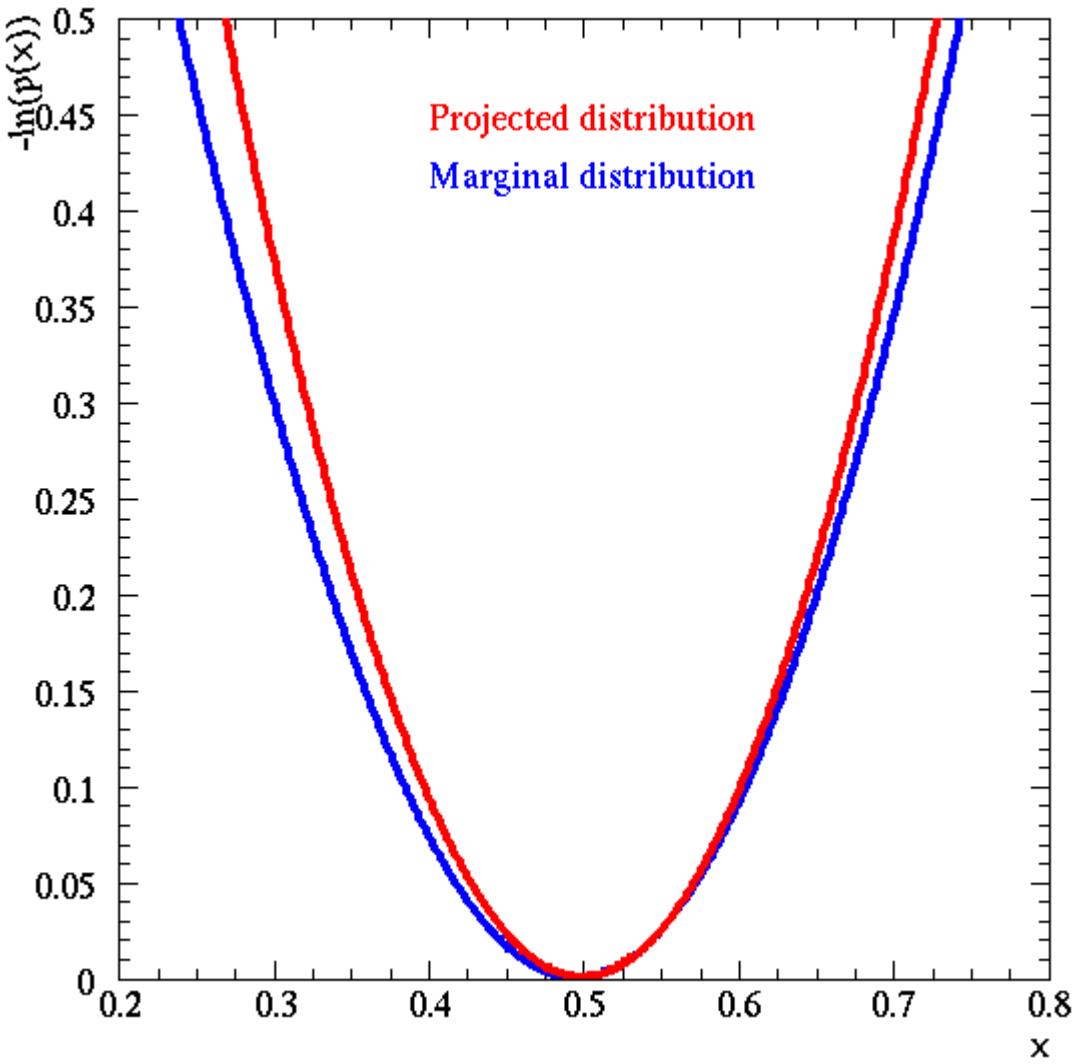
If I plot -ln P(x), and use the Δ ln L = +0.5 to define the 1σ error range, I get:

marginal distribution:

$$x = 0.49 \pm 0.25$$

projected distribution:

$$x = 0.49^{+0.23}_{-0.22}$$



Conclusions on marginalization

The best solution is to integrate the posterior PDF over the nuisance parameters.

If that's too much trouble, maximize the PDF as a function of the nuisance parameters to get the projected PDF. Then calculate the Fisher information matrix for the nuisance parmeters, and use:

$$f_{marg}(x) \propto f_{proj}(x)(\det \mathbf{I}(x))^{-1/2}$$

This gives a better approximation than the projected PDF alone.

If you're dealing with likelihoods (for example, in a frequentist analysis), the analogous form is:

$$-\ln L(x) \approx \min_{y}[-\ln L(x, y)] + \frac{1}{2}\ln \det \mathbf{I}(x)$$

where *I* is the matrix of pairwise partial derivatives of -ln L(x,y) with respect to all the nuisance parameters, evaluated at the values of those nuisance parameters that minimize -ln L(x,y) for the given x.

A confession: this is seldom done

For many years I wondered what the justification was for using function minimization as an approximation for doing an integral over a nuisance parameter. Frequentist statistics tells you to just do the projection, but doesn't motivate why this work.

The answer is that it comes from the Laplacian approximation to the Bayesian posterior PDF.

That being said, I have never actually seen a frequentist calculate det(**I**) and use that as a correction to the likelihood when marginalizing. The reason is two-fold:

1) If the joint likelihood is really Gaussian, then **I** ≈ constant.
2) To be perfectly honest, hardly anyone has heard of this correction, or could justify why you should remove nuisance parameters by projection! Not that this makes it OK ...

It's all about the minimization

The whole key to applying Laplace's approximation for the PDF or the likelihood is the ability to locate the global minimum. While finding a minimum in N-dimensional parameter space may be easier than doing an N-dimensional integral, it ain't exactly a piece of cake.

Many good descriptions of minimization routines exist. See Gregory Ch. 11, or Numerical Recipes Ch. 10.

Not every function has a single peak or a Gaussian-like minimum!

Monte Carlo Integration

Suppose we have some nasty multi-dimensional integral (possibly over some nuisance parameters) that we cannot do analytically or by Laplace's approximation (perhaps it's not that Gaussian).

Monte Carlo integration is a numerical technique when all else fails ...

The basic idea is to randomly sample points uniformly over the region of integration, and to calculate the average value of the function over this region. Then

$$\int f\, dV \approx \langle f \rangle V \pm V \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}}$$

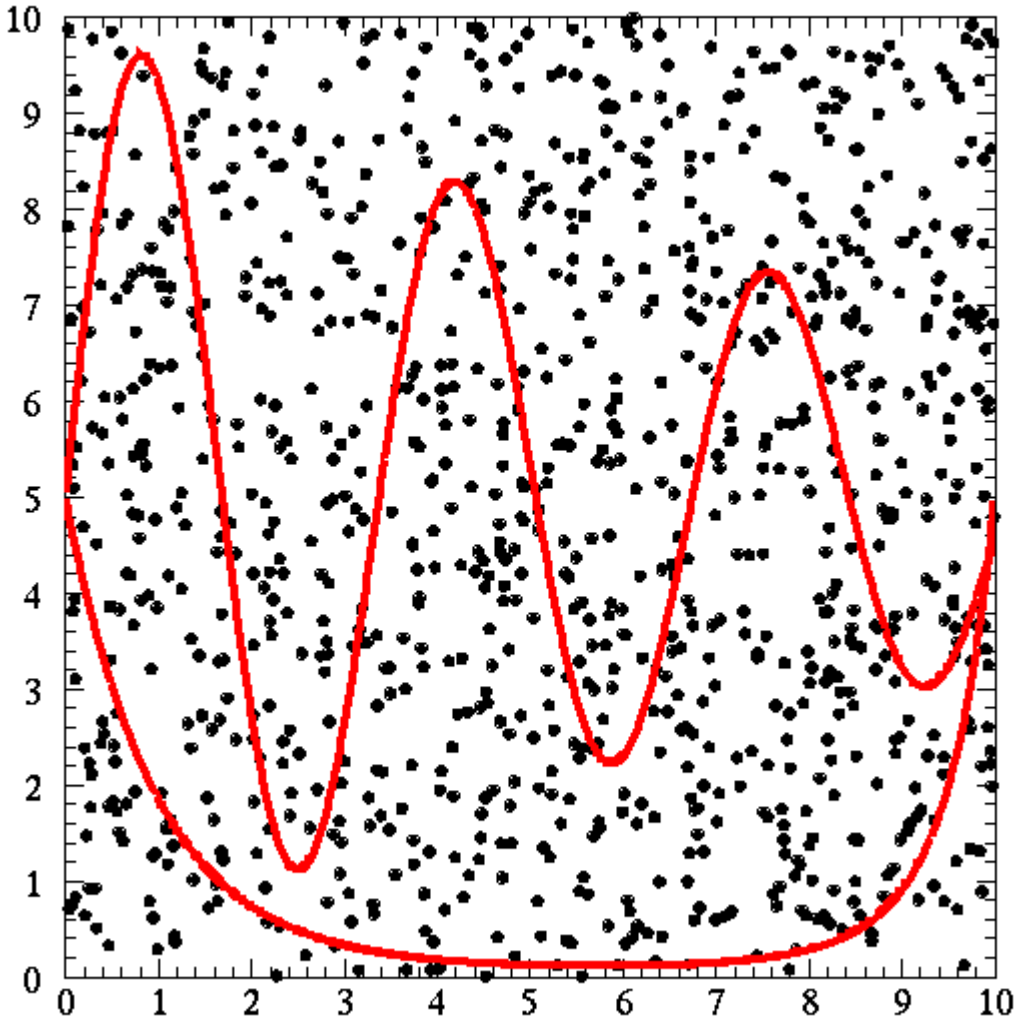
A case where Monte Carlo integration helps

Suppose we want to evaluate the area between the two red curves. Neither is analytically integrable.

Scatter points uniformly, see what fraction are inside the curves, and multiply that fraction by the total area. Fairly straightforward.

$$I = \int_A dx\, dy$$

1000 points is enough to give I=43.3 ± 1.6



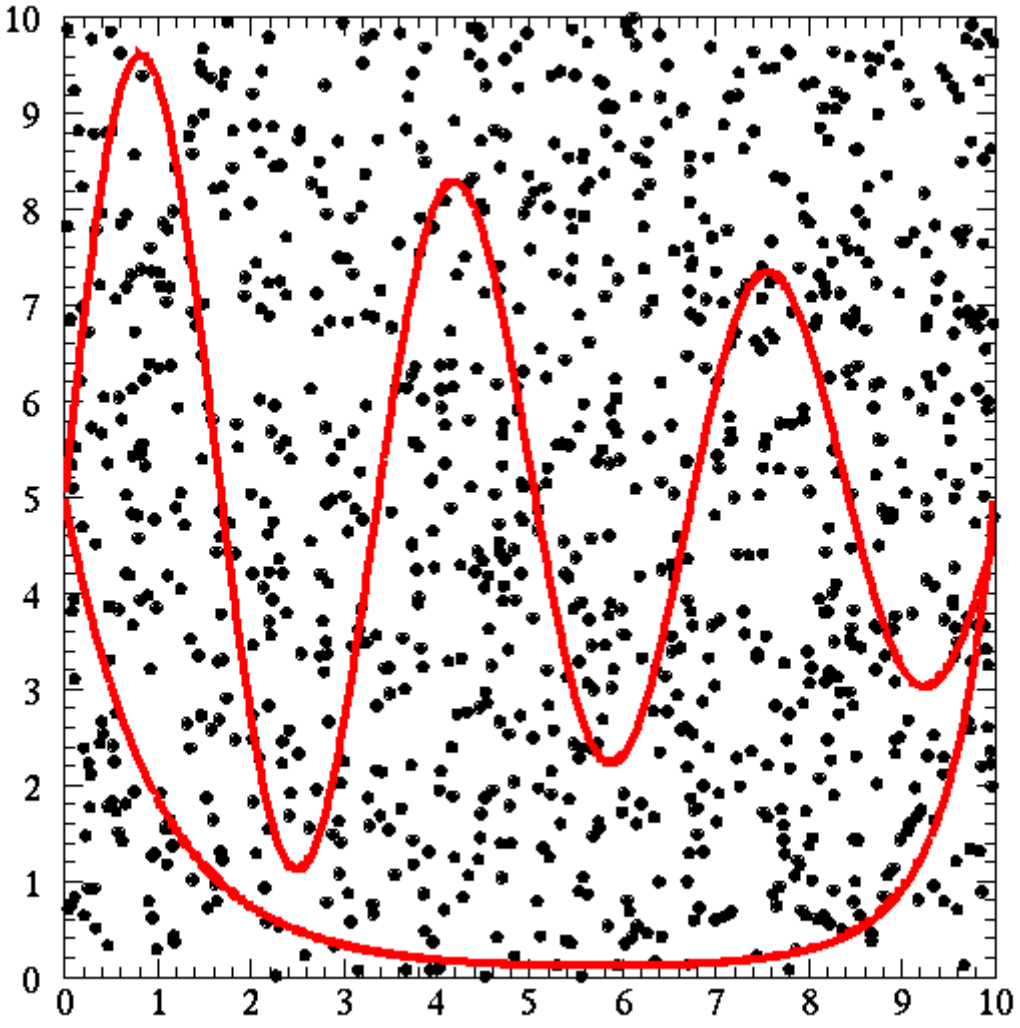
Slightly more complicated

Now suppose we want to evaluate this integral

$$I = \int_A \exp[-0.1(x^2+y^2)]\,dx\,dy$$

we can do it the same way---we evaluate the function of the integrand if the random point is inside the region, and evaluate it as 0 outside. The average value of the function, times the total area sampled over, will equal I.

With 1000 points, I estimate I=3.50±0.32



Importance sampling

There's a different way to view this integral

$$I = \int_A \exp[-0.1(x^2+y^2)]\,dx\,dy = \int f(x,y) \times \left[\exp[-0.1(x^2+y^2)]\,dx\,dy\right]$$

where f(x,y)=1 if (x,y)∈A and 0 otherwise. If the stuff in brackets were a PDF, we would interpret this as the expectation value of f, given the PDF. Of course the term in brackets isn't normalized like a proper PDF, so we rather have that

$$I = \langle f \rangle \int_0^{10}\int_0^{10} dy\,dx \exp[-0.1(x^2+y^2)]$$

and <f> is the expectation value of f sampling from a PDF given by $\propto$ exp[-0.1(x²+y²)]

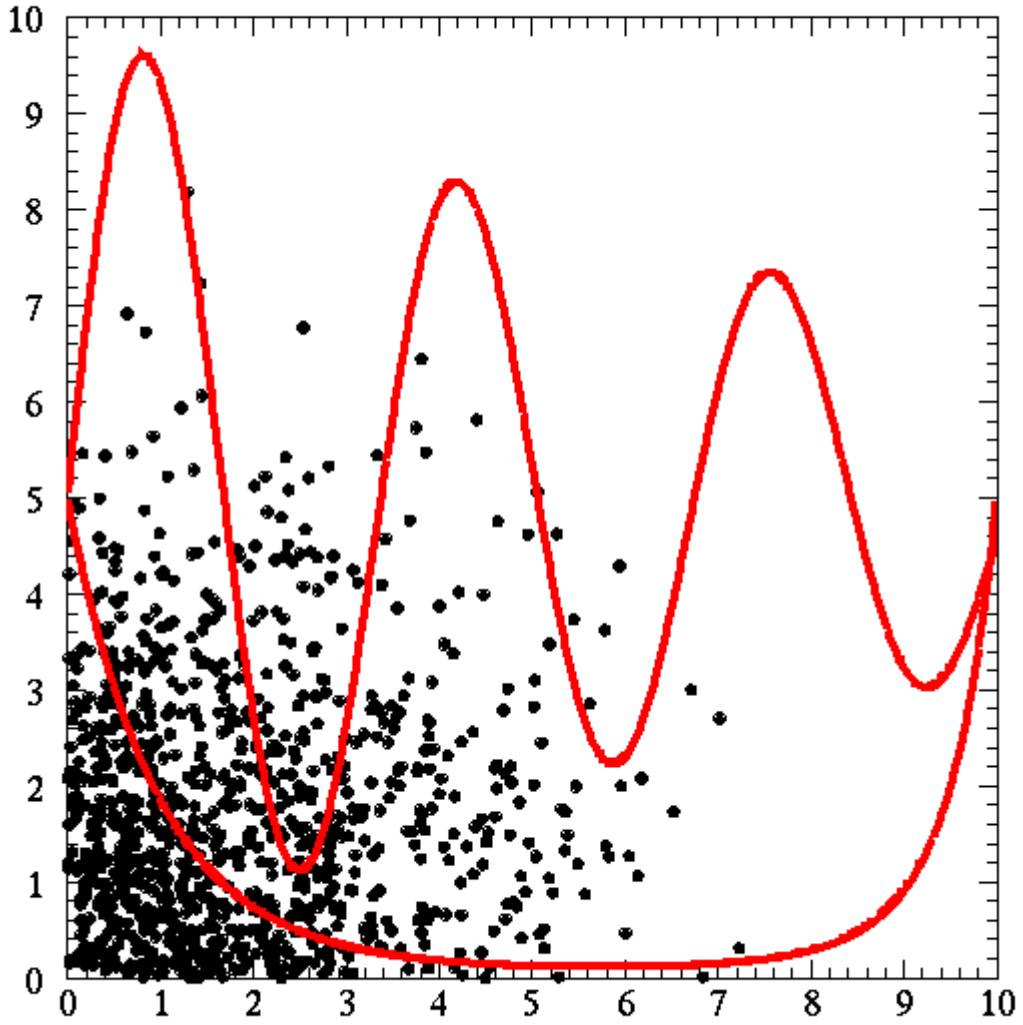
We can do the integral part analytically. Then we can calculate the average value of f by sampling N times from the sampling PDF, and dividing the sum by N.

Importance sampling result

With 1000 points, I estimate I=3.50±0.13. This is much more accurate than the I=3.50±0.32 that I got with uniform sampling and the same number of points.

Why better? Fewer "wasted" points evaluating the function where it's known to be small, and more effort spent where the function is large.

Of course I had to know enough about the function I was integrating to do this factorization.



Application to Bayesian analysis

Consider a typical Bayesian analysis integral over a nuisance parameter:

$$p(x|D,I)=\int dy\; p(x,y|D,I)\propto\int dy\; p(y|I)\,p(x|I)\,p(D|x,y,I)$$

It will often be the case that p(D|x,y,I) will be a complicated function of y, but you certainly know the prior, and can probably sample from it by drawing values of Y from the prior p(y|I).

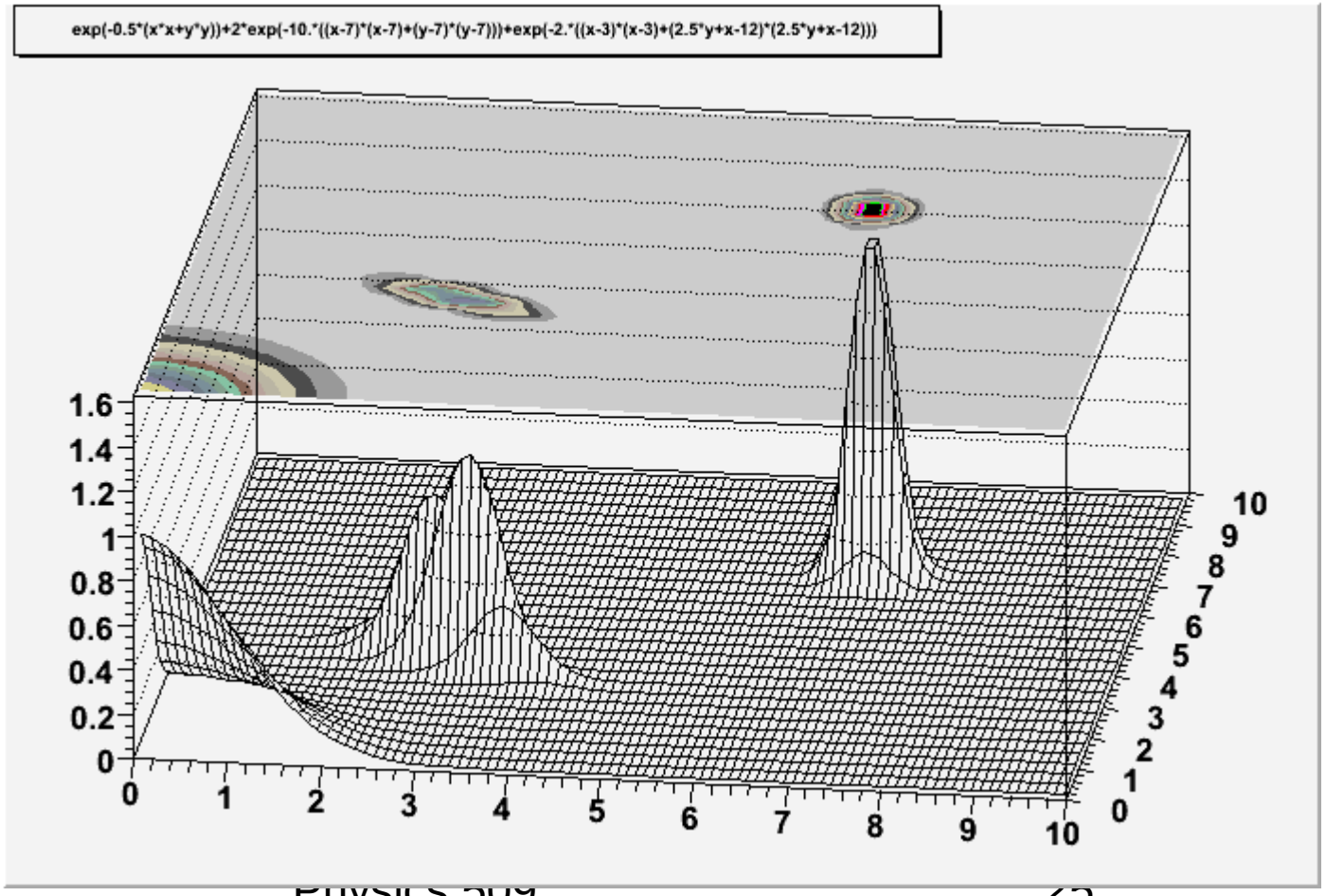
In general, to get more efficient Monte Carlo integration, try to factor the integral so that the function you evaluate is as flat as possible:

$$\int_A dx\; f(x)=\int_A dx\, g(x)\frac{f(x)}{g(x)}=\left\langle\frac{f}{g}\right\rangle\cdot\int_A dx\, g(x)$$

If f(x)/g(x) is very flat, then you get great accuracy, but of course you have to be able to do the integral over g(x)!

A terrible PDF to deal with:

Consider the following scary PDF shown below. You want to sample from it, but it's disconnected and ugly. Rejection method will be very inefficient!



Metropolis-Hastings

The Metropolis-Hastings algorithm is a tool for sampling from a complicated PDF. All it requires you to know about the shape of the PDF is the ability of calculate the PDF's value at any given point. The basic idea is to do a "weighted" random walk.

1) Pick some set of starting values X_0 for the PDF variables.

2) Generate a new proposed set of values Y for these variables using some proposal distribution: $q(Y|X_i)$

For example, $q(Y|X_i)$ might be a multidimensional Gaussian centered at X_i.

3) Calculate the Metropolis ratio *r*:

$$r = \frac{p(Y)\,q(X_i|Y)}{p(X_i)\,q(Y|X_i)}$$

4) Calculate a uniform random number U from 0 to 1. If U<r, then replace X_i with Y (X_{i+1}=Y). If U>r, then X_{i+1}=X_i (just repeat the last set of values)

5) Throw away the first part of the sequence as "burn-in". The rest should correctly sample p(X)

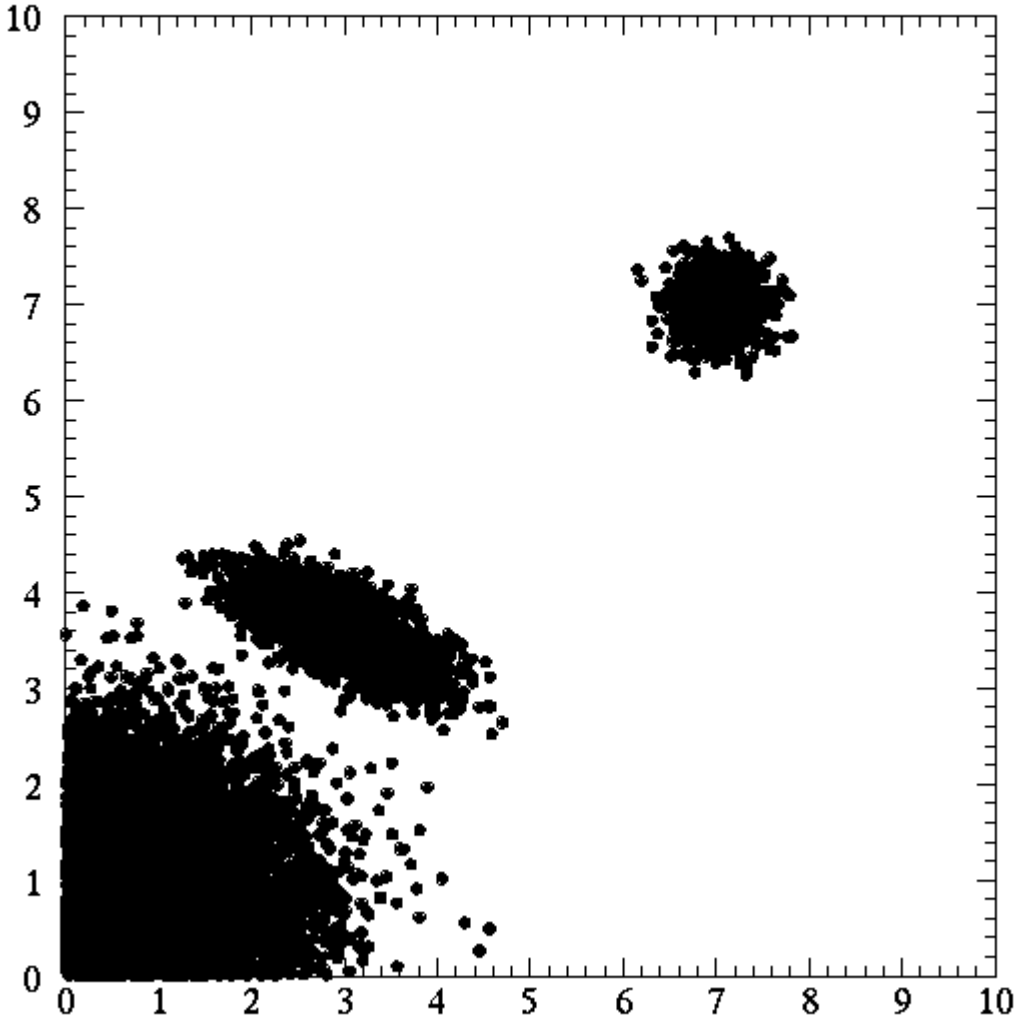
Metropolis-Hastings results

Proposal distribution---a Gaussian of width two centered at the current point:

$X_{i+1}=X_i$ + 2 gasdev
$Y_{i+1}=Y_i$ + 2 gasdev

Acceptance rate for proposed new points: ~9%

Not so efficient, but whole of PDF is seemingly being sampled.

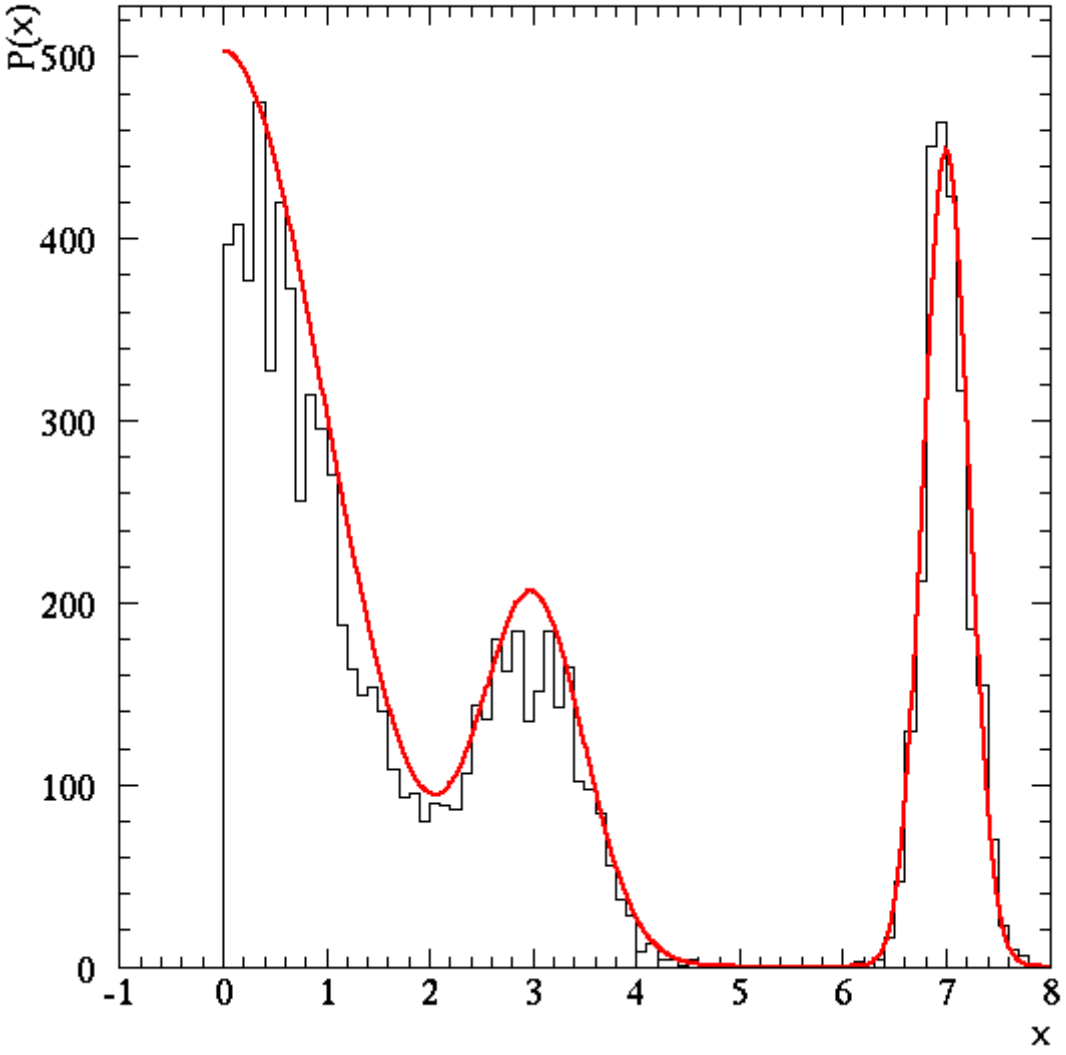


Metropolis-Hastings results: marginal distribution

For this proposal distribution, I compute the marginal distribution for x.

Red curve is the exact analytic solution---a close match!

Note that in spite of having 100,000 samples, the curve doesn't look very smooth. This is because each point is re-used many times (small acceptance probability for new points.)



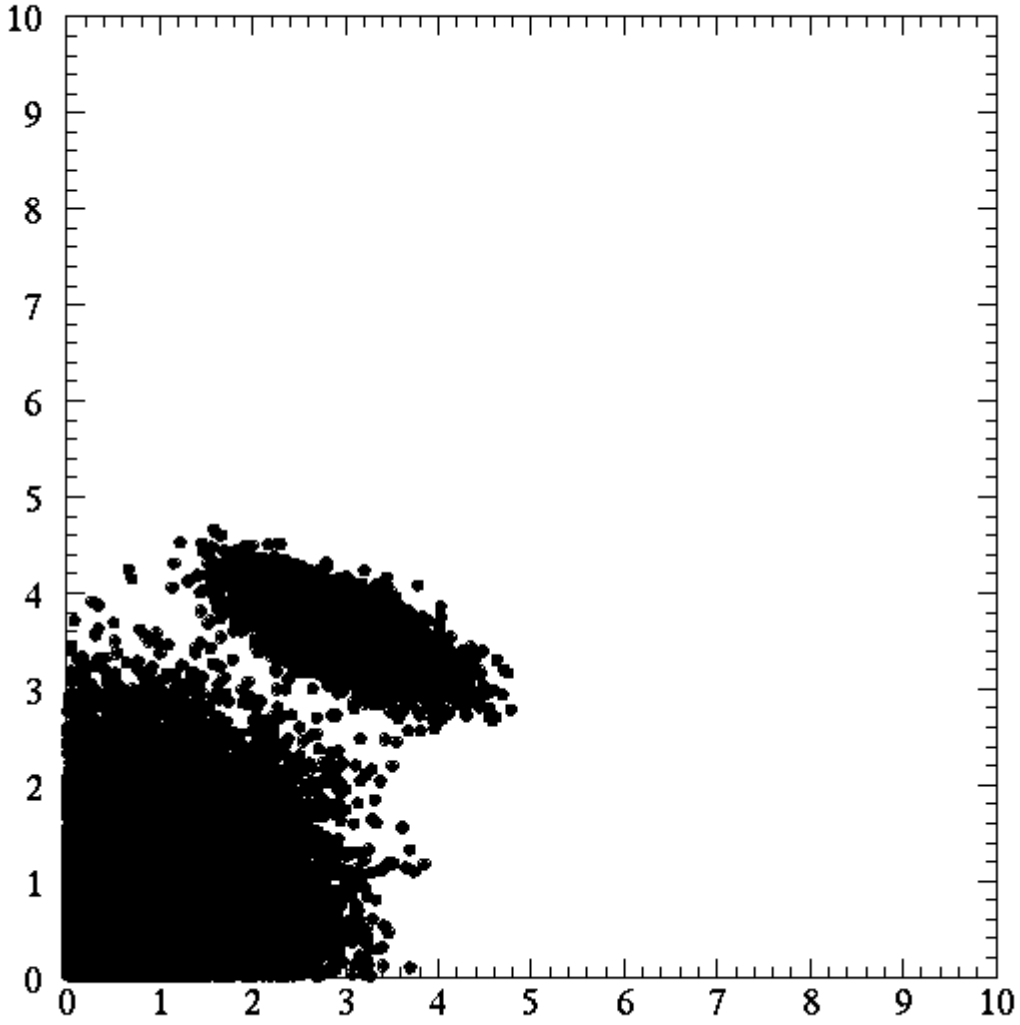
Metropolis-Hastings results with finer steps

Proposal distribution---a Gaussian of width one centered at the current point:

Acceptance rate for proposed new points: ~26%

The peak at (7,7) is not being sampled! The algorithm is not likely to randomly jump over the valley in between for this case.

What do you do?



Tempered Metropolis-Hastings

Choosing the best proposal distribution is tricky. If the random steps are too small, you won't sample the whole distribution. If the steps are too big, the algorithm isn't efficient, and individual points get repeated too many times.

You really run into trouble when there are many separated peaks in the distribution---the same kind of problem where you have trouble finding a maximum/minimum.

Various "tempering" algorithms try to correct for this by adaptively altering the effective step size. For example, consider altering the posterior distribution by

$$p(x|I)\,p(D|x,I) \rightarrow p(x|I)\exp[\beta \ln p(D|x,I)]$$

If β=1, this is the normal distribution. As $\beta \rightarrow 0$, it get progressively flatter. We call β the temperature parameter---think of β as 1/kT.

Tempered Metropolis-Hastings 2

$$p(x|I)\,p(D|x,I) \rightarrow p(x|I)\exp[\beta \ln p(D|x,I)]$$

When β is small (temperature large), the distribution is flatter, and you will more readily sample different parts of parameters space.

In a parallel tempering simulation, you run multiple copies of the simulation, each at a different temperature. The high temperature versions will be good for looking at the global structure---high acceptance probability for moves to very different parts of parameter space. The low temperature versions will be better for sampling fine structure. (Think "coarse grid search" and "fine grid search").

In parallel tempering, you have a rule that on every iteration you have some small probability of proposing that a higher temperature and a lower temperature simulation "swap" their current sets of parameter values. If a swap is proposed, you then decide whether to accept or not based on a probability ratio (see Gregory Sec 12.5)

Tips for Metropolis-Hastings

1) Try different proposal distributions (both coarse-grained and fine-grained) to make sure you're sampling all of the relevant parameter space.

2) Be careful with “burn-in”. The first several 10?/100?/1000? iterations will not have reached an equilibrium condition yet. You can partly check this by calculating the autocorrelation function, or at least by checking whether the PDF or likelihood value at the peak has reached a plateau.

3) Always remember that successive events are correlated---do not use output for time-correlated studies (although maybe a random shuffle would help)

4) Routines are easy to code, but take a long time to adjust to give sensible results. Be cautious!