PART 4: PRACTICAL BAYESIAN METHODS

Matthew Pitkin
GraWIToN School

25 March 2015

University of Glasgow

In this part of the course we will discuss a practical method for Bayesian parameter estimation:

- Markov chain Monte Carlo (MCMC)

This method is now very commonly used to perform parameter estimation for multi-dimensional posterior parameter spaces.

Markov chain Monte Carlo (MCMC)

- Markov chain - a sequence of random variables where the probability of a subsequent state only depends on the current state (they are "memoryless")
- Monte Carlo - an algorithm relying on repeated random sampling

So, a MCMC is a class of algorithms for drawing random samples that have a Markovian property.

We are particularly interesting in the **Metropolis-Hastings algorithm**. This can be used to efficiently draw samples from an underlying probability distribution.

The Metropolis-Hastings algorithm (simplified to a 1D case):

- choose some initial random point in the parameter space, x_1, and calculate posterior $p(x_1|d, I)$
- centre a new pdf $q(x'|x_1, I)$, the **proposal distribution**, at x_1
- randomly draw a point from $x_2 \sim q(x'|x_1, I)$, and calculate posterior $p(x_2|d, I)$

Now compute the ratio

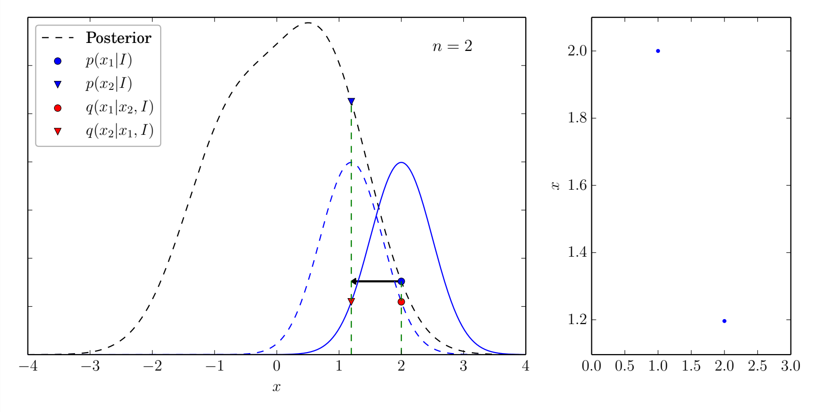
$$R = \frac{p(x_2|d, I)}{p(x_1|d, I)} \frac{q(x_2|x_1, I)}{q(x_1|x_2, I)}$$

The Metropolis-Hastings algorithm:

- accept the point x_2 if $R > 1$ (go uphill in posterior)
- if $R < 1$
  - accept the point x_2 with a probability R (go downhill in posterior)
  - otherwise reject the new point x_2 and set $x_2 = x_1$

This process gets repeated many times to build up a chain of samples.

Note: in general the proposal distribution will be Gaussian, so will be symmetric and $\frac{q(x_2|x_1,I)}{q(x_1|x_2,I)} = 1$. This ratio is required to maintain **detailed balance**, i.e. there should be the same probability to go forward along the chain as to run it in reverse.

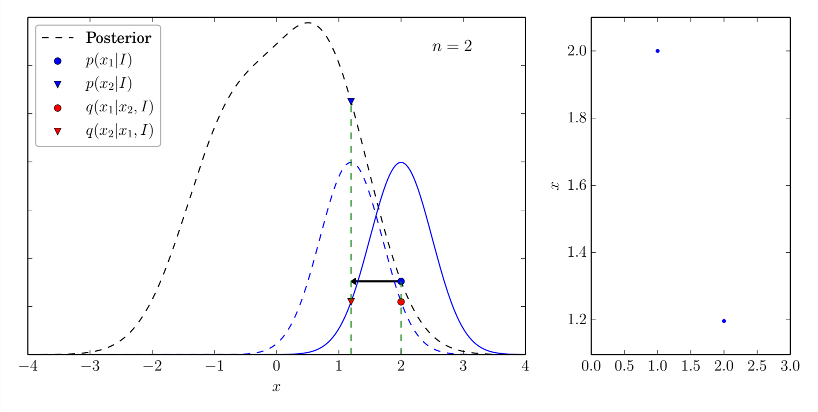

Posterior
$p(x_1|I)$
$p(x_2|I)$
$q(x_1|x_2, I)$
$q(x_2|x_1, I)$
$n = 2$
−4 −3 −2 −1 0 1 2 3 4
x
2.0
1.8
1.6
1.4
1.2
x
0.0 0.5 1.0 1.5 2.0 2.5 3.0

So the Metropolis Algorithm generally (but not always) moves uphill, towards the peak of the posterior pdf.

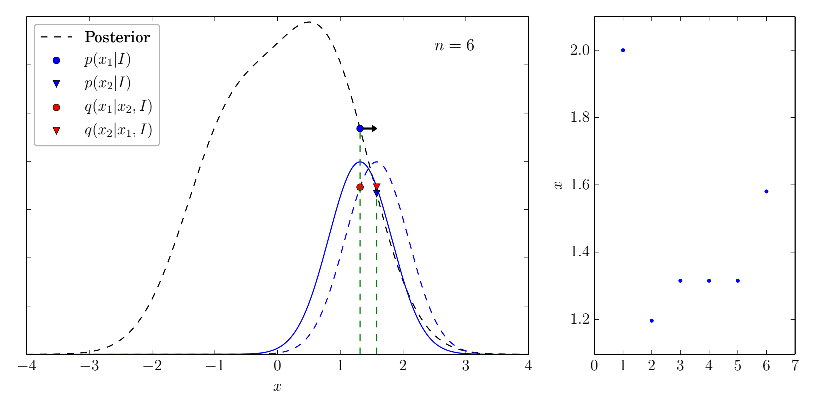
Remarkable facts:

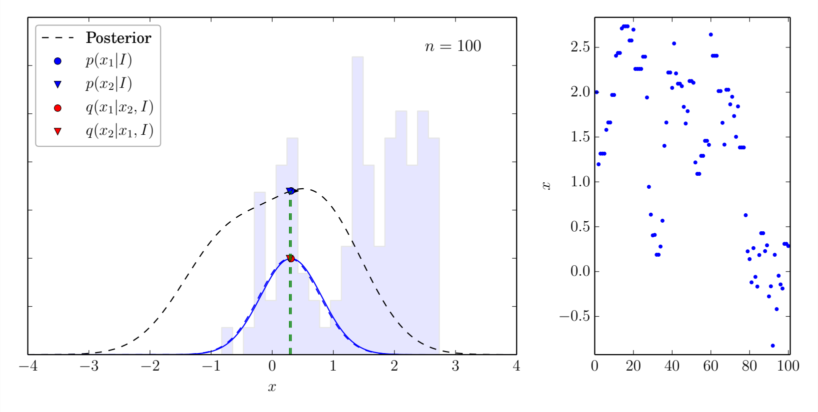
- The sequence of points $\{x_1, x_2, \ldots, x_n\}$ represent samples from the marginalised posterior $p(x|d, I)$
- We can make a histogram of $\{x_1, x_2, \ldots, x_n\}$ and use it to compute the mean and variance of x

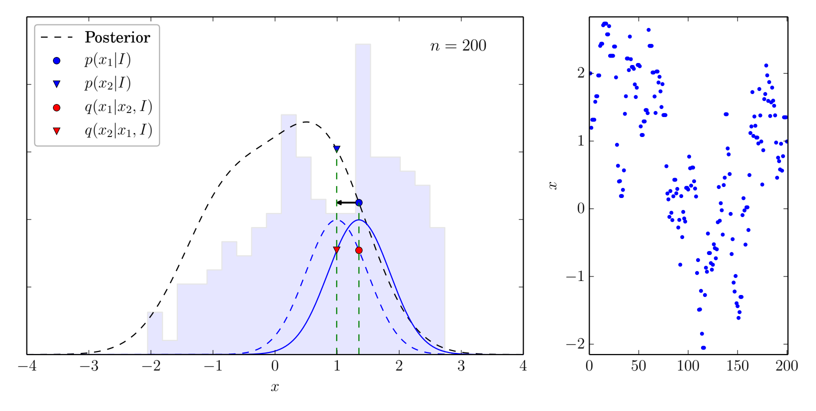
MCMC DEMONSTRATION

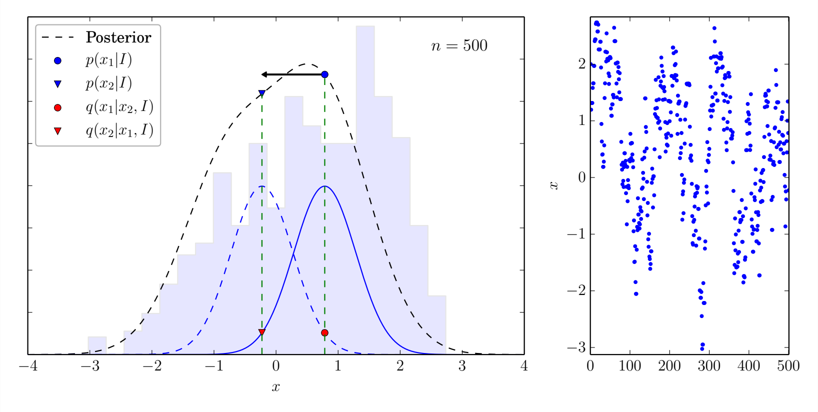


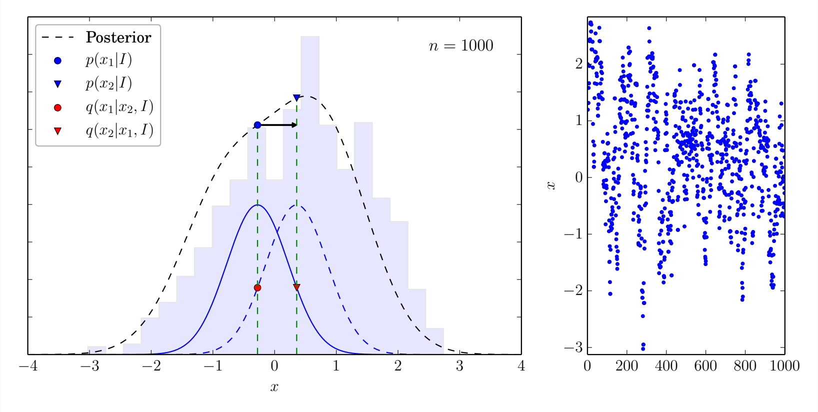
MCMC DEMONSTRATION



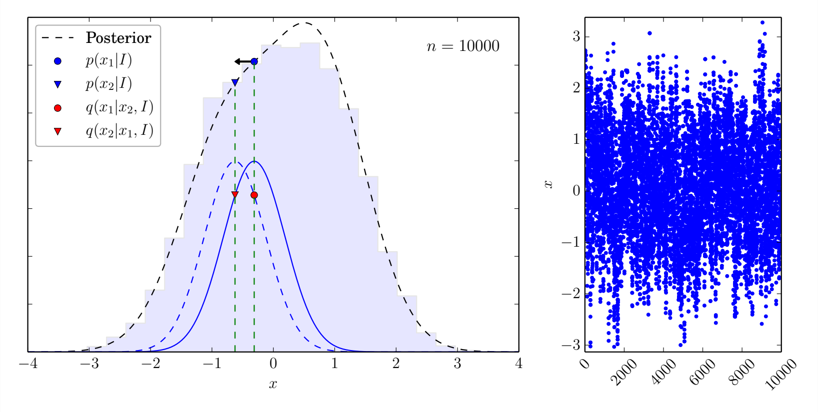

Posterior
$p(x_1|I)$
$p(x_2|I)$
$q(x_1|x_2, I)$
$q(x_2|x_1, I)$
$n = 100$
−4 −3 −2 −1 0 1 2 3 4
x
2.5 2.0 1.5 1.0 0.5 0.0 −0.5
x
0 20 40 60 80 100

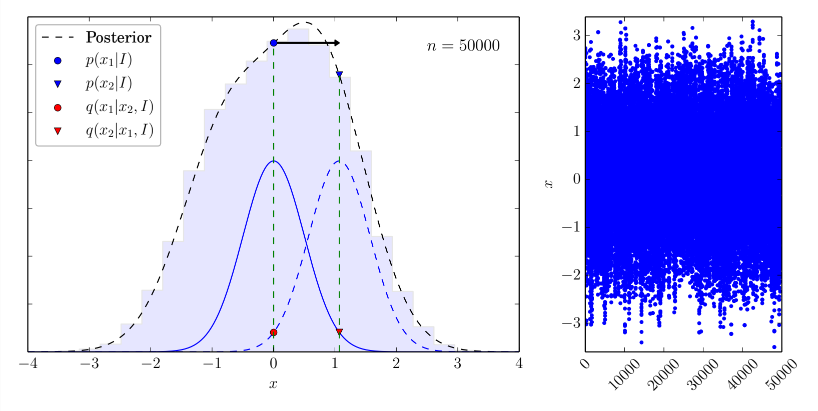

Posterior
$p(x_1|I)$
$p(x_2|I)$
$q(x_1|x_2, I)$
$q(x_2|x_1, I)$
$n = 200$
−4
−3
−2
−1
0
1
2
3
4
x
2
1
0
−1
−2
0
50
100
150
200
x


Posterior
$p(x_1|I)$
$p(x_2|I)$
$q(x_1|x_2, I)$
$q(x_2|x_1, I)$
$n = 500$
−4 −3 −2 −1 0 1 2 3 4
x
2 1 0 −1 −2 −3
0 100 200 300 400 500
x



Posterior
$p(x_1|I)$
$p(x_2|I)$
$q(x_1|x_2, I)$
$q(x_2|x_1, I)$
$n = 1000$
−4
−3
−2
−1
0
1
2
3
4
x
2
1
0
−1
−2
−3
0
200
400
600
800
1000
x


Posterior
$p(x_1|I)$
$p(x_2|I)$
$q(x_1|x_2, I)$
$q(x_2|x_1, I)$
$n = 10000$
-4 -3 -2 -1 0 1 2 3 4
x
x
3 2 1 0 -1 -2 -3
0 2000 4000 6000 8000 10000


Posterior
$p(x_1|I)$
$p(x_2|I)$
$q(x_1|x_2, I)$
$q(x_2|x_1, I)$
$n = 50000$
−4
−3
−2
−1
0
1
2
3
4
x
3
2
1
0
−1
−2
−3
x
0
10000
20000
30000
40000
50000

If we don't choose a start point close to the bulk of the posterior it may take time for the chain to converge on that area, so (given a non-infinite amount of samples) a number of initial points may need to be *discarded* during the **burn-in**.

(A couple of) diagnostic methods to assess convergence are:

- Geweke's test - test whether the means of an early part of the chain are consistent with the means for a later part
- Gelman-Rubin test - test whether the variances of two parallel chains are consistent

Although just checking by eyeball is often a good way to proceed.

MCMC chains are not necessarily uncorrelated (not entirely Markovian!), so you can calculate the autocorrelation length of the chains and thin them accordingly. This can also be used as a test of convergence, in that the autocorrelation length may be large during the burn in phase.

If you have an infinite chain (almost) any choice of start position and proposal would produce the marginalised posterior, but we can only practically produce a finite number of samples, e.g. $100\,000$s or $1\,000\,000$s

For efficient sampling we could have:

- a proposal that quite closely matches the posterior being sampled (e.g. via adjusting the proposal during **burn-in** based on covariance of currently collected samples)
- use a parameterisation where the parameters are (close to) independent
- simulated annealing (flattening the likelihood by “raising the temperature” during a **burn-in** period)

Simple MCMC methods can have problems when:

- posterior is very tightly constrained within a very small part of the allowed parameter space
- posterior is multi-modal
- posterior has oddly shaped degeneracies between parameters

(A couple of) methods to address these are e.g.

- parallel tempering
- ensemble samplers

There are lots of other advanced methods suitable for dealing with more complex situations, e.g.:

- Nested sampling (evaluate *evidence* integrals)
- Principle component analysis (PCA) (data compression)
- Reversible Jump MCMC
- Hierarchical models
- Approximate Bayesian Computation (ABC)
- Non-parameteric methods
  - Gaussian processes
  - Dirichlet processes
- Bayesian neural networks

It is a large and rapidly evolving field!