Debugging Samplers

Making MCMC Work in Practice

Micha Elsner

Machine Learning Reading Group
Brown University

July 8, 2010

Markov Chain Monte Carlo

- Posterior inference in graphical models
- Easy to design a theoretically correct algorithm...
  - (but sometimes harder to get a good one)
- Popular techniques:
  - Metropolis-Hastings
  - Gibbs Sampling

So how does it work in real life?

I'm going to assume you've seen the **core algorithms** and **basic math**.
We're going to cover **diagnostics and development techniques**, mostly for **directed graphical models**.

Before you start

It's worth asking: why are you building your own sampler?

Off-the-shelf tools

Increasingly powerful, flexible, and efficient
BUT...
As researchers, we do sometimes need additional capability
Or as students, we want to learn hands-on

Anyway, check out:

- FACTORIE (UMass)
- Hierarchical Bayes Compiler (Hal Daume)
- Church (MIT)
- Bayes Net Toolbox (Kevin Murphy)
- etc...

What can go wrong?

Everything!

This talk is all about diagnosing errors.
Debugging MCMC is tricky because the programs are stochastic,
and errors occur at *many levels* of representation.

What can go wrong?

Everything!

This talk is all about diagnosing errors.
Debugging MCMC is tricky because the programs are stochastic,
and errors occur at *many levels* of representation.

- **Model error:** your model doesn't describe the data

What can go wrong?

Everything!

This talk is all about diagnosing errors.
Debugging MCMC is tricky because the programs are stochastic,
and errors occur at *many levels* of representation.

- **Model error:** your model doesn't describe the data
- **Search error:** you get stuck in a bad region

What can go wrong?

Everything!

This talk is all about diagnosing errors.
Debugging MCMC is tricky because the programs are stochastic,
and errors occur at *many levels* of representation.

- **Model error:** your model doesn't describe the data
- **Search error:** you get stuck in a bad region
- **Math error:** your math doesn't encode the model/search you designed

What can go wrong?

Everything!

This talk is all about diagnosing errors.
Debugging MCMC is tricky because the programs are stochastic,
and errors occur at *many levels* of representation.

- **Model error:** your model doesn't describe the data
- **Search error:** you get stuck in a bad region
- **Math error:** your math doesn't encode the model/search you designed
- **Code error:** your code doesn't implement the math you intended

Overview

Preliminaries: example model and inference

Synthetic data

Analysing likelihood

Search errors

Overview

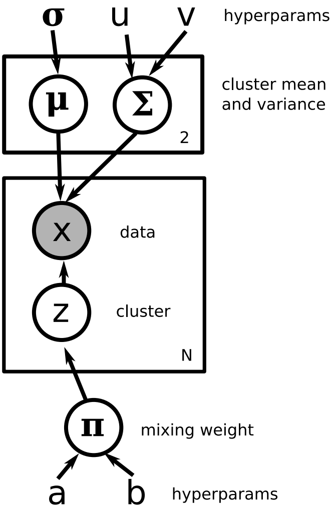
Preliminaries: example model and inference

Synthetic data

Analysing likelihood

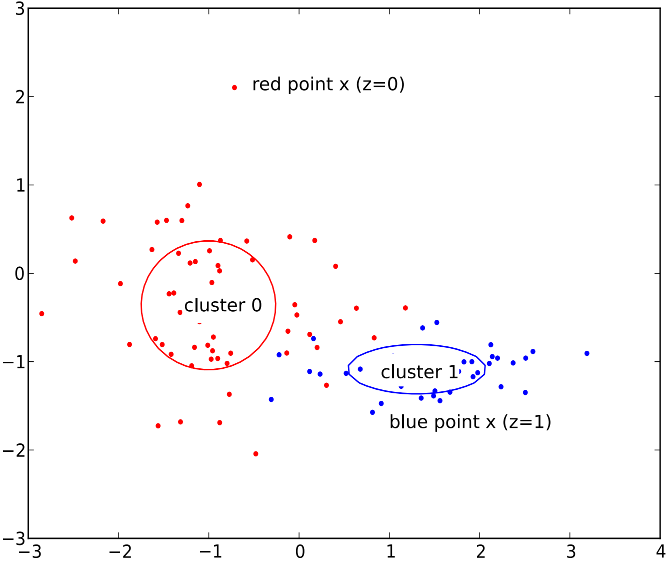
Search errors

Case study: Two-component Gaussian mixture



- $\mu \sim N(0, \sigma)$
- $\Sigma \sim InvGamma(u, v)$
- $\pi \sim Beta(a, b)$
- $z_i \sim Bernoulli(\pi)$
- $x_i \sim N(\mu_{z_i}, \Sigma_{z_i})$

Some data



red point x (z=0)
cluster 0
cluster 1
blue point x (z=1)
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
4

Case study

The model:

- Should look very familiar...
- Similar to most Bayesian clustering models:
  - Full mixture of Gaussians
  - LDA (mixture of multinomial)

Case study

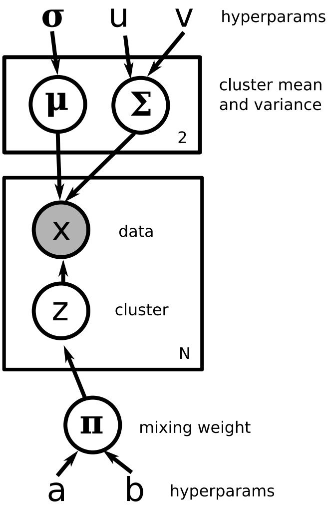
The model:

- Should look very familiar...
- Similar to most Bayesian clustering models:
  - Full mixture of Gaussians
  - LDA (mixture of multinomial)

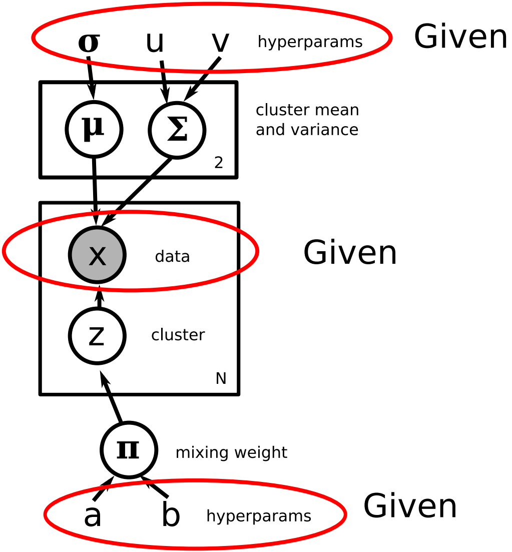
Our sampler: designed to showcase some popular methods.

- z: Collapsed Gibbs
  - Integrate out π (using conjugate Beta prior)
- μ, Σ: Metropolis-Hastings
  - Since our priors are conjugate...
  - We'd use Gibbs in real life
  - MH just for example

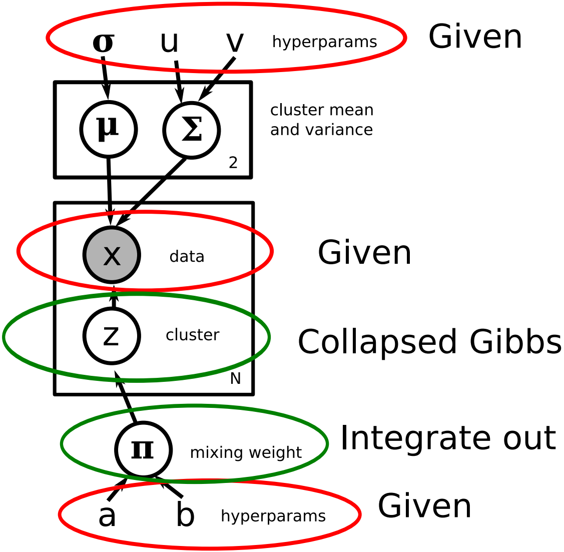
Case study: inference methods



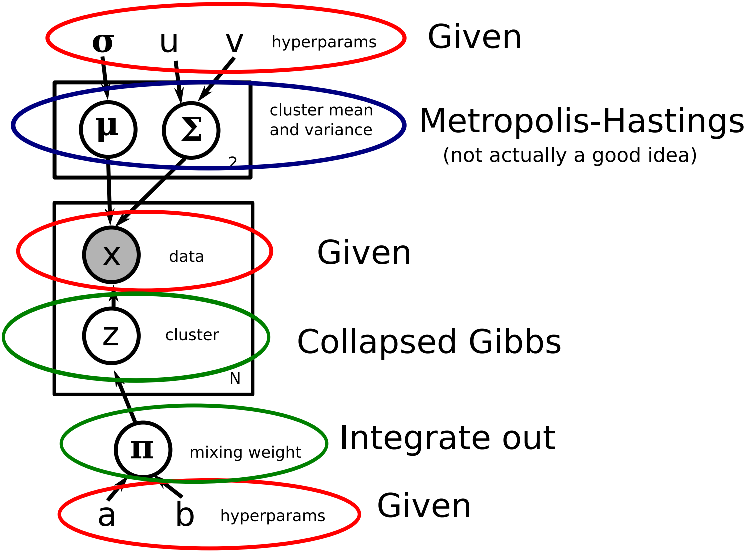
Case study: inference methods



Case study: inference methods



Case study: inference methods



Overview

Step 1: Sample a dataset

Synthetic data: why?

Want to work in an environment *we control*...

- Guaranteed to be distributed according to the model
- We know values for all the hidden variables
- We can have as much data as we want

None of these are true for the data you actually care about!

In a **directed** graphical model, it's easy to sample a dataset.
For **undirected** models, it's hard...

...and often requires MCMC.

Sampling data

Sampling data is *much* easier than sampling conditioned on values.

- Make up values for the hyperparameters...

```
u = .01
v = .01
```

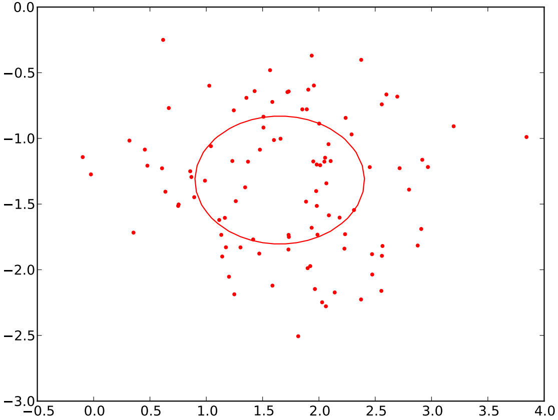
- Then start at nodes without parents...

```
pi = betaRand(u, v)
>>> 1.5235197013120821e-14
```

- And continue until you reach the leaves

Once you can sample data...

Generate a few datasets and check their empirical statistics:



No blue points? What's going on?

Sensitivity to hyperparameters

Model error: Bad hyperparameter values

- Your data doesn't look the way you expect:
- Statistics reach extreme values...
  - Like one cluster getting all the points
- ...or don't spread out enough
  - Like all the cluster means grouping around the origin
- A common problem with sparse priors
  - Like stick-breaking, Chinese restaurant, etc
  - A little sparsity is good...
  - But large clusters grow more attractive
  - ... and can snowball quickly!

Fixing the problem

Find (and type in) better values!

- It's usually best to move closer to uniform

```
#uniform beta prior
u = 1
v = 1
pi = betaRand(u, v)
>>> 0.1943
```

Or you could try sampling the hyperparameters...

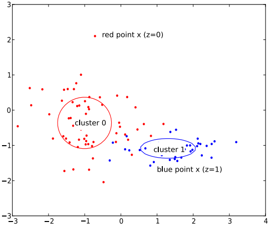
- This lessens the impact of your decisions
- But prevents you from expressing your prior beliefs

Do you recover the parameters?

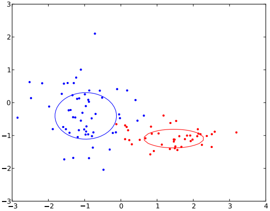
Ideally, you will get close to the truth:

- Often, you can't visualize the parameters...
- But you can check a few by eye...
- Compute statistics:
  - Rand distance and other clustering metrics
    - (Meila '03)
  - KL divergence
- Or project into 2d using MDS

Truth:



Sampled:



Identifiability

Did you notice that the model switched the red and blue clusters?

Model error: Non-identifiability : many parameterizations of your model define the same distribution

For instance, switching the red cluster and the blue cluster does not change $p(x)$

- Most clustering models are non-identifiable in this way
- Can happen in other models too
  - Linear classifier with linearly dependent features
- Not a problem if you just care about densities
- But can make it tricky to check recovery of the parameters
- Or analyse their values (like classifier feature weights)...

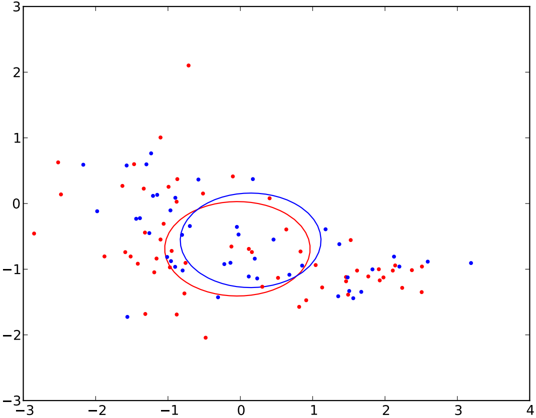
Forcing identifiability

To make your model identifiable:

- Eliminate redundant parameters
  - Can be difficult
- OR Break symmetries in the prior
  - For instance, set the mean of μ_0 further left...
  - Or restrict $\pi \geq .5$ to force red to be larger

Something's broken...

If you didn't recover the parameters... how can you tell what's wrong?



Overview

Preliminaries: example model and inference

Synthetic data

Analysing likelihood

Search errors

Plotting the likelihood: step 1

Calculate the likelihood of your sampled dataset.

- This is straightforward:
- Every time you make a random decision...
  - Calculate its probability

```
logLikelihood = 0
#uniform beta prior
u = 1
v = 1
pi = betaRand(u, v)
logLikelihood += log(betaPdf(pi, u, v))
```

Plotting the likelihood: step 2

Now calculate the joint likelihood of the state of your sampler.

- This works *mostly* as above...
- But there's a caveat!

Plotting the likelihood: step 2

Now calculate the joint likelihood of the state of your sampler.

- This works *mostly* as above...
- But there's a caveat!

Math error: Integrating out parameters creates dependence!

If you're using collapsed Gibbs, you probably use:

$$z_i \sim P(z_i \mid x_i, z_{-i}; \mu, \Sigma)$$

You may be tempted to follow up with:

$$\mathit{logLikelihood} \mathrel{+}= \mathit{log}\ P(z_i \mid x_i, z_{-i}; \mu, \Sigma)$$

This is wrong!

Integrating out a parameter

$$P(z|x,\mu,\Sigma,a,b) \propto P(x|z,\mu,\Sigma)P(z|a,b)$$

Integrating out a parameter

$$P(z|x, \mu, \Sigma, a, b) \propto P(x|z, \mu, \Sigma) P(z|a, b)$$

$$
\begin{aligned}
P(z|a, b) &= \int_\pi P(z|\pi) P(\pi|a, b) d\pi && \text{(def. of the model)} \\
&\propto \int_\pi \pi^{a+\#(z=0)} (1-\pi)^{b+\#(z=1)} d\pi && \text{conjugacy}
\end{aligned}
$$

Integrating out a parameter

$$P(z|x,\mu,\Sigma,a,b) \propto P(x|z,\mu,\Sigma)P(z|a,b)$$

$$\begin{aligned} P(z|a,b) &= \int_\pi P(z|\pi)P(\pi|a,b)d\pi && \text{(def. of the model)} \\ &\propto \int_\pi \pi^{a+\#(z=0)}(1-\pi)^{b+\#(z=1)}d\pi && \text{conjugacy} \end{aligned}$$

By definition:

$$Beta(x;c,d) = \frac{\Gamma(c+d)}{\Gamma(c)\Gamma(d)}x^{c-1}(1-x)^{d-1}$$

$$\therefore \int_x x^{c-1}(1-x)^{d-1}dx = \frac{\Gamma(c)\Gamma(d)}{\Gamma(c+d)}$$

Integrating out a parameter

$$P(z|x,\mu,\Sigma,a,b) \propto P(x|z,\mu,\Sigma)P(z|a,b)$$

$$\begin{aligned}
P(z|a,b) &= \int_\pi P(z|\pi)P(\pi|a,b)d\pi && \text{(def. of the model)} \\
&\propto \int_\pi \pi^{a+\#(z=0)}(1-\pi)^{b+\#(z=1)}d\pi && \text{conjugacy}
\end{aligned}$$

By definition:

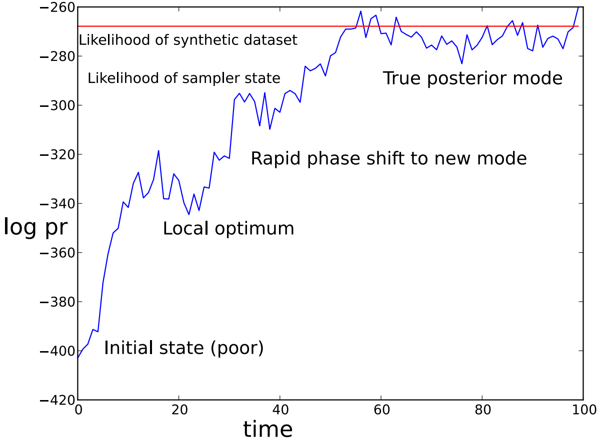
$$Beta(x;c,d) = \frac{\Gamma(c+d)}{\Gamma(c)\Gamma(d)} x^{c-1}(1-x)^{d-1}$$

$$\therefore \int_x x^{c-1}(1-x)^{d-1}dx = \frac{\Gamma(c)\Gamma(d)}{\Gamma(c+d)}$$

Then choose $c = a + \#(z=0)+1$, $d = a + \#(z=1)+1$

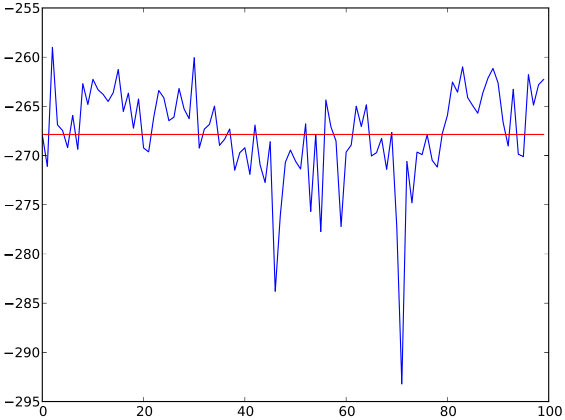
Plotting your likelihood

The likelihood plot usually looks like this:



Plotting your likelihood

But a correct sampler could produce plots like this too.
It's hard to tell expected oscillation from errors by eye.
This happens if the posterior is flat or you start near a mode.



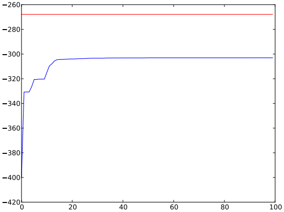
Greedy MCMC

Does the likelihood oscillate because of stochasticity?
Or is it just **broken**?

Greedy MCMC

Replace stochastic acceptance rule with:

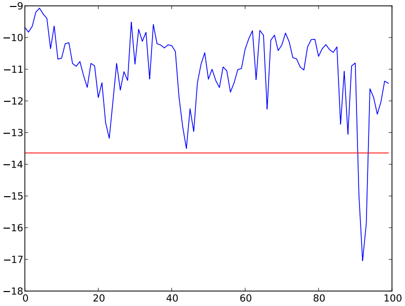
- Metropolis-Hastings: accept if $p_{new} > p_{old}$
- Gibbs: $z_i \leftarrow argmax p(z_i | z_{-i})$
- **Prone to local maxima; don't use in practice**



Diagnosing errors from the likelihood plot

Likelihoods significantly *above* truth:

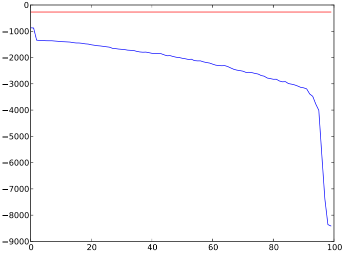
- Not enough data– due to variance, the posterior mode is far from truth (actually how I made this plot)
- OR **Model error: Non-identifiability**
- OR **Math error: Computing the likelihood wrong**



Diagnosing errors from the likelihood plot (2)

Likelihoods going *down*:

- No good reason for this– it HAS to be a bug
- **Math error: Recheck your derivations**
- **Code error: Did you flip a sign? Invert a ratio?**



- This plot:
- The Metropolis ratio: $A = \frac{p(x_{new})}{p(x_{old})}$
- Not the Metropolis ratio! $A = \frac{p(x_{old})}{p(x_{new})}$

Isolating the error

Unit test sampling for one variable

Fix the other variables to their true values.
Sample the target...
Check the parameters and likelihood.

For instance:

- Fix μ, Σ and sample z
- Or even fix $\mu, \Sigma, z_{0..n-1}$ and sample z_n

Always worth checking– even if the joint likelihood is going up, individual components could still be broken.

Overview

Preliminaries: example model and inference

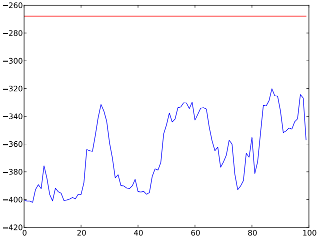
Synthetic data

Analysing likelihood

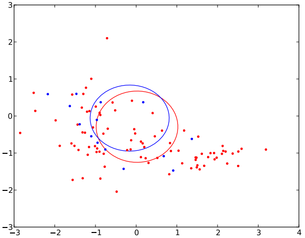
Search errors

Bad, but plausible solutions

Likelihood:



Parameters:



- **Could just be a bug, or...**
- **Search error: Local maximum**
- **Search error: Slow convergence (not mixing)**

Local maxima

Search error: Local maximum

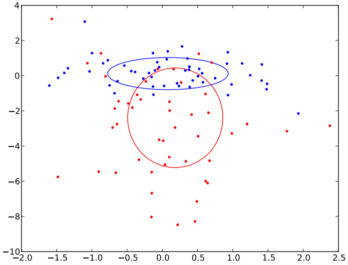
MCMC is just local search... it can get stuck in a posterior mode

- In the *infinite limit* it always escapes
- But you can't wait that long!
- Types of moves and proposals affect how long it takes

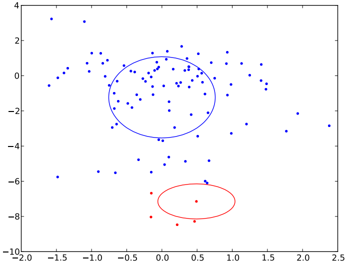
A local maximum

Deliberately caused by terrible initialization

True solution:



Local max:



Testing for a local max

Test for local maxima by:

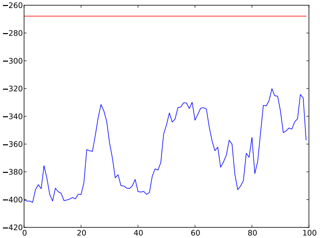
- Checking your initialization
- Trying multiple datasets
- Reducing the amount of data (flattens the posterior)
- Running for longer
  - No easy way to predict escape time though

Fixing the problem

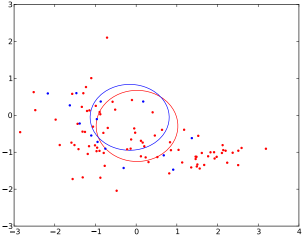
- Easy: fix the initializer
  - Ex: set z_i uniformally $0/1$, both clusters standard normal
  - Avoid saddle points/maxima
  - (Try to break symmetries)
  - Put parameters somewhere near plausible values
  - Can set incrementally (sequential sampling)
- Easy: some form of annealing
  - Replace $x \sim p(x)$ with $x \sim p(x)^t$
  - Decrease t at each iteration
  - $t >> 1$ flattens the initial posterior a lot
- Harder: block or collapsed sampling
  - Gives longer-distance moves
- Hard: complex MH proposals
  - Like cluster split-merge

Convergence problems apart from maxima

Likelihood:



Parameters:



Search error: Stuck near initial position

Parameters don't explain the data

Metropolis-Hastings acceptance ratio

Acceptance ratio

Number of times proposal accepted / Number of samples

- All rejections: no mobility
- More tricky to see why all acceptances is bad
  - It isn't always (Gibbs)
  - But proposal $x_{new} = x_{old}$ also always accepts!
  - Can signal low exploration
- Folk wisdom: good ratio is $\sim \frac{2}{3}$

Our MH algorithm: closer look

Using symmetric (random-walk) proposal on each coordinate i of μ and Σ:

$$\mu^i_{new} \sim N(\mu^i_{old}, \sigma_q)$$

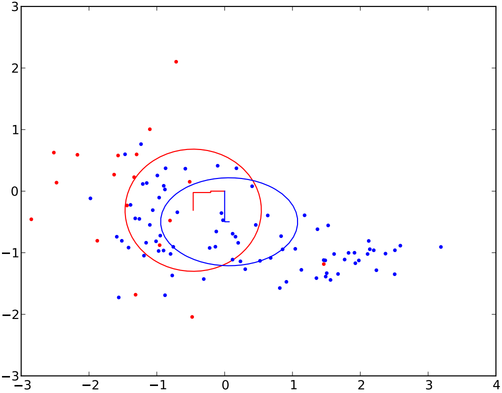
(q term in MH ratio cancels)
Performance depends on σ_q

Proposal explores too widely

Large σ_q:

- Acceptance ratio for means .022
- Acceptance ratio for variances .015

(Lines show position of means throughout sampling run; the means take long steps, but not very often)

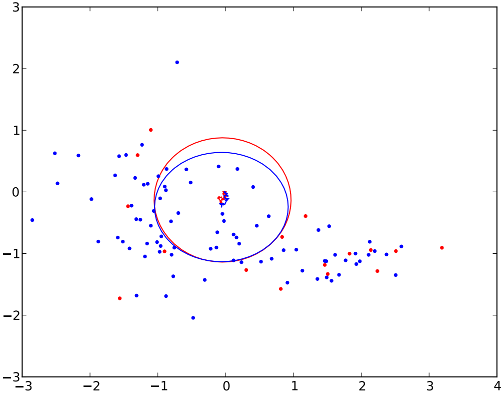


Proposal is too conservative

Small σ_q:

- Acceptance ratio for means 95.5
- Acceptance ratio for variances 82.5

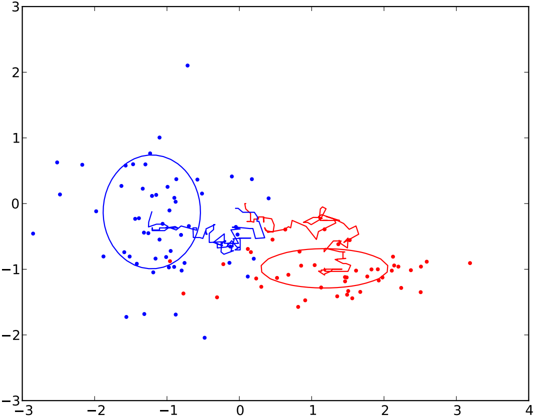
(Lines show position of means throughout sampling run; the means take many steps, but don't move far enough)



Reasonable proposal

Medium σ_q:

- Acceptance ratio for means 74.8
- Acceptance ratio for variances 47

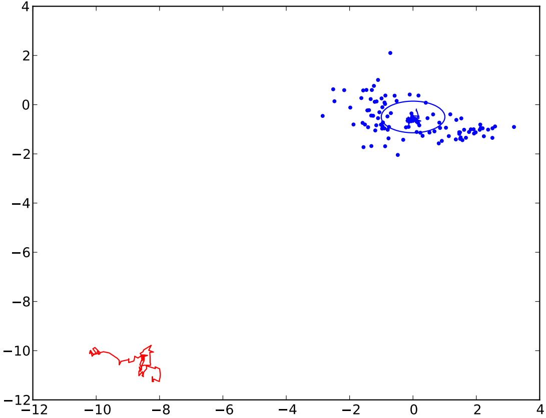


Initialization is still important!

Random walk proposals fail unless there is a strong gradient in the likelihood ratio

(In other words, the model should care a lot about parameter differences near the current point)

Otherwise, **you will just wander at random**



Conclusions

- Control your environment
  - Sample datasets
  - Fix variables to their true values
  - Replace stochasticity with greed

Conclusions

- Control your environment
  - Sample datasets
  - Fix variables to their true values
  - Replace stochasticity with greed
- Make sure what you expect to happen is happening
  - Likelihood increases
  - True likelihood is maximal
  - Parameters are recovered

Acknowledgements

Thanks to Dae-Il Kim and Deepak Santhanam...
To Sharon Goldwater, Dan Grollman, Tom Griffiths, David McClosky, Stefan Roth and Frank Wood for teaching me MCMC in the first place...
And to the Brown MLRG.