Sampling 2 Markov Chain Marte Carlo (MCMC)

Sampling 1 Summary

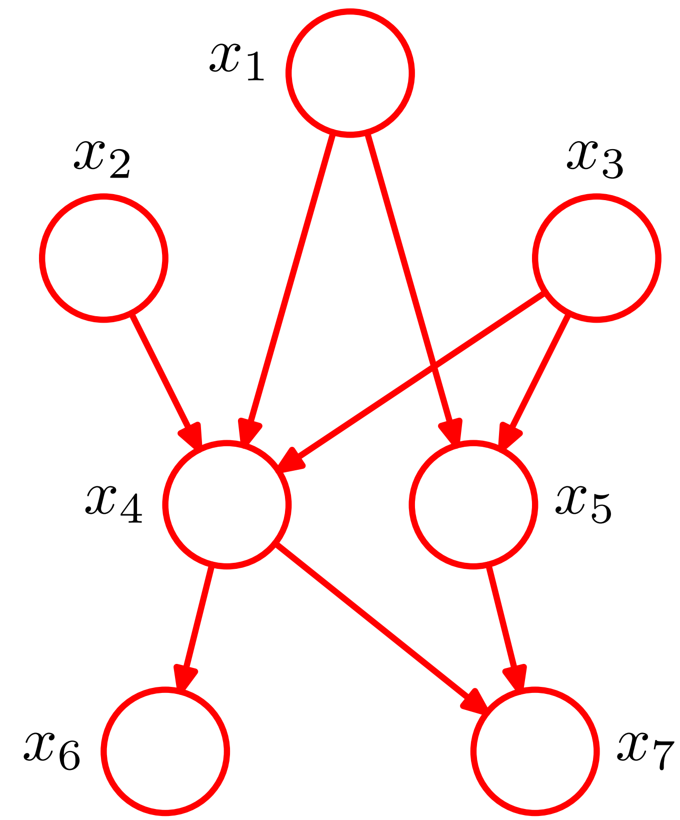
One dimensional pdfs with inverses can be sampled

Factored distributions can be sampled <movies>

Ancestral sampling

Faced with sampling from $p(\mathbf{x})$

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$



E.g.
Sampling x_1, x_2 and x_3
allows the sampling of x_4

So if we have a way of sampling the conditional distributions,
then we can use the graph structure to drastically reduce the work

Sampling Lecture 1 Summary

Factored distributions can be sampled : Ancestral sampling

One dimensional pdfs with inverses can be sampled

The trick:One can use the *uniform distribution* to create closed form algebraic formulas

Gaussians can be sampled

Use the trick to sample 2d then transform

But what about arbitrary distributions?

There are methods for the case where the distribution is known up to a scaling factor

Rejection sampling is simple but expensive for his dimensional pdfs

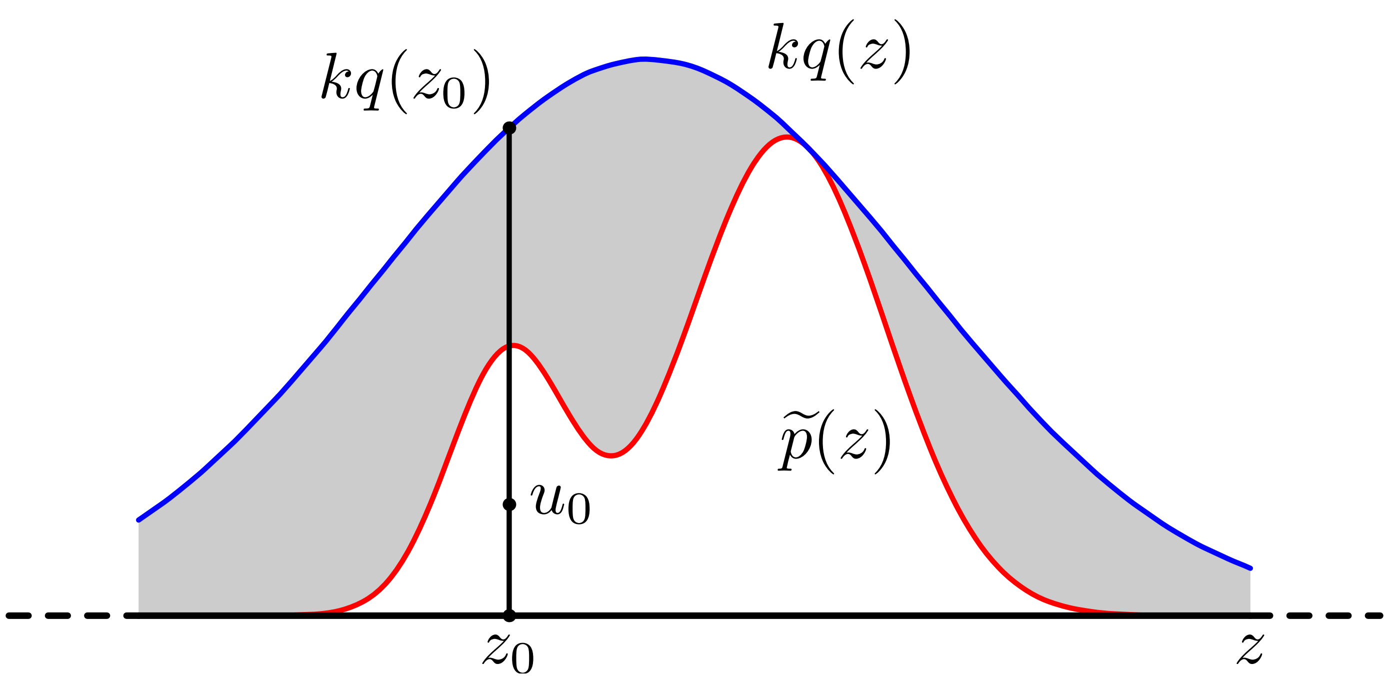
Because almost all samples are rejected

Importance sampling takes advantage of computing functions but expensive for his dimensional pdfs

Because most samples are either unimportant or low probability

Rejection Sampling

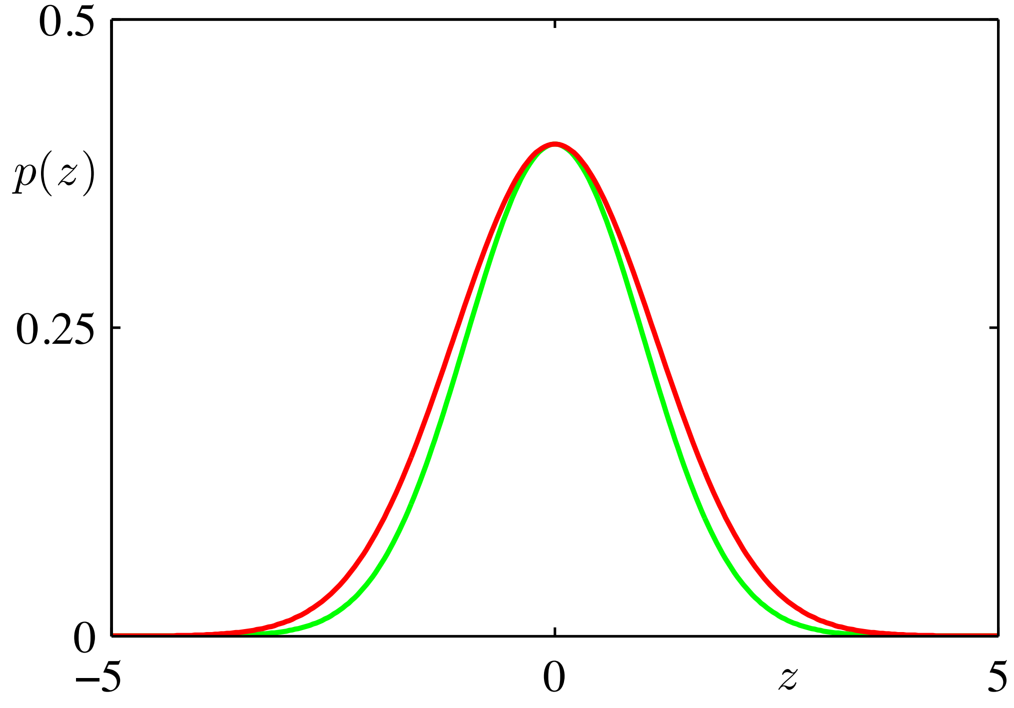
Know $\hat{p}(z)$ but don't know the normalization factor



$$
\begin{aligned}
p(\text{accept}) &= \int \{\widetilde{p}(z)/kq(z)\}\, q(z)\, \mathrm{d}z \\
&= \frac{1}{k}\int \widetilde{p}(z)\, \mathrm{d}z.
\end{aligned}
$$

The curse of dimensionality revisited

Example: Rejection sampling



Sampling Lecture 2
Markov Chain Monte Carlo (MCMC)

A very general framework that allows
sampling from a large class of distributions that scales well
with the dimensionality of the sample space

Previously the samples were independent, but now …

1. Keep track of samples $\mathbf{z}^{(\tau)}$
2. Proposal distribution $q(\mathbf{z}|\mathbf{z}^{(\tau)})$ depends on current state
3. The samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)} \ldots$ form a *Markov Chain*

Metropolis algorithm

Assume symmetry i.e. $q(\mathbf{z}_a|\mathbf{z}_b) = q(\mathbf{z}_a|\mathbf{z}_b)$

Candidate sample $\mathbf{z}^*$ is accepted with probability

$$A(\mathbf{z}^*, \mathbf{z}^\tau) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*}{\tilde{p}(\mathbf{z}^\tau}\right)$$

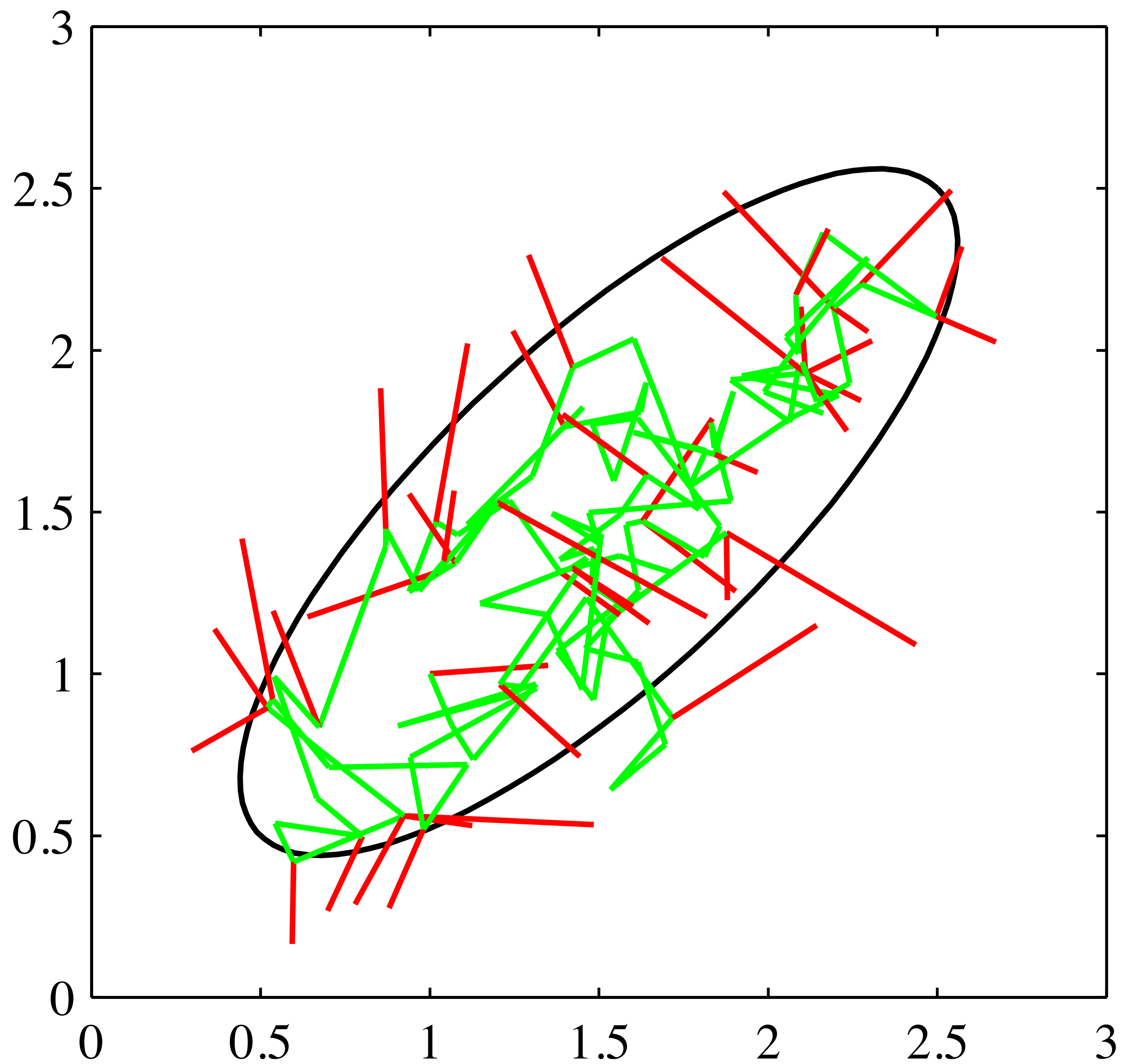
How to do this?

If accept $\quad \mathbf{z}^{(\tau+1)} = \mathbf{z}^*$

Else $\quad \mathbf{z}^{(\tau+1)} = \mathbf{z}^\tau$

Consequences ?

Metropolis Algorithm sampling a Gaussian



Proposal distribution is
symmetric N(0,0.2)

Proposal distribution is
is centered on the last sample

So it moves with the Markov process

Markov Chains

Markov property $$p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(m)}) = p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)})$$

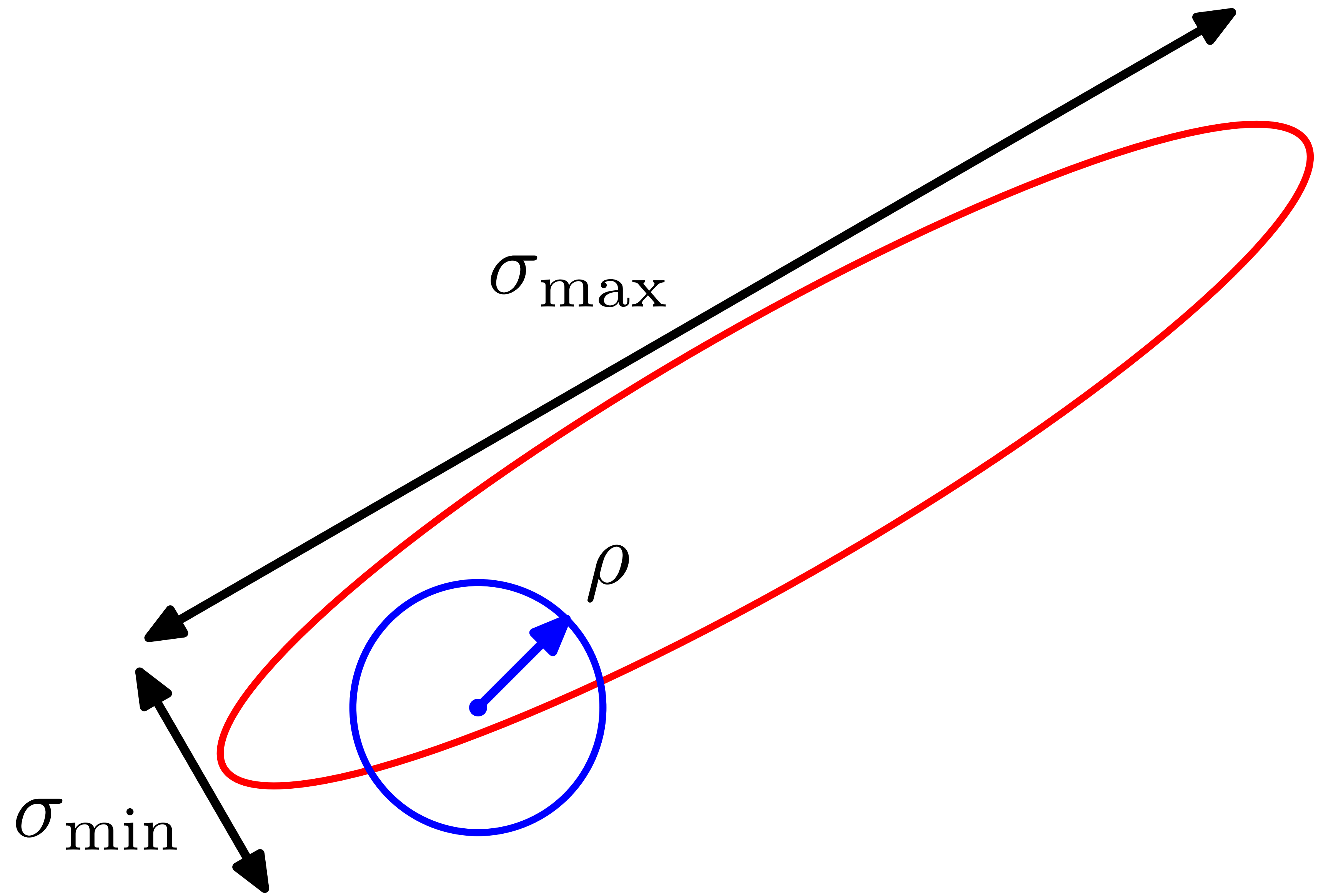
Marginal Probability $$p(\mathbf{z}^{(m+1)}) = \sum_{\mathbf{z}^{(m)}} p(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)})p(\mathbf{z}^{(m)})$$

Invariant $$p^{\star}(\mathbf{z}) = \sum_{\mathbf{z}'} T(\mathbf{z}',\mathbf{z})p^{\star}(\mathbf{z}').$$

Detailed balance:
sufficient condition for invariance $$p^{\star}(\mathbf{z})T(\mathbf{z},\mathbf{z}') = p^{\star}(\mathbf{z}')T(\mathbf{z}',\mathbf{z})$$

It works because… $$\sum_{\mathbf{z}'} p^{\star}(\mathbf{z}')T(\mathbf{z}',\mathbf{z}) = \sum_{\mathbf{z}'} p^{\star}(\mathbf{z})T(\mathbf{z},\mathbf{z}') = p^{\star}(\mathbf{z})\sum_{\mathbf{z}'} p(\mathbf{z}'|\mathbf{z}) = p^{\star}(\mathbf{z}).$$

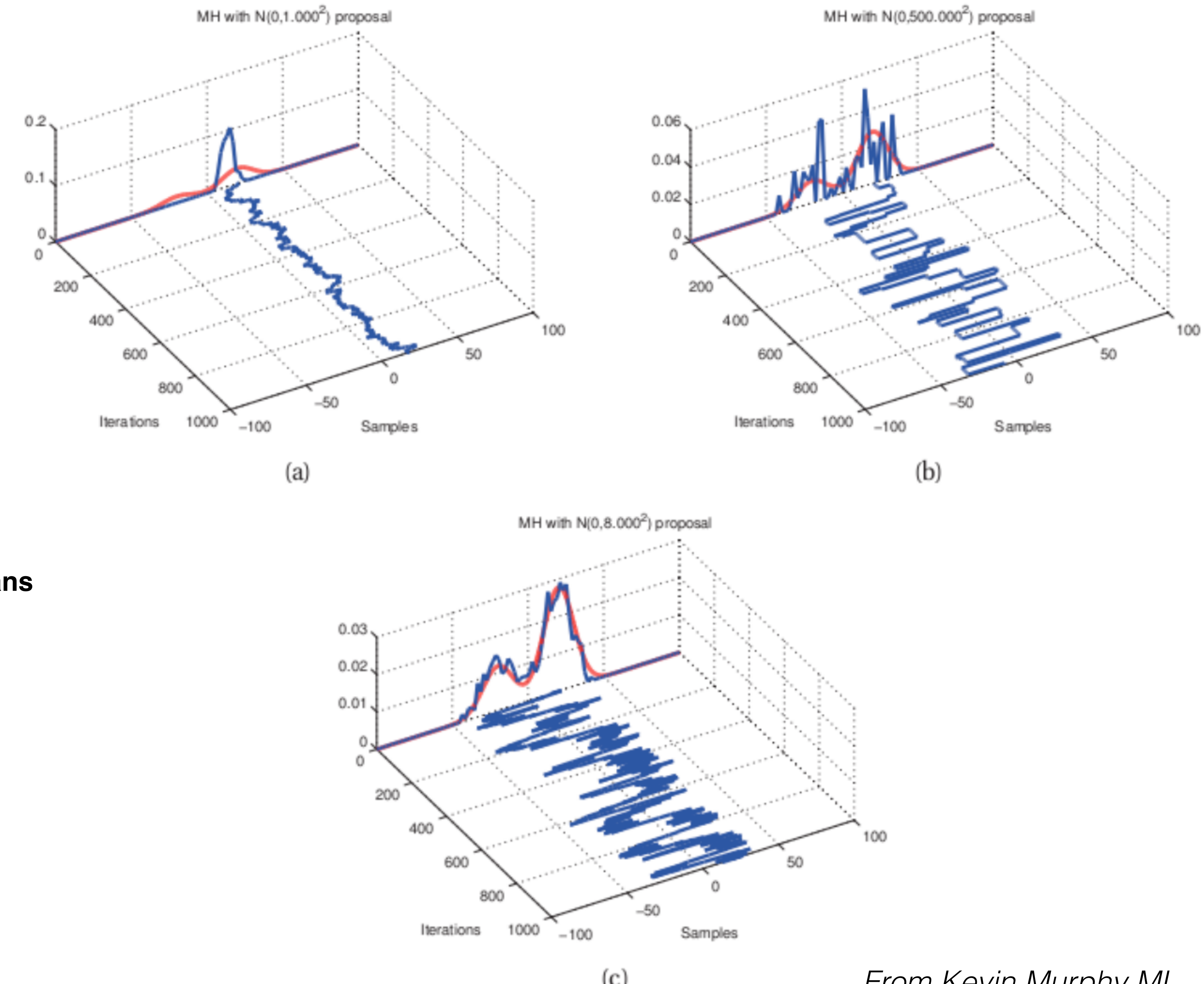
Issues when sampling with Metropolis-Hastings



If proposal Gaussian is too small, it takes a long time to walk elongated distribution

If proposal Gaussian is too large, many proposed samples are rejected

Experiments sampling from a sum of two Gaussians



Distribution to be sampled:
A weighted sum of two gaussians

$$\mu = \{-20, 20\}$$

$$\pi = \{0.3, 0.7\}$$

$$\sigma = \{100, 100\}$$

From Kevin Murphy ML

Metropolis Hastings Algorithm

When proposal distribution is not symmetric have to make adjustments

$$A_k(\mathbf{z}^\star, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\widetilde{p}(\mathbf{z}^\star) q_k(\mathbf{z}^{(\tau)}|\mathbf{z}^\star)}{\widetilde{p}(\mathbf{z}^{(\tau)}) q_k(\mathbf{z}^\star|\mathbf{z}^{(\tau)})}\right)$$

Make identifications

$$\mathbf{z} = \mathbf{z}^{(T)}$$

$$\mathbf{z}' = \mathbf{z}^\star$$

$$\begin{aligned} p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z})A_k(\mathbf{z}', \mathbf{z}) &= \min\left(p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z}), p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}')\right) \\ &= \min\left(p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}'), p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z})\right) \\ &= p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}')A_k(\mathbf{z}, \mathbf{z}') \end{aligned}$$

With the appropriate identifications $p^\star(\mathbf{z})T(\mathbf{z}, \mathbf{z}') = p^\star(\mathbf{z}')T(\mathbf{z}', \mathbf{z})$

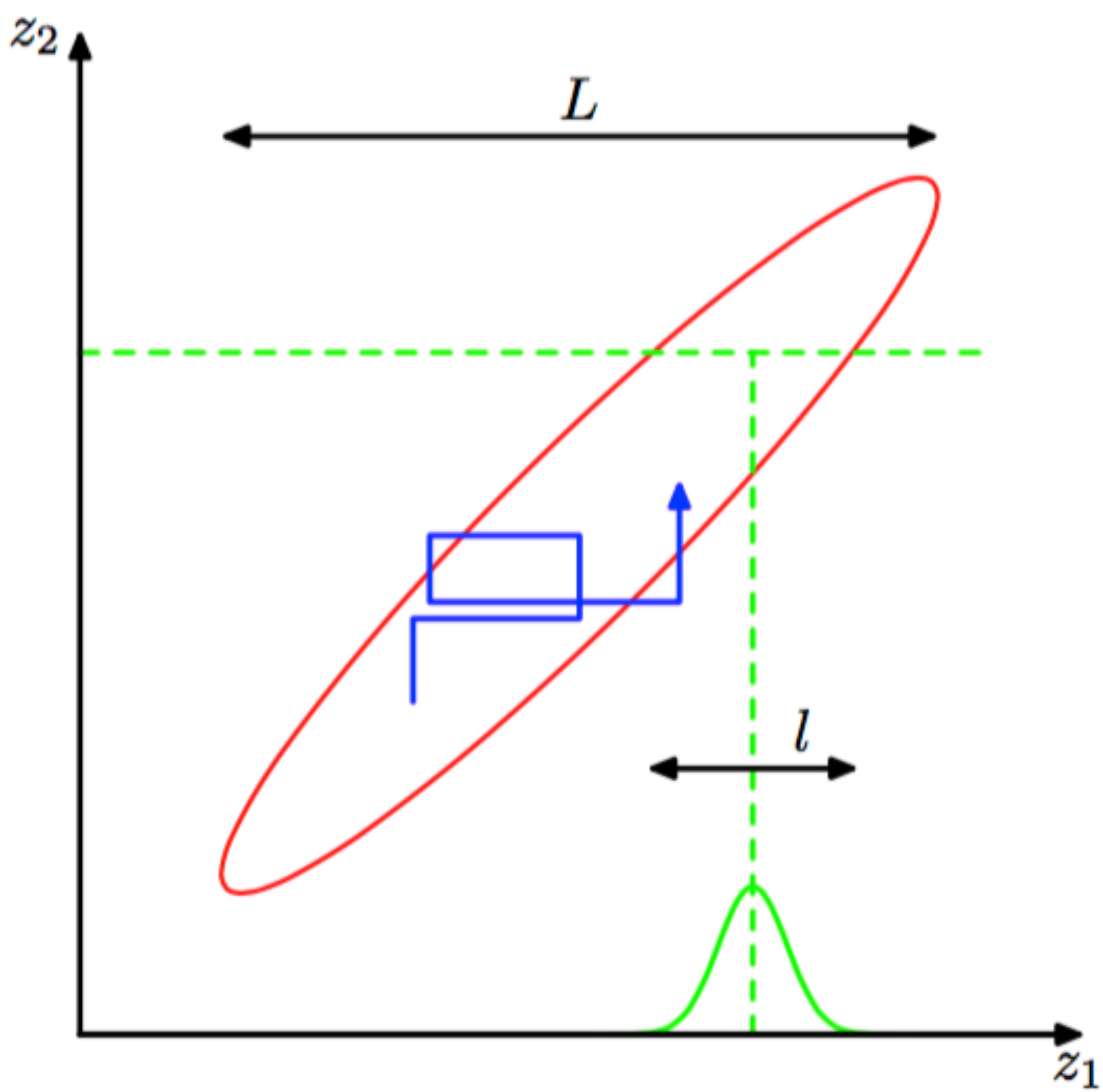
Why would we want an asymmetric proposal distribution?

Gibbs Sampling

1. Initialize $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
   - Sample $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - Sample $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.

     $\vdots$
   - Sample $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$.

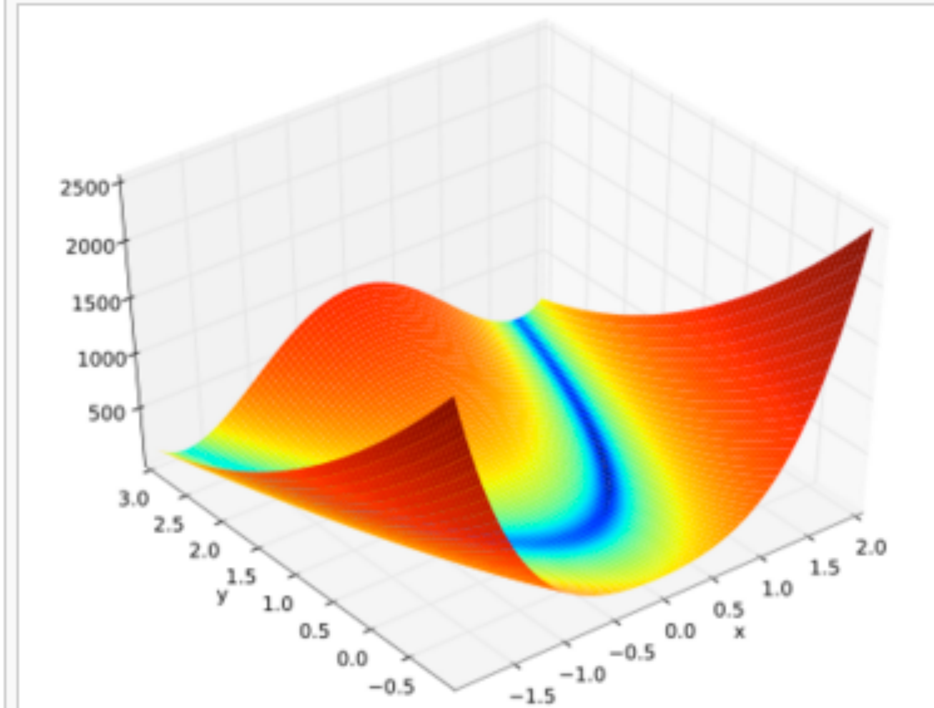
     $\vdots$
   - Sample $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$.


z_2
L
l
z_1

Rosenbrock function

From Wikipedia, the free encyclopedia

In mathematical optimization, the **Rosenbrock function** is a non-convex function used as a performance test problem for optimization algorithms introduced by Howard H. Rosenbrock in 1960.[1] It is also known as **Rosenbrock's valley** or **Rosenbrock's banana function**.



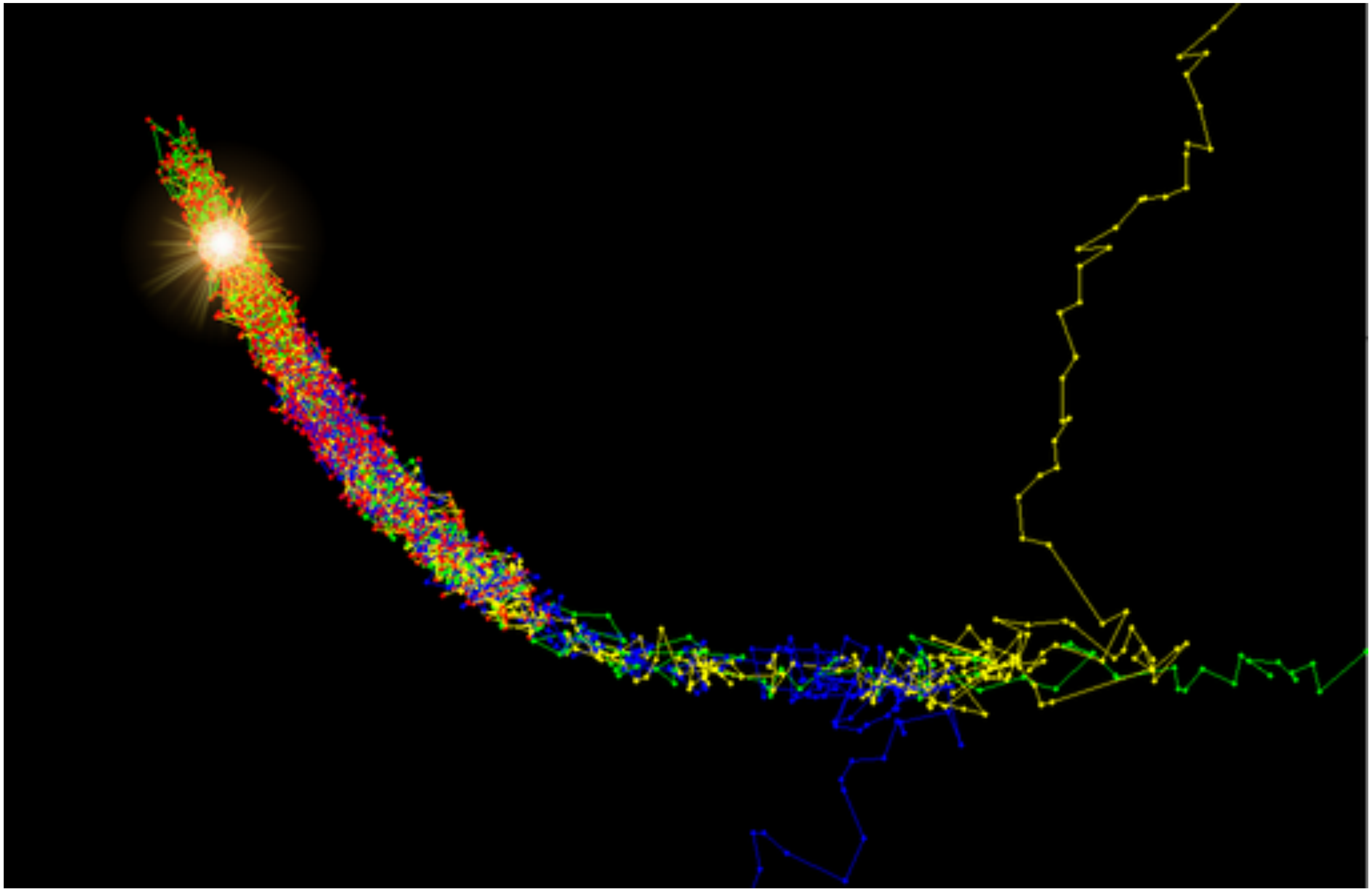
Plot of the Rosenbrock function of two variables.

The global minimum is inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial. To converge to the global minimum, however, is difficult.

The function is defined by

$$f(x,y) = (a-x)^2 + b(y-x^2)^2$$

It has a global minimum at $(x,y)=(a,a^2)$, where $f(x,y)=0$. Usually $a=1$ and $b=100$.



Over-relaxation for Gibbs

$$z_i' = \mu_i + \alpha(z_i - \mu_i) + \sigma_i(1 - \alpha_i^2)^{1/2}\nu$$

$$E[\nu] = 0 \qquad var[\nu] = \mathbf{I}$$

$$E[z_i^{'}] = ?$$

Metropolis Algorithm

Still need a proposal distribution q(**z**)

The candidate sample is then accepted with probability

$$A(\mathbf{z}^\star, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\widetilde{p}(\mathbf{z}^\star)}{\widetilde{p}(\mathbf{z}^{(\tau)})}\right).$$

Why does this work?

Homework

One and two dimensional simulations of the Metropolis Algorithm using Gausian mixtures as the target distribution. Given the distribution

$$p(x) = 0.3N(-25, 10) + 0.7N(20, 10)$$

where $N(\mu, \sigma)$ is a Gaussian with mean μ and standard deviation σ,

1. Implement the Metropolis algorithm for sampling $p(x)$.
2. show the results of using correct and incorrect variances for your proposal distribution.
3. Extend your results to a two dimensional Gaussian mixture $p(\mathbf{x})$ with means μ and covaiance matrix $\mathbf{\Sigma}$ that you choose.
4. Try Gibbs sampling on the example in 3