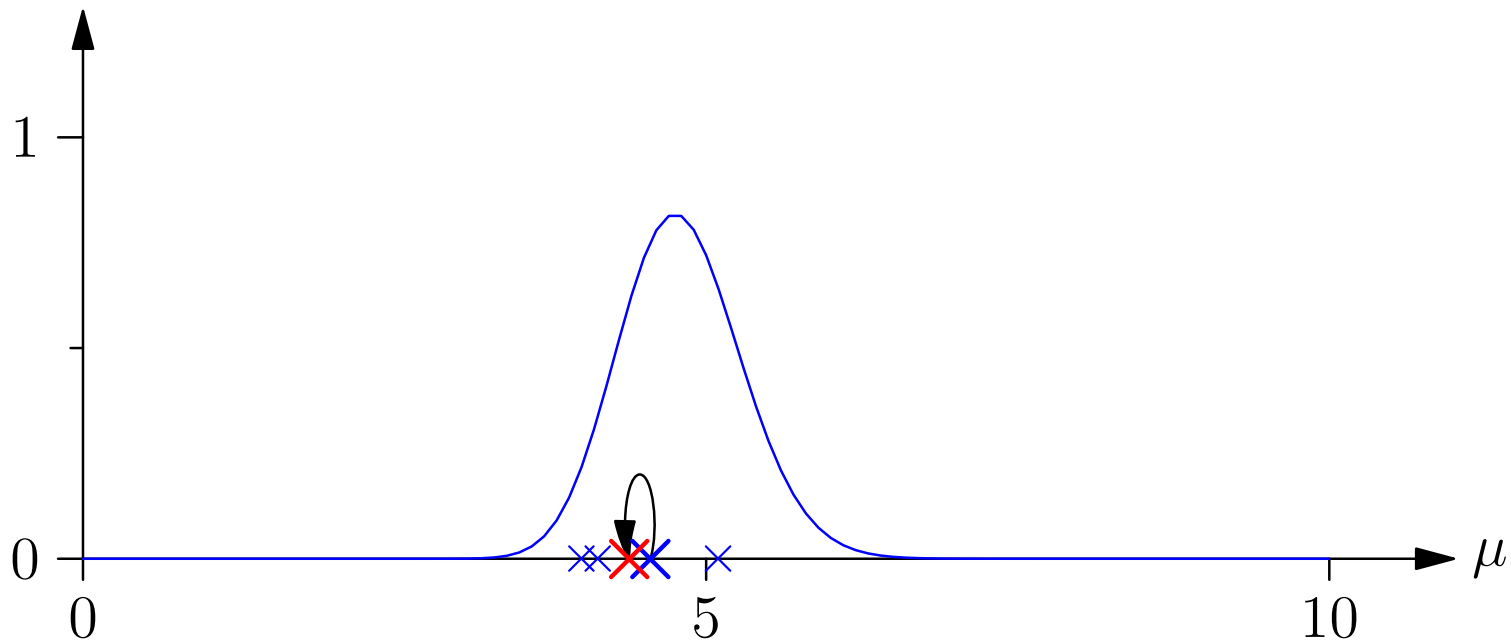


Advanced Machine Learning

MCMC

$$\mathcal{D} = \{4, 4, 6, 4, 2, 2, 5, 9, 5, 4, 3, 2, 5, 4, 4, 11, 6, 2, 3, 11\}$$

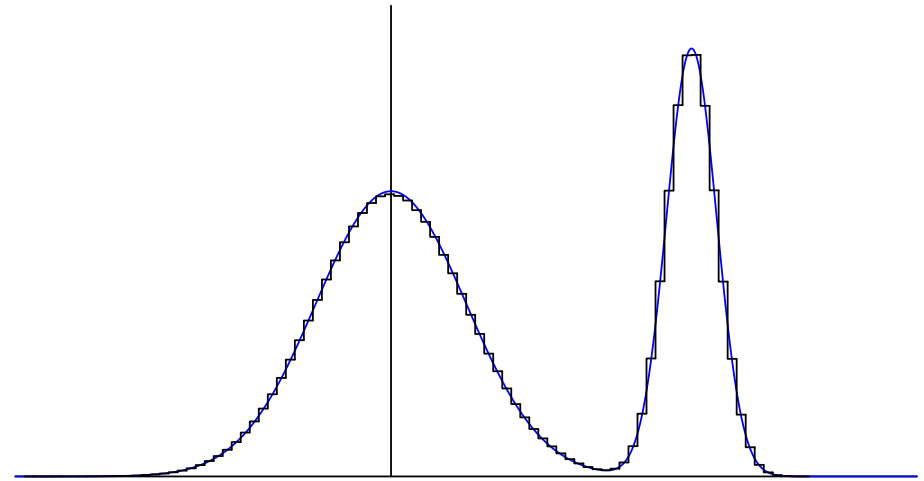


Monte Carlo methods, MCMC, Variational Methods

Outline

1. **Sampling**
2. Random Number Generation
3. MCMC

$T = 10000000$, acceptance rate = 0.897



Bayesian Inference Gets Hard

- We saw that in some cases if we had a simple likelihood (normal, binomial, Poisson, multinomial) you can choose a conjugate prior (gamma-normal/Wishart, beta, gamma, Dirichlet) so that the posterior has the same form as the prior■
- Very often we are working with more complex models where no conjugate prior exists■
- The posterior is not described by a known distribution■
- We have to work a lot harder—particularly with multivariate distributions■

Bayesian Inference

- Recall our problem is that we are given some data \mathcal{D} ■

- Our posterior is given by

$$\mathbb{P}(\boldsymbol{\theta}|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}|\boldsymbol{\theta}) \mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\mathcal{D})} \quad \text{or} \quad f(\boldsymbol{\theta}|\mathcal{D}) = \frac{f(\mathcal{D}|\boldsymbol{\theta}) f(\boldsymbol{\theta})}{f(\mathcal{D})}$$

- Where $\boldsymbol{\theta}$ are the parameters we are trying to infer■
- But our likelihood (and/or prior) might be quite complicated■
- Typically we don't have a closed form representation for our posterior distribution■

Histograms, Samples and Means

- We could represent our posterior as a histogram, although for multivariate distributions (i.e. when we are modelling more than one variable) a histogram can be unwieldy■
- A sample from the posterior distribution is often sufficient e.g. in our topic models (LDA) a typical set of topics is what we are after■
- However, when samples vary a lot, often the most useful quantities are expectation, e.g.

$$\mathbb{E}[\Theta]$$

$$\mathbb{E}[\Theta_i \Theta_j] - \mathbb{E}[\Theta_i] \mathbb{E}[\Theta_j]$$

$$\mathbb{E}[\Theta_i^2] - \mathbb{E}[\Theta_i]^2$$

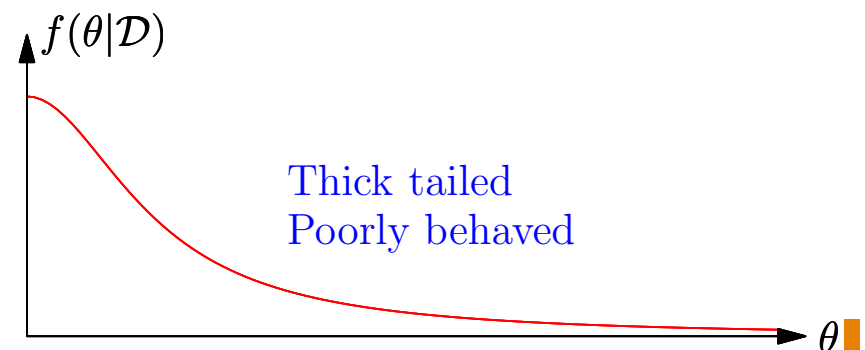
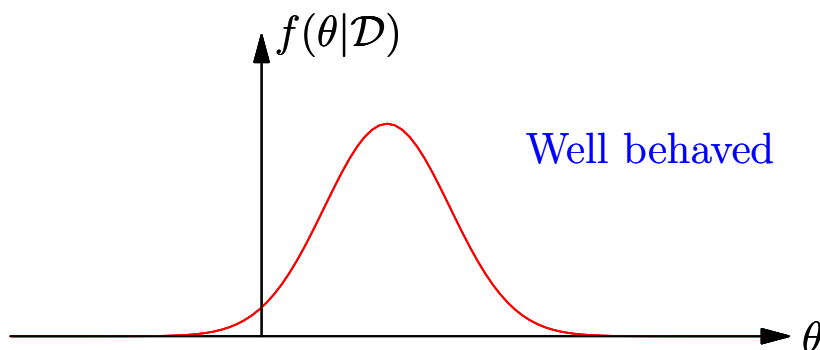
$$\mathbb{E}[\Theta \Theta^T] - \mathbb{E}[\Theta] \mathbb{E}[\Theta]^T \quad \blacksquare$$

Sample Estimation

- If we can draw independent **deviates** (aka **variates**), Θ_i , from our posterior distribution then we can obtain an estimate of our expectation

$$\mathbb{E}[g(\Theta)] \approx \frac{1}{n} \sum_{i=1}^n g(\Theta_i)$$

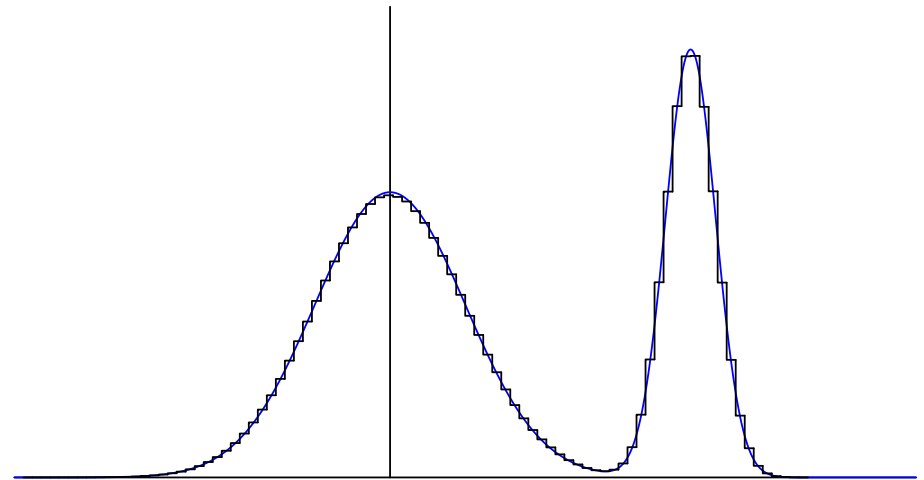
- Provided our posterior distribution is well behaved the relative error in our estimate will drop off as $1/\sqrt{n}$



Outline

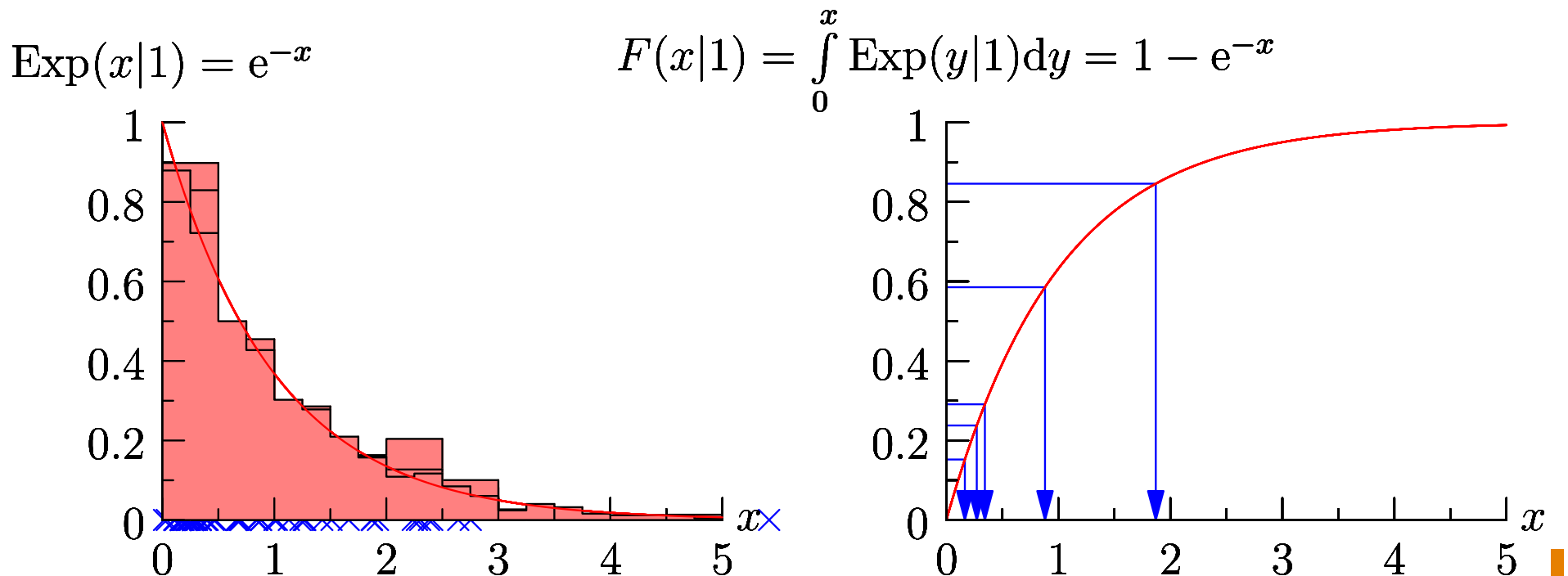
1. Sampling
2. **Random Number Generation**
3. MCMC

$T = 10000000$, acceptance rate = 0.897



Drawing Random Samples

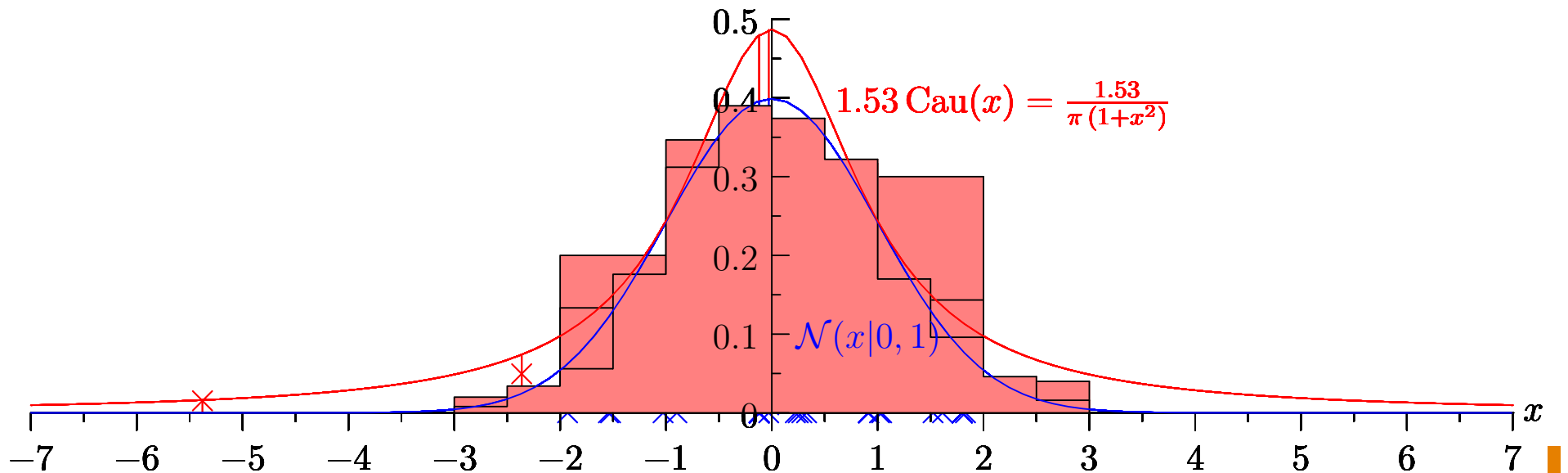
- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**
- For some very simple distributions we can use the **transformation methods** to transform a uniform distribution



Rejection Method

- The transformation method only works when we can easily compute the inverse *cumulative distribution function* (CDF)■
- A more general technique is the **rejection method** where we generate deviates from $g_Y(y)$ such that $cg_Y(x) \geq f_X(x)$ ■
- To draw deviates from $f_X(x)$ we draw a deviate $Y \sim g_Y$ and then accept the deviate with probability $f_X(Y)/(cg_Y(Y))$ ■
- The expected rejection rate is $c - 1$ ■
- Need to choose a good distribution $g_Y(y)$ ■

Drawing Normal Deviates



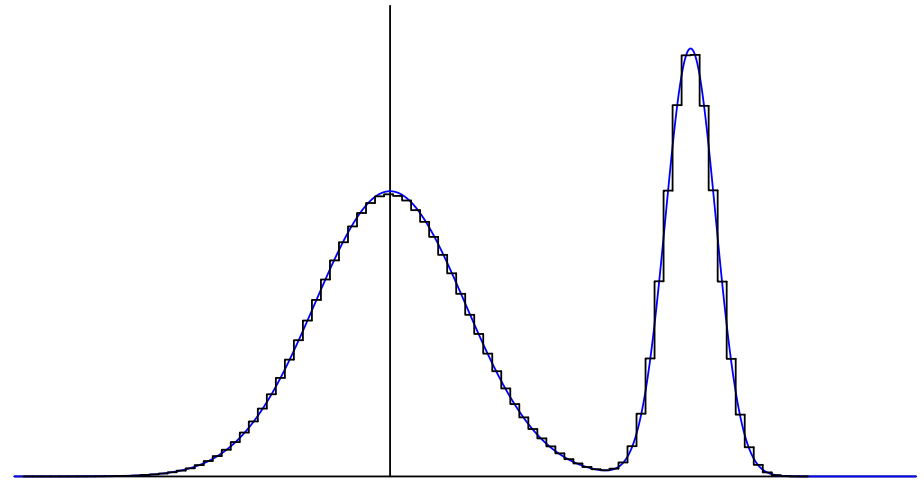
Problems with Rejection

- The rejection method is very general and often the method of choice (although for normal deviates there is a clever transformation method which is faster)■
- However, for complicated probability distributions it can be difficult to find a good proposal distribution $g_Y(y)$ ■
- This is particular true for multivariate distributions■
- If the proposal distribution is poor c might be very high and the number of rejections is stupidly high■

Outline

1. Sampling
2. Random Number Generation
3. **MCMC**

$T = 10000000$, acceptance rate = 0.897



Detailed Balance

- Suppose we have a set of states \mathcal{S} and want to draw sample from a probability distribution $\pi = (\pi_i | i \in \mathcal{S})$ ■
- We invent a dynamical system with a transition probability M_{ij} from state j to state i such that

$$M_{ij}\pi_j = M_{ji}\pi_i$$

- This is known as **detailed balance**■
- Summing both sides over j

$$\sum_j M_{ij}\pi_j = \sum_j M_{ji}\pi_i \quad \text{■} = \pi_i \quad \text{■} \quad \mathbf{M}\pi = \pi \quad \text{■}$$

Convergence of MCMC

- Suppose we start from a state $x(0) = \sum_i c_i v^{(i)}$ where the $v^{(i)}$'s are an eigenvectors of the transition matrix \mathbf{M} with eigenvalues λ_i
- If I apply \mathbf{M} many times then

$$x(t) = \mathbf{M}^t x(0) = \mathbf{M}^t \sum_i c_i v^{(i)} = \sum_i \lambda_i^t c_i v^{(i)}$$

- And $\lim_{t \rightarrow \infty} x(t) = v^*$ where v^* is the eigenvector with the maximum eigenvalue
- Now $\|\mathbf{M}v\|_1 \leq \|\mathbf{M}\|_1 \|v\|_1 = \|v\|_1$ so the maximum eigenvalue is 1 with eigenvector π (\mathbf{M} is known as a **stochastic matrix**)

Metropolis Algorithm

- A very easy way to achieve detailed balance is starting from state j choose a “neighbouring” state, i with equal probability■
- We accept the move if either
 - ★ $\pi_i > \pi_j$ or
 - ★ we make the move with a probability π_i/π_j ■
- If $\pi_i > \pi_j$ then $M_{ij} = 1$ and $M_{ji} = \pi_j/\pi_i$. Thus

$$M_{ij}\pi_j = \pi_j \quad M_{ji}\pi_i = \frac{\pi_j}{\pi_i}\pi_i = \pi_j$$

- Note that we require the state i to have the same number of neighbours as state j so that detailed balance is satisfied■

Continuous Variables

- If we are working with continuous variables θ then the equation for detailed balance for the transition probability $W(\theta \rightarrow \theta')$ is

$$W(\theta \rightarrow \theta')\pi(\theta) = W(\theta' \rightarrow \theta)\pi(\theta') \blacksquare$$

- where $\pi(\theta)$ is the probability distribution we wish to sample from \blacksquare
- The update rule is to choose a nearby value θ' , compute $r = \pi(\theta')/\pi(\theta)$ and accept the update with probability $\min(1, r)$ \blacksquare
- We require that the probability of choosing θ from θ' is the same as the reverse \blacksquare

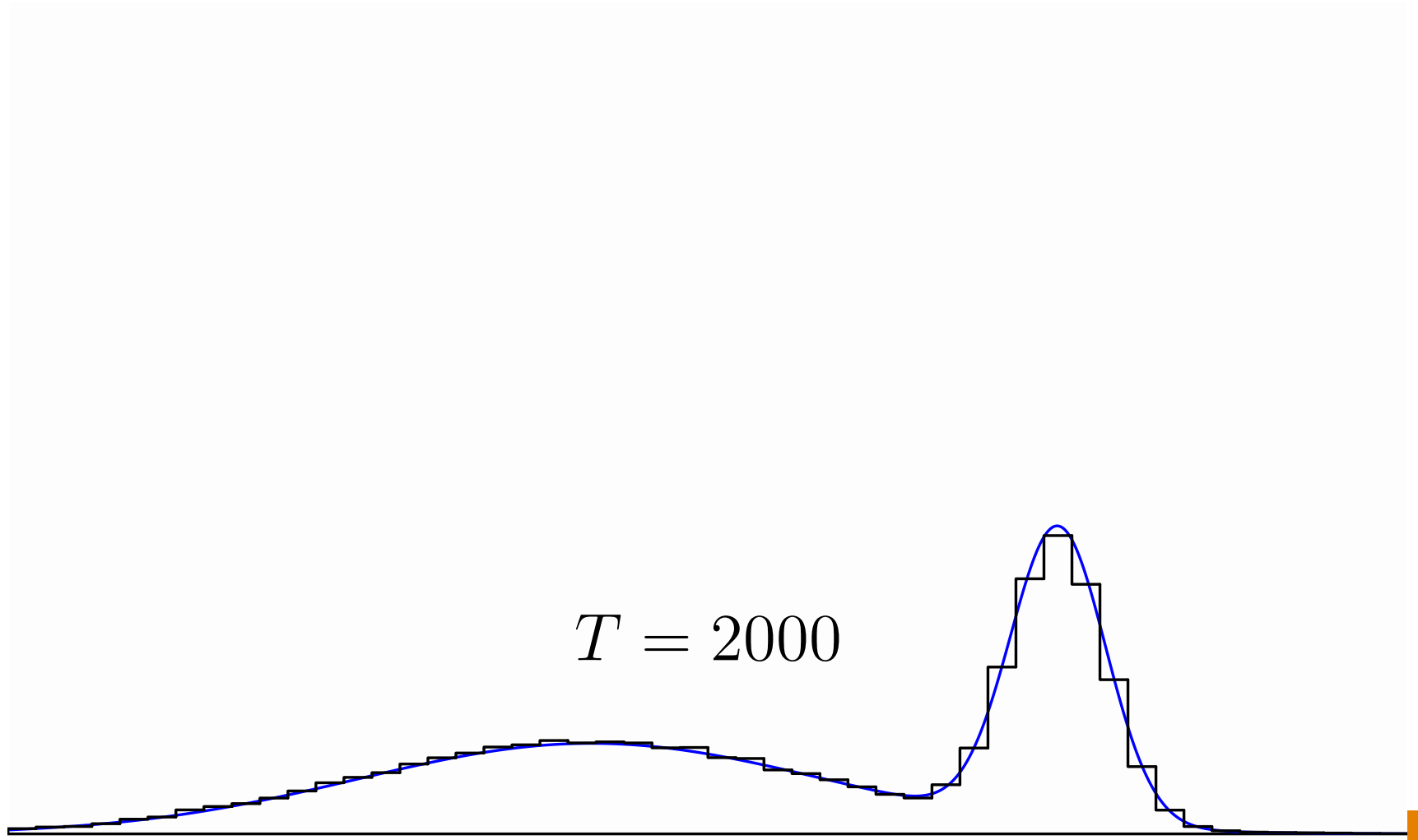
What Makes MCMC Nice

- Because we are free to choose where we move (and choose close by neighbours) $\pi(\boldsymbol{\theta}') \approx \pi(\boldsymbol{\theta})$ so that moves are not too infrequent■
- Also very importantly the updates depend only on the ratio $\pi(\boldsymbol{\theta}')/\pi(\boldsymbol{\theta})$ ■
- We only need to know our probabilities up to a multiplicative scaling factor■
- For sampling from the posterior we only need to know the likelihood and prior $\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})$ ■ (or $f(\mathcal{D}|\boldsymbol{\theta})f(\boldsymbol{\theta})$)■
- We don't need to know $\mathbb{P}(\mathcal{D})$ which we generally don't know■

What Makes MCMC Nasty

- It can take a long time until our states occur with the probability π (i.e. we have forgotten our initial state)■
- We don't even know how long we have to wait■
- Even when we have reached this *equilibration time* each sample is correlated with the previous sample■
- To get a good approximation to the posterior expectation requires running for many times the equilibration time■
- Note, if we are just finding sample averages then we can use all samples after equilibrating even if they are not independent■

Burn-In



Proposals and Metropolis-Hastings

- We have some freedom in choosing a new proposal θ' from our current position θ —a good choice can increase the acceptance rate making the MCMC more efficient■
- We define the proposal distribution $p(\theta'|\theta)$ ■
- For the standard Metropolis algorithm to work we require $p(\theta'|\theta) = p(\theta|\theta')$ ■
- In some cases (e.g when $\theta_i \geq 0$) this can be hard to achieve■
- We can modify our update rule to accept a move with probability

$$\min \left(1, \frac{p(\theta|\theta') f(\mathcal{D}|\theta') f(\theta')}{p(\theta'|\theta) f(\mathcal{D}|\theta) f(\theta)} \right) \blacksquare$$

Traffic Rate

- Consider monitoring the flow of traffic where we have data

$$\mathcal{D} = (N_1, N_2, \dots, N_n)$$

where N_i is the number of car that past on day i ■

- We assume $N_i \sim \text{Poi}(\mu)$ and want to infer μ ■
- The Poisson distribution has a beta conjugate prior■
- We don't have any prior knowledge on μ so we use a non-informative prior $\text{Gam}(\mu|0,0) = 1/\mu$ ■
- Note that we can solve this problem exactly—however, lets compare with MCMC■

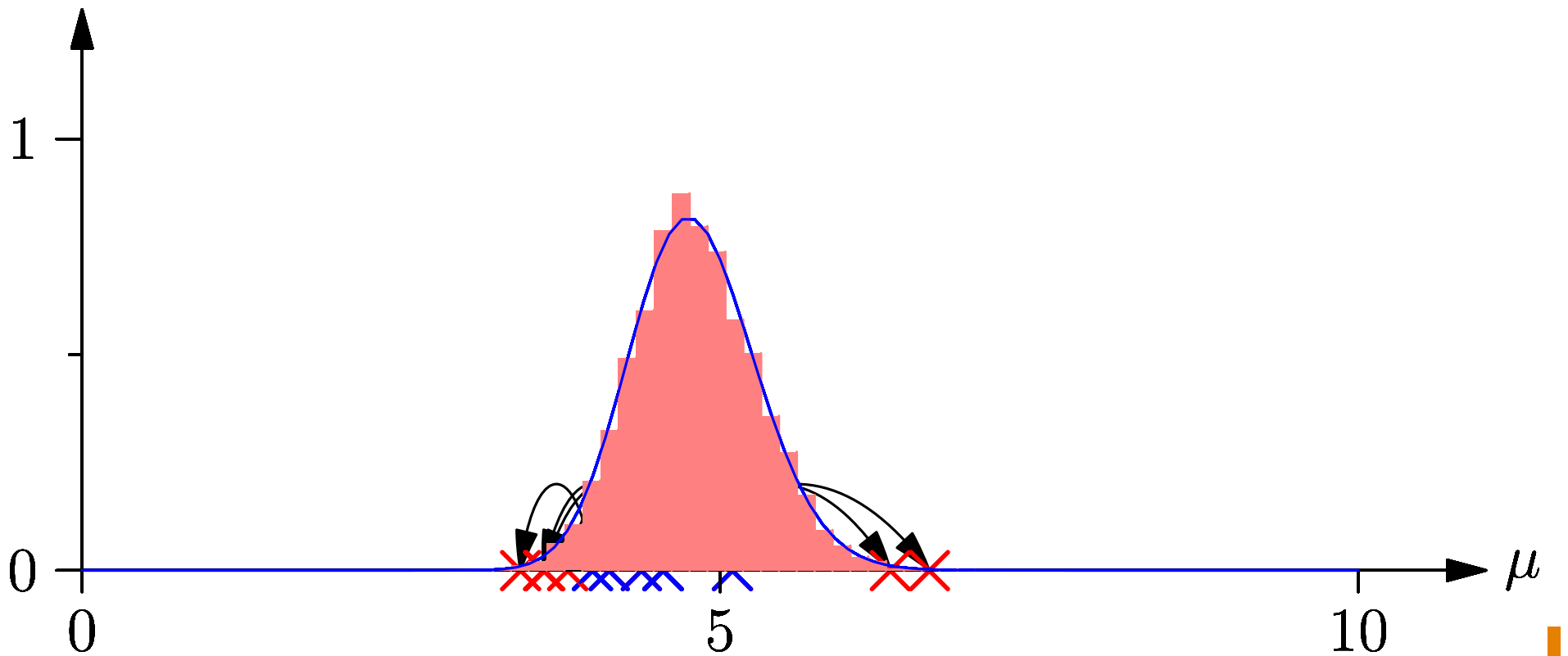
Proposal Distribution

- If we can choose our proposal distribution $p(\mu'|\mu)$ to be close to the posterior distribution then our acceptance rate would be close to 1■
- We choose $p(\mu'|\mu) = \text{Gam}(\mu'|\mu, \mu^2)$ which has $\mathbb{E}[\mu'] = \mu$ and variance 1■
- We update with probability $\min(1, r)$ where

$$\begin{aligned} r &= \frac{\text{Gam}(\mu|\mu'^2, \mu') \frac{1}{\mu'} \prod_{i=1}^n \text{Poi}(N_i|\mu')}{\text{Gam}(\mu'|\mu^2, \mu) \frac{1}{\mu} \prod_{i=1}^n \text{Poi}(N_i|\mu)} \blacksquare \\ &= \frac{\mu \text{Gam}(\mu|\mu'^2, \mu')}{\mu' \text{Gam}(\mu'|\mu^2, \mu)} e^{-n(\mu' - \mu) + \sum_{i=1}^n N_i \log\left(\frac{\mu'}{\mu}\right)} \blacksquare \end{aligned}$$

MCMC in Practice

$$\mathcal{D} = \{4, 4, 6, 4, 2, 2, 5, 9, 5, 4, 3, 2, 5, 4, 4, 11, 6, 2, 3, 11\}$$



MCMC Details

- To compute correct histograms you need to count samples where no move is made multiple times■
- On modern computers its quite quick to compute millions of samples■
- The code is not very difficult to write (although care is need to get everything correct)■
- This can be used on complicated problems such as topic models (LDA) with thousands of parameters■
- The accuracy of MCMC is slow if it takes a long time to sample the posterior distribution■

The MCMC Industry

- MCMC provides a means to accurately sample from very complex models■
- There have been many advanced techniques developed to improve MCMC performance■
- E.g. hybrid MCMC simulates a dynamics to find good proposals with similar probability far from the starting point■
- Often it seems that MCMC is complicated because there are so many optimisations, but often simple implementations are sufficient■

Conclusions

- As soon as we use complex models we are no longer able to compute the posterior in closed form■
- Monte Carlo techniques and particularly MCMC are a very general method for computing samples from the posterior■
- These techniques have been highly developed, but very frequently even simple implementations are sufficient to do good inference■
- Variational methods provide an approximate closed form solution to problems with complex likelihoods■
- Variational methods are mathematically challenging, but are potentially far faster to compute than MCMC■