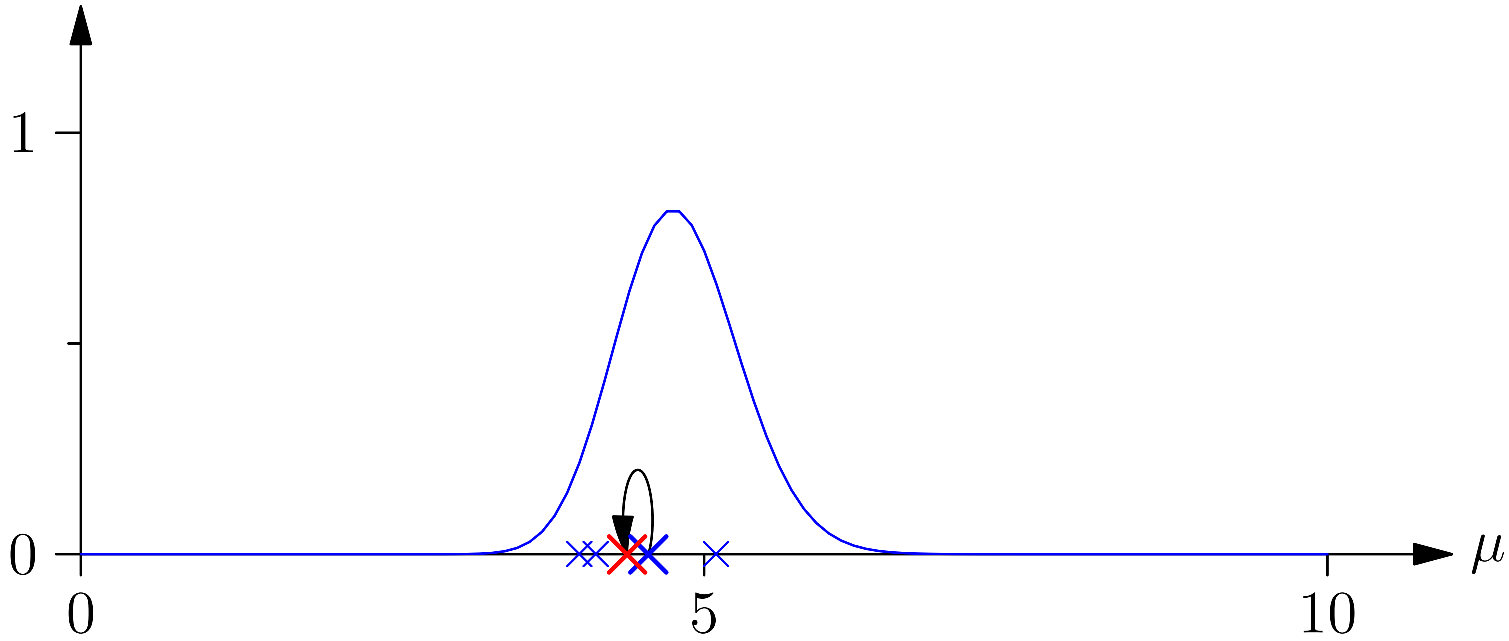
Advanced Machine Learning

MCMC

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

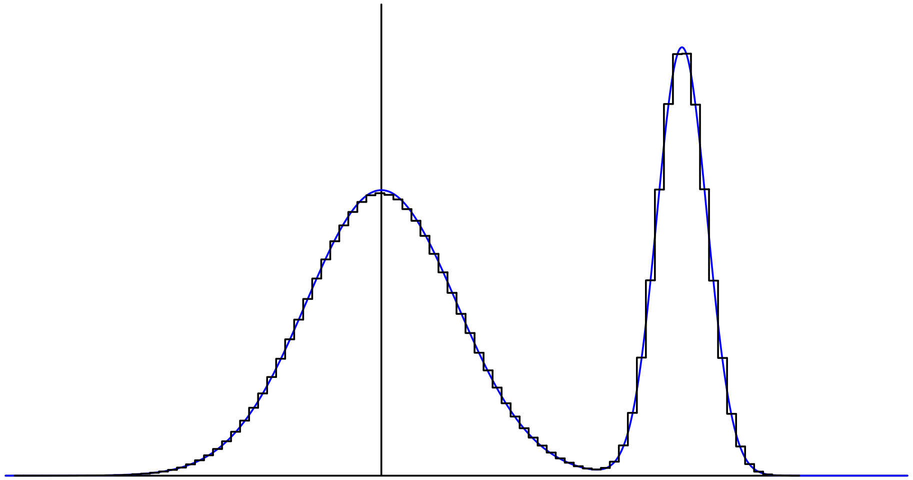


Monte Carlo methods, MCMC, Variational Methods

Outline

1. **Sampling**
2. MCMC
3. Variational Methods

$T = 10000000$, acceptence 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, Dirchlet) 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 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, Dirchlet) 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 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, Dirchlet) 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 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, Dirchlet) 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

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

- 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 distibution

Bayesian Inference

- Recall our problem is that we are given some data $\mathcal{D}$
- Our posterior is given by

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

- 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 distibution

Bayesian Inference

- Recall our problem is that we are given some data $\mathcal{D}$
- Our posterior is given by

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

- 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 distibution

Bayesian Inference

- Recall our problem is that we are given some data $\mathcal{D}$
- Our posterior is given by

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

- 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 distibution

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.

$$\mathbb{E}[\boldsymbol{\Theta}] \qquad \mathbb{E}\left[\Theta_i^2\right] - \mathbb{E}[\Theta_i]^2$$

$$\mathbb{E}[\Theta_i\,\Theta_j] - \mathbb{E}[\Theta_i]\,\mathbb{E}[\Theta_j] \qquad \mathbb{E}\left[\boldsymbol{\Theta}\,\boldsymbol{\Theta}^{\mathsf{T}}\right] - \mathbb{E}[\boldsymbol{\Theta}]\,\mathbb{E}[\boldsymbol{\Theta}]^{\mathsf{T}}$$

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.

$$\mathbb{E}[\boldsymbol{\Theta}] \qquad \mathbb{E}\left[\Theta_i^2\right] - \mathbb{E}[\Theta_i]^2$$

$$\mathbb{E}[\Theta_i\,\Theta_j] - \mathbb{E}[\Theta_i]\,\mathbb{E}[\Theta_j] \qquad \mathbb{E}\left[\boldsymbol{\Theta}\,\boldsymbol{\Theta}^{\mathsf{T}}\right] - \mathbb{E}[\boldsymbol{\Theta}]\,\mathbb{E}[\boldsymbol{\Theta}]^{\mathsf{T}}$$

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.

$$\mathbb{E}[\boldsymbol{\Theta}] \qquad \mathbb{E}\left[\Theta_i^2\right] - \mathbb{E}[\Theta_i]^2$$

$$\mathbb{E}[\Theta_i\,\Theta_j] - \mathbb{E}[\Theta_i]\,\mathbb{E}[\Theta_j] \qquad \mathbb{E}\left[\boldsymbol{\Theta}\,\boldsymbol{\Theta}^{\mathsf{T}}\right] - \mathbb{E}[\boldsymbol{\Theta}]\,\mathbb{E}[\boldsymbol{\Theta}]^{\mathsf{T}}$$

Sample Estimation

- If we can draw independent **deviates** (aka **variates**), $\boldsymbol{\Theta}_i$, from our posterior distribution then we can obtain an estimate of our expectation

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

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

Sample Estimation

- If we can draw independent **deviates** (aka **variates**), $\boldsymbol{\Theta}_i$, from our posterior distribution then we can obtain an estimate of our expectation

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

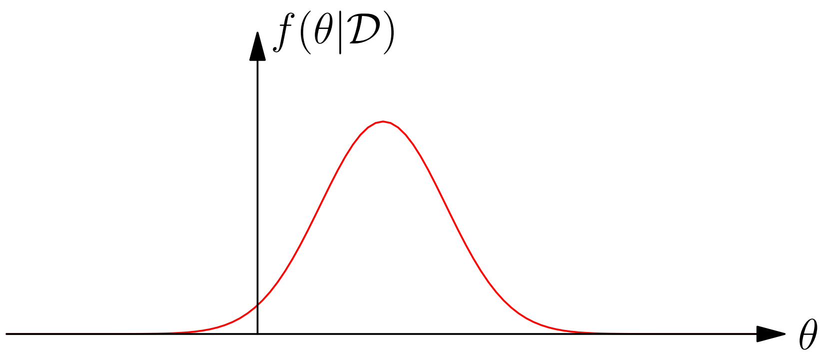
- Provided our posterior distribution is well behaved the relative error in our estimate will drop off as $1/\sqrt{n}$

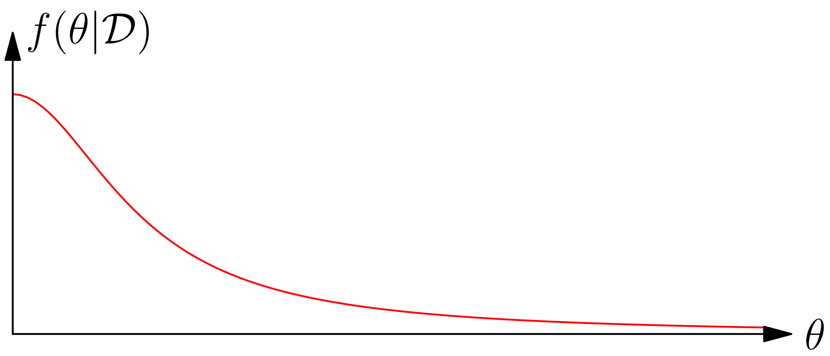
Sample Estimation

- If we can draw independent **deviates** (aka **variates**), $\boldsymbol{\Theta}_i$, from our posterior distribution then we can obtain an estimate of our expectation

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

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



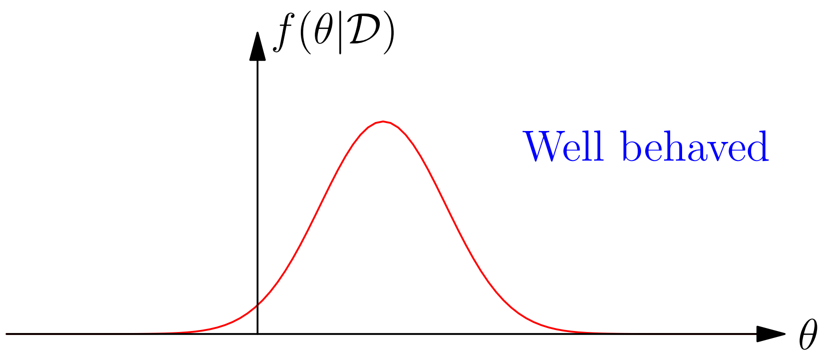


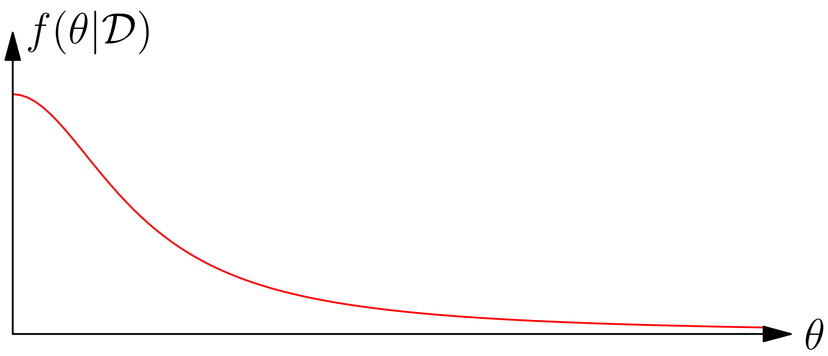
Sample Estimation

- If we can draw independent **deviates** (aka **variates**), $\boldsymbol{\Theta}_i$, from our posterior distribution then we can obtain an estimate of our expectation

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

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



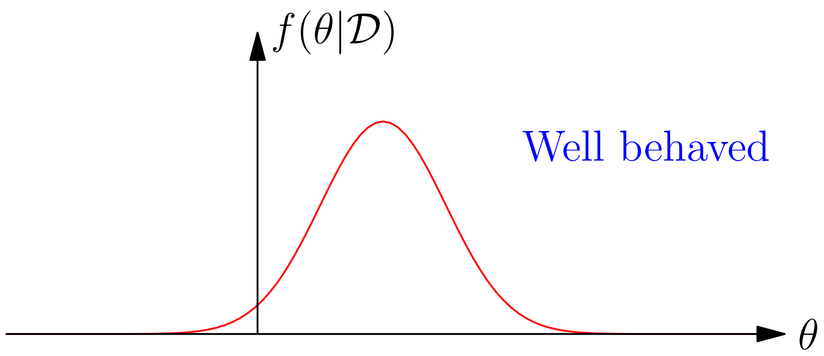


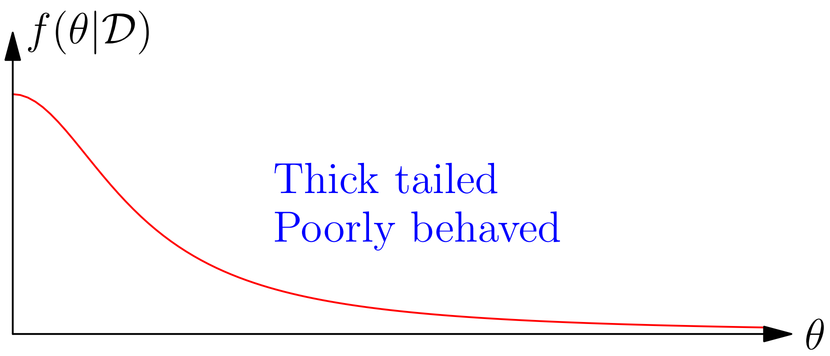
Sample Estimation

- If we can draw independent **deviates** (aka **variates**), $\boldsymbol{\Theta}_i$, from our posterior distribution then we can obtain an estimate of our expectation

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

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





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

Drawing Random Samples

- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**

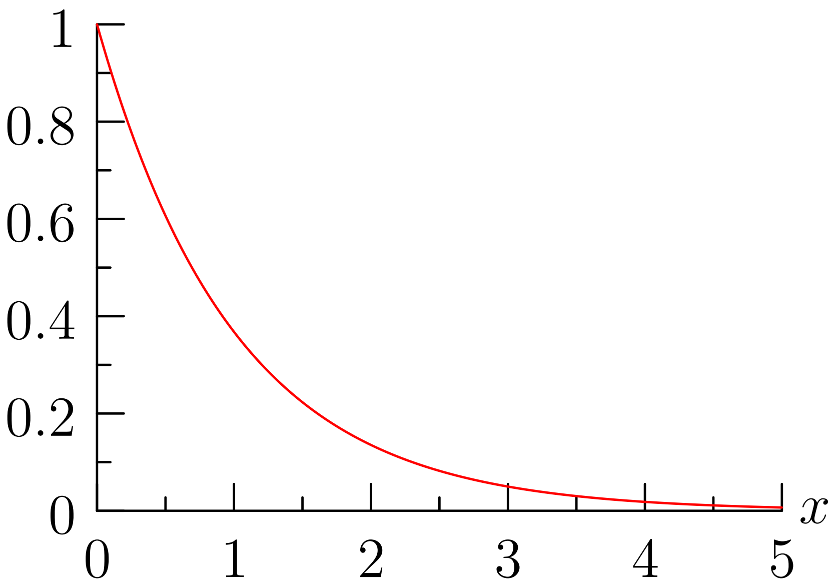
- For some very simple distributions we can use the transformation methods to transform a uniform distribution

Drawing Random Samples

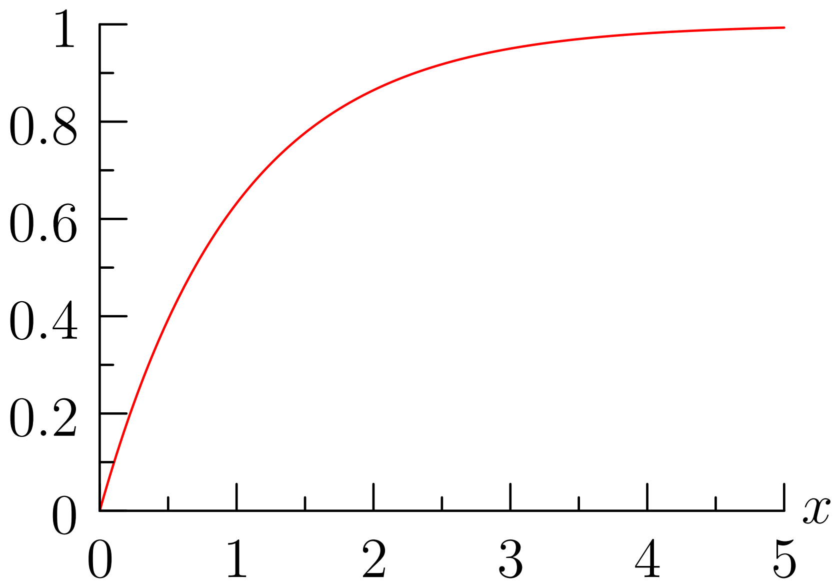
- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**

- For some very simple distributions we can use the transformation methods to transform a uniform distribution

$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$



$$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$$

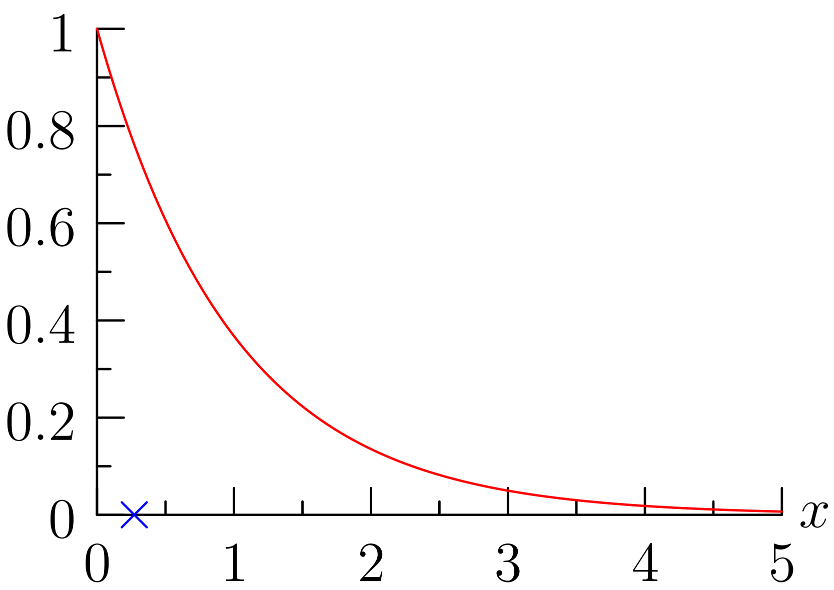


Drawing Random Samples

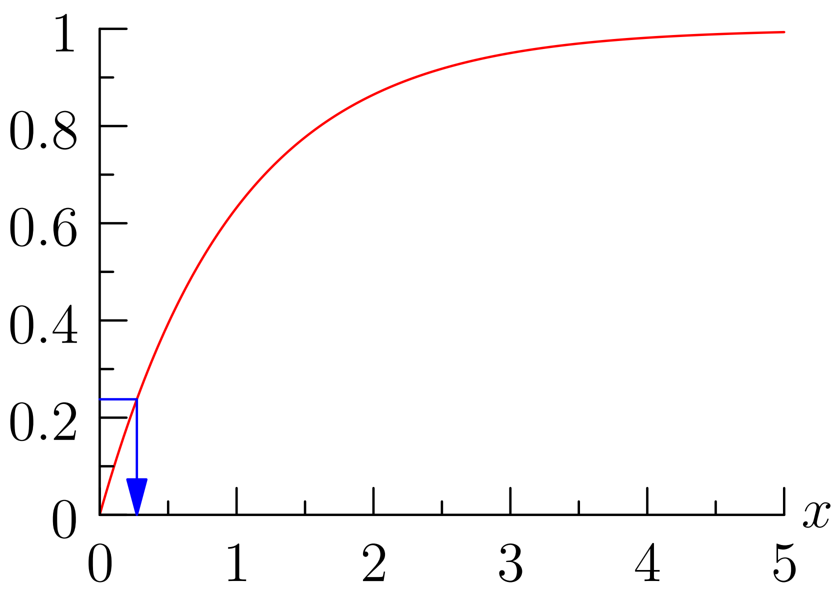
- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**

- For some very simple distributions we can use the transformation methods to transform a uniform distribution

$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$



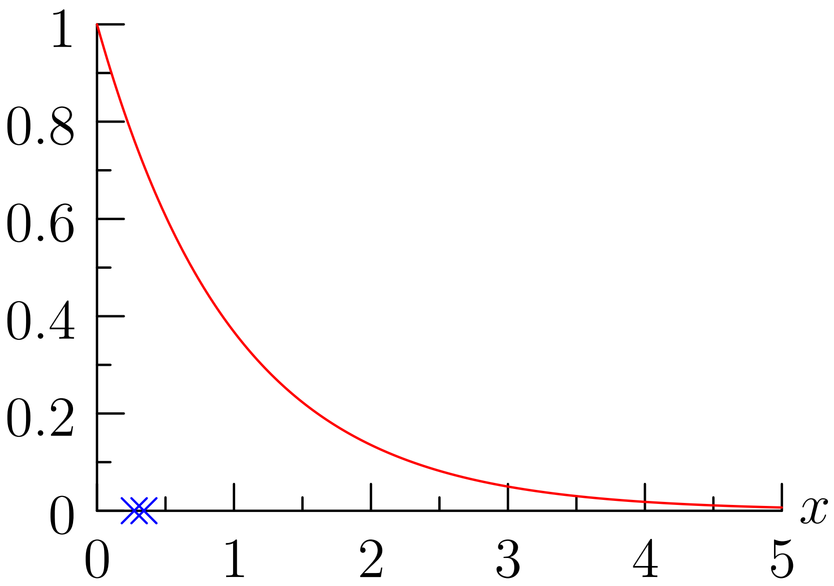
$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$



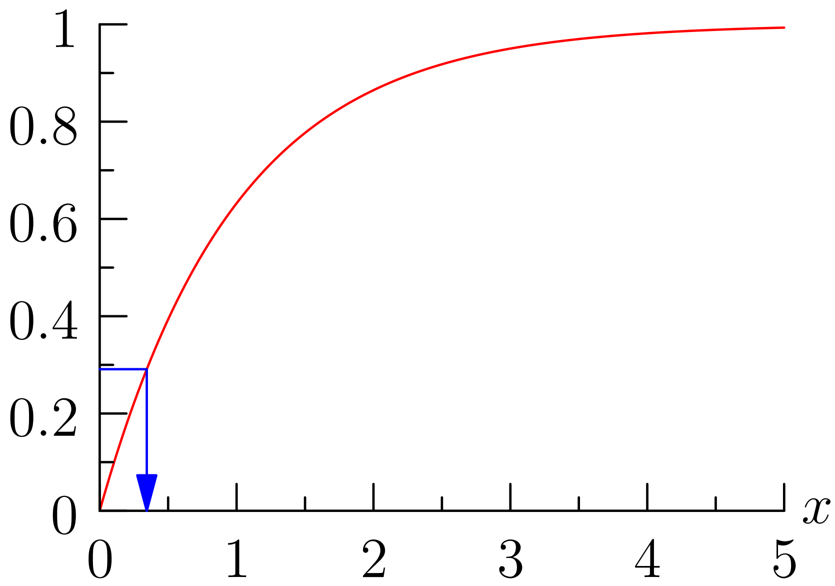
Drawing Random Samples

- 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

$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$



$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$

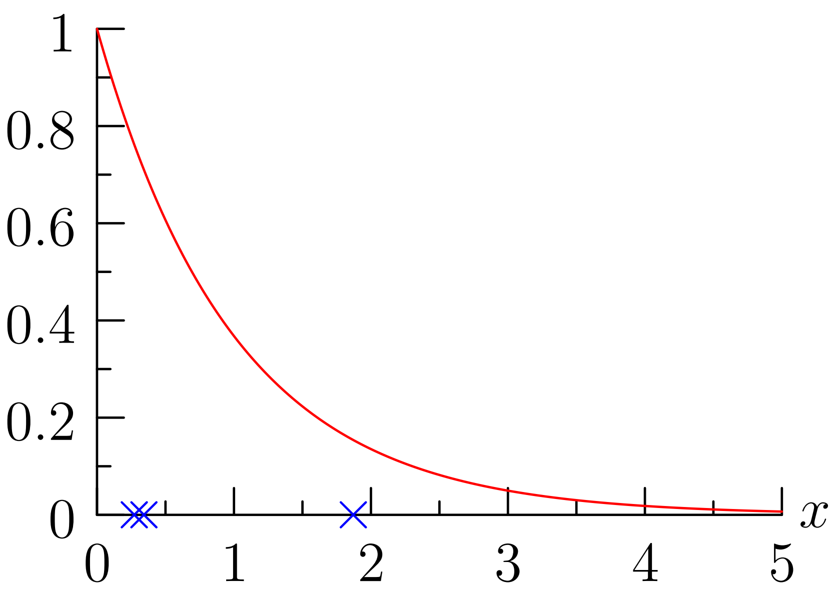


Drawing Random Samples

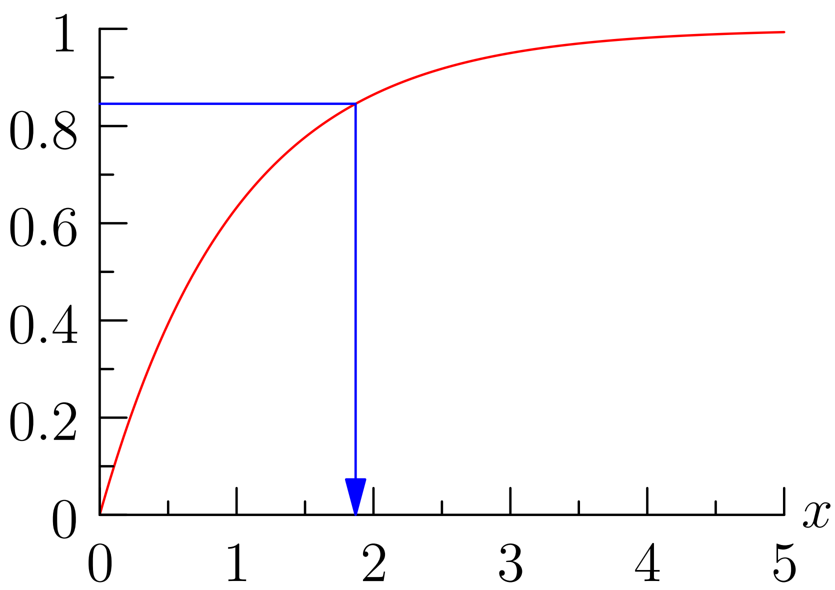
- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**

- For some very simple distributions we can use the transformation methods to transform a uniform distribution

$$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$$



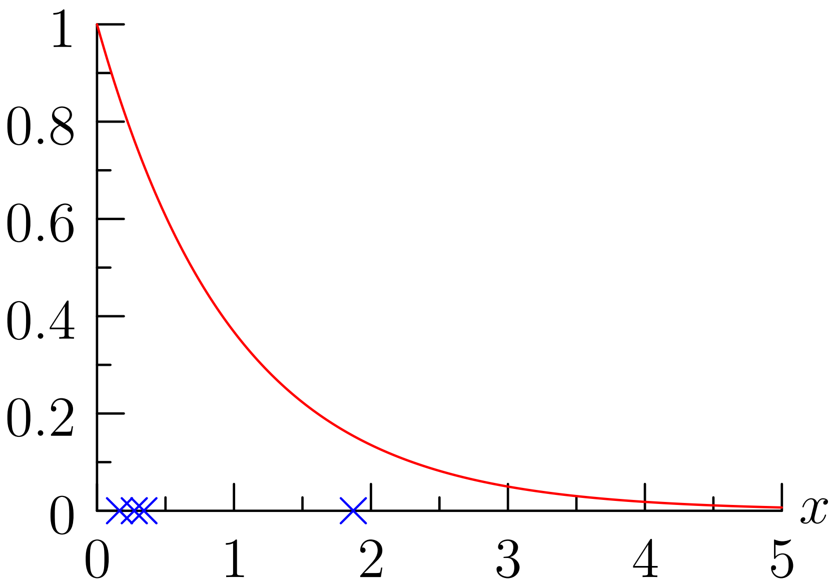
$$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$$



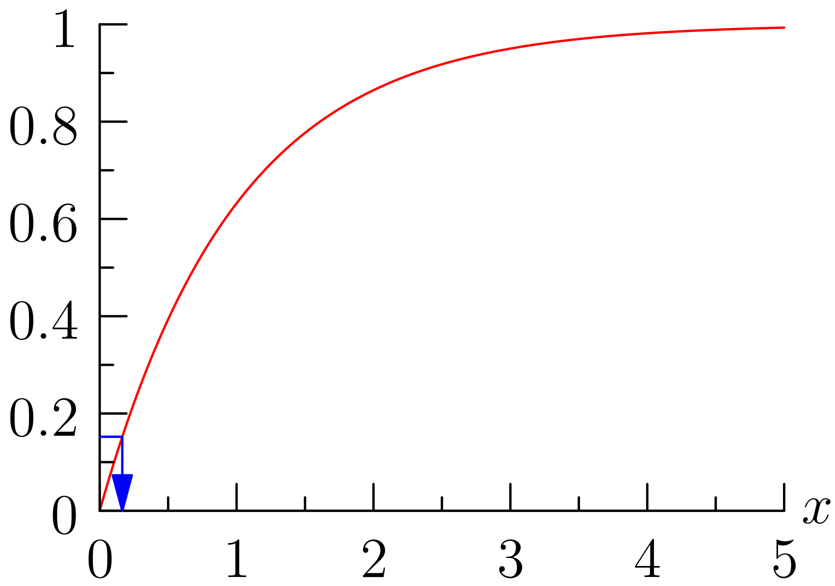
Drawing Random Samples

- 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

$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$



$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$

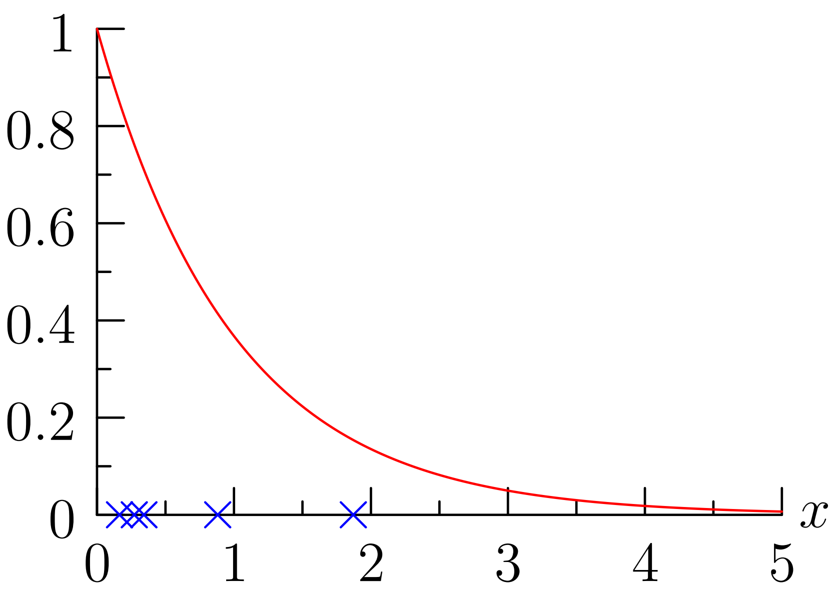


Drawing Random Samples

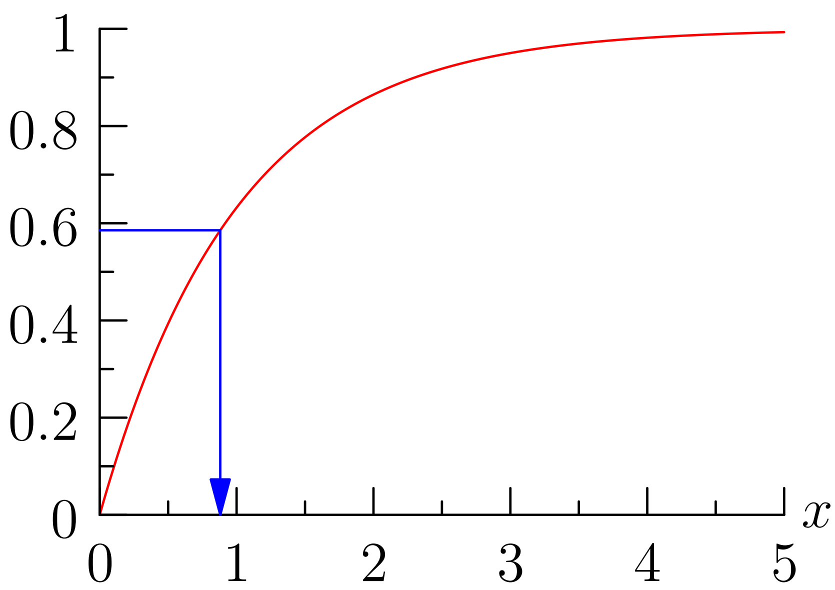
- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**

- For some very simple distributions we can use the transformation methods to transform a uniform distribution

$$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$$



$$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$$

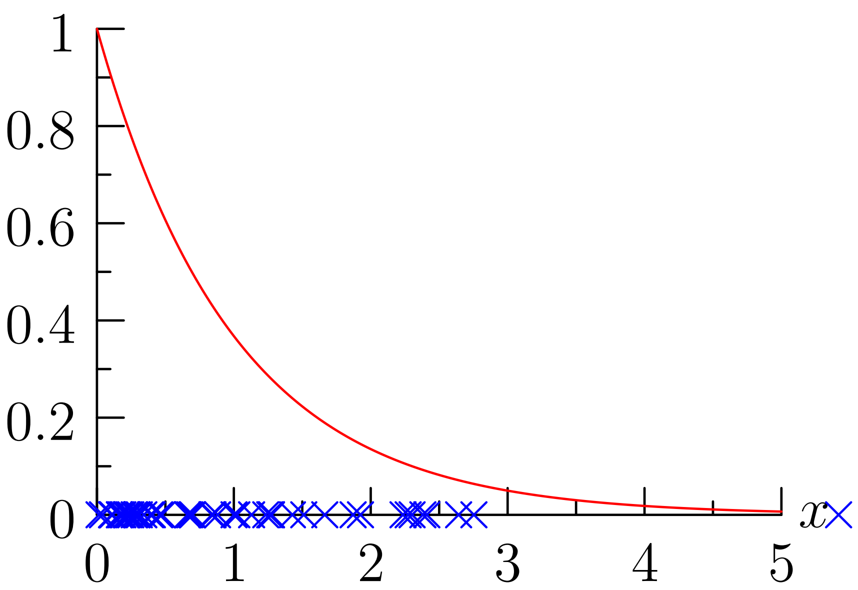


Drawing Random Samples

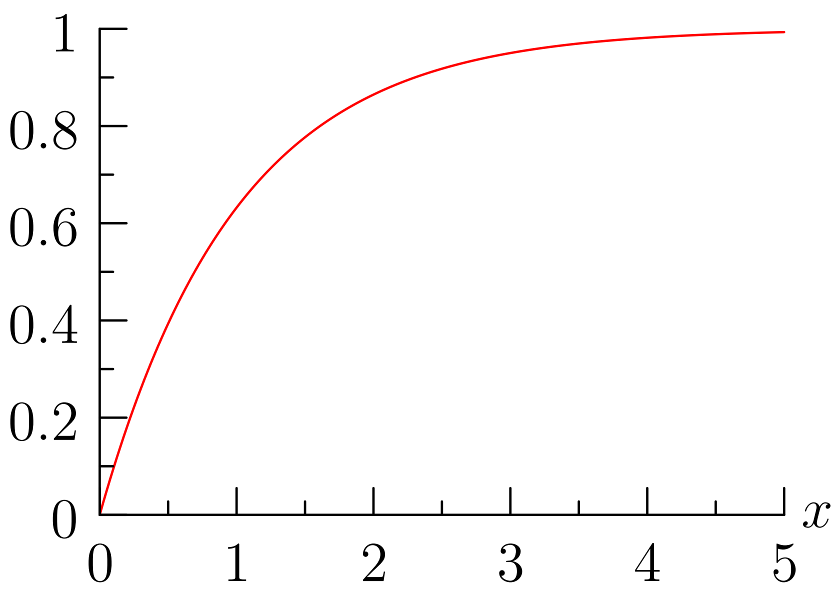
- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**

- For some very simple distributions we can use the transformation methods to transform a uniform distribution

$$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$$



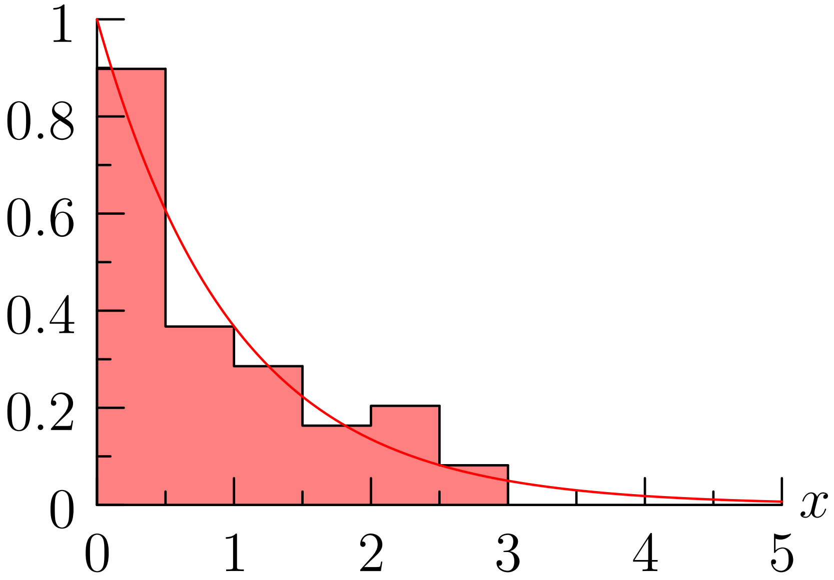
$$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$$



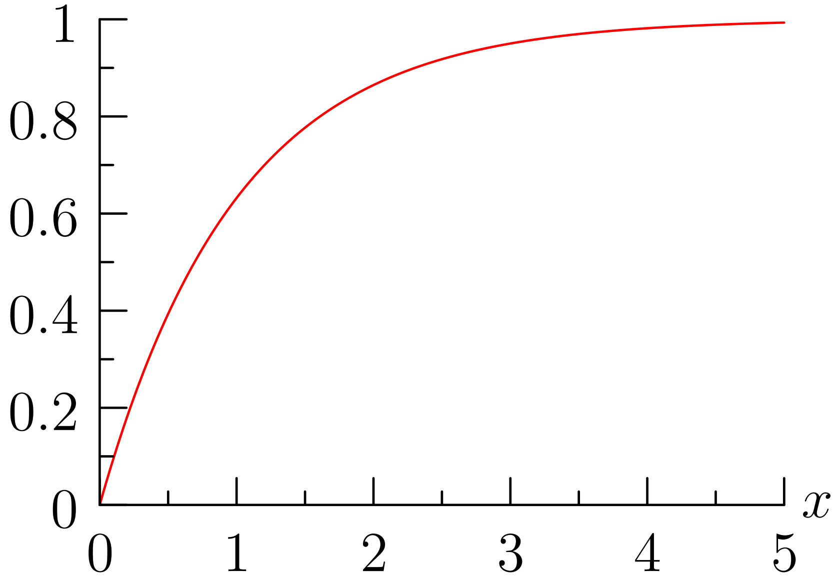
Drawing Random Samples

- 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

$$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$$



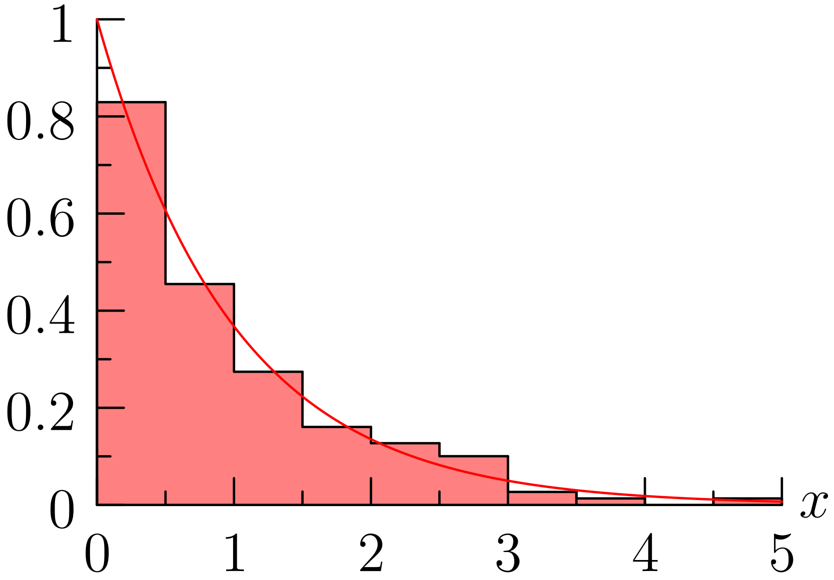
$$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$$



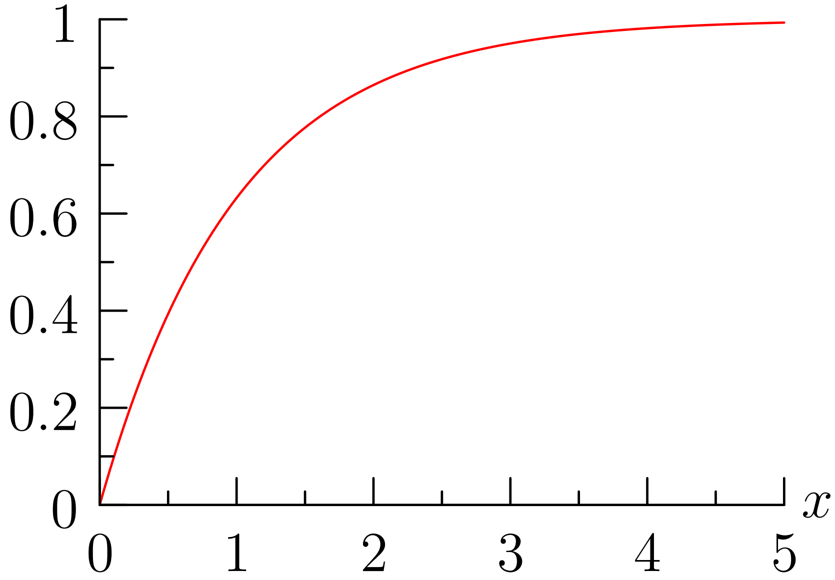
Drawing Random Samples

- 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

$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$



$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$

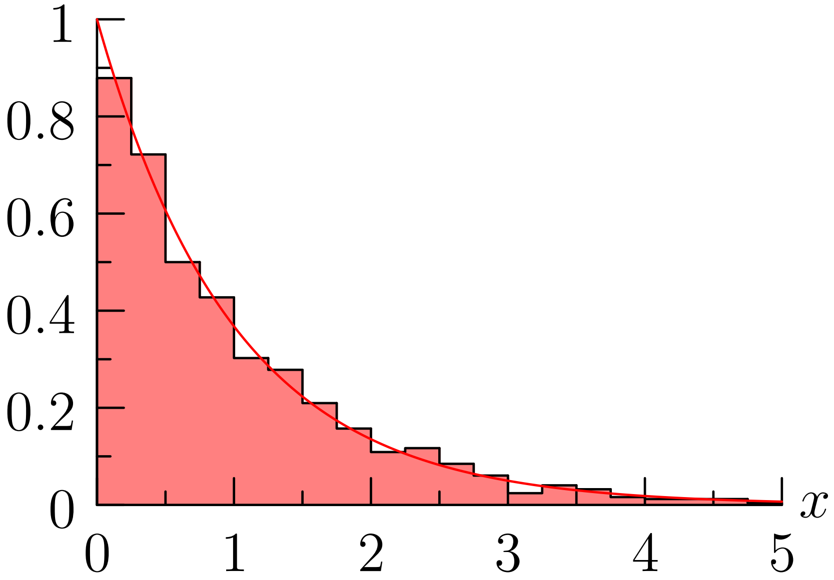


Drawing Random Samples

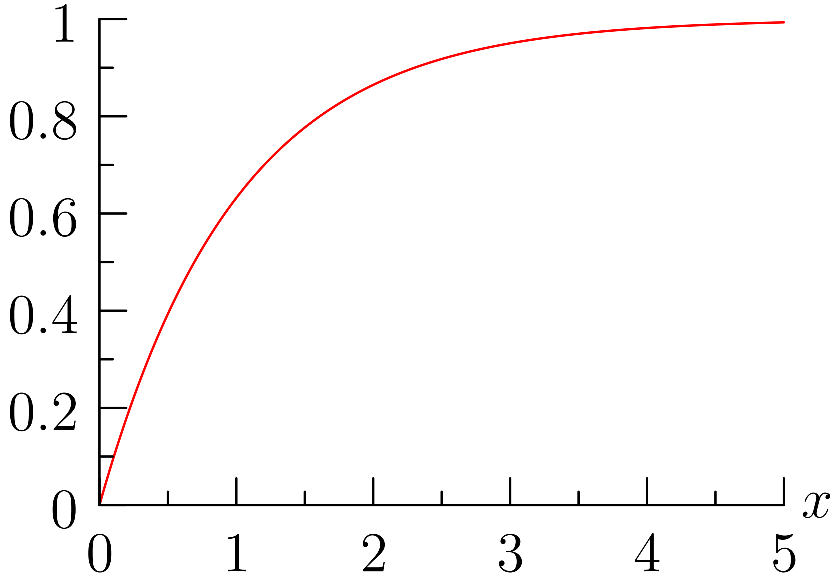
- Drawing (pseudo) random variables from a distribution is known as **Monte Carlo**

- For some very simple distributions we can use the transformation methods to transform a uniform distribution

$$\mathrm{Exp}(x|1) = \mathrm{e}^{-x}$$



$$F(x|1) = \int_0^x \mathrm{Exp}(y|1)\mathrm{d}y = 1 - \mathrm{e}^{-x}$$



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 $c\, g_Y(x) \geq f_X(x)$
- To draw deviates from $f_X(x)$ we draw a deviate $Y \sim g_Y$ and then accept the deviate with probability $f_X(Y)/(c\, g_Y(Y))$
- The expected rejection rate is $c - 1$
- Need to choose a good distribution $g_Y(y)$

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 $c\, g_Y(x) \geq f_X(x)$
- To draw deviates from $f_X(x)$ we draw a deviate $Y \sim g_Y$ and then accept the deviate with probability $f_X(Y)/(c\, g_Y(Y))$
- The expected rejection rate is $c - 1$
- Need to choose a good distribution $g_Y(y)$

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 $c\, g_Y(x) \geq f_X(x)$
- To draw deviates from $f_X(x)$ we draw a deviate $Y \sim g_Y$ and then accept the deviate with probability $f_X(Y)/(c\, g_Y(Y))$
- The expected rejection rate is $c - 1$
- Need to choose a good distribution $g_Y(y)$

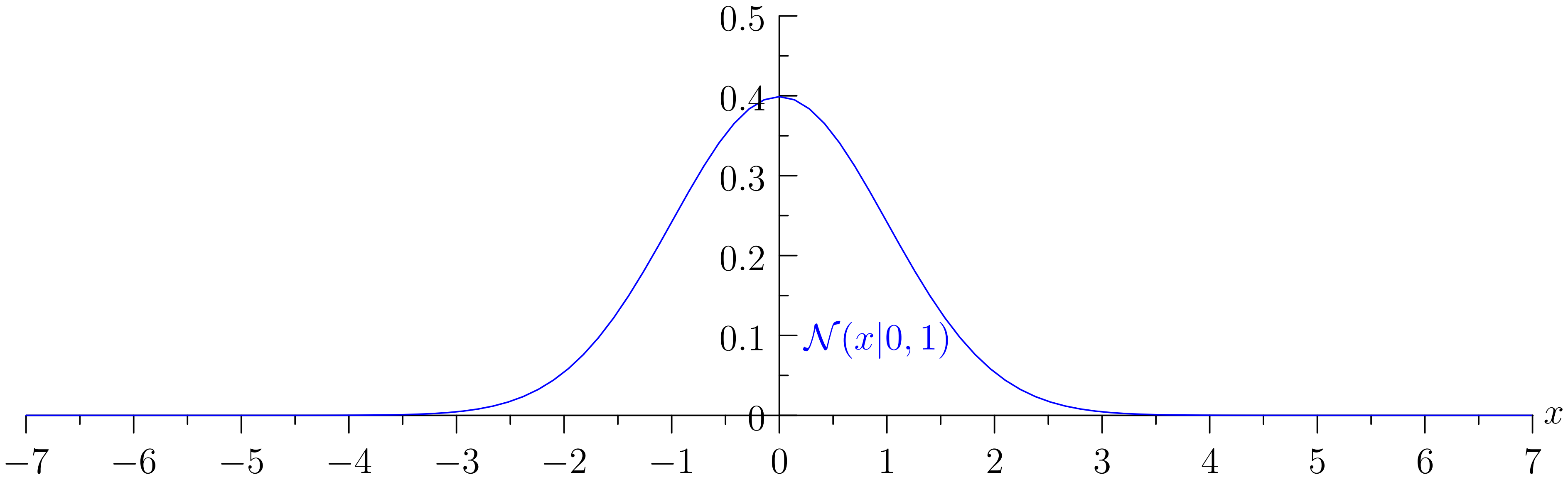
Rejection Method

- 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 $c\, g_Y(x) \geq f_X(x)$
- To draw deviates from $f_X(x)$ we draw a deviate $Y \sim g_Y$ and then accept the deviate with probability $f_X(Y)/(c\, g_Y(Y))$
- The expected rejection rate is $c - 1$
- Need to choose a good distribution $g_Y(y)$

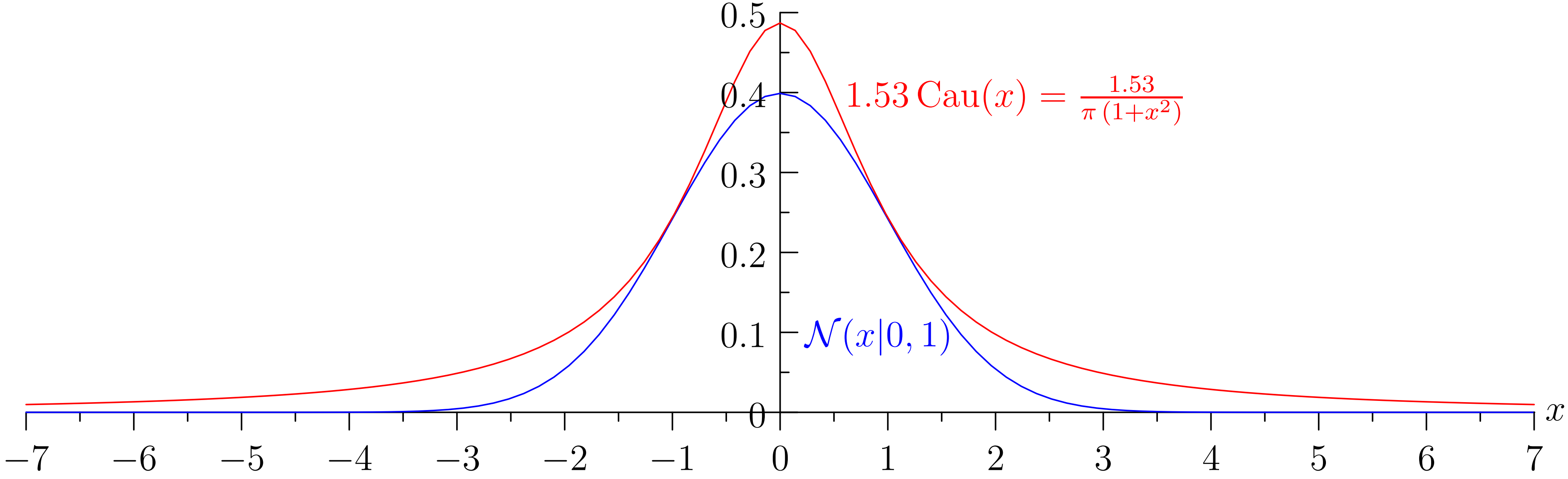
Rejection Method

- 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 $c\, g_Y(x) \geq f_X(x)$
- To draw deviates from $f_X(x)$ we draw a deviate $Y \sim g_Y$ and then accept the deviate with probability $f_X(Y)/(c\, g_Y(Y))$
- The expected rejection rate is $c - 1$
- Need to choose a good distribution $g_Y(y)$

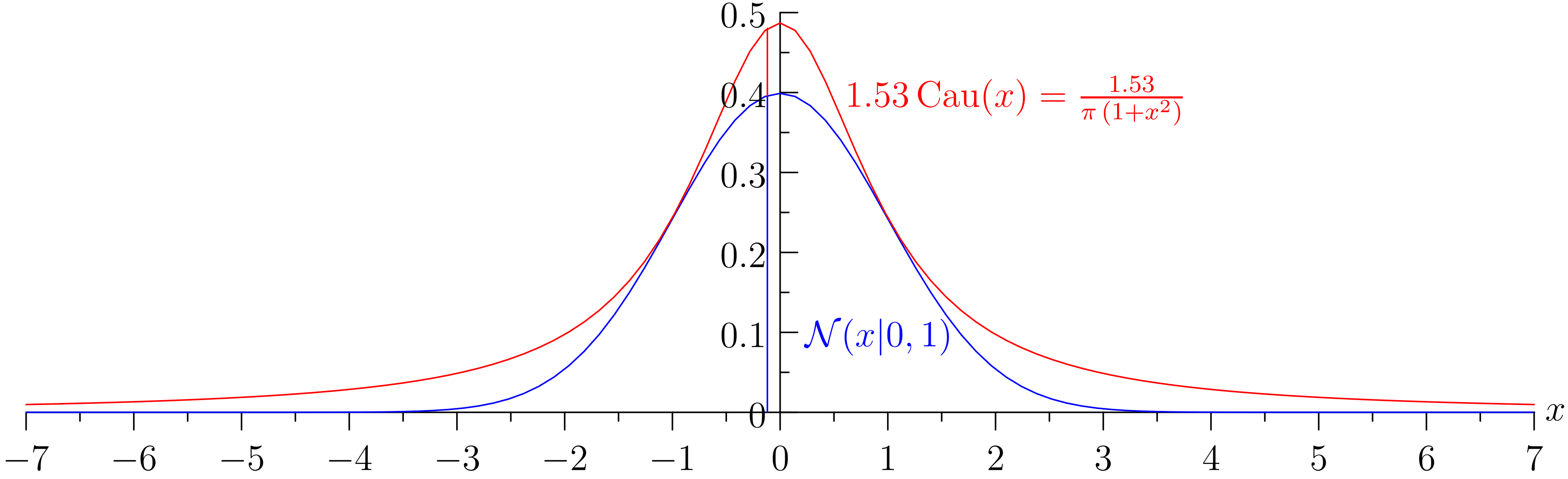
Drawing Normal Deviates



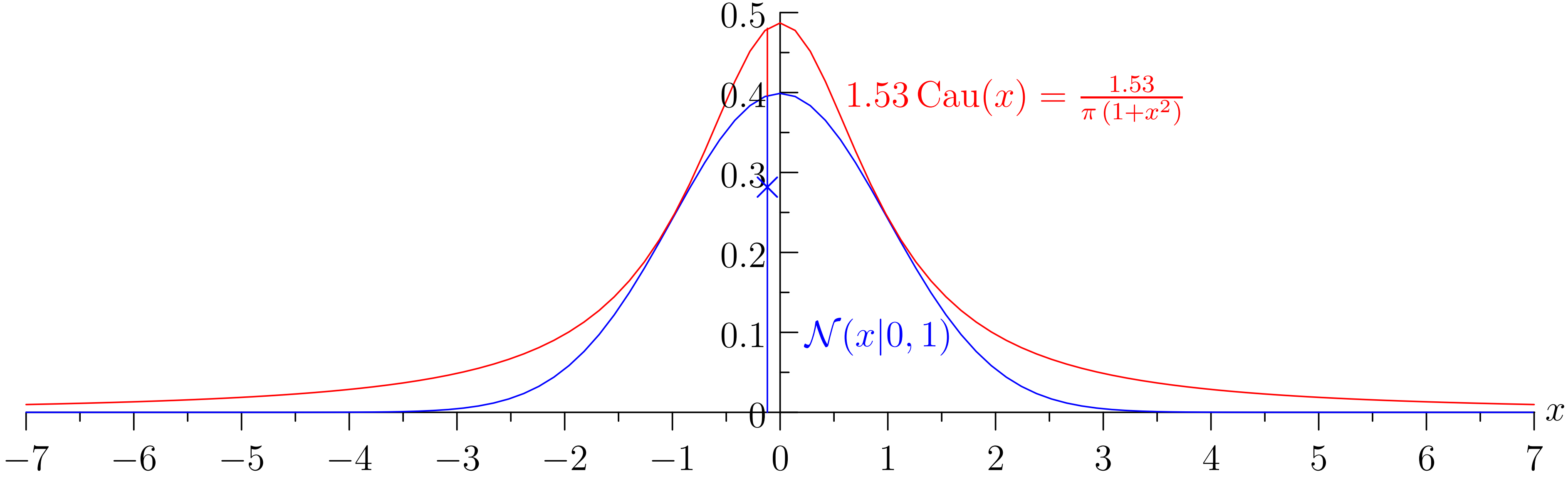
Drawing Normal Deviates



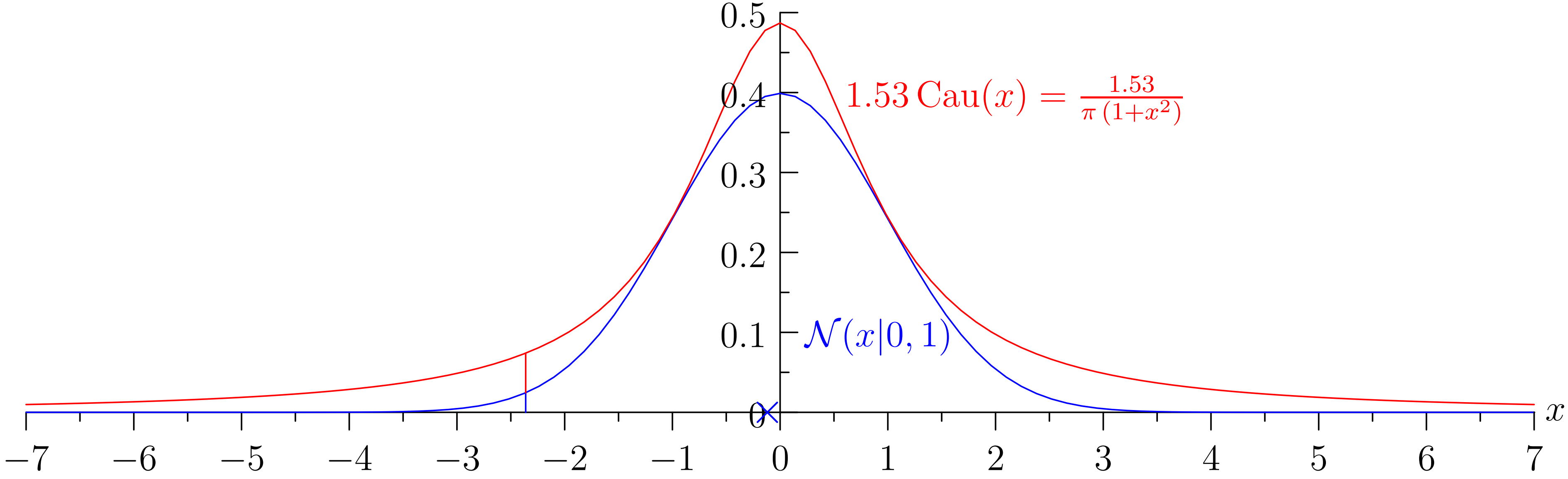
Drawing Normal Deviates



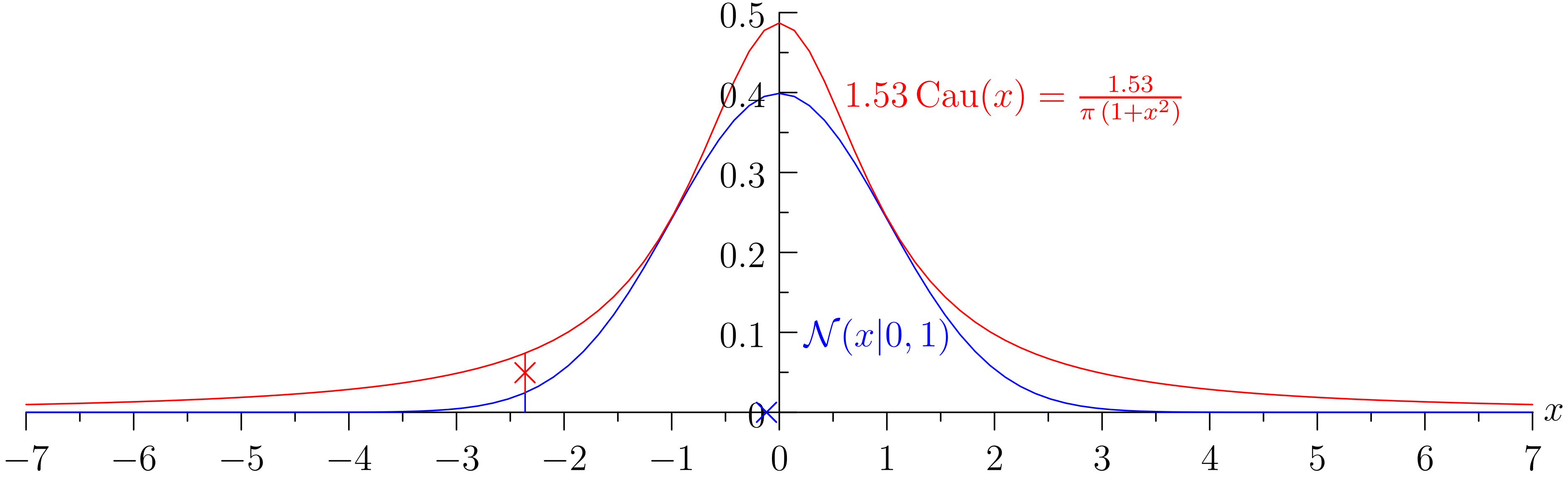
Drawing Normal Deviates



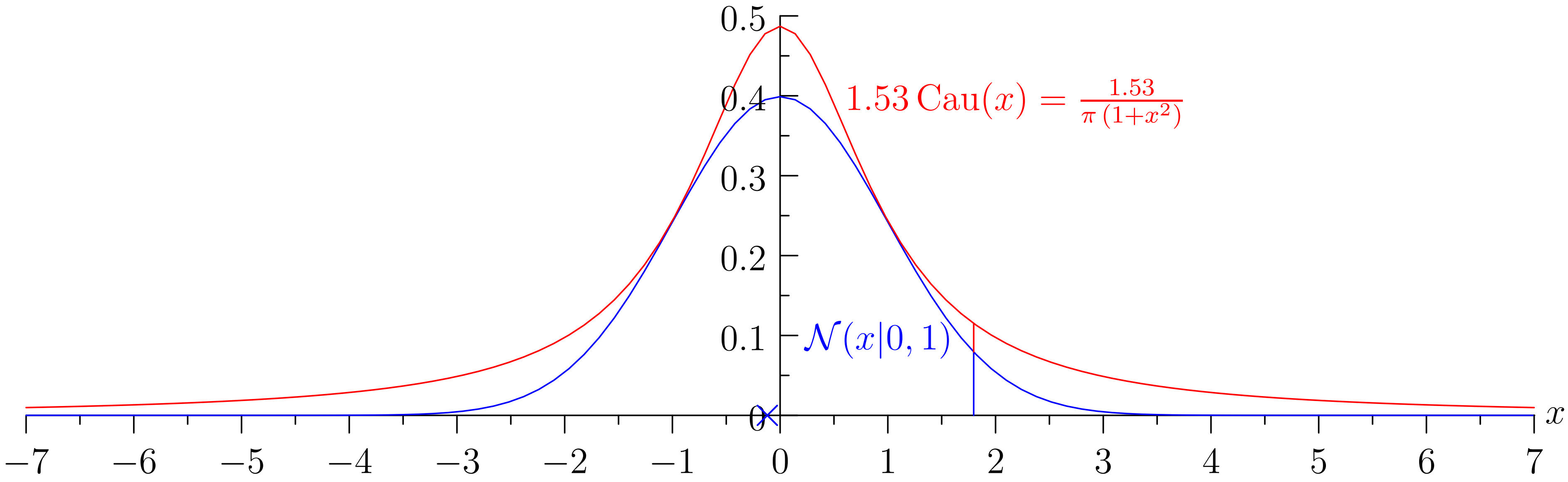
Drawing Normal Deviates



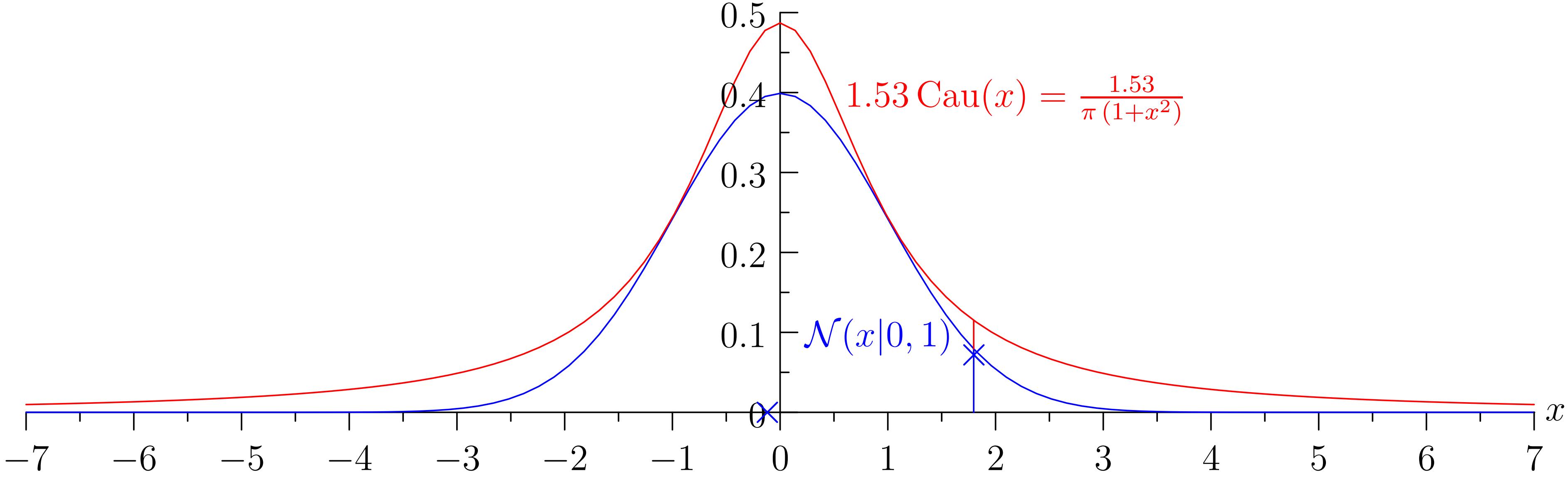
Drawing Normal Deviates



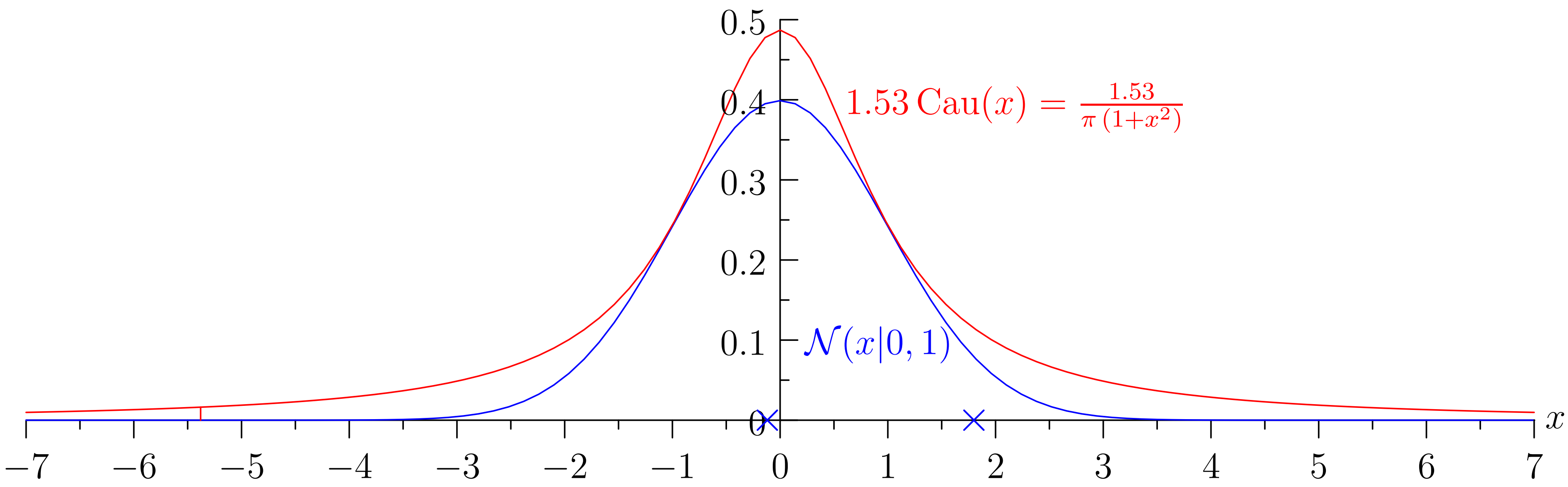
Drawing Normal Deviates



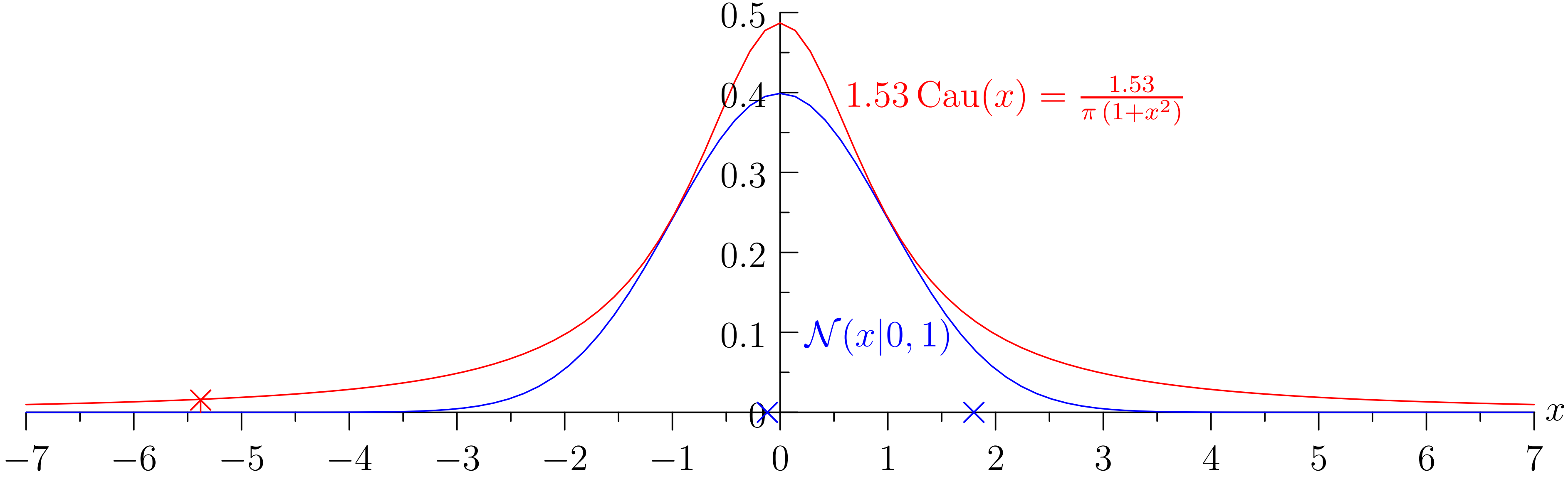
Drawing Normal Deviates



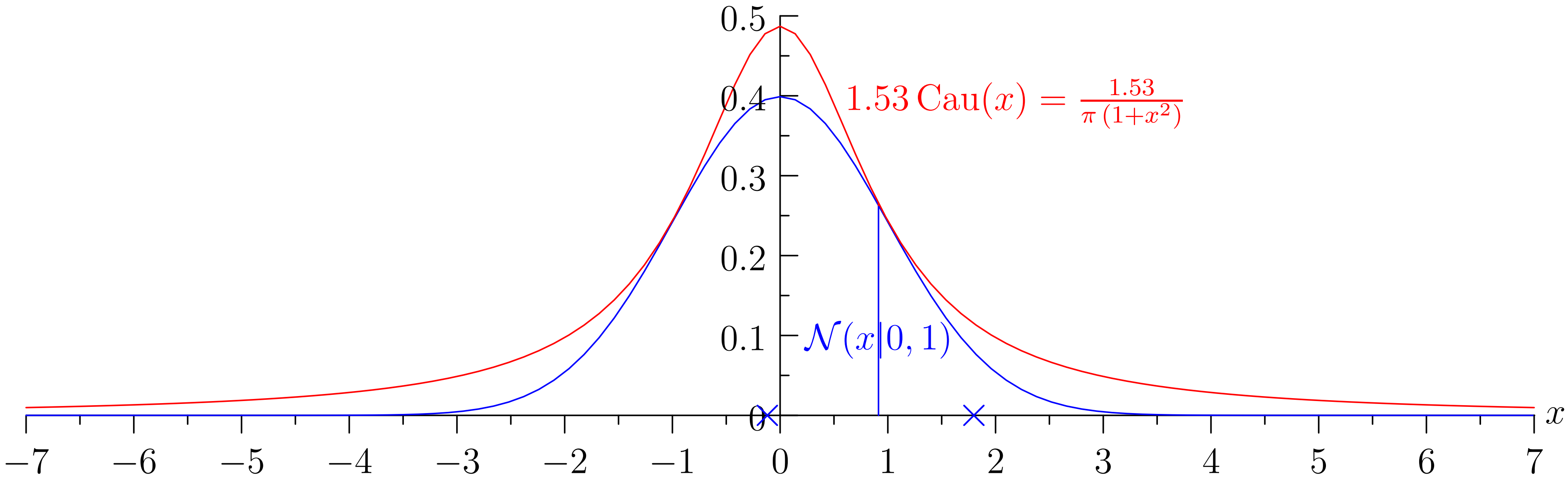
Drawing Normal Deviates



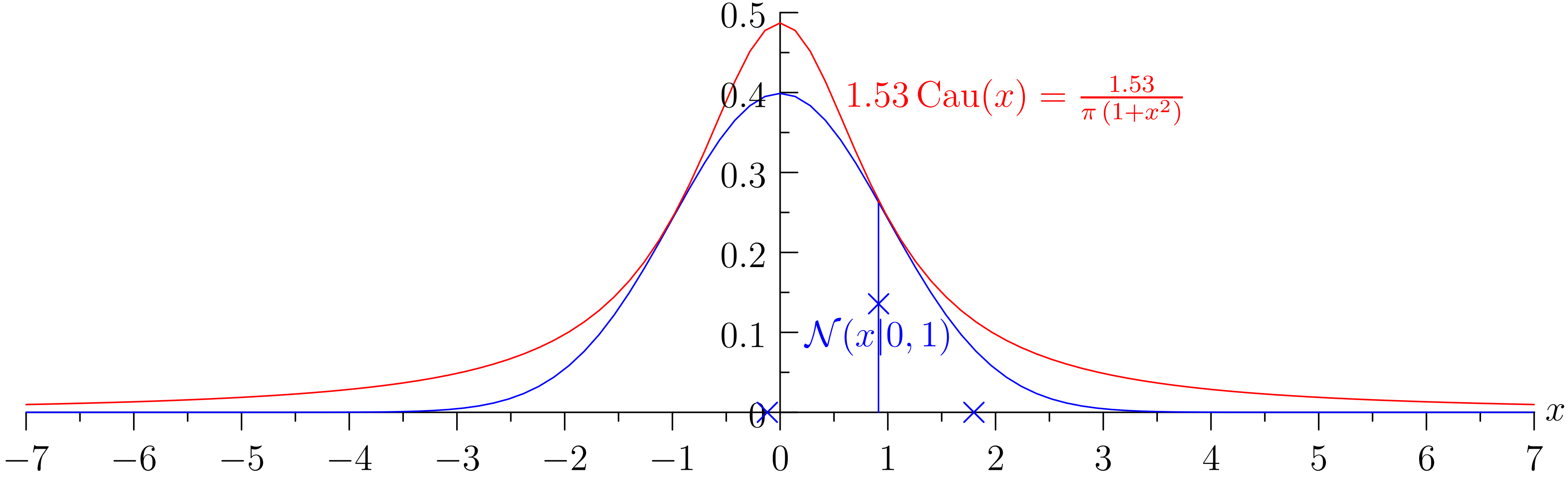
Drawing Normal Deviates



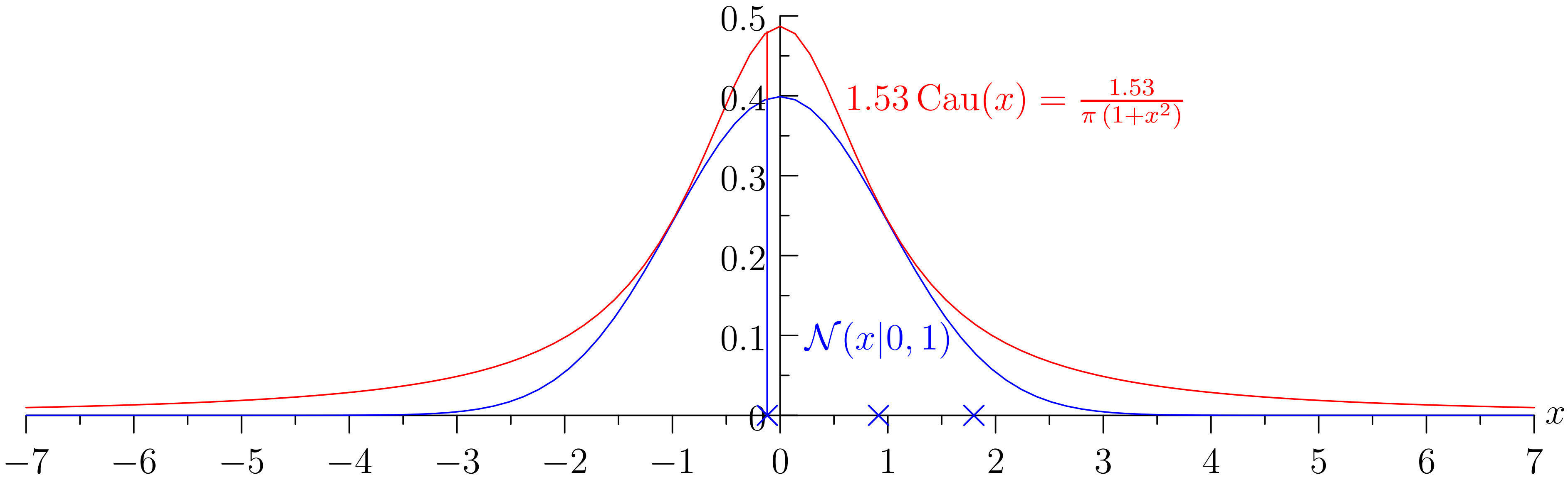
Drawing Normal Deviates



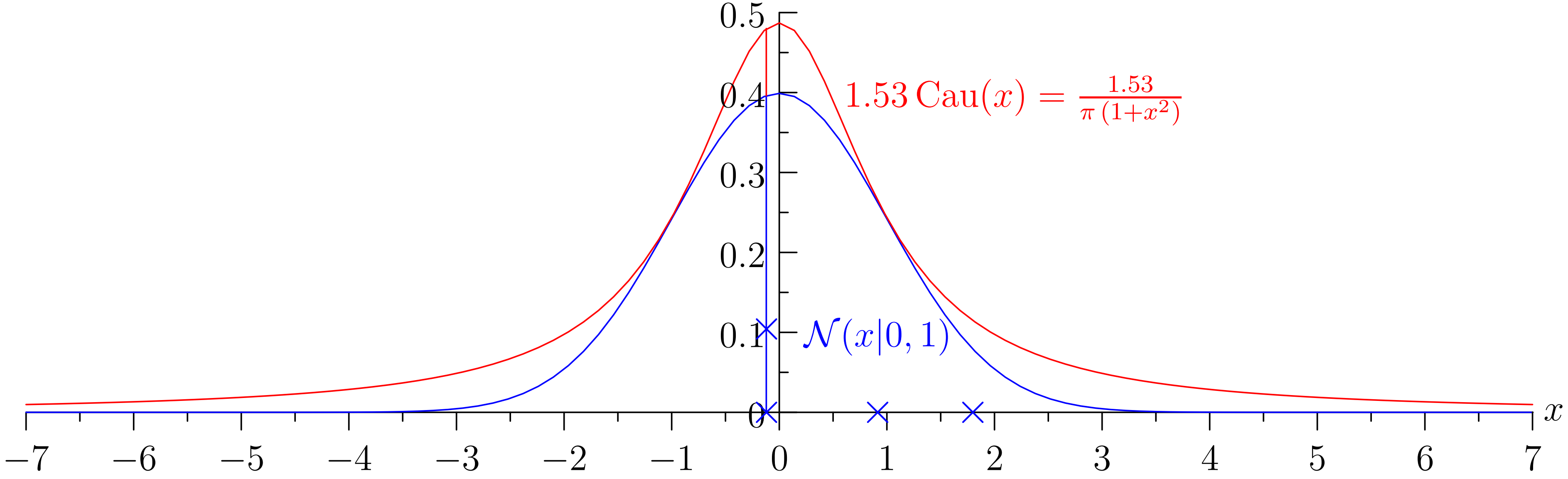
Drawing Normal Deviates



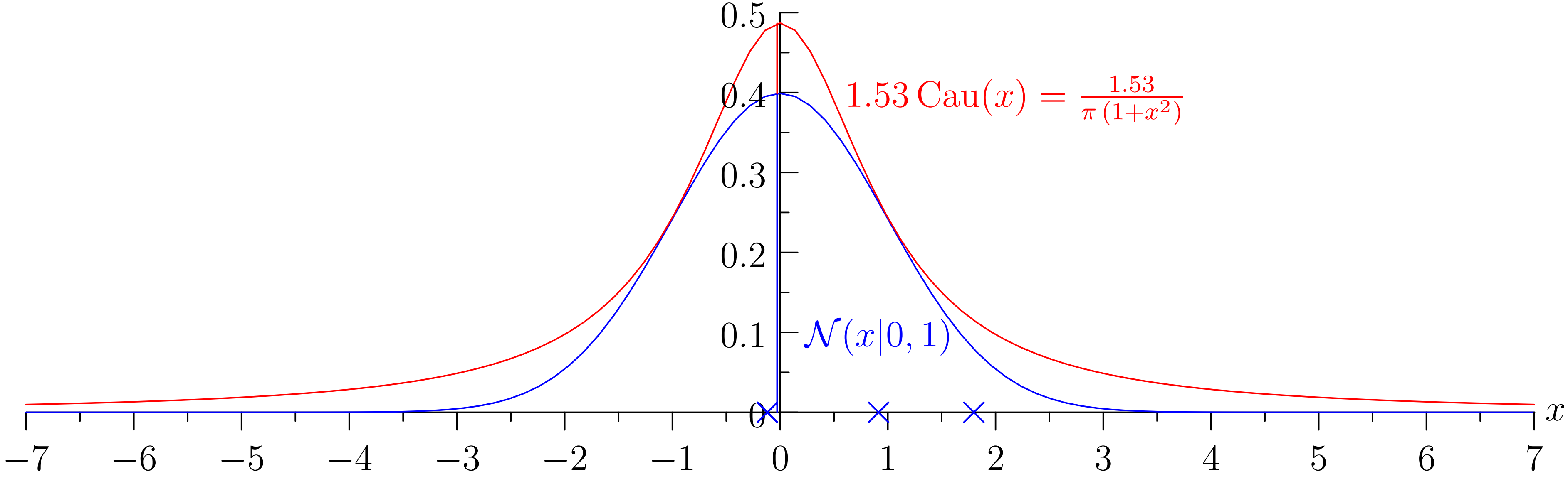
Drawing Normal Deviates



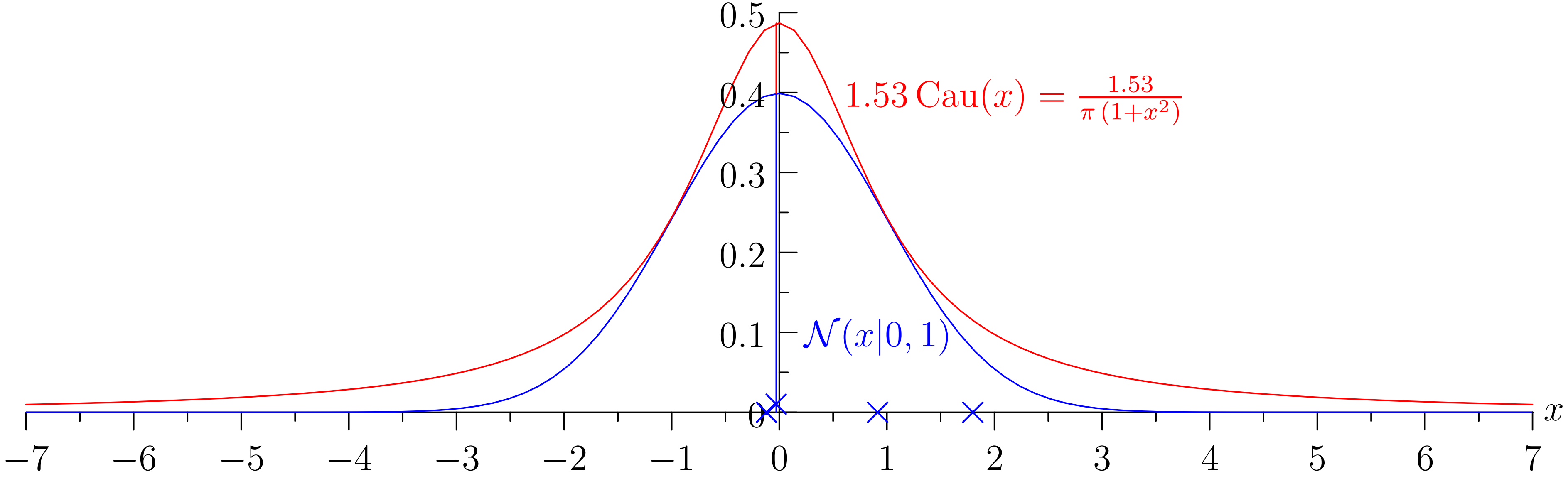
Drawing Normal Deviates



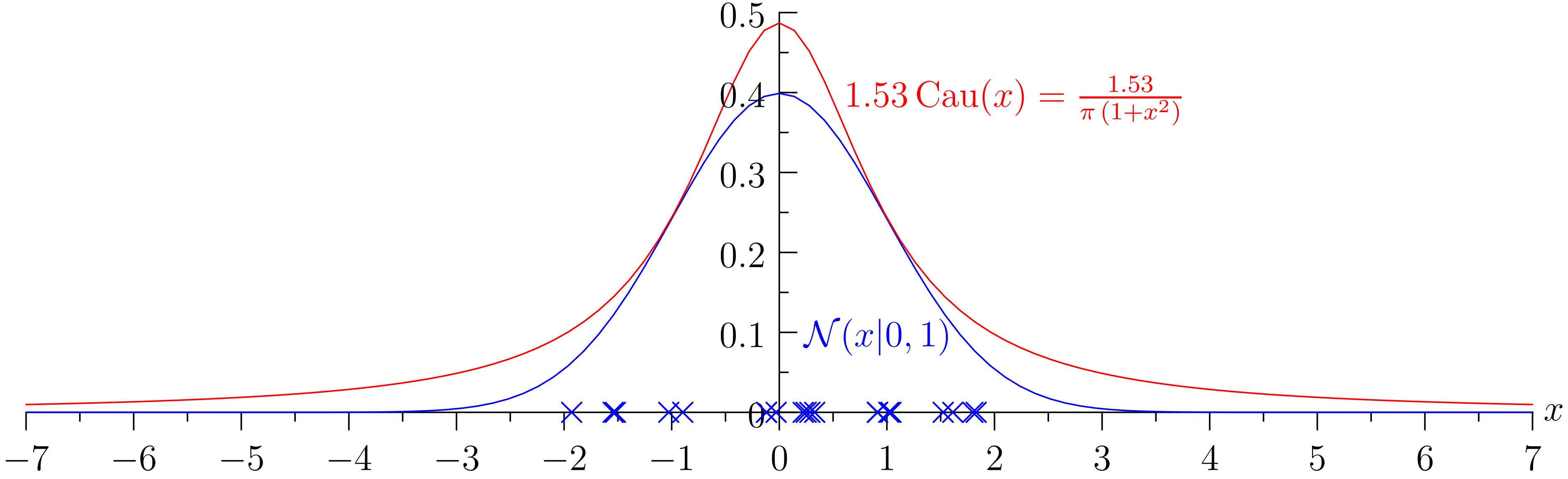
Drawing Normal Deviates



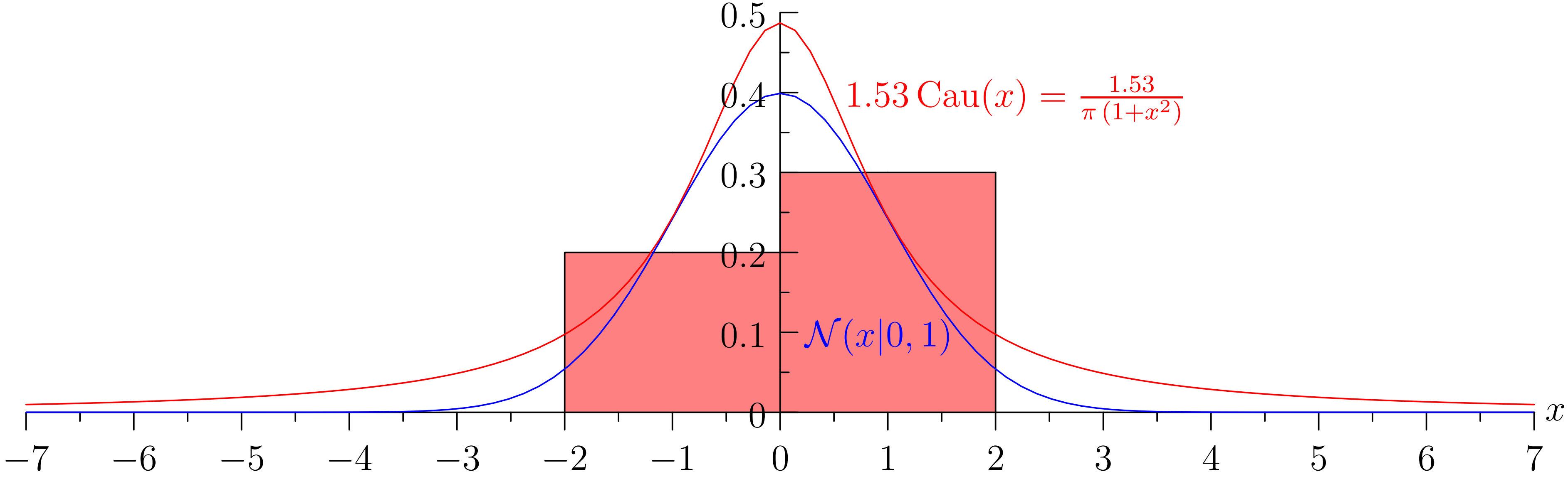
Drawing Normal Deviates



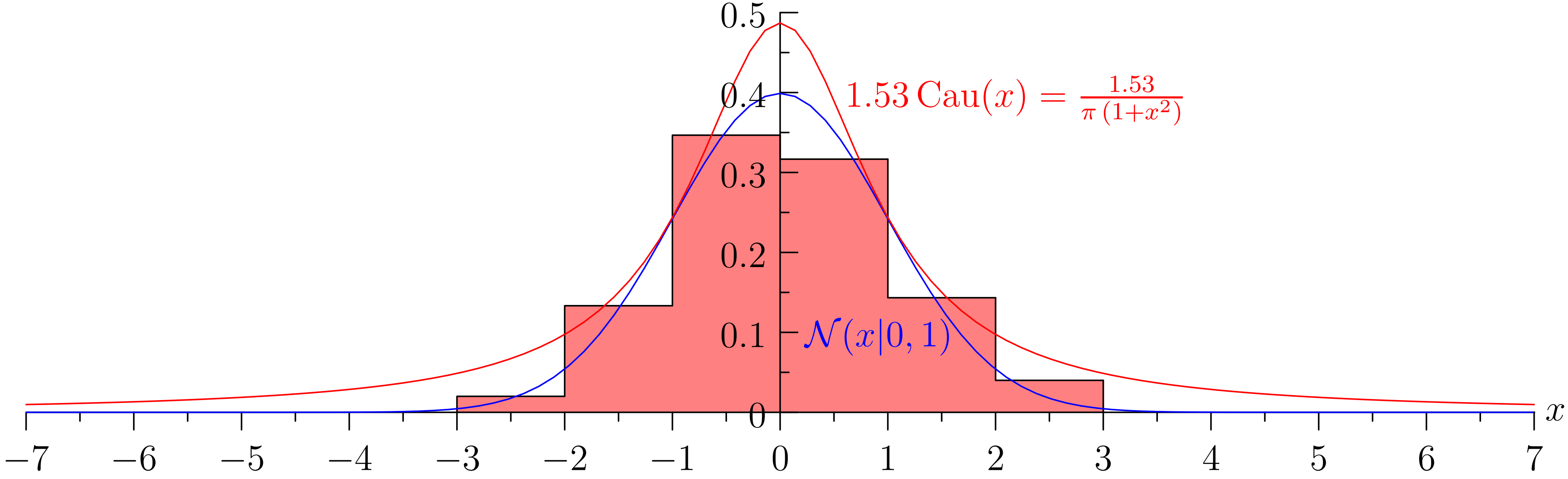
Drawing Normal Deviates



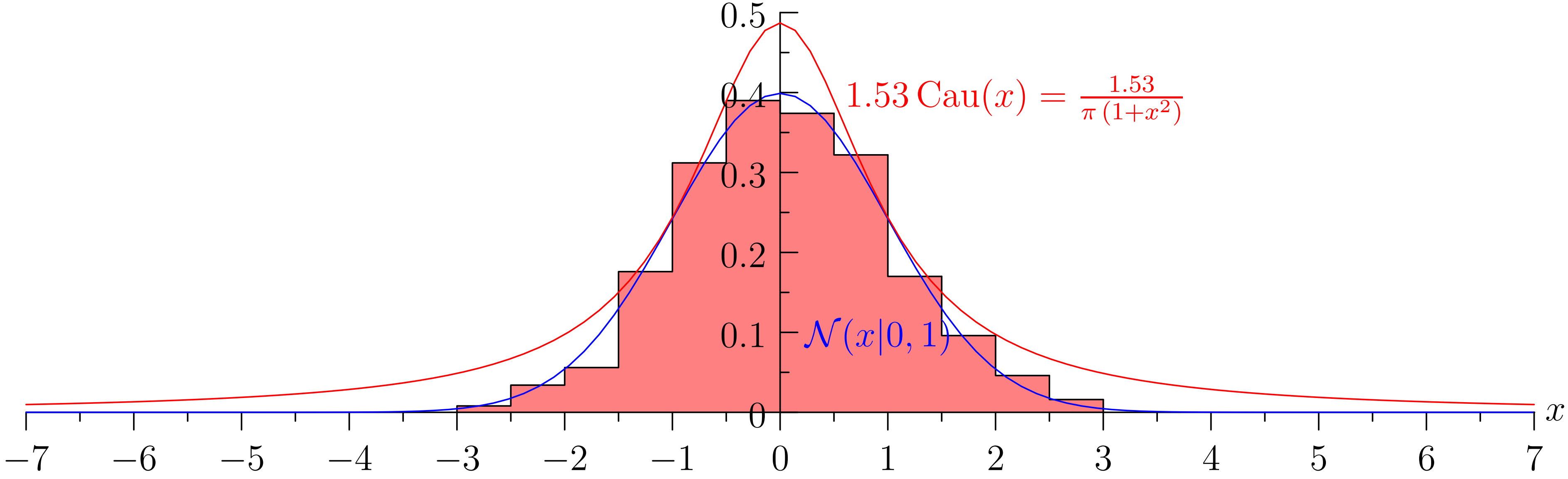
Drawing Normal Deviates



Drawing Normal Deviates



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

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

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

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)

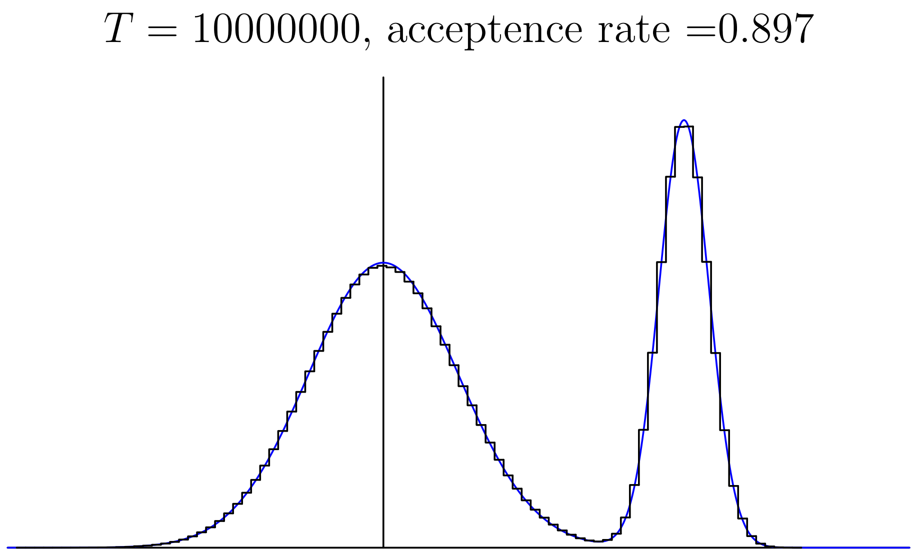
- However, for complicated probability distributions it can be difficult to find a good proposal distribution $g_Y(y)$

- 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. **MCMC**
3. Variational Methods



Detailed Balance

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

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

- Then (with mild conditions on M_{ij}) if we start from any state, eventually, the probability of being in state i is π_i
- This condition is known as **detailed balance**

Detailed Balance

- Suppose we have a set of states $\mathcal{S}$ and want to draw sample from a probability distribution $\boldsymbol{\pi} = (\pi_i | i \in \mathcal{S})$

- If we choose a transition probability M_{ij} from state j to state i such that

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

- Then (with mild conditions on M_{ij}) if we start from any state, eventually, the probability of being in state i is π_i

- This condition is known as **detailed balance**

Detailed Balance

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

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

- Then (with mild conditions on M_{ij}) if we start from any state, eventually, the probability of being in state i is π_i
- This condition is known as **detailed balance**

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
- Note that we require the state i to have the same number of neighbours as state j so that detailed balance is satisfied
- This is the basis of **Markov Chain Monte Carlo**

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
- Note that we require the state i to have the same number of neighbours as state j so that detailed balance is satisfied
- This is the basis of **Markov Chain Monte Carlo**

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
- Note that we require the state i to have the same number of neighbours as state j so that detailed balance is satisfied
- This is the basis of **Markov Chain Monte Carlo**

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
- Note that we require the state i to have the same number of neighbours as state j so that detailed balance is satisfied
- This is the basis of **Markov Chain Monte Carlo**

Continuous Variables

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

$$W(\boldsymbol{\theta} \to \boldsymbol{\theta}')\,\pi(\boldsymbol{\theta}) = W(\boldsymbol{\theta}' \to \boldsymbol{\theta})\,\pi(\boldsymbol{\theta}')$$

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

Continuous Variables

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

$$W(\boldsymbol{\theta} \to \boldsymbol{\theta}')\,\pi(\boldsymbol{\theta}) = W(\boldsymbol{\theta}' \to \boldsymbol{\theta})\,\pi(\boldsymbol{\theta}')$$

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

Continuous Variables

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

$$W(\boldsymbol{\theta} \to \boldsymbol{\theta}')\,\pi(\boldsymbol{\theta}) = W(\boldsymbol{\theta}' \to \boldsymbol{\theta})\,\pi(\boldsymbol{\theta}')$$

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

Continuous Variables

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

$$W(\boldsymbol{\theta} \to \boldsymbol{\theta}')\,\pi(\boldsymbol{\theta}) = W(\boldsymbol{\theta}' \to \boldsymbol{\theta})\,\pi(\boldsymbol{\theta}')$$

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

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 $\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\,\mathbb{P}(\boldsymbol{\theta})$
- We don't need to know $\mathbb{P}(\mathcal{D})$ which we generally don't know

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 $\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\, \mathbb{P}(\boldsymbol{\theta})$
- We don't need to know $\mathbb{P}(\mathcal{D})$ which we generally don't know

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 $\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\,\mathbb{P}(\boldsymbol{\theta})$
- We don't need to know $\mathbb{P}(\mathcal{D})$ which we generally don't know

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 $\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\,\mathbb{P}(\boldsymbol{\theta})$
- We don't need to know $\mathbb{P}(\mathcal{D})$ which we generally don't know

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 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

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

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

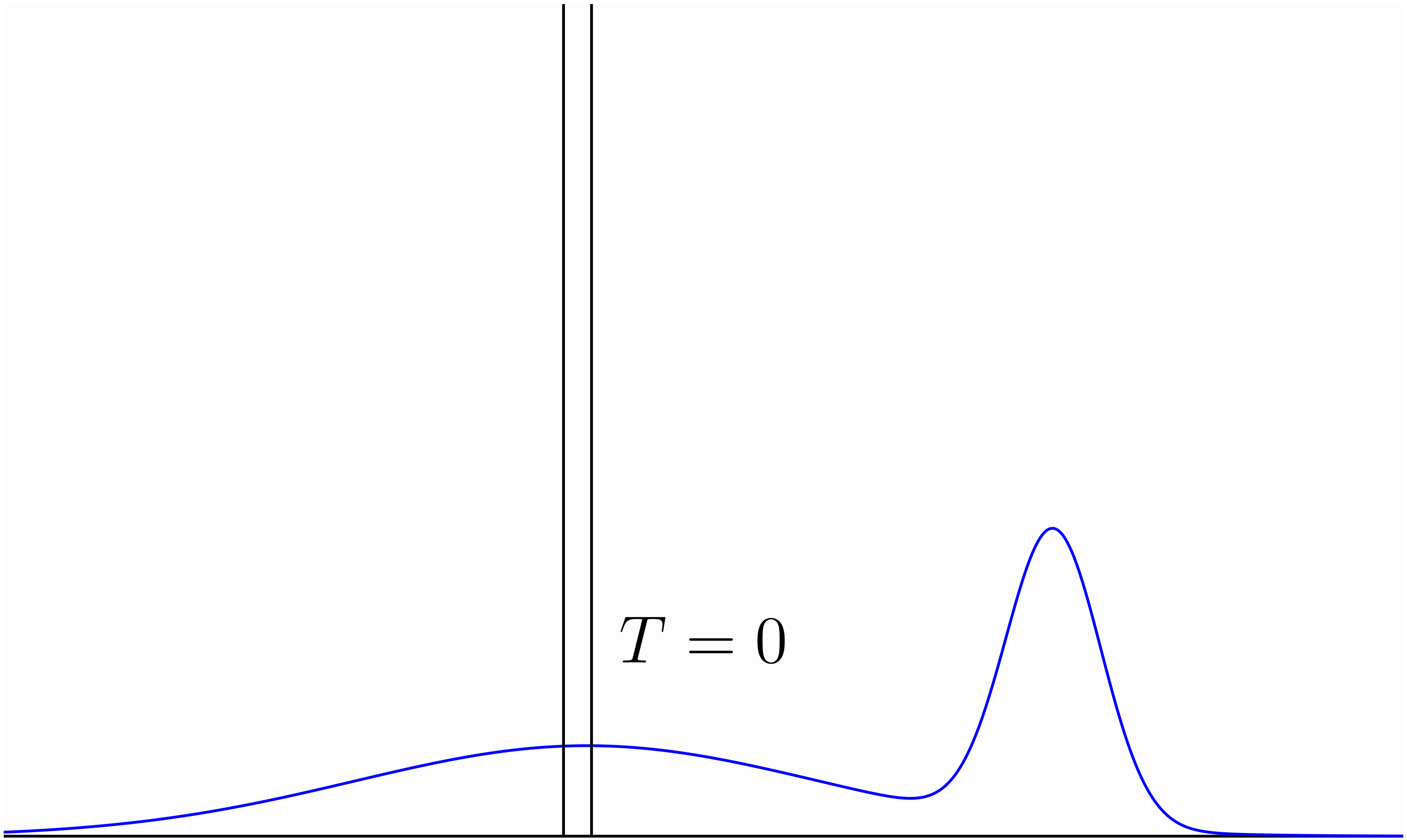
What Makes MCMC Nasty

- 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

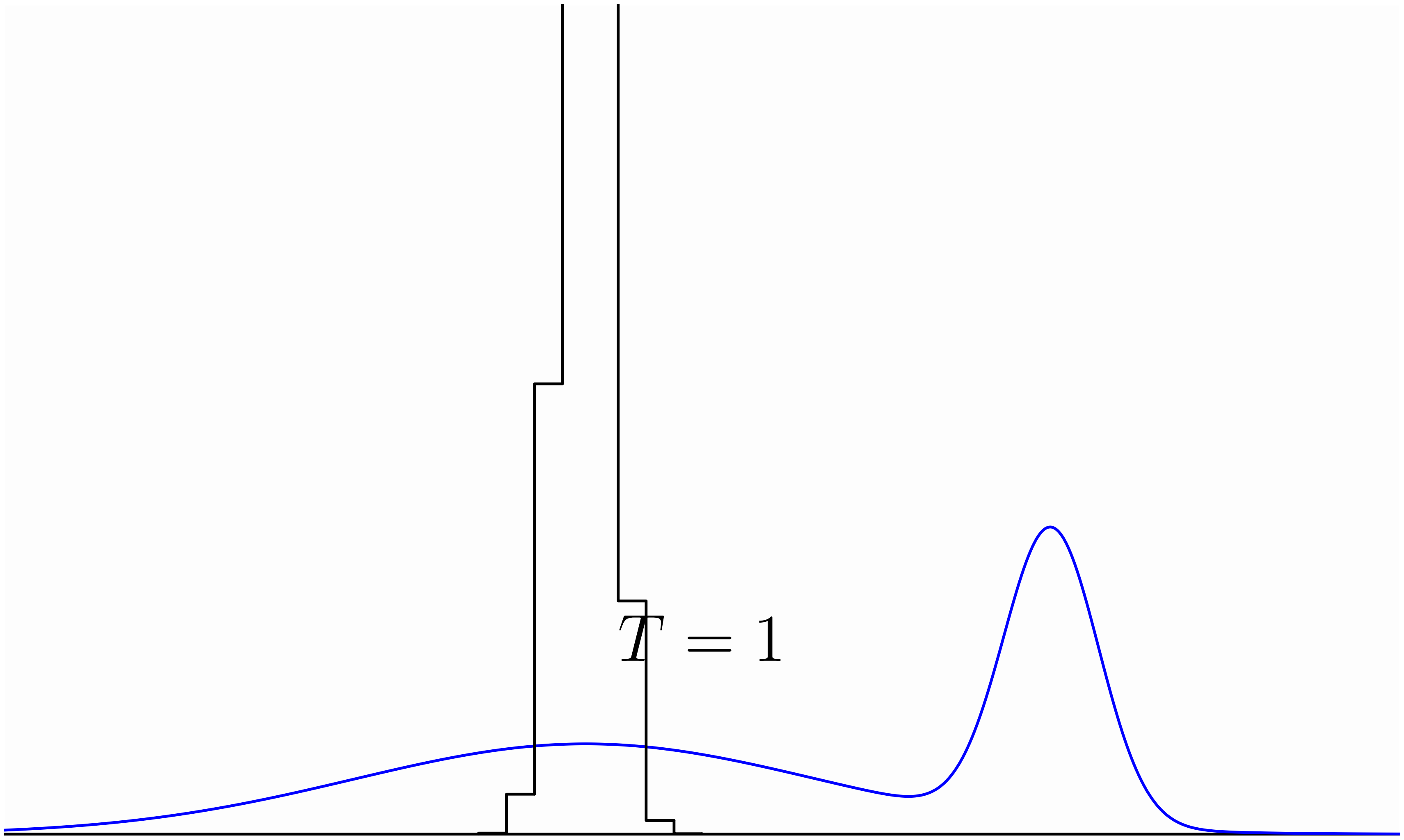
What Makes MCMC Nasty

- 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

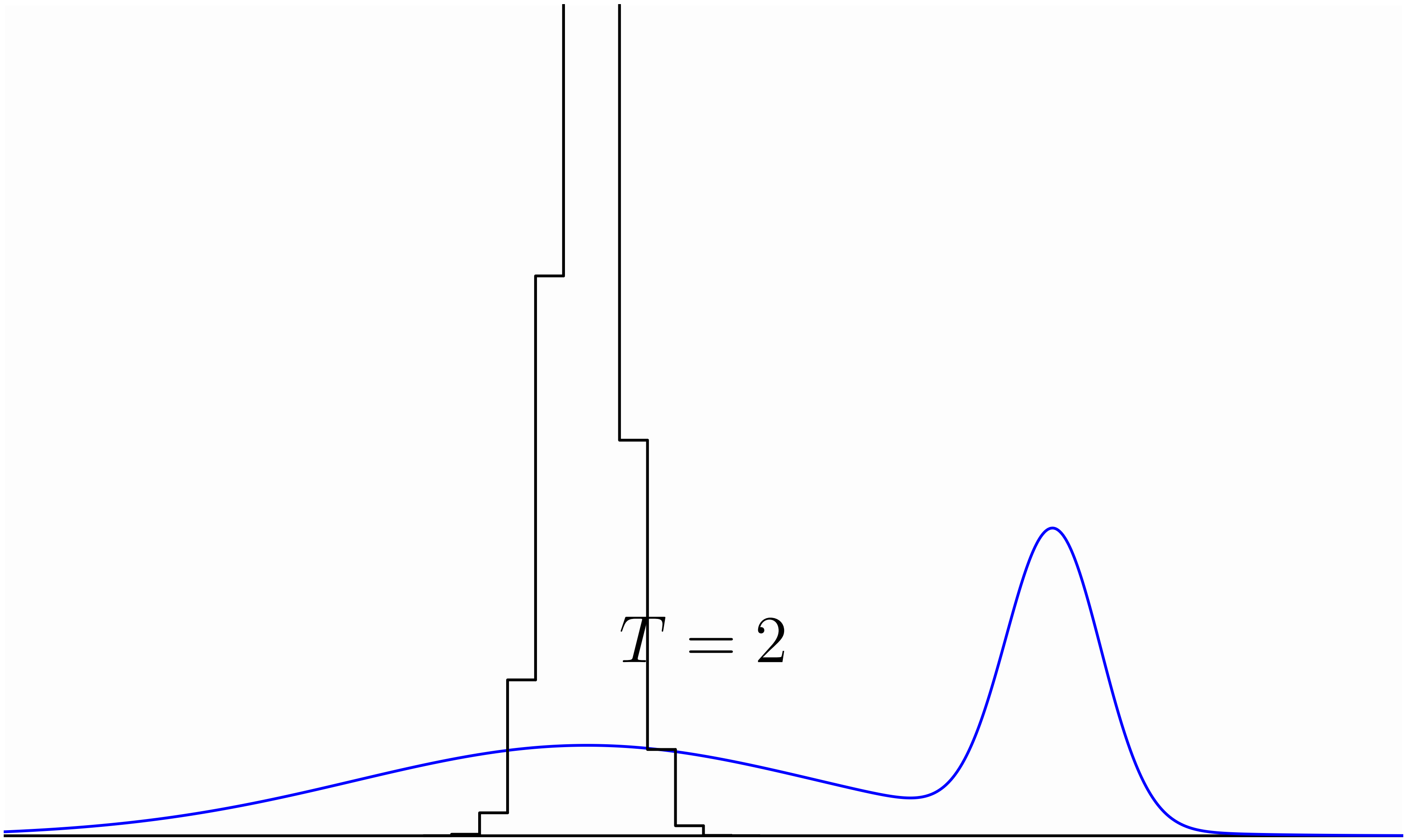
Burn-In



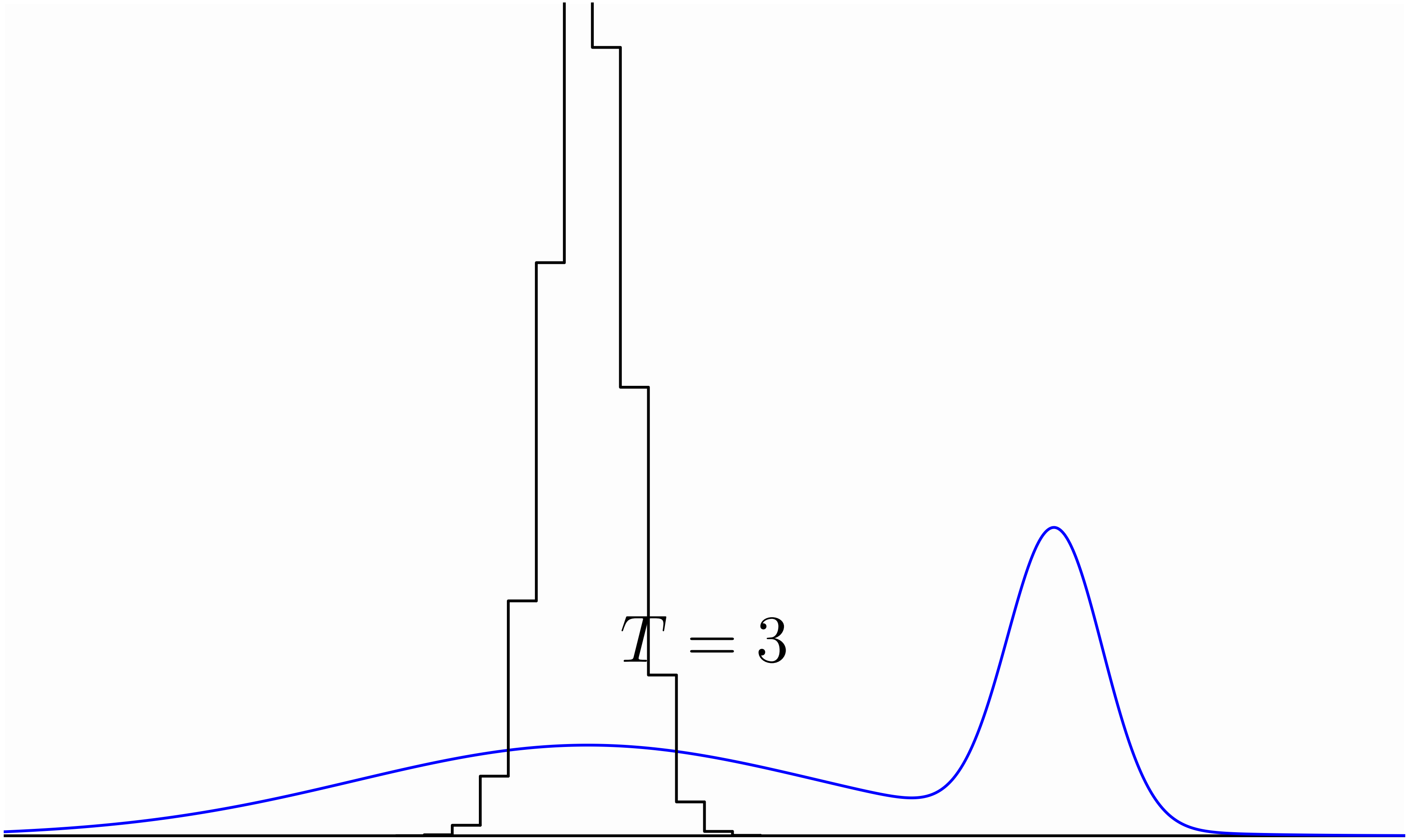
Burn-In



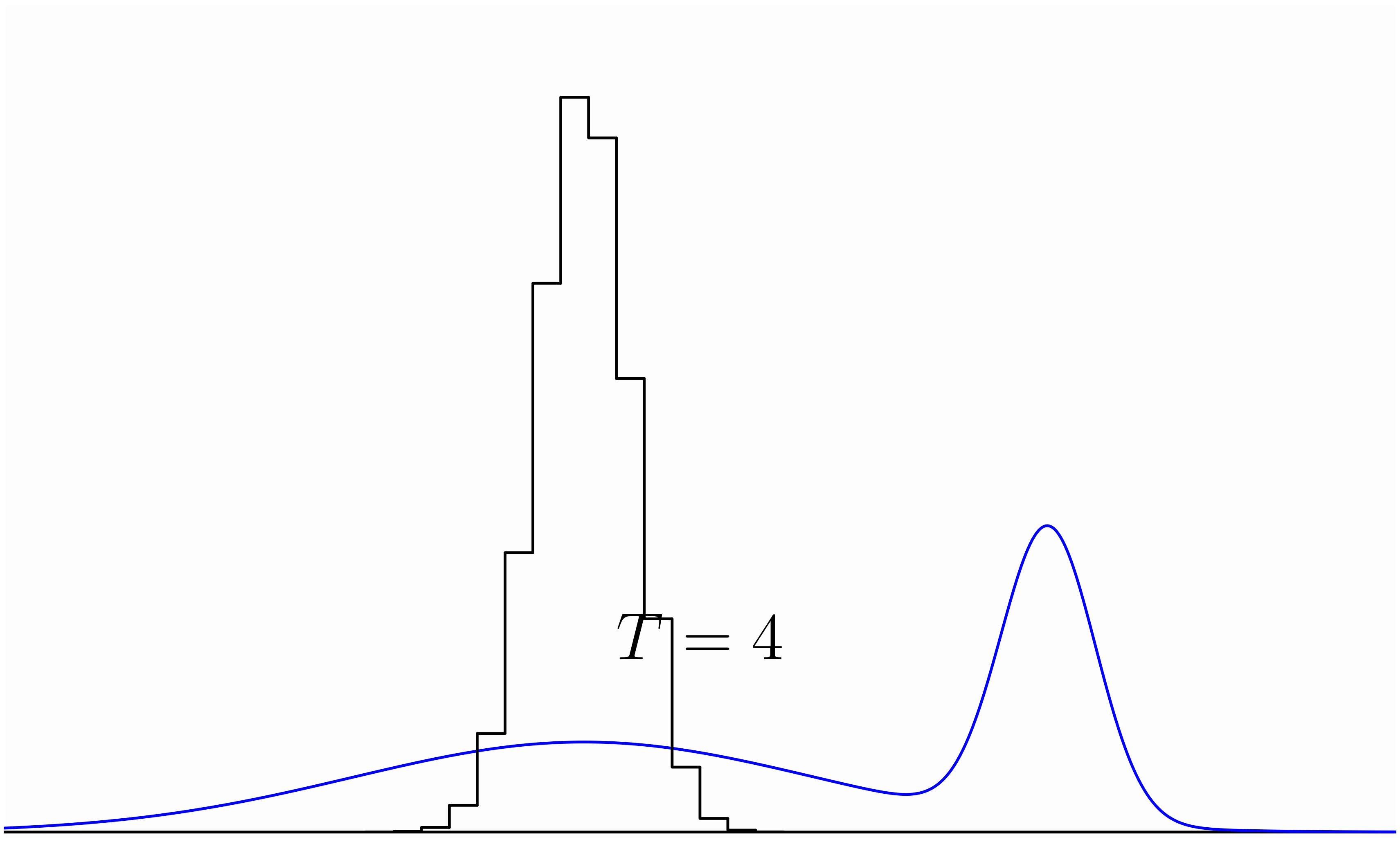
Burn-In



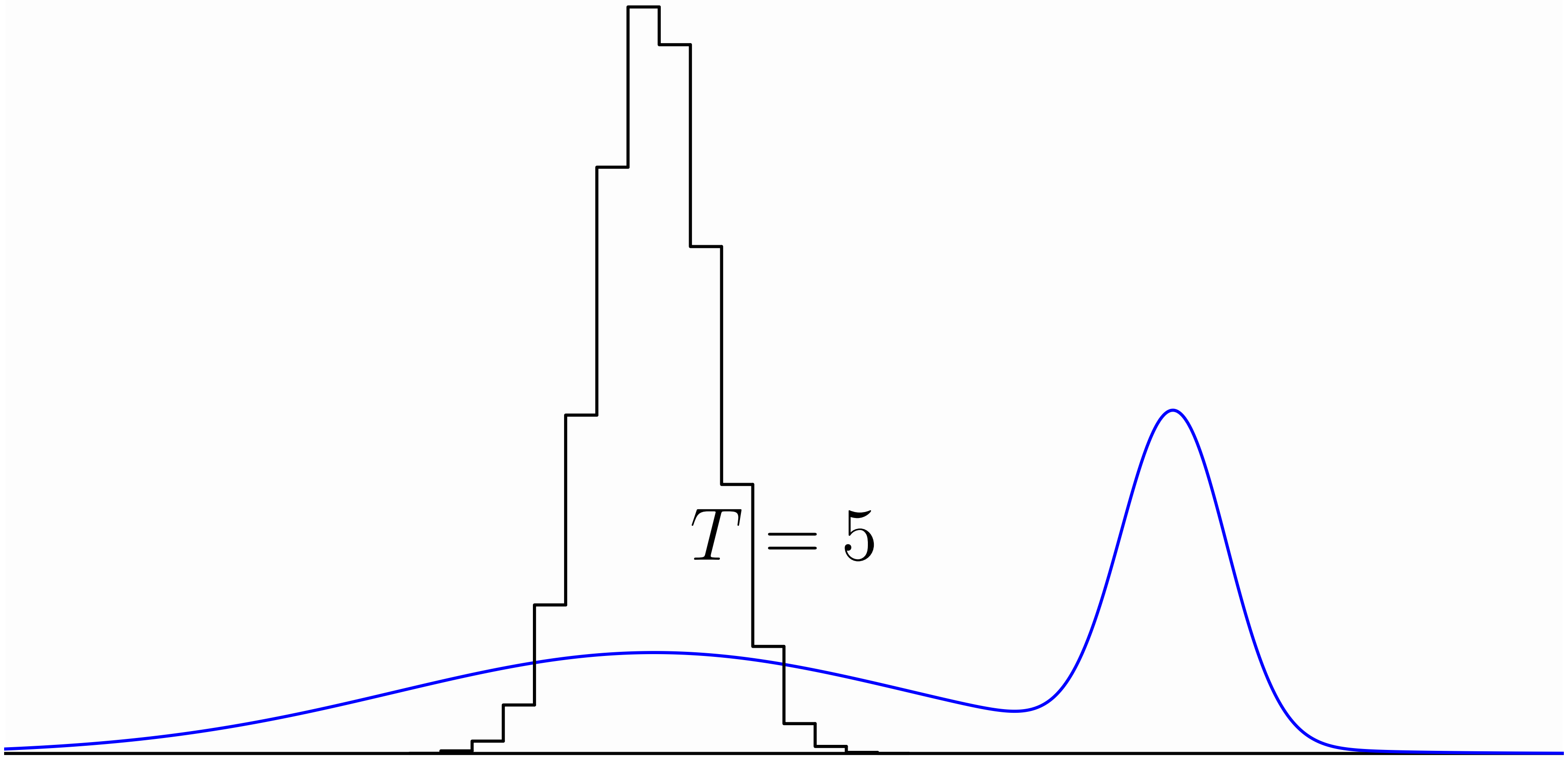
Burn-In



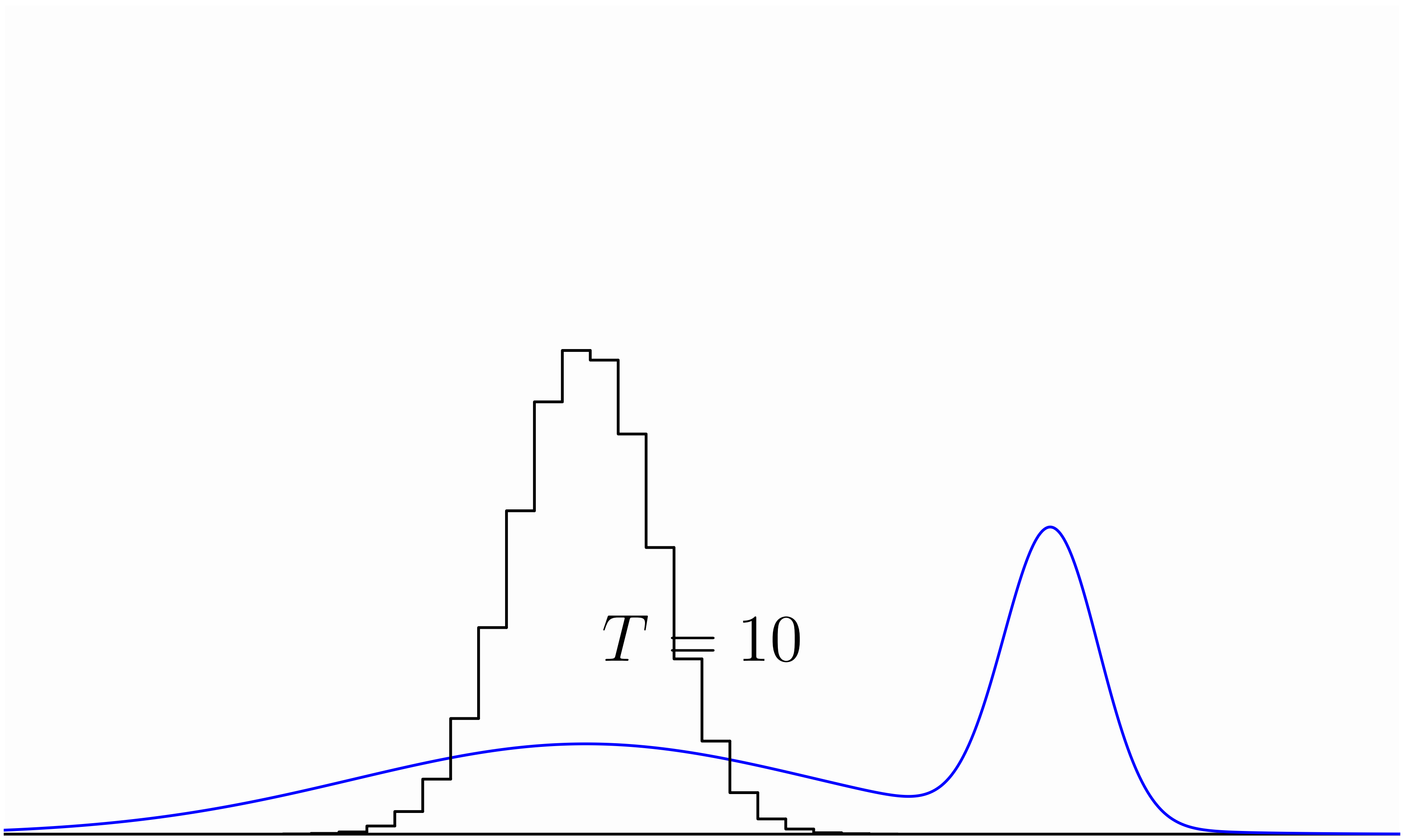
Burn-In



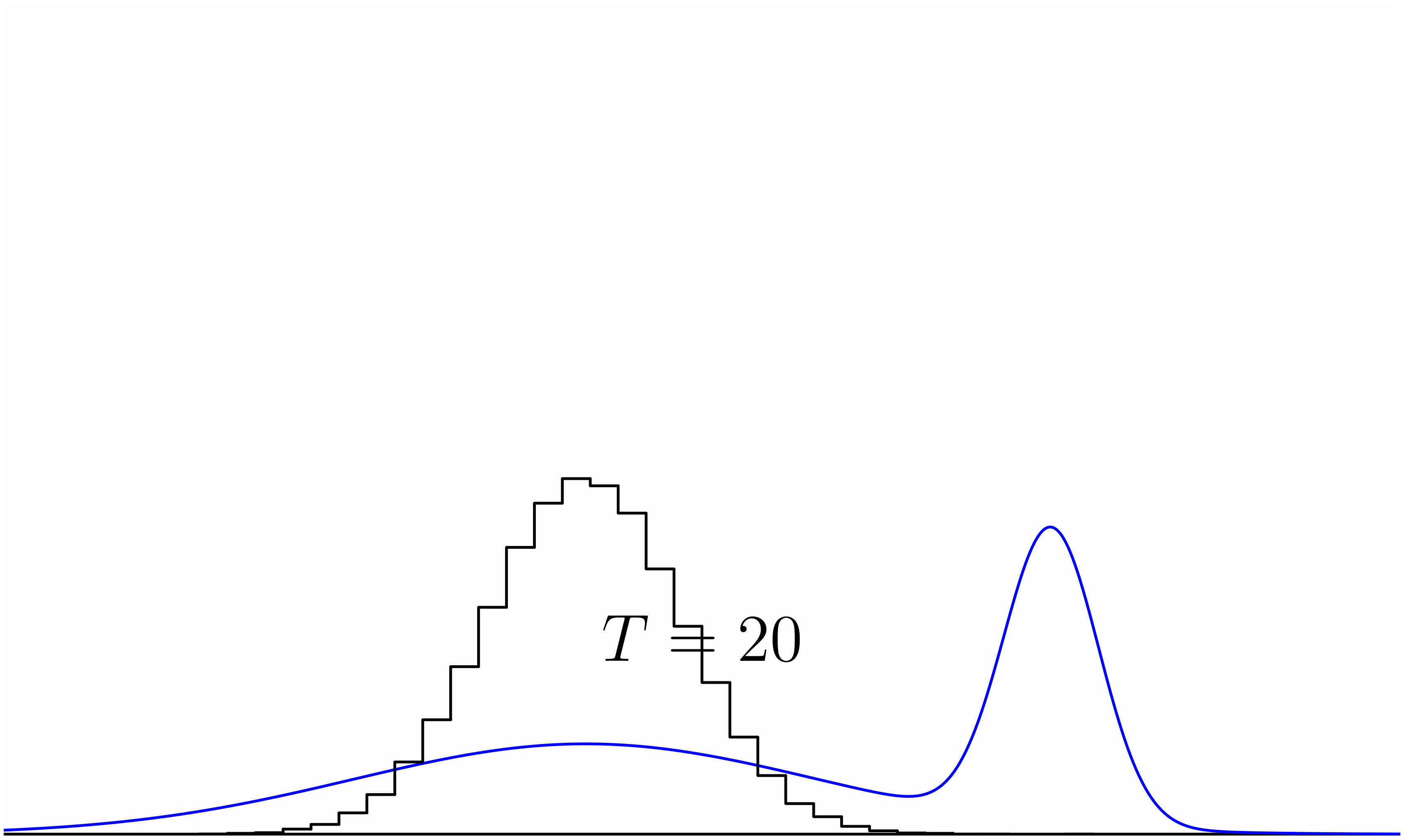
Burn-In



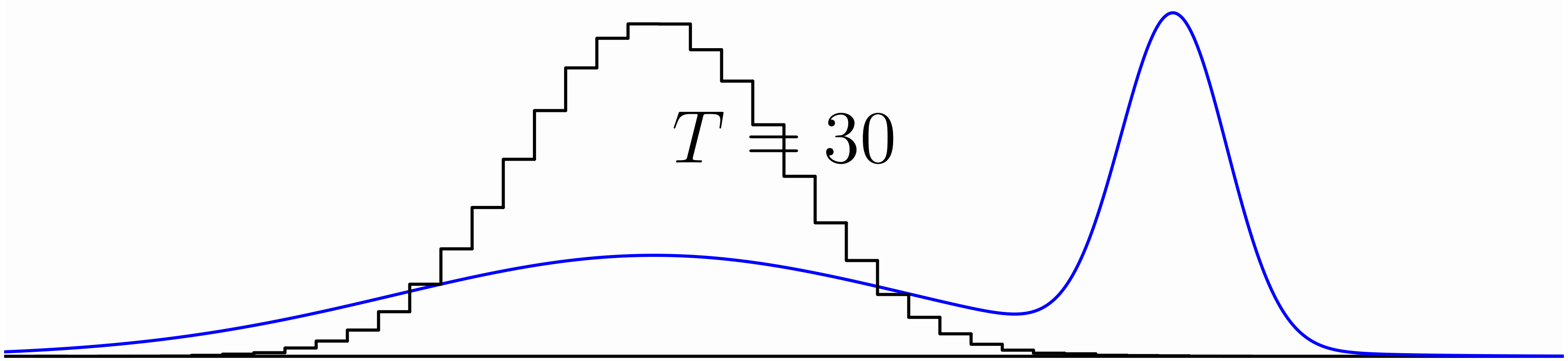
Burn-In



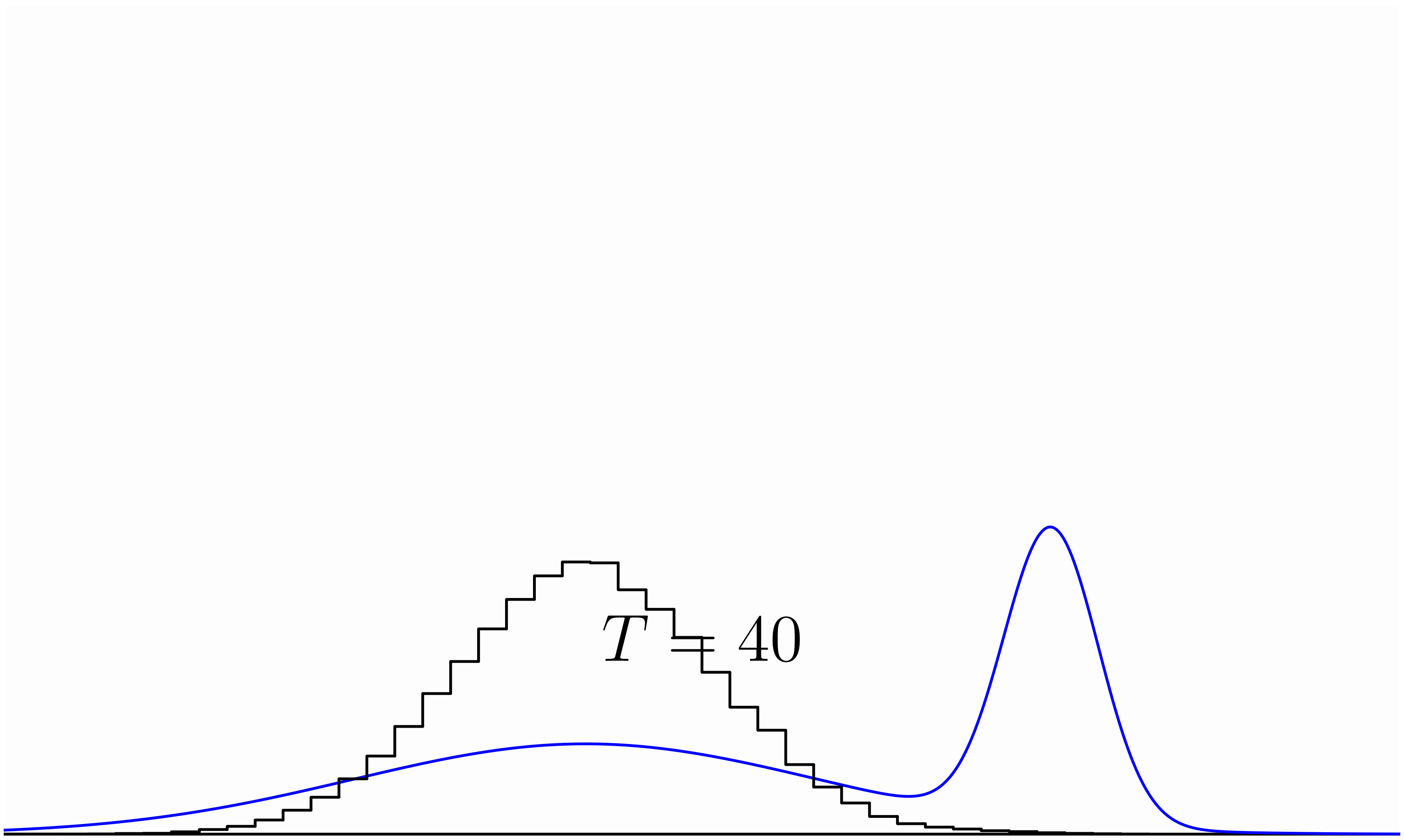
Burn-In



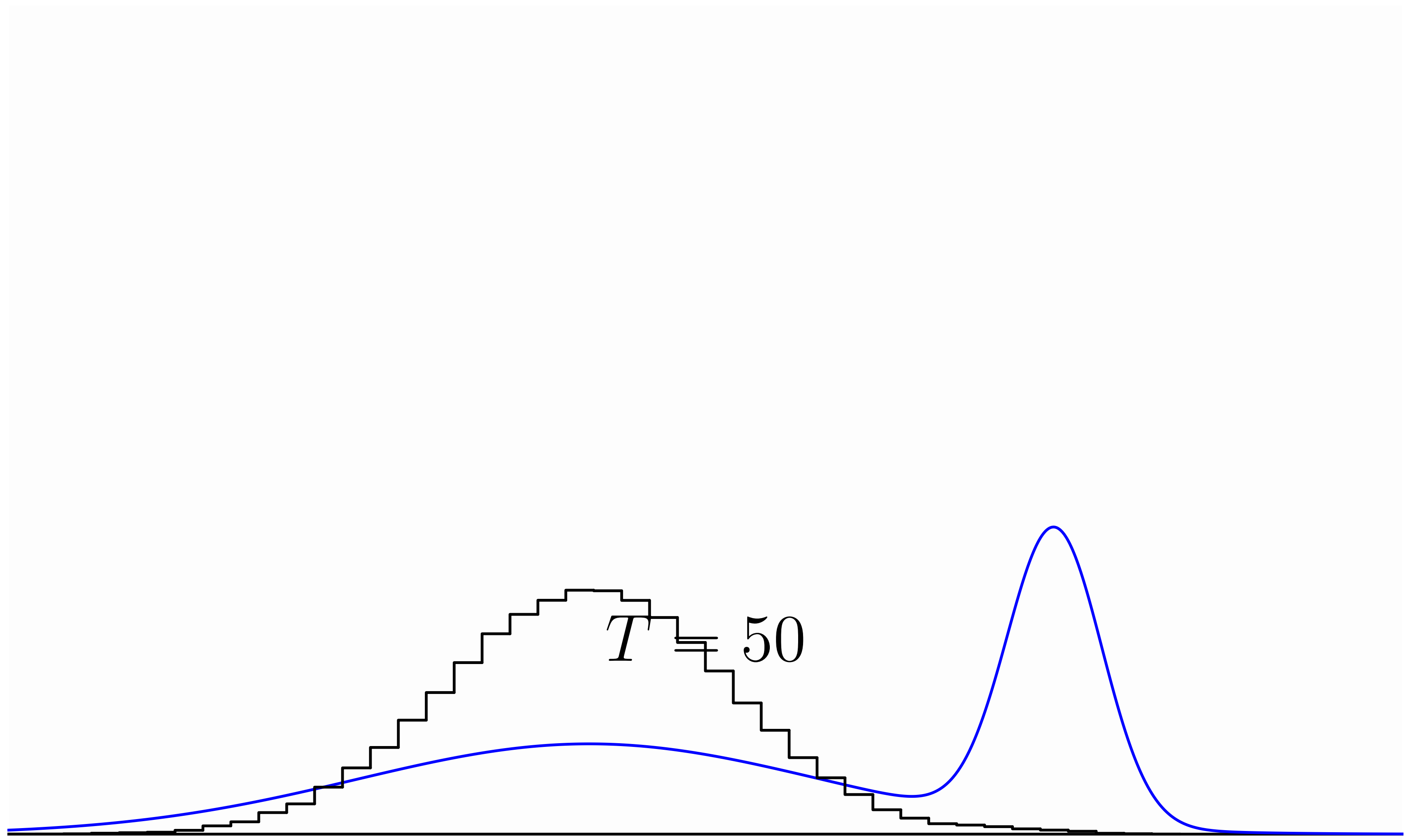
Burn-In



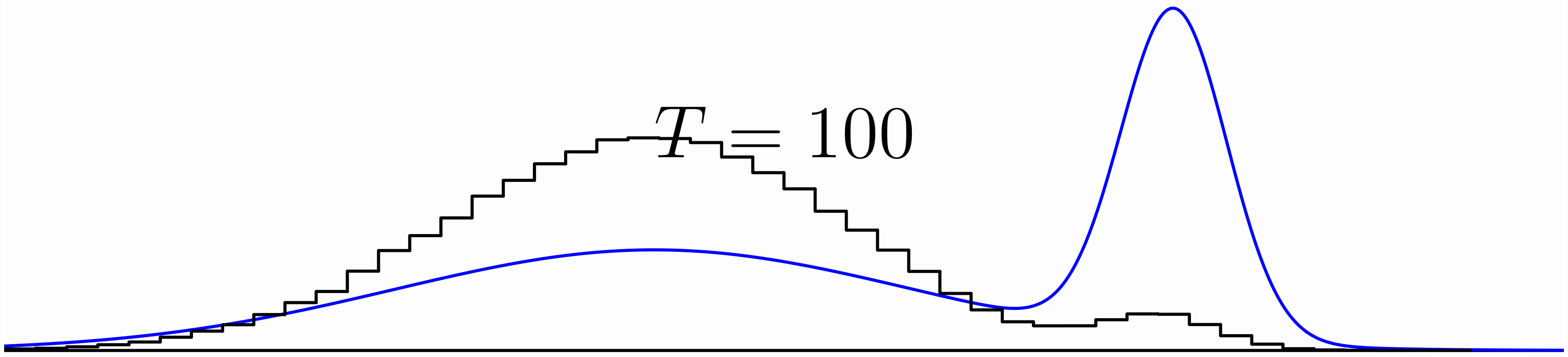
Burn-In



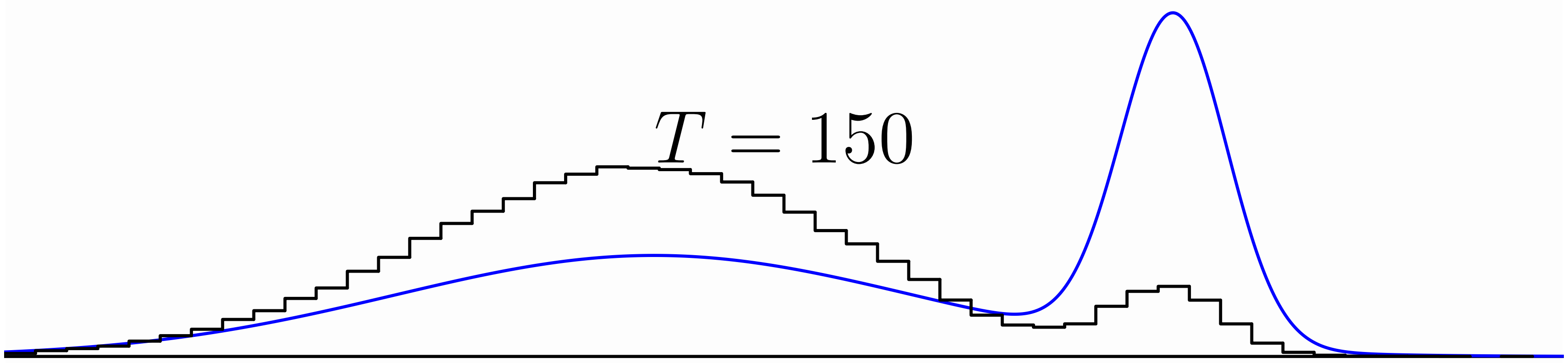
Burn-In



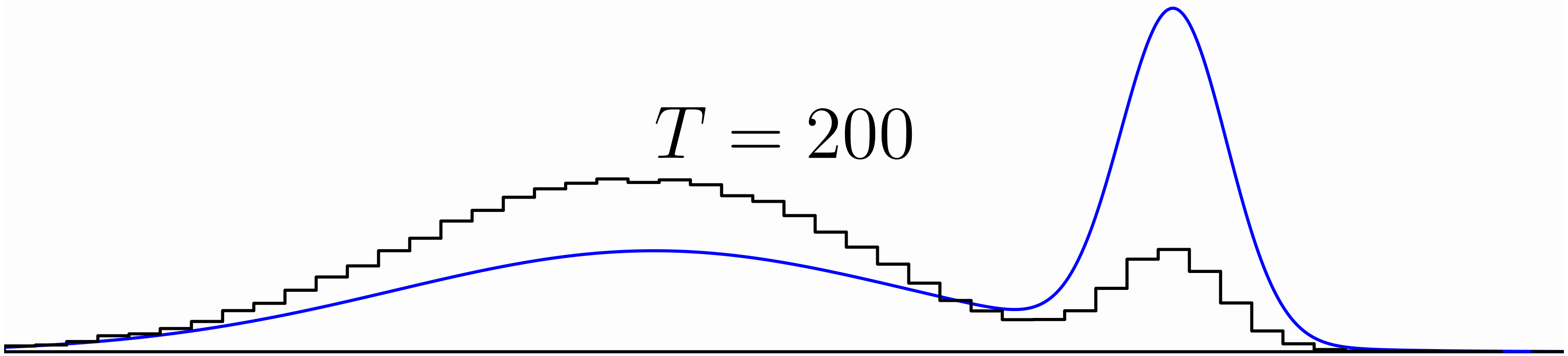
Burn-In



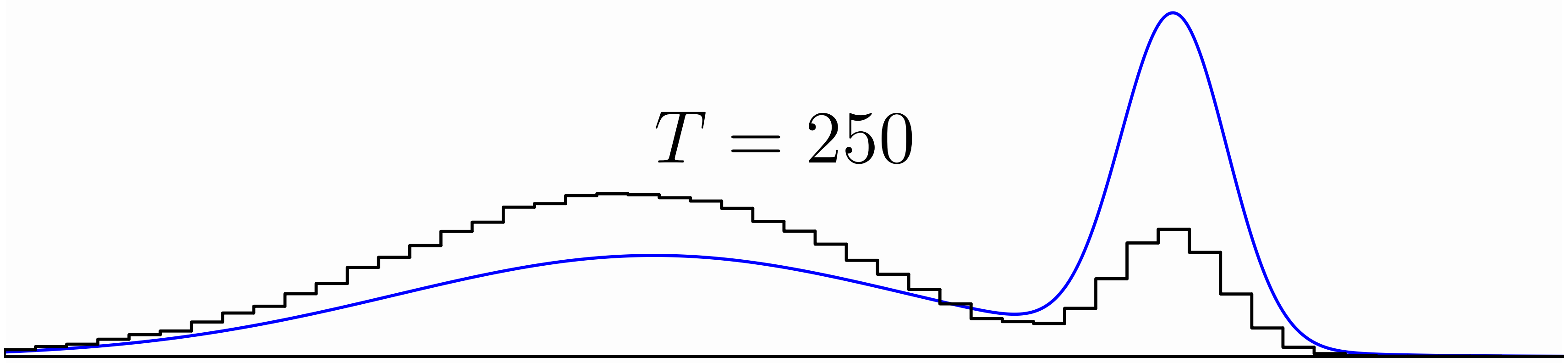
Burn-In



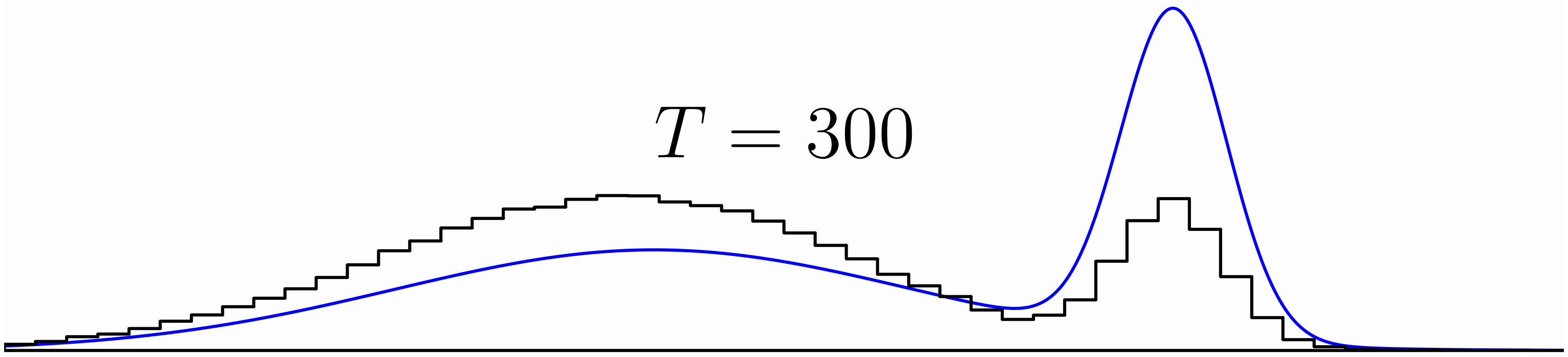
Burn-In



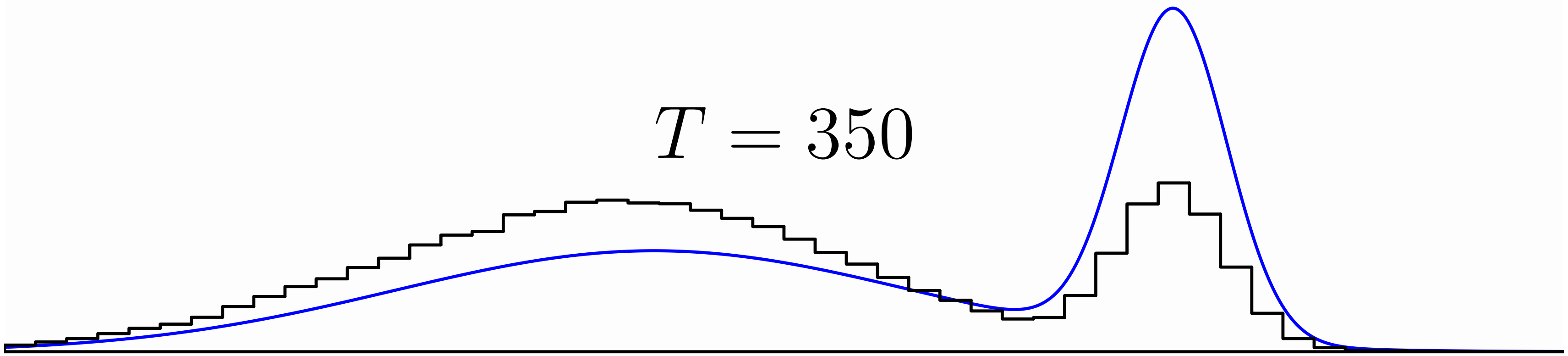
Burn-In



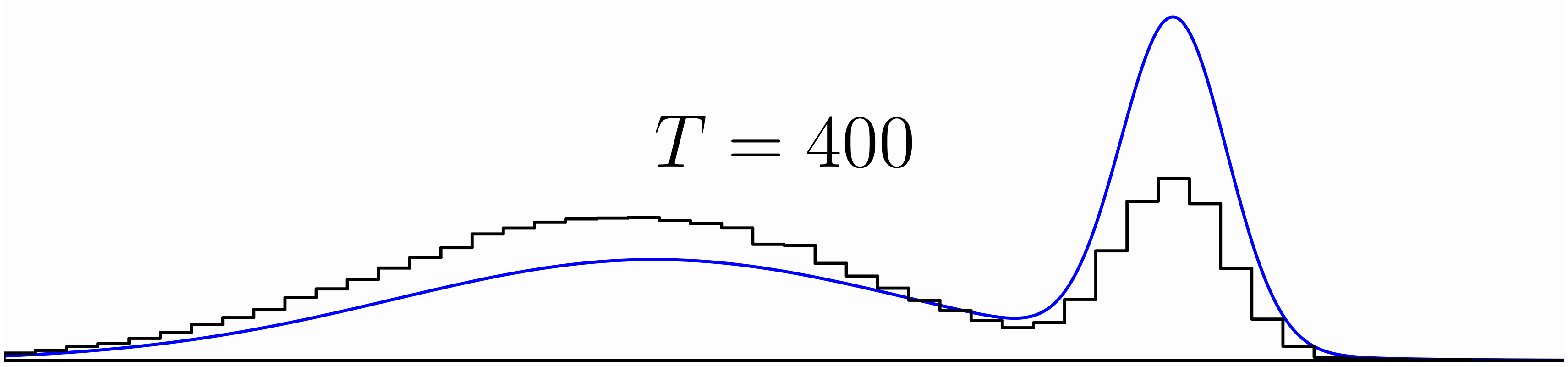
Burn-In



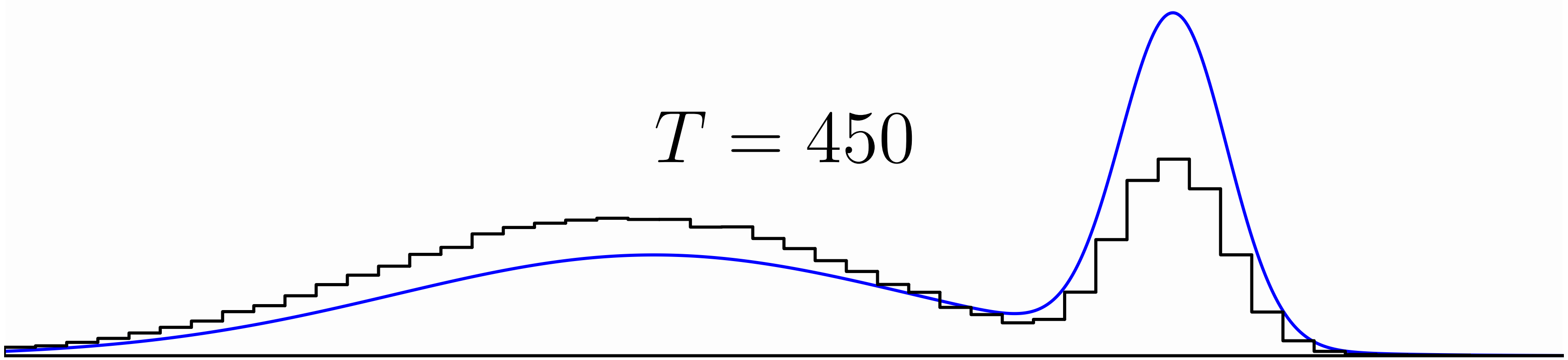
Burn-In



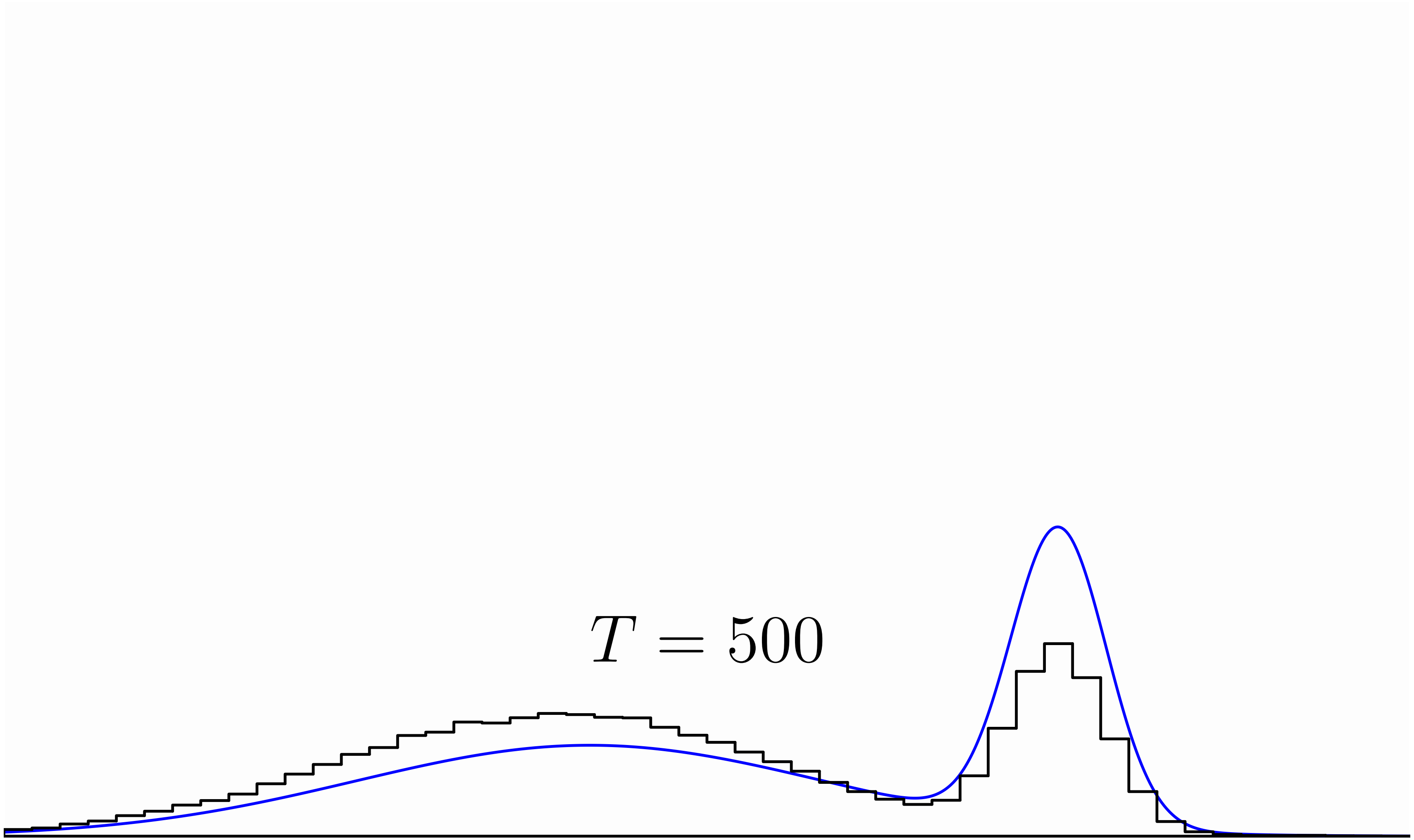
Burn-In



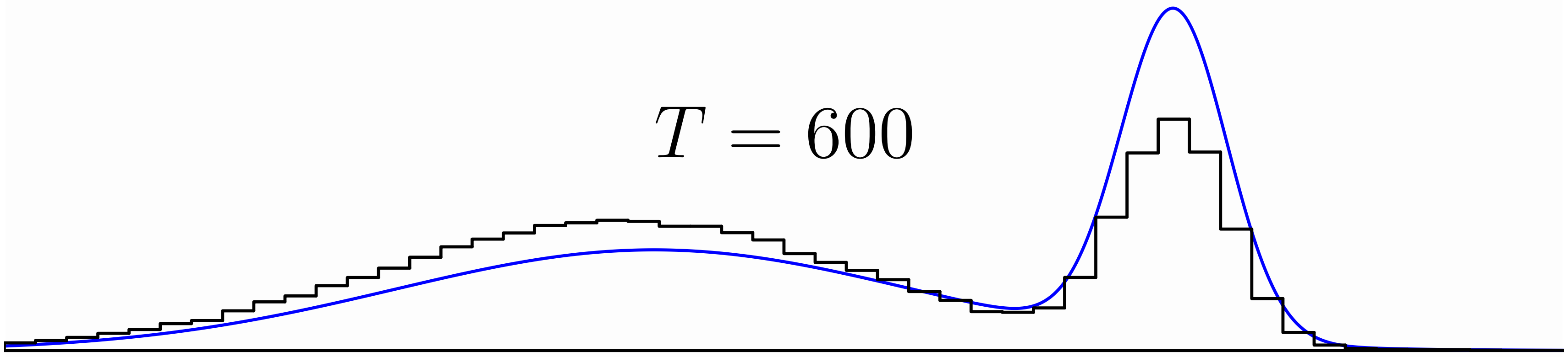
Burn-In



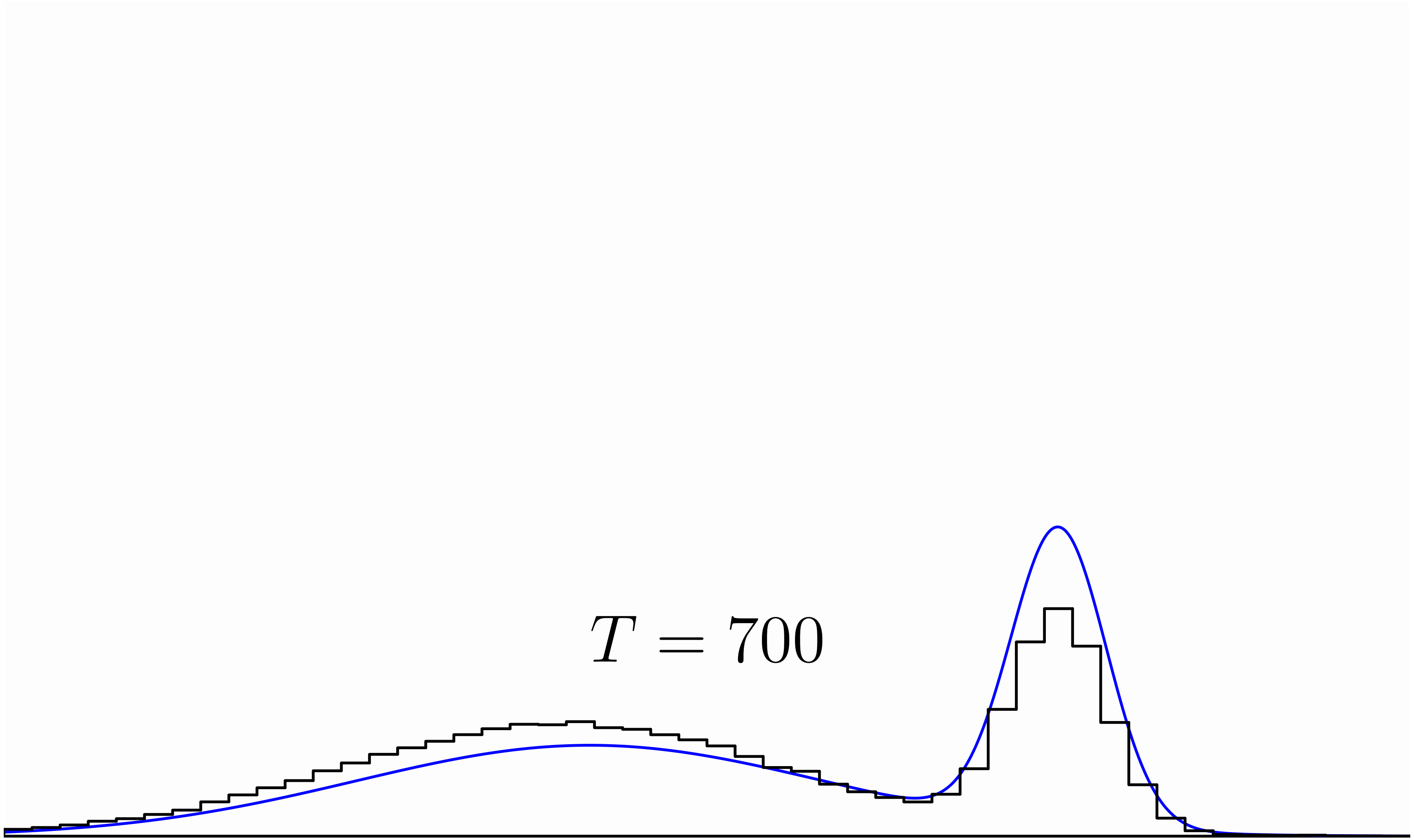
Burn-In



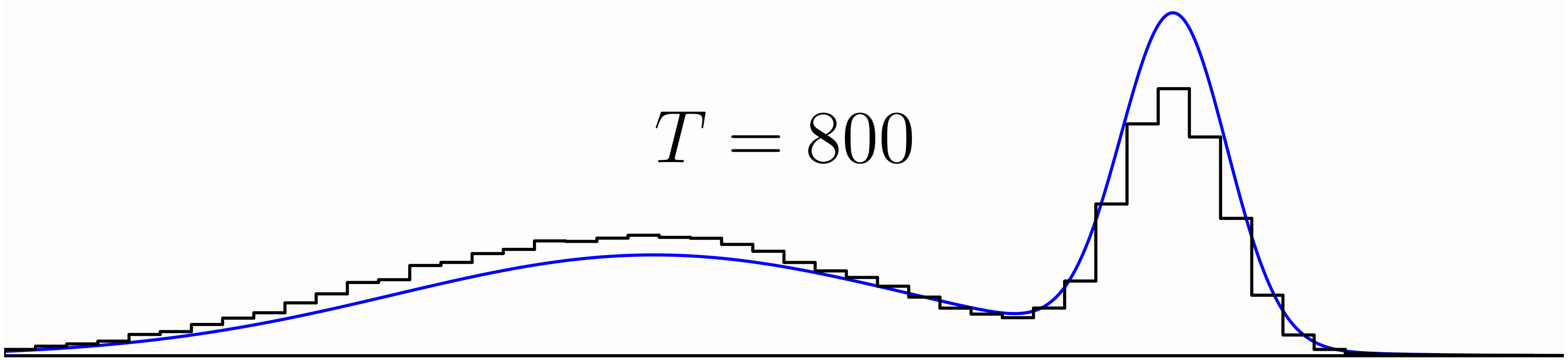
Burn-In



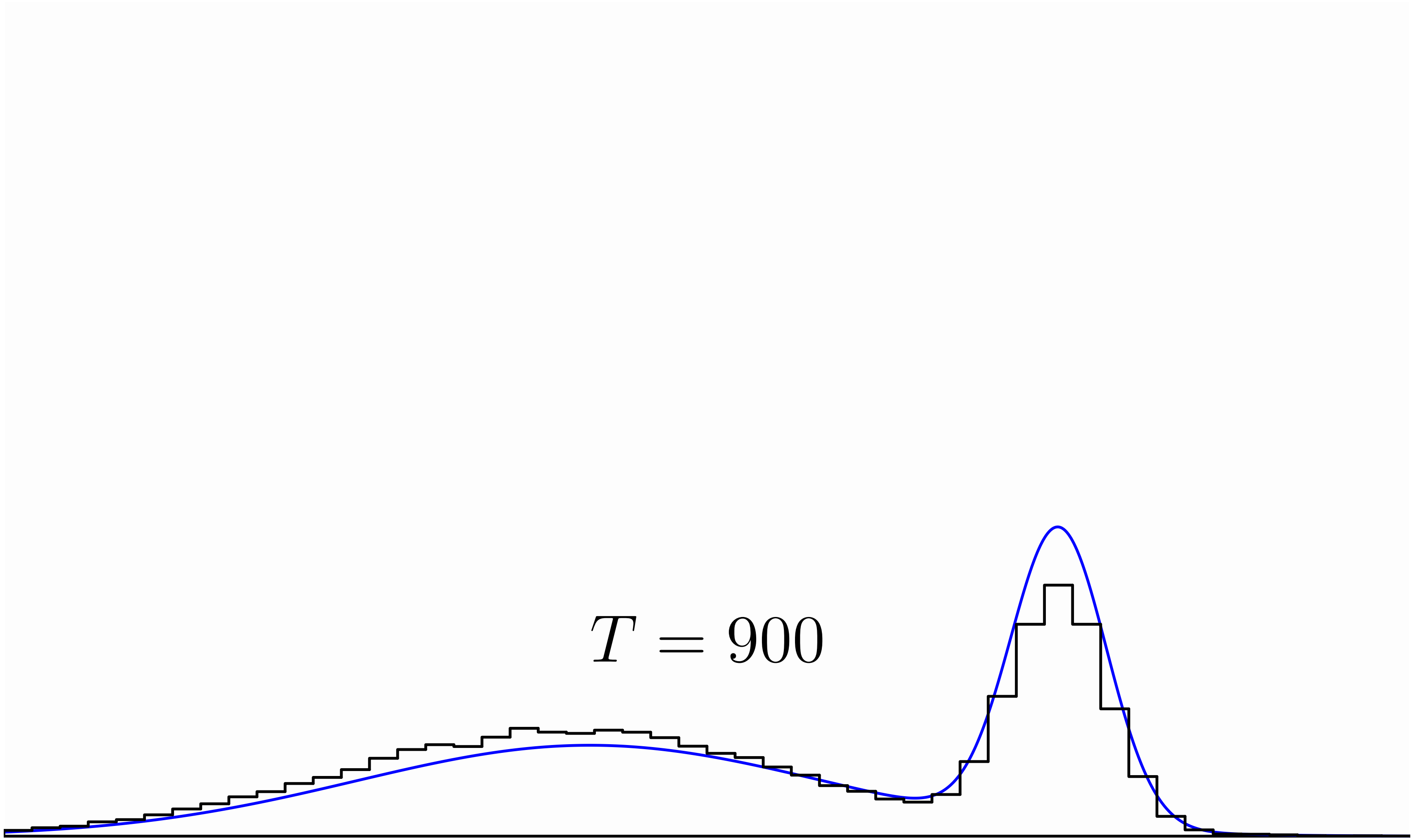
Burn-In



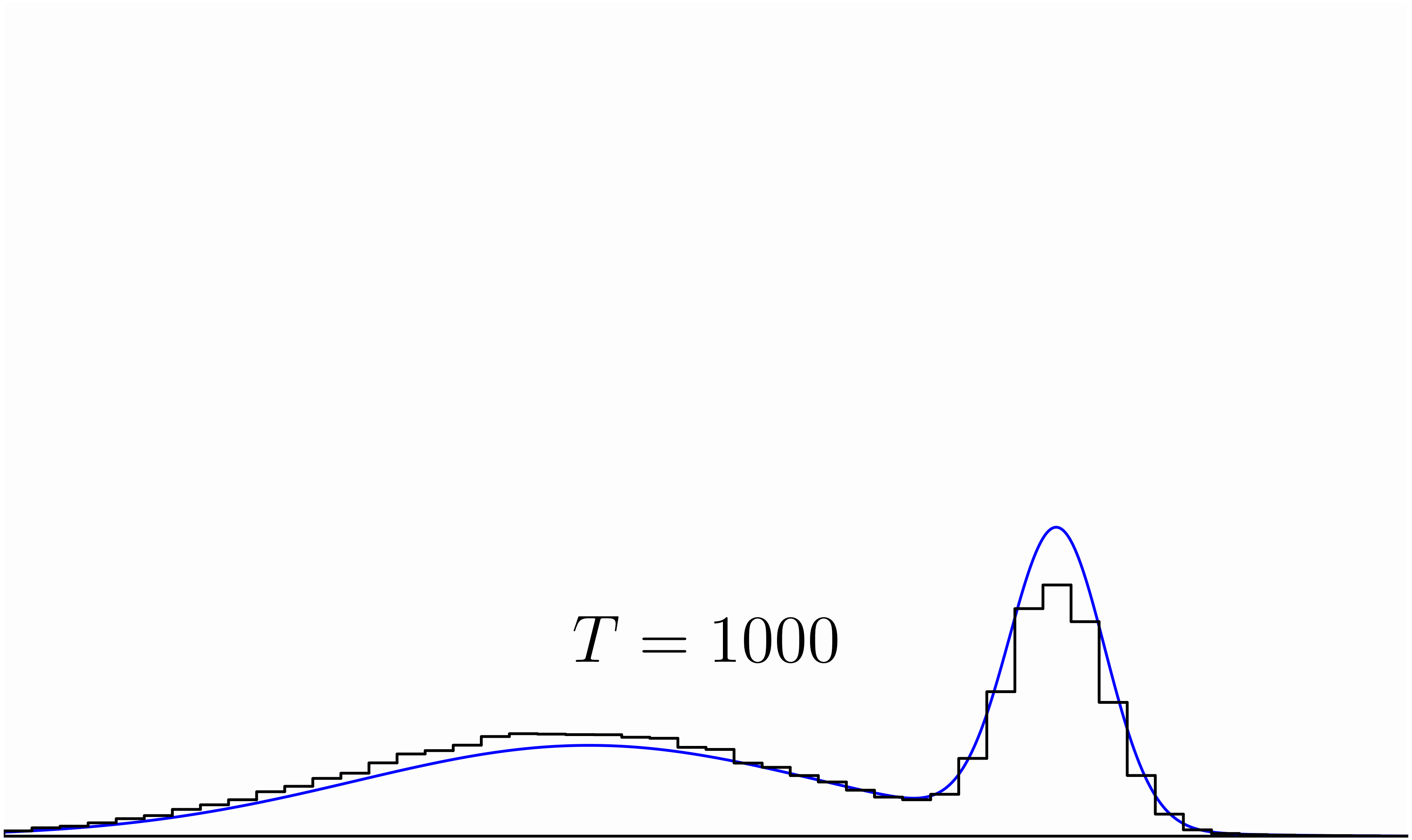
Burn-In



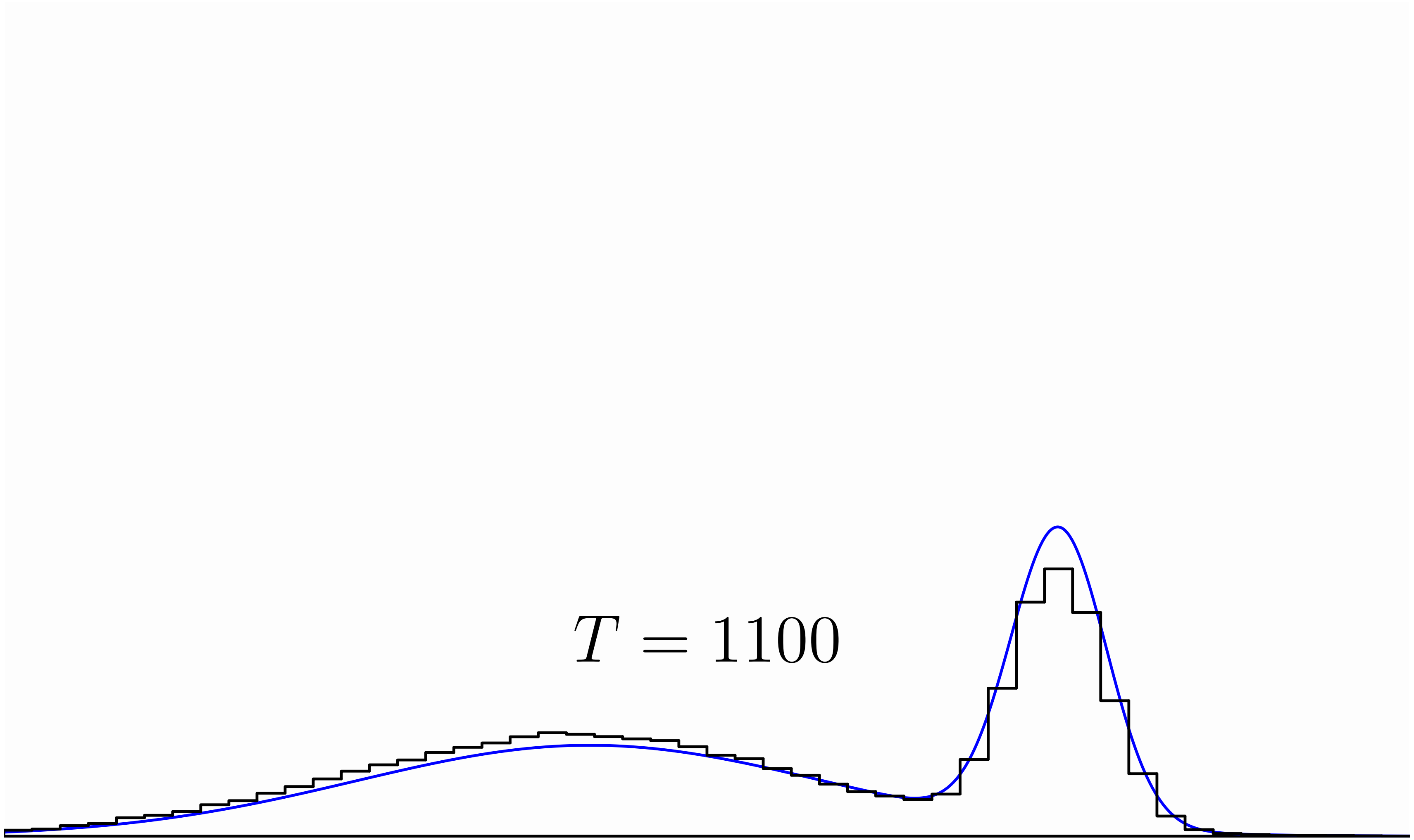
Burn-In



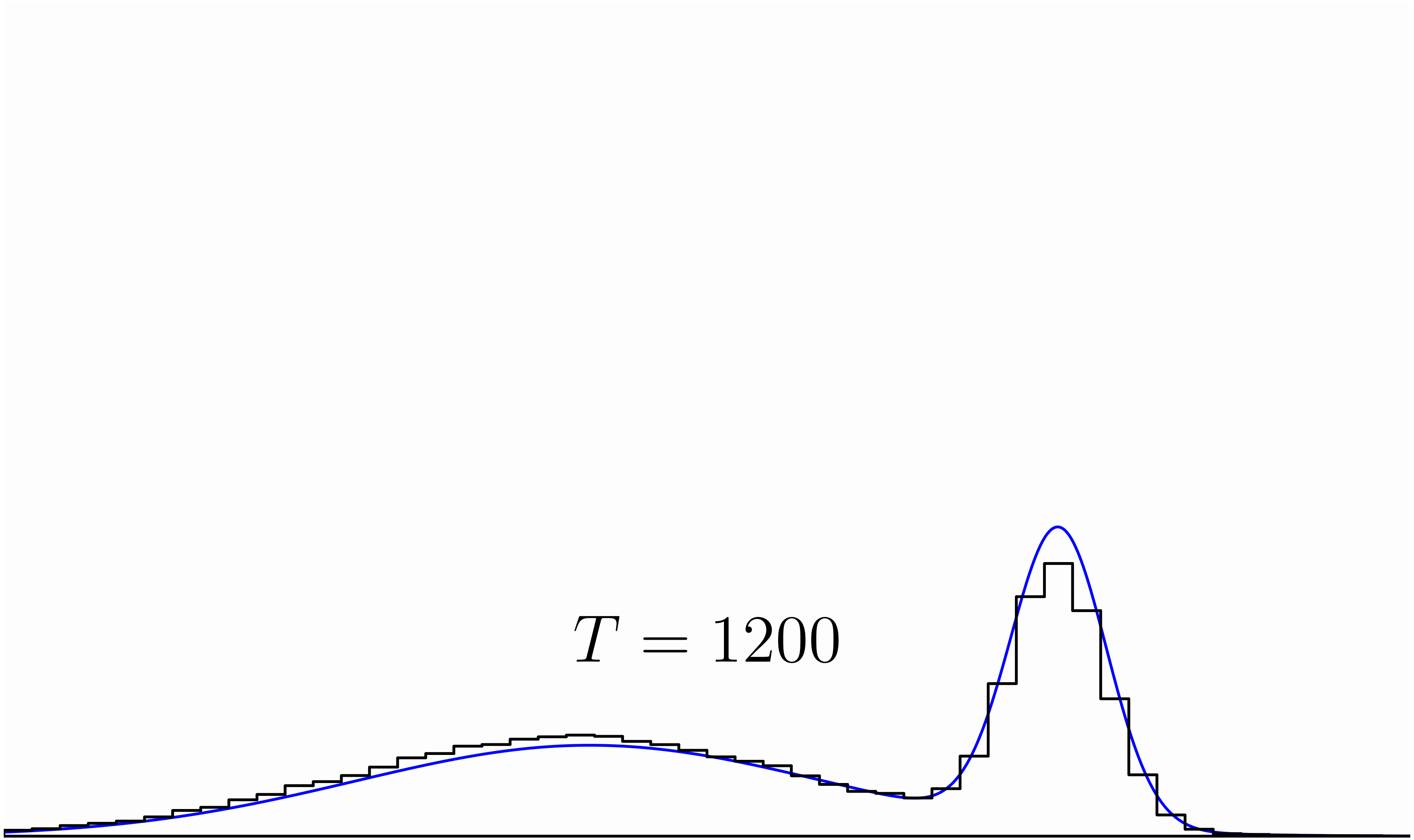
Burn-In



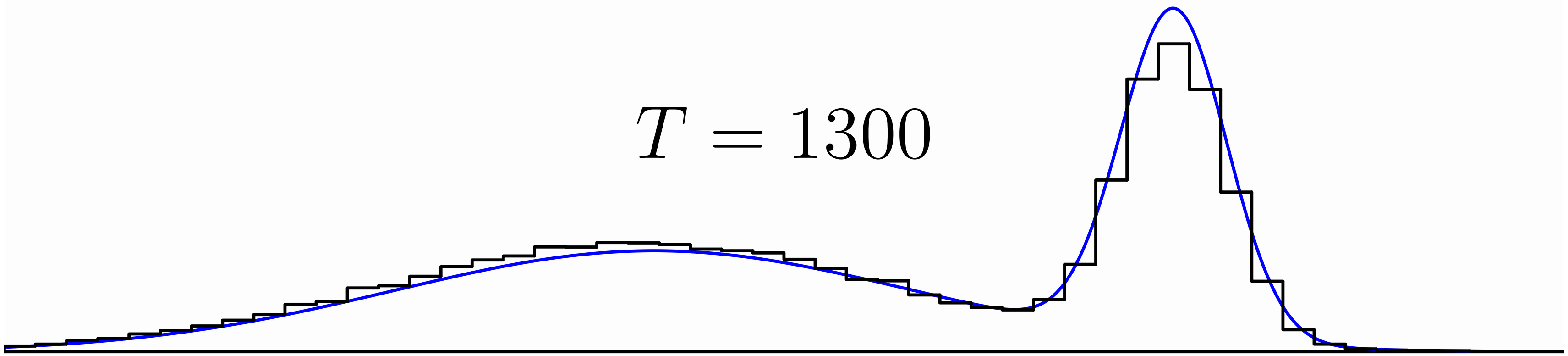
Burn-In



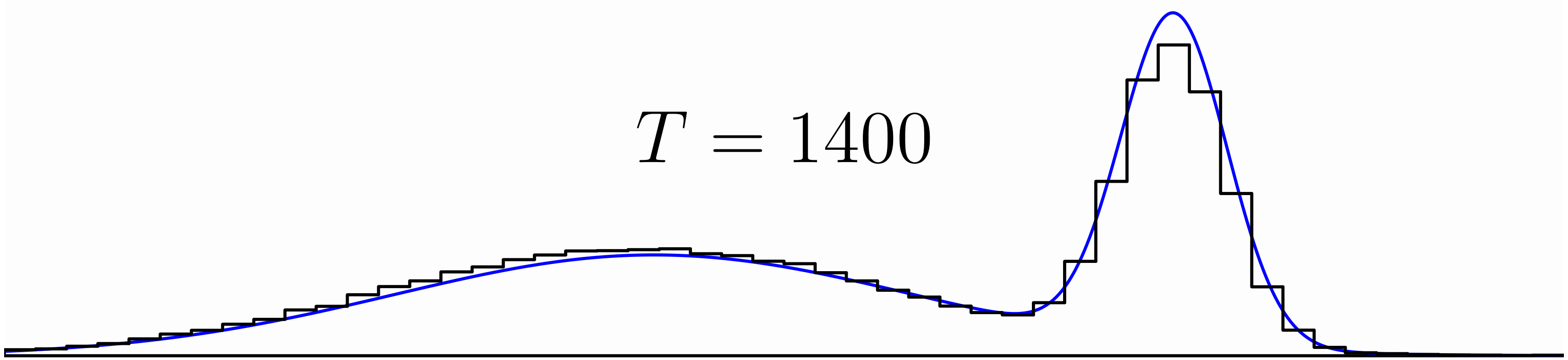
Burn-In



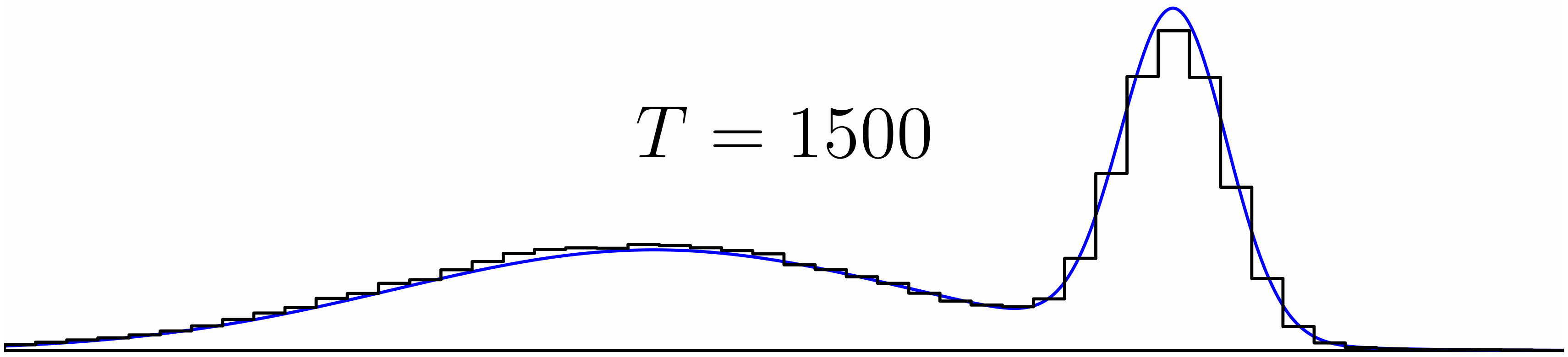
Burn-In



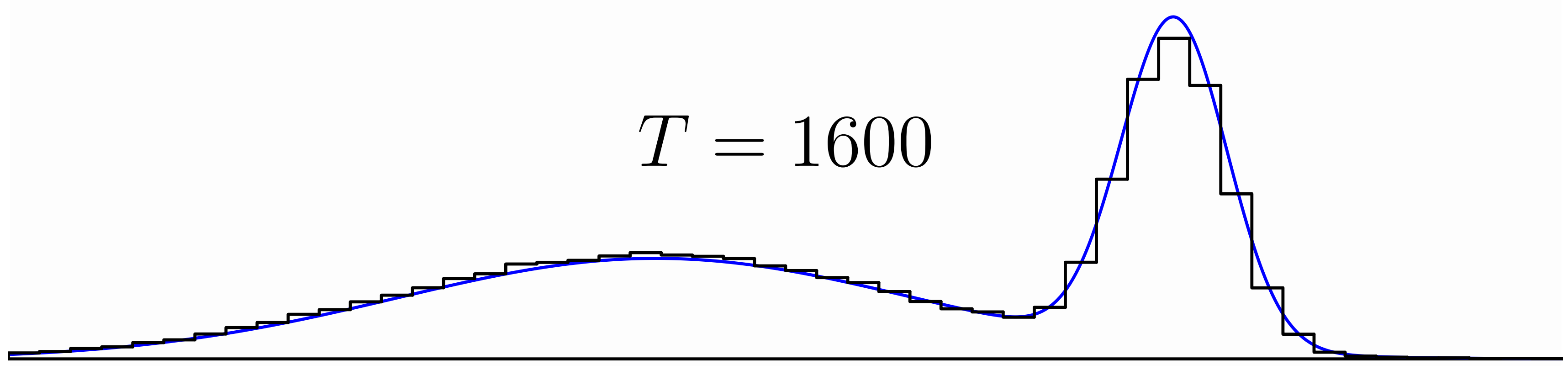
Burn-In



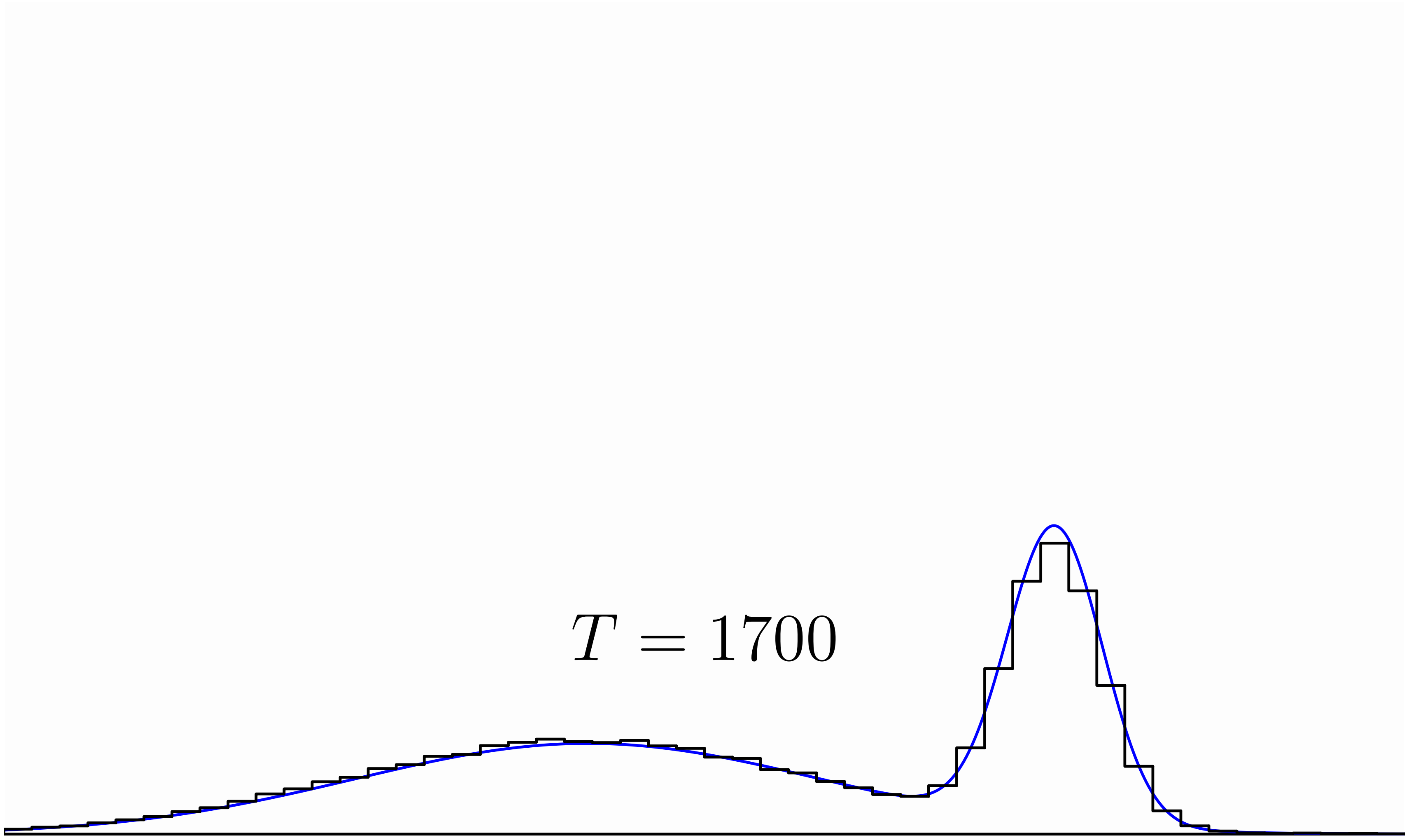
Burn-In



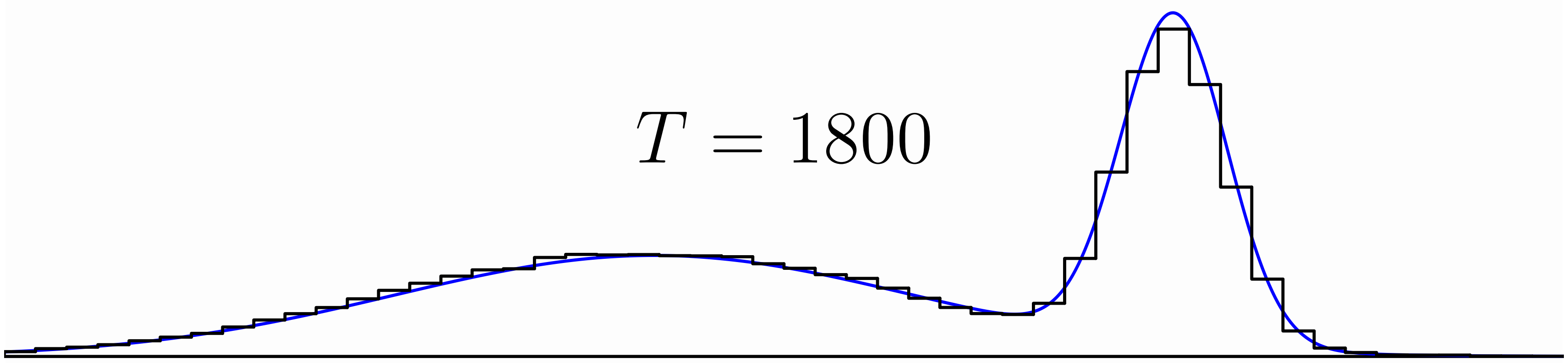
Burn-In



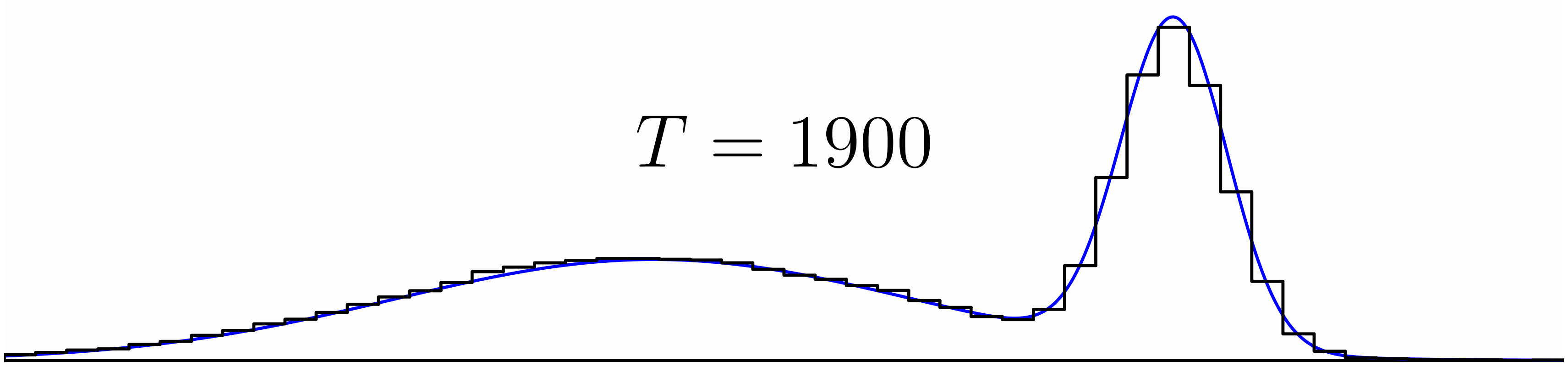
Burn-In



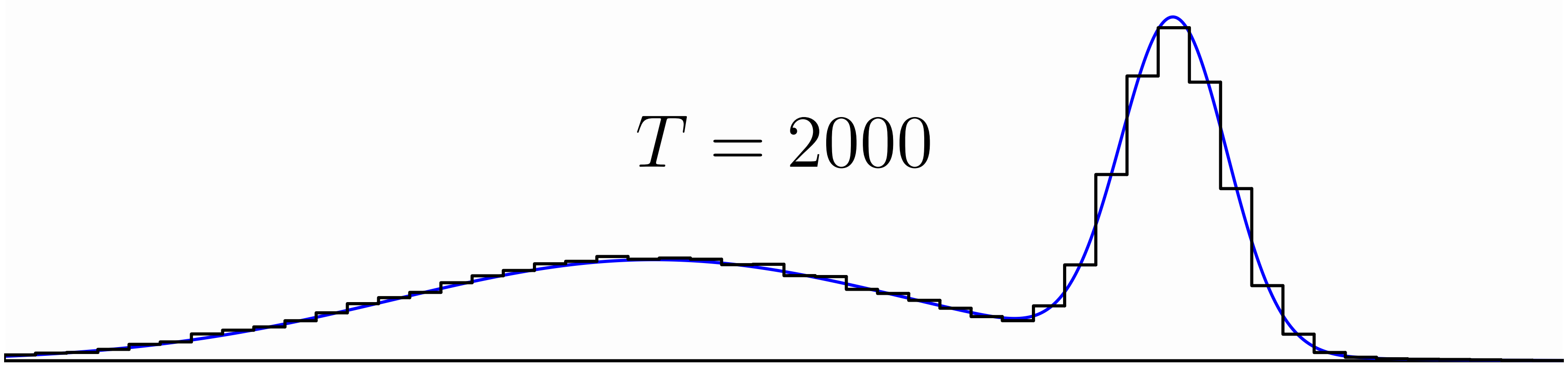
Burn-In



Burn-In



Burn-In



Proposals and Metropolis-Hastings

- We have some freedom in choosing a new proposal $\boldsymbol{\theta}'$ from our current position $\boldsymbol{\theta}$—a good choice can increase the acceptance rate making the MCMC more efficient

- We define the proposal distribution $p(\boldsymbol{\theta}'|\boldsymbol{\theta})$

- For the standard Metropolis algorithm to work we require $p(\boldsymbol{\theta}'|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{\theta}')$

- 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

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

Proposals and Metropolis-Hastings

- We have some freedom in choosing a new proposal $\boldsymbol{\theta}'$ from our current position $\boldsymbol{\theta}$—a good choice can increase the acceptance rate making the MCMC more efficient

- We define the proposal distribution $p(\boldsymbol{\theta}'|\boldsymbol{\theta})$

- For the standard Metropolis algorithm to work we require $p(\boldsymbol{\theta}'|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{\theta}')$

- 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

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

Proposals and Metropolis-Hastings

- We have some freedom in choosing a new proposal $\boldsymbol{\theta}'$ from our current position $\boldsymbol{\theta}$—a good choice can increase the acceptance rate making the MCMC more efficient
- We define the proposal distribution $p(\boldsymbol{\theta}'|\boldsymbol{\theta})$
- For the standard Metropolis algorithm to work we require $p(\boldsymbol{\theta}'|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{\theta}')$
- 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

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

Proposals and Metropolis-Hastings

- We have some freedom in choosing a new proposal $\boldsymbol{\theta}'$ from our current position $\boldsymbol{\theta}$—a good choice can increase the acceptance rate making the MCMC more efficient

- We define the proposal distribution $p(\boldsymbol{\theta}'|\boldsymbol{\theta})$

- For the standard Metropolis algorithm to work we require $p(\boldsymbol{\theta}'|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{\theta}')$

- 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

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

Proposals and Metropolis-Hastings

- We have some freedom in choosing a new proposal $\boldsymbol{\theta}'$ from our current position $\boldsymbol{\theta}$—a good choice can increase the acceptance rate making the MCMC more efficient

- We define the proposal distribution $p(\boldsymbol{\theta}'|\boldsymbol{\theta})$

- For the standard Metropolis algorithm to work we require $p(\boldsymbol{\theta}'|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{\theta}')$

- 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

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

Traffic Rate

- Consider monitoring the flow of traffic where we have data

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

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

- We assume $N_i \sim \mathrm{Poi}(\mu)$ 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 $\mathrm{Gam}(\mu|0,0) = 1/\mu$
- Note that we can solve this problem exactly—however, lets compare with MCMC

Traffic Rate

- Consider monitoring the flow of traffic where we have data

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

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

- We assume $N_i \sim \mathrm{Poi}(\mu)$ 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 $\mathrm{Gam}(\mu|0,0) = 1/\mu$
- Note that we can solve this problem exactly—however, lets compare with MCMC

Traffic Rate

- Consider monitoring the flow of traffic where we have data

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

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

- We assume $N_i \sim \mathrm{Poi}(\mu)$ 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 $\mathrm{Gam}(\mu|0,0) = 1/\mu$
- Note that we can solve this problem exactly—however, lets compare with MCMC

Traffic Rate

- Consider monitoring the flow of traffic where we have data

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

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

- We assume $N_i \sim \mathrm{Poi}(\mu)$ 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 $\mathrm{Gam}(\mu|0,0) = 1/\mu$
- Note that we can solve this problem exactly—however, lets compare with MCMC

Traffic Rate

- Consider monitoring the flow of traffic where we have data

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

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

- We assume $N_i \sim \mathrm{Poi}(\mu)$ 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 $\mathrm{Gam}(\mu|0, 0) = 1/\mu$
- 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 $p(\mu'|\mu) = \mathrm{Gam}(\mu'|\mu, \mu^2)$ which has $\mathbb{E}[\mu'] = \mu$ and variance 1

- We update with probability $\min(1, r)$ where

$$r = \frac{\mathrm{Gam}(\mu|\mu'^2, \mu')\frac{1}{\mu'}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu')}{\mathrm{Gam}(\mu'|\mu^2, \mu)\frac{1}{\mu}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu)}$$

$$= \frac{\mu\,\mathrm{Gam}(\mu|\mu'^2, \mu')}{\mu'\,\mathrm{Gam}(\mu'|\mu^2, \mu)}\mathrm{e}^{-n(\mu'-\mu)+\sum\limits_{i=1}^{n} N_i \log\left(\frac{\mu'}{\mu}\right)}$$

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 $p(\mu'|\mu) = \mathrm{Gam}(\mu'|\mu, \mu^2)$ which has $\mathbb{E}[\mu'] = \mu$ and variance 1

- We update with probability $\min(1, r)$ where

$$r = \frac{\mathrm{Gam}(\mu|\mu'^2, \mu')\frac{1}{\mu'}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu')}{\mathrm{Gam}(\mu'|\mu^2, \mu)\frac{1}{\mu}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu)}$$

$$= \frac{\mu\,\mathrm{Gam}(\mu|\mu'^2, \mu')}{\mu'\,\mathrm{Gam}(\mu'|\mu^2, \mu)}\mathrm{e}^{-n(\mu'-\mu)+\sum\limits_{i=1}^{n} N_i \log\left(\frac{\mu'}{\mu}\right)}$$

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 $p(\mu'|\mu) = \mathrm{Gam}(\mu'|\mu, \mu^2)$ which has $\mathbb{E}[\mu'] = \mu$ and variance 1

- We update with probability $\min(1, r)$ where

$$r = \frac{\mathrm{Gam}(\mu|\mu'^2, \mu')\frac{1}{\mu'}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu')}{\mathrm{Gam}(\mu'|\mu^2, \mu)\frac{1}{\mu}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu)}$$

$$= \frac{\mu\,\mathrm{Gam}(\mu|\mu'^2, \mu')}{\mu'\,\mathrm{Gam}(\mu'|\mu^2, \mu)}\mathrm{e}^{-n(\mu'-\mu)+\sum\limits_{i=1}^{n} N_i \log\left(\frac{\mu'}{\mu}\right)}$$

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 $p(\mu'|\mu) = \mathrm{Gam}(\mu'|\mu, \mu^2)$ which has $\mathbb{E}[\mu'] = \mu$ and variance 1

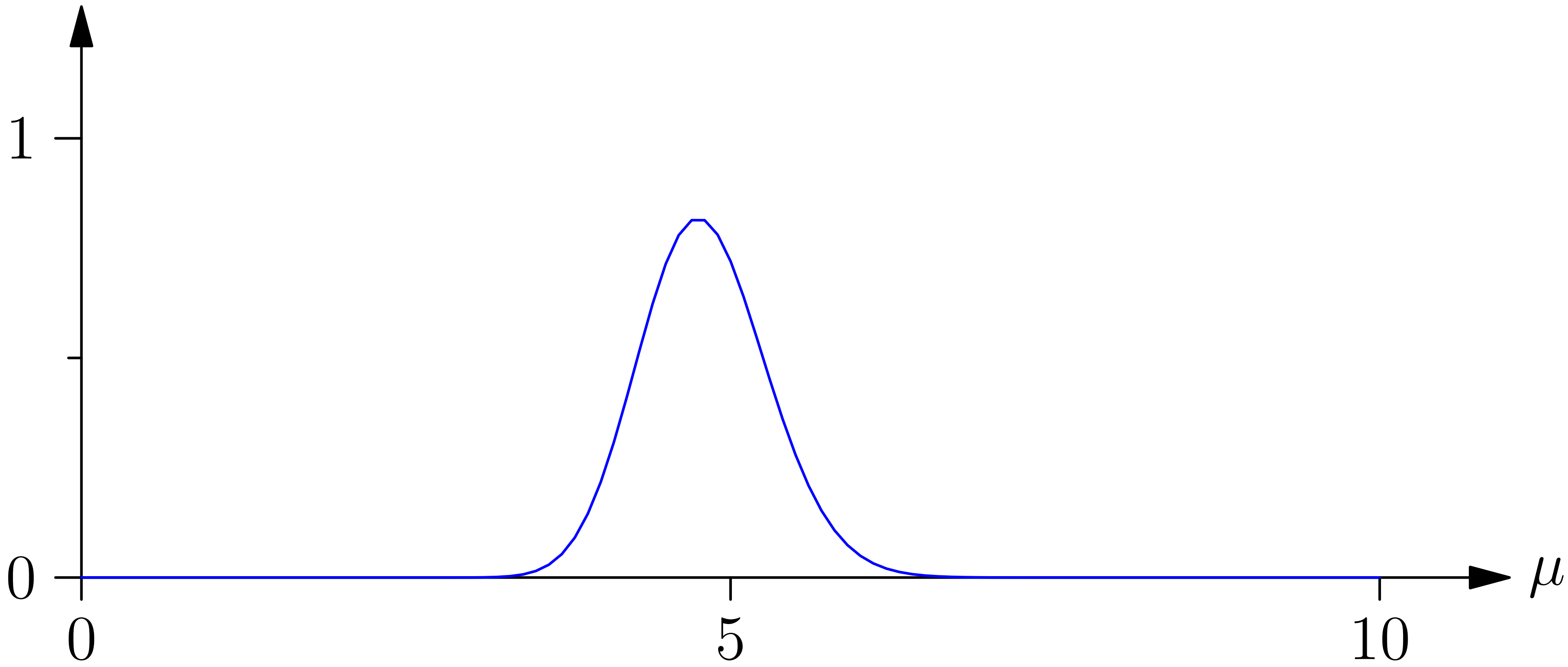
- We update with probability $\min(1, r)$ where

$$r = \frac{\mathrm{Gam}(\mu|\mu'^2, \mu')\frac{1}{\mu'}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu')}{\mathrm{Gam}(\mu'|\mu^2, \mu)\frac{1}{\mu}\prod_{i=1}^{n}\mathrm{Poi}(N_i|\mu)}$$

$$= \frac{\mu\,\mathrm{Gam}(\mu|\mu'^2, \mu')}{\mu'\,\mathrm{Gam}(\mu'|\mu^2, \mu)}\mathrm{e}^{-n(\mu'-\mu)+\sum_{i=1}^{n} N_i \log\left(\frac{\mu'}{\mu}\right)}$$

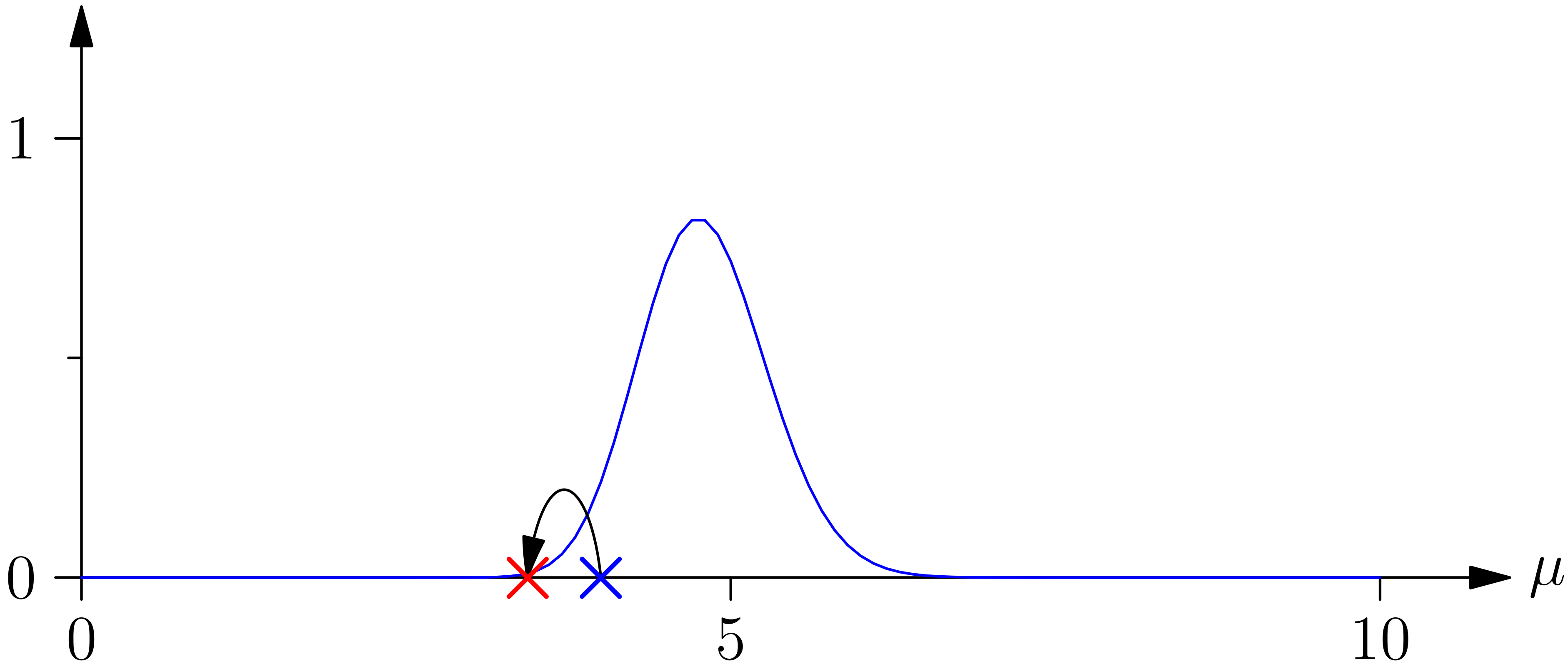
MCMC in Practice

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



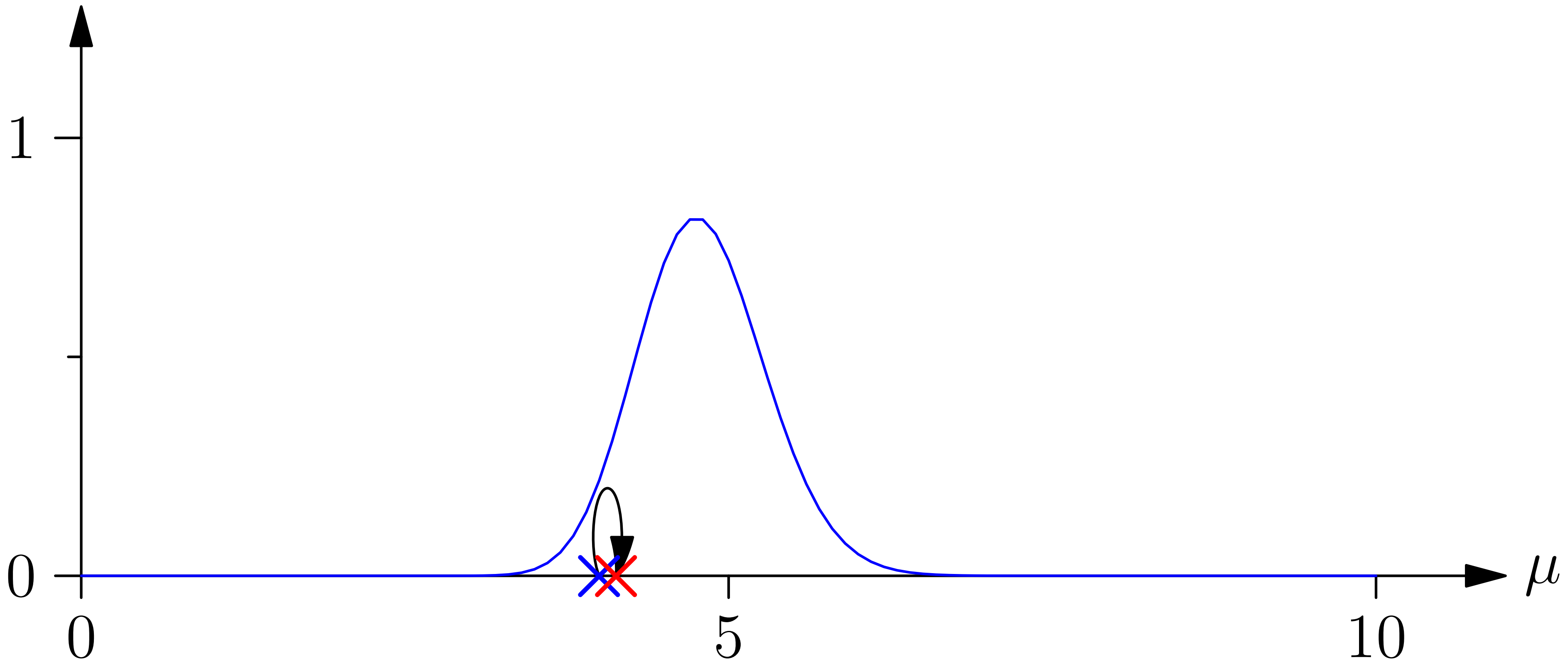
MCMC in Practice

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



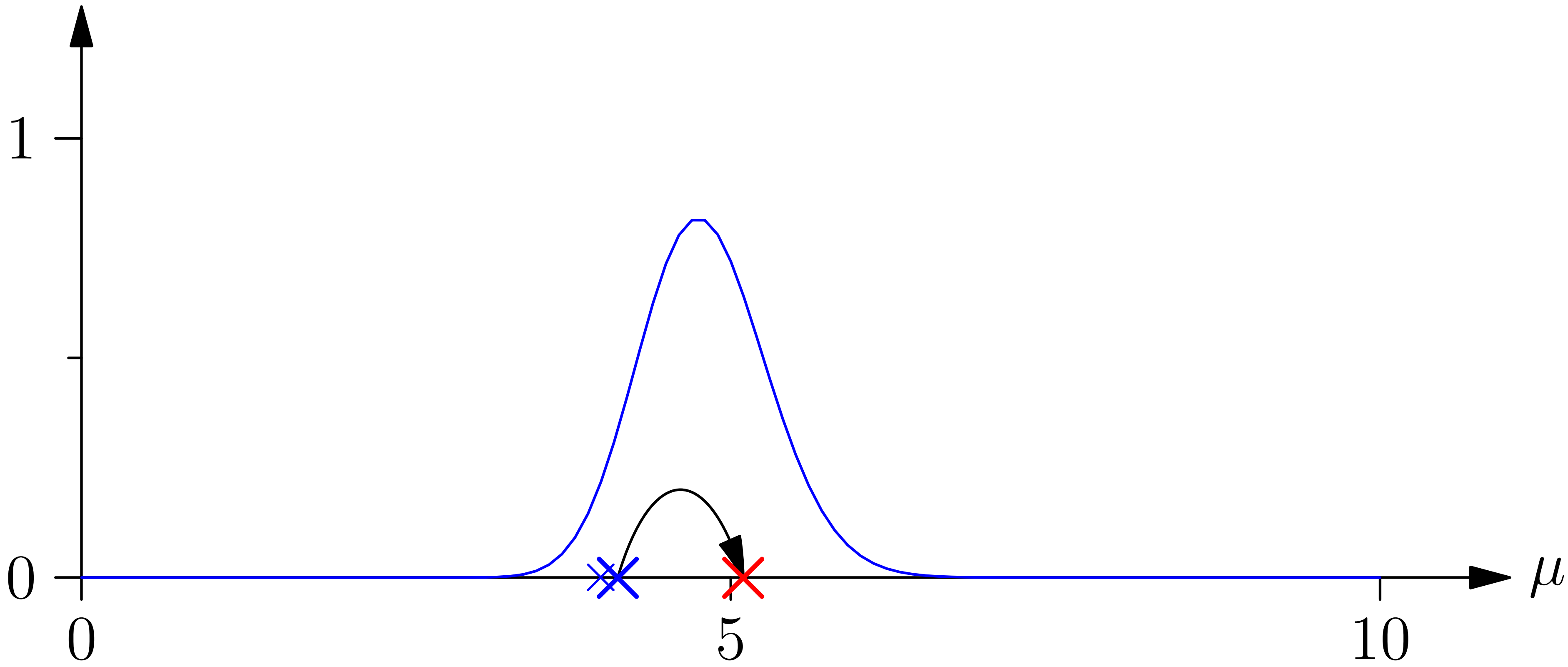
MCMC in Practice

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



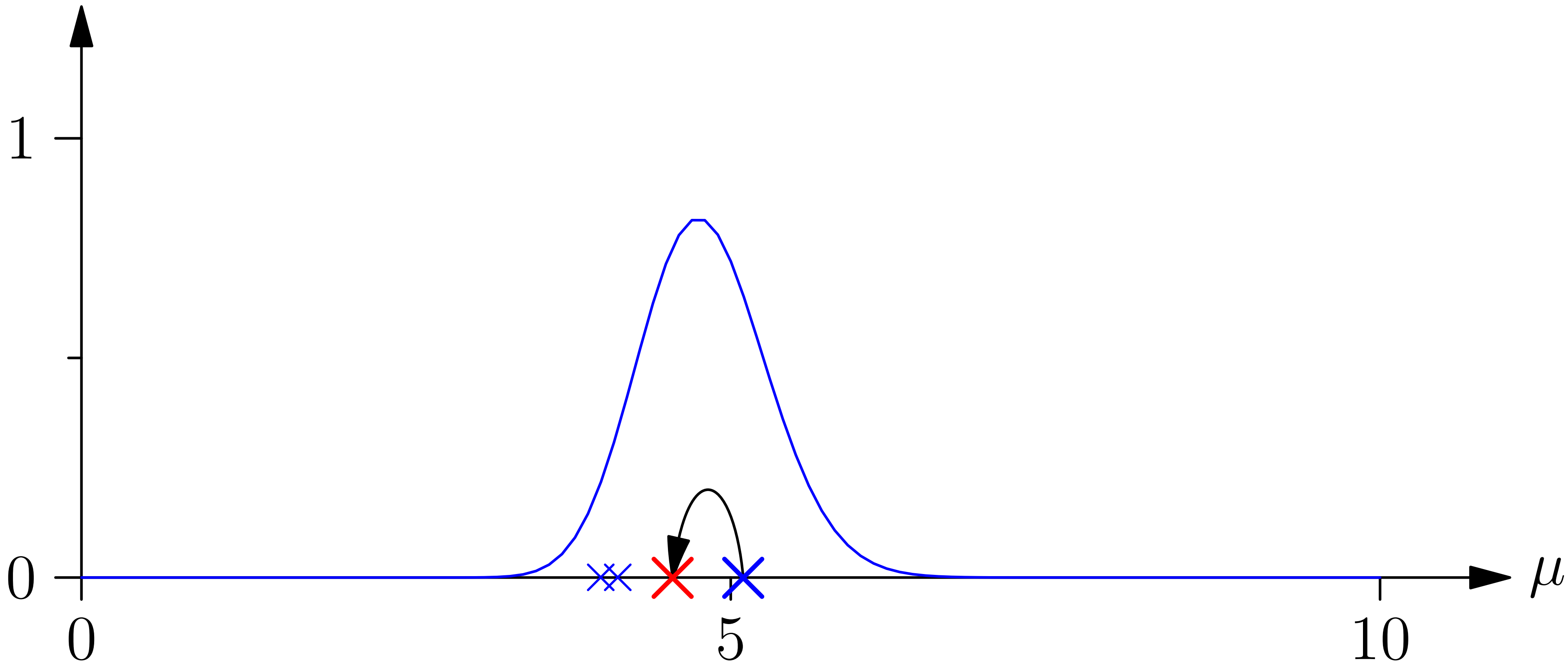
MCMC in Practice

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



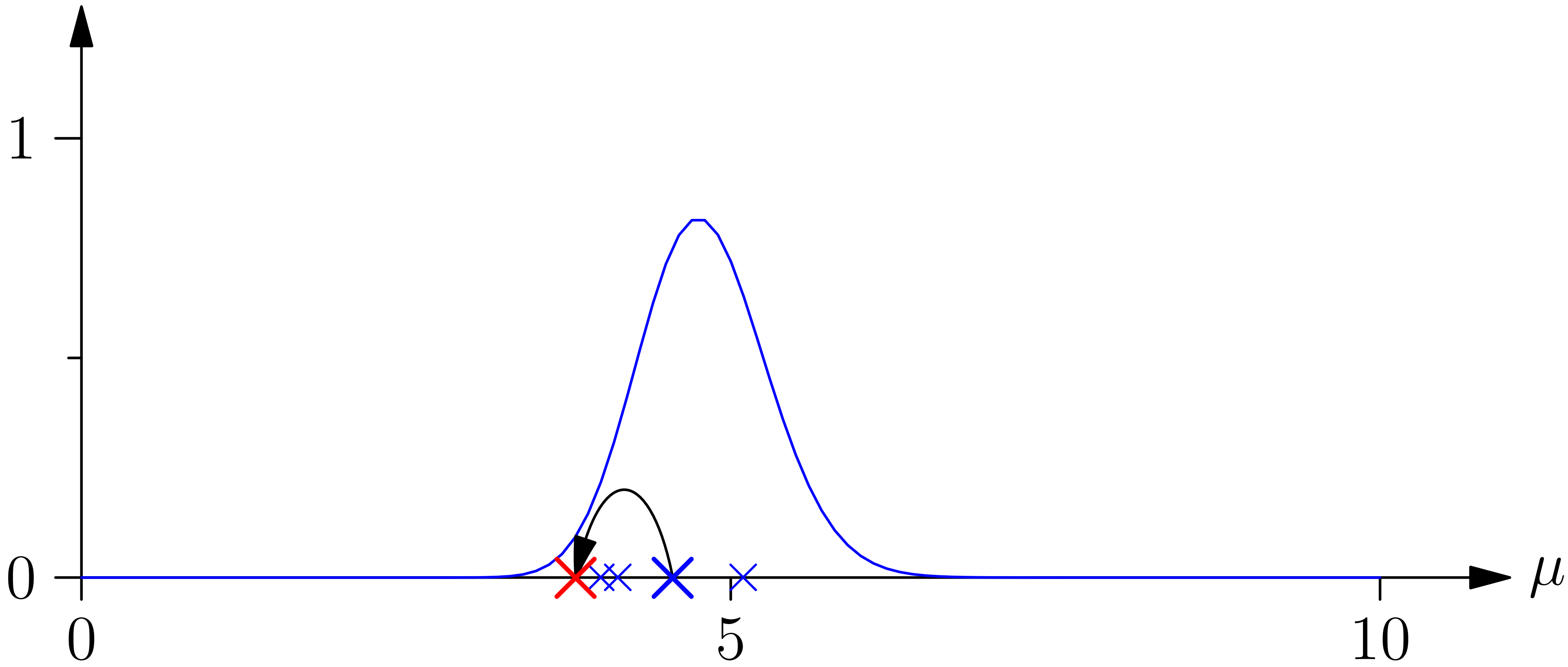
MCMC in Practice

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



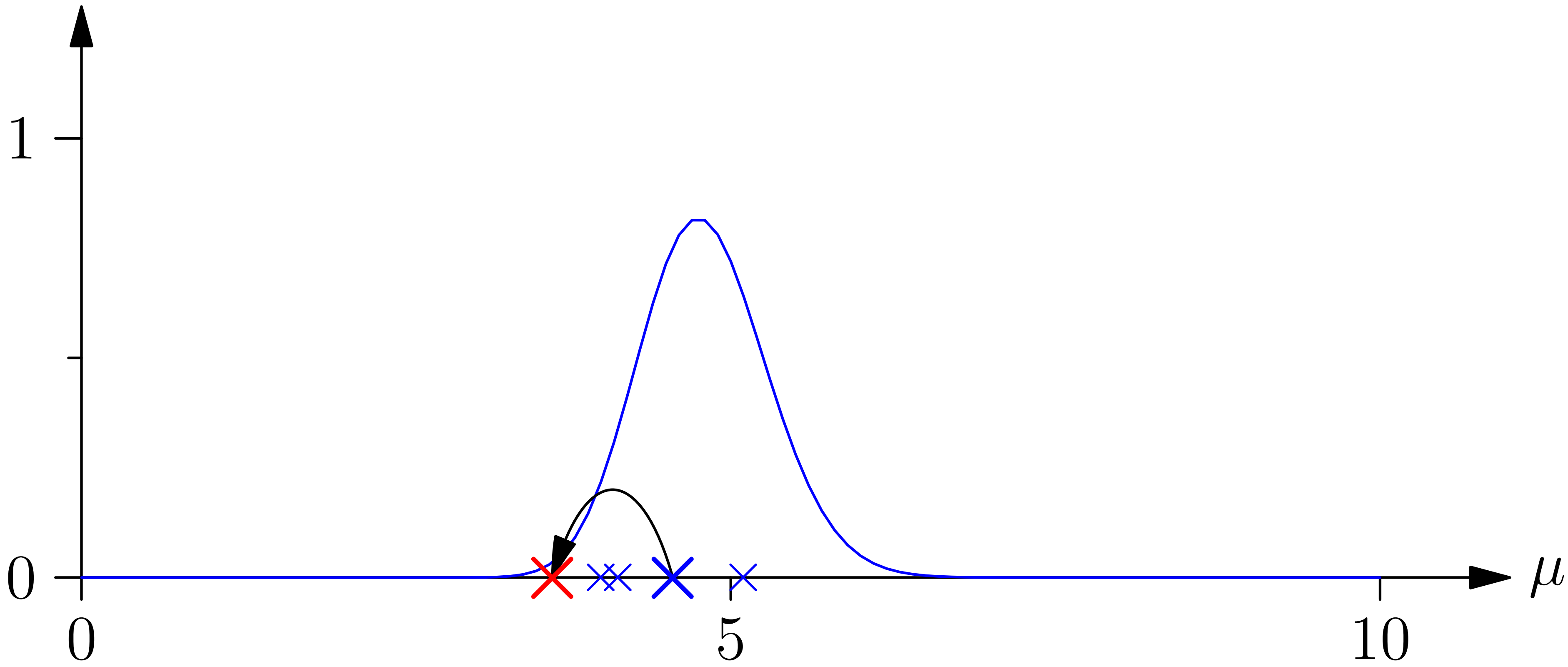
MCMC in Practice

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



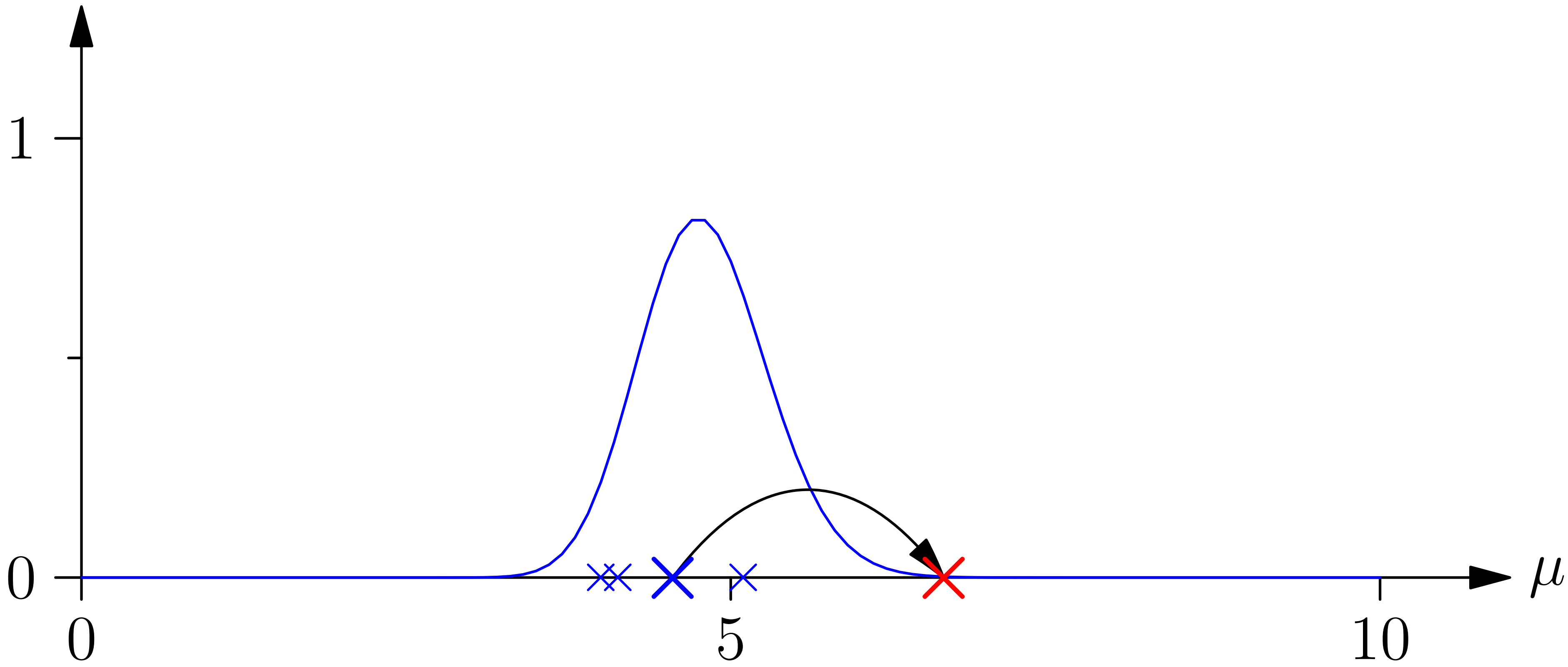
MCMC in Practice

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



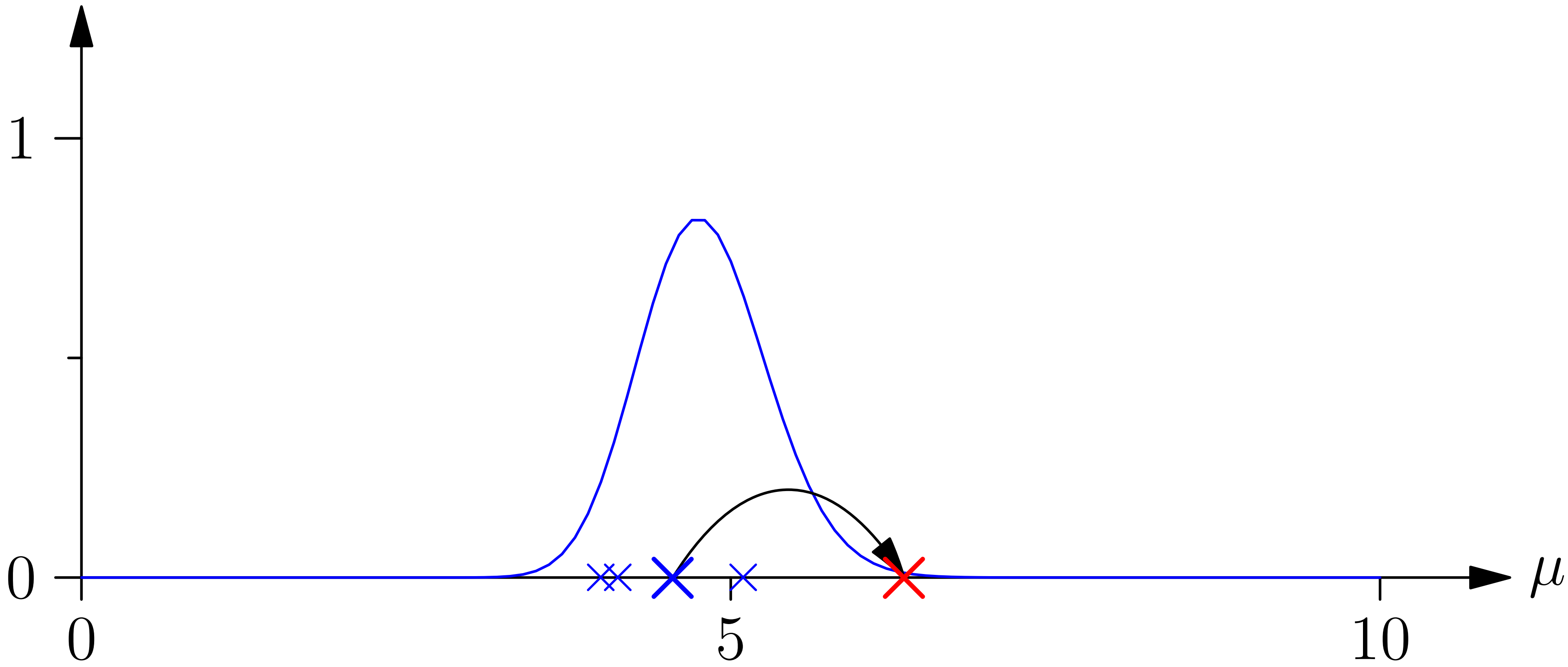
MCMC in Practice

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



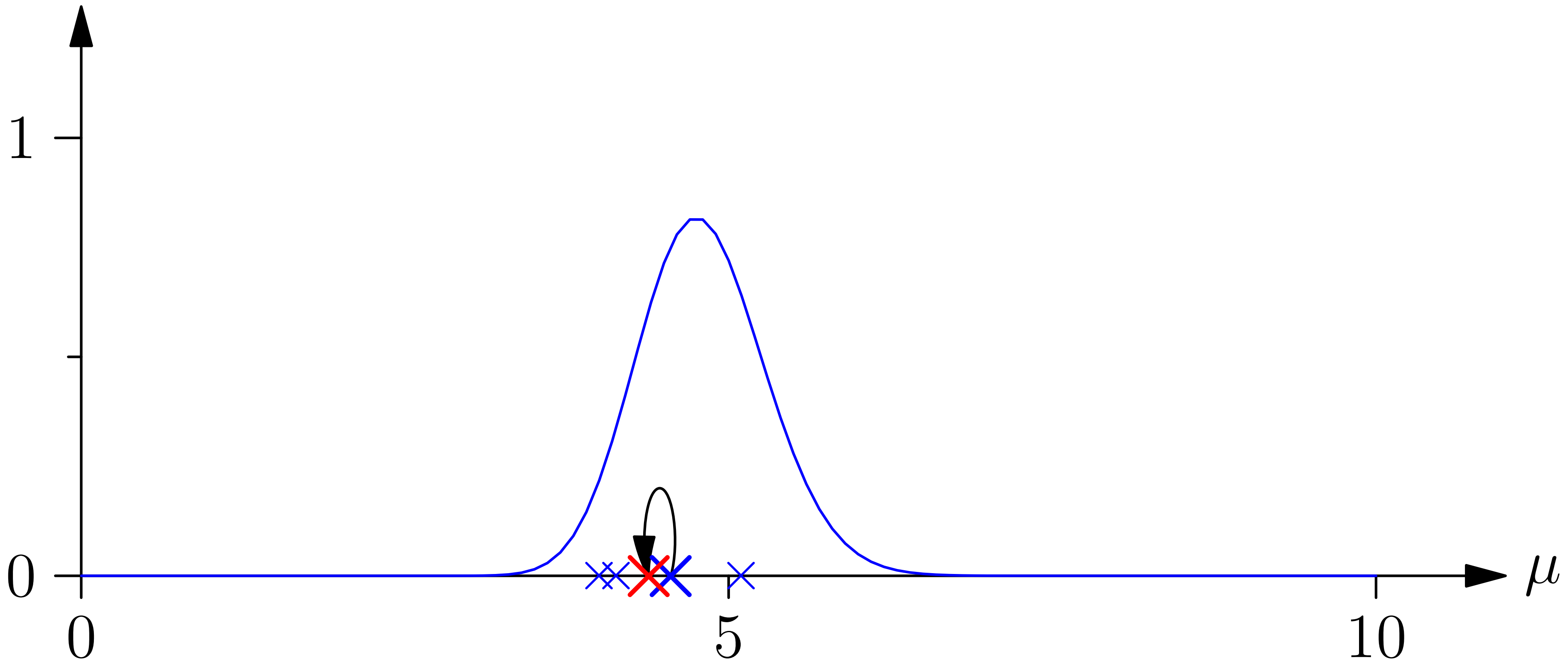
MCMC in Practice

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



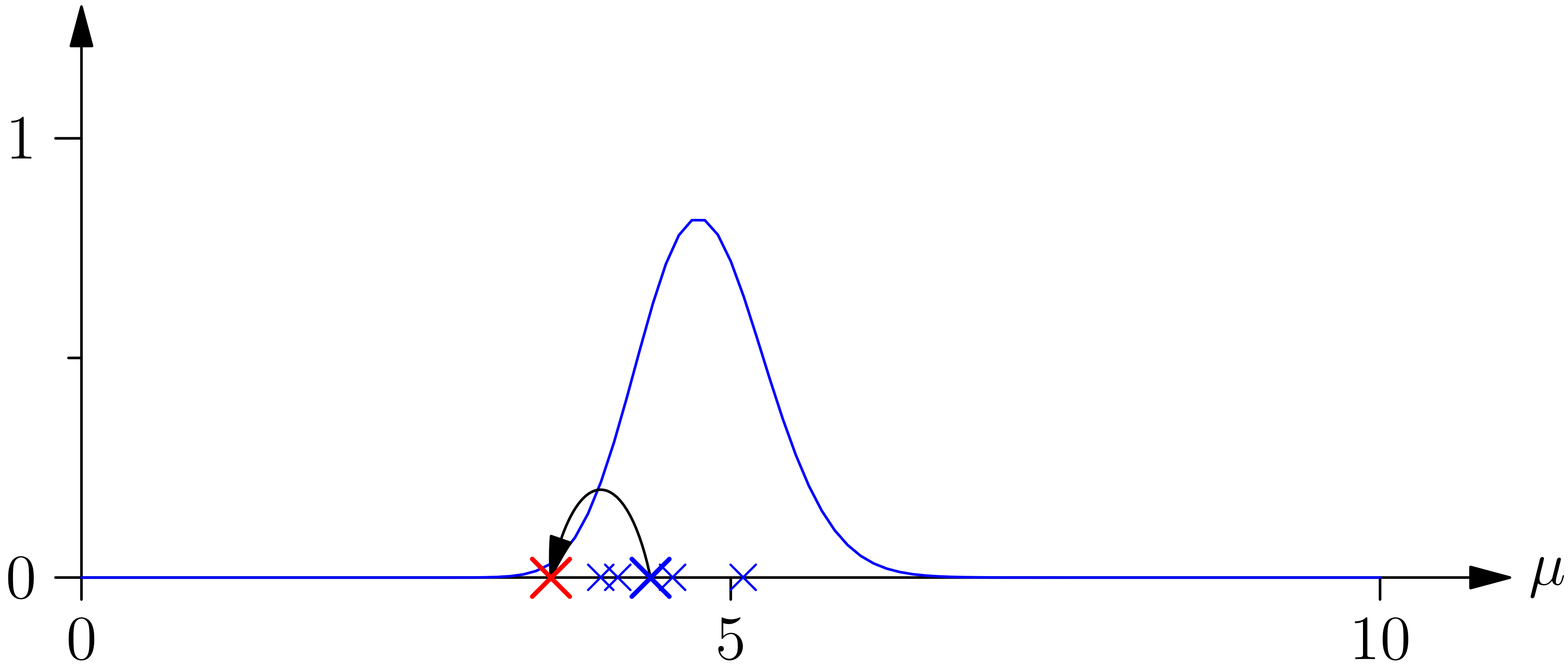
MCMC in Practice

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



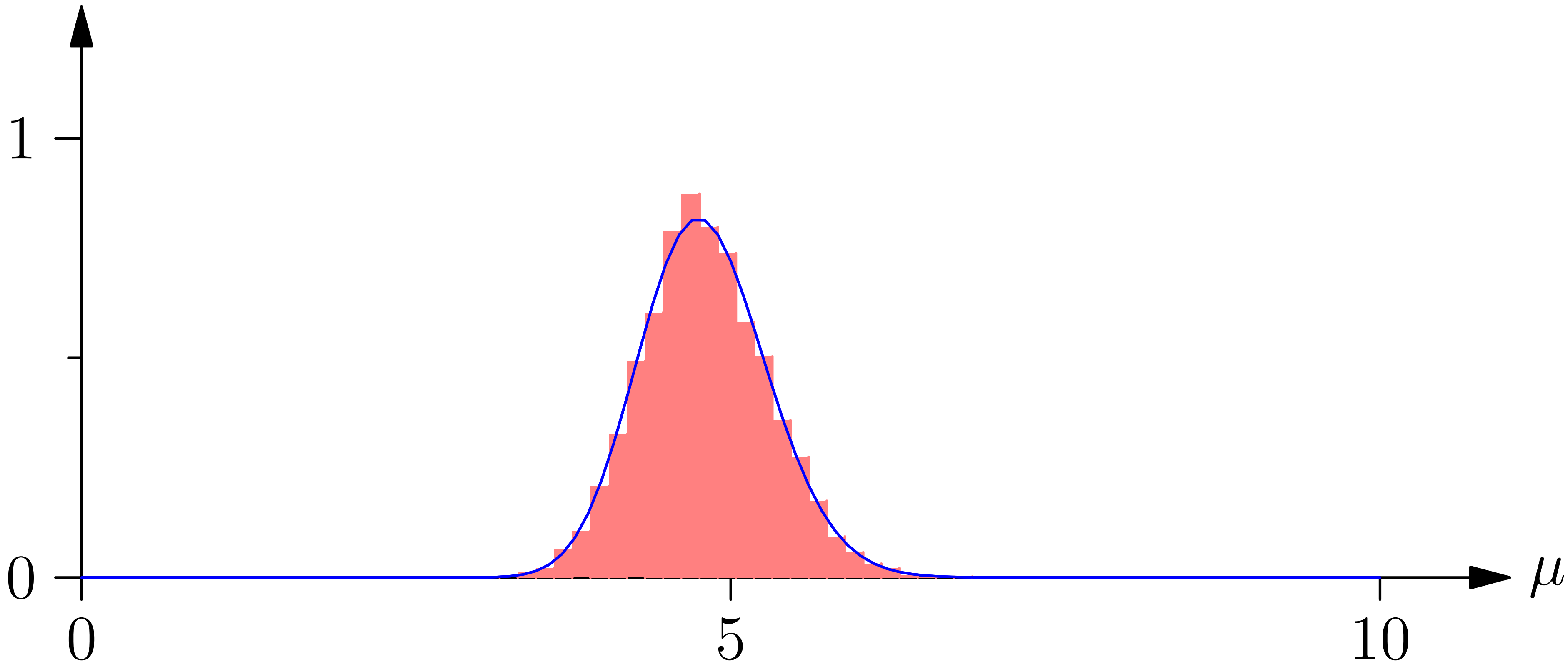
MCMC in Practice

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



MCMC 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

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

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

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

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

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

The MCMC Industry

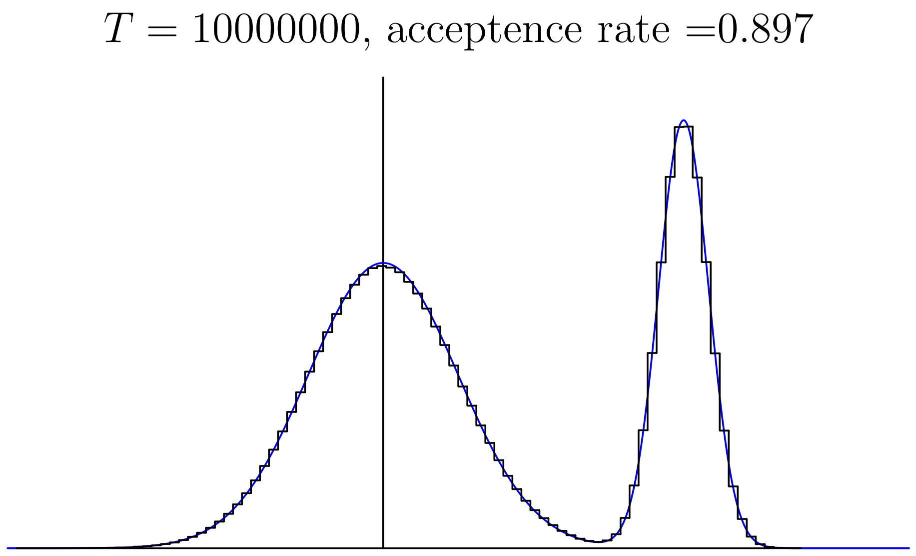
- MCMC provides a means to accurately sample from very complex models
- There have been many advanced techniques developed to improve MCMC performance
- E.g. hybrid MCMC simulates a dynamics to find good proposals with similar probability far from the starting point
- Often it seems that MCMC is complicated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

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

Outline

1. Sampling
2. MCMC
3. **Variational Methods**



MAP Solutions

- The simplest alternative to MCMC is to find a set of parameters $\boldsymbol{\theta}$ that maximise $f(\mathcal{D}|\boldsymbol{\theta})\, f(\boldsymbol{\theta})$
- This is the Maximum Aposteriori (MAP) approximation
- It can give good results if the posterior is unimodal and fairly well concentrated
- However, you are throwing away most of the probabilistic information
- In full Bayes (using MCMC) you can, for example, measure your uncertainty which MAP doesn't let you do

MAP Solutions

- The simplest alternative to MCMC is to find a set of parameters $\boldsymbol{\theta}$ that maximise $f(\mathcal{D}|\boldsymbol{\theta})\, f(\boldsymbol{\theta})$
- This is the Maximum Aposteriori (MAP) approximation
- It can give good results if the posterior is unimodal and fairly well concentrated
- However, you are throwing away most of the probabilistic information
- In full Bayes (using MCMC) you can, for example, measure your uncertainty which MAP doesn't let you do

MAP Solutions

- The simplest alternative to MCMC is to find a set of parameters $\boldsymbol{\theta}$ that maximise $f(\mathcal{D}|\boldsymbol{\theta})\, f(\boldsymbol{\theta})$
- This is the Maximum Aposteriori (MAP) approximation
- It can give good results if the posterior is unimodal and fairly well concentrated
- However, you are throwing away most of the probabilistic information
- In full Bayes (using MCMC) you can, for example, measure your uncertainty which MAP doesn't let you do

MAP Solutions

- The simplest alternative to MCMC is to find a set of parameters $\boldsymbol{\theta}$ that maximise $f(\mathcal{D}|\boldsymbol{\theta})\, f(\boldsymbol{\theta})$
- This is the Maximum Aposteriori (MAP) approximation
- It can give good results if the posterior is unimodal and fairly well concentrated
- However, you are throwing away most of the probabilistic information
- In full Bayes (using MCMC) you can, for example, measure your uncertainty which MAP doesn't let you do

MAP Solutions

- The simplest alternative to MCMC is to find a set of parameters $\boldsymbol{\theta}$ that maximise $f(\mathcal{D}|\boldsymbol{\theta})\, f(\boldsymbol{\theta})$
- This is the Maximum Aposteriori (MAP) approximation
- It can give good results if the posterior is unimodal and fairly well concentrated
- However, you are throwing away most of the probabilistic information
- In full Bayes (using MCMC) you can, for example, measure your uncertainty which MAP doesn't let you do

Variational Methods

- A second method is to approximate the posterior distribution by a simpler (typically factored distribution)

$$f(\boldsymbol{\theta}|\mathcal{D}) \approx g(\boldsymbol{\theta}|\boldsymbol{\phi}) = \prod_i g(\theta_i|\phi_i)$$

- We then choose $\boldsymbol{\phi}$ to minimise the “distance” between our distribution and the true posterior

$$\mathrm{KL}\big(g(\boldsymbol{\theta}|\boldsymbol{\phi})\big\|f(\boldsymbol{\theta}|\mathcal{D})\big)$$

- Where $\mathrm{KL}(g\|f)$ is the *Kullback-Leibler (KL) divergence*

$$\mathrm{KL}(g\|f) = \int g(\boldsymbol{\theta}|\boldsymbol{\phi}) \log\left(\frac{f(\boldsymbol{\theta}|\mathcal{D})}{g(\boldsymbol{\theta}|\boldsymbol{\phi})}\right) \mathrm{d}\boldsymbol{\theta}$$

Variational Methods

- A second method is to approximate the posterior distribution by a simpler (typically factored distribution)

$$f(\boldsymbol{\theta}|\mathcal{D}) \approx g(\boldsymbol{\theta}|\boldsymbol{\phi}) = \prod_i g(\theta_i|\phi_i)$$

- We then choose $\boldsymbol{\phi}$ to minimise the “distance” between our distribution and the true posterior

$$\mathrm{KL}\big(g(\boldsymbol{\theta}|\boldsymbol{\phi})\big\|f(\boldsymbol{\theta}|\mathcal{D})\big)$$

- Where $\mathrm{KL}(g\|f)$ is the *Kullback-Leibler (KL) divergence*

$$\mathrm{KL}(g\|f) = \int g(\boldsymbol{\theta}|\boldsymbol{\phi}) \log\left(\frac{f(\boldsymbol{\theta}|\mathcal{D})}{g(\boldsymbol{\theta}|\boldsymbol{\phi})}\right) \mathrm{d}\boldsymbol{\theta}$$

Variational Methods

- A second method is to approximate the posterior distribution by a simpler (typically factored distribution)

$$f(\boldsymbol{\theta}|\mathcal{D}) \approx g(\boldsymbol{\theta}|\boldsymbol{\phi}) = \prod_i g(\theta_i|\phi_i)$$

- We then choose $\boldsymbol{\phi}$ to minimise the "distance" between our distribution and the true posterior

$$\mathrm{KL}\big(g(\boldsymbol{\theta}|\boldsymbol{\phi})\big\|f(\boldsymbol{\theta}|\mathcal{D})\big)$$

- Where $\mathrm{KL}(g\|f)$ is the *Kullback-Leibler (KL) divergence*

$$\mathrm{KL}(g\|f) = \int g(\boldsymbol{\theta}|\boldsymbol{\phi}) \log\left(\frac{f(\boldsymbol{\theta}|\mathcal{D})}{g(\boldsymbol{\theta}|\boldsymbol{\phi})}\right) \mathrm{d}\boldsymbol{\theta}$$

KL-divergence

- The KL-divergence satisfies
  - $\mathrm{KL}(g\|f) \geq 0$
  - $\mathrm{KL}(f\|f) = 0$
- It acts as a measure of the difference between probability distributions although $\mathrm{KL}(g\|f) \neq \mathrm{KL}(f\|g)$
- We therefore choose $g(\boldsymbol{\theta}|\boldsymbol{\phi})$ to minimise $\mathrm{KL}(g\|f)$
- This is a bit technical (I skip all details), but often results in quite neat algorithms that are far faster than MCMC although approximate

KL-divergence

- The KL-divergence satisfies
  - $\star$ $\mathrm{KL}(g\|f) \geq 0$
  - $\star$ $\mathrm{KL}(f\|f) = 0$
- It acts as a measure of the difference between probability distributions although $\mathrm{KL}(g\|f) \neq \mathrm{KL}(f\|g)$
- We therefore choose $g(\boldsymbol{\theta}|\boldsymbol{\phi})$ to minimise $\mathrm{KL}(g\|f)$
- This is a bit technical (I skip all details), but often results in quite neat algorithms that are far faster than MCMC although approximate

KL-divergence

- The KL-divergence satisfies
  - $\mathrm{KL}(g\|f) \geq 0$
  - $\mathrm{KL}(f\|f) = 0$
- It acts as a measure of the difference between probability distributions although $\mathrm{KL}(g\|f) \neq \mathrm{KL}(f\|g)$
- We therefore choose $g(\boldsymbol{\theta}|\boldsymbol{\phi})$ to minimise $\mathrm{KL}(g\|f)$
- This is a bit technical (I skip all details), but often results in quite neat algorithms that are far faster than MCMC although approximate

KL-divergence

- The KL-divergence satisfies
  - $\mathrm{KL}(g\|f) \geq 0$
  - $\mathrm{KL}(f\|f) = 0$
- It acts as a measure of the difference between probability distributions although $\mathrm{KL}(g\|f) \neq \mathrm{KL}(f\|g)$
- We therefore choose $g(\boldsymbol{\theta}|\boldsymbol{\phi})$ to minimise $\mathrm{KL}(g\|f)$
- This is a bit technical (I skip all details), but often results in quite neat algorithms that are far faster than MCMC although approximate

Variational versus MCMC

- Variational approximations have an elegance
- They can produce good answers (although not always)
- They can be extended (e.g. by minimising $\mathrm{KL}(f\|g)$ rather than $\mathrm{KL}(g\|f)$—this is known as *belief propagation*)
- MCMC is less elegant, but is a controlled approximation (we get better results by increasing the number of iterations)
- MCMC is slower, but on modern computers this isn't usually a problem

Variational versus MCMC

- Variational approximations have an elegance
- They can produce good answers (although not always)
- They can be extended (e.g. by minimising $\mathrm{KL}(f\|g)$ rather than $\mathrm{KL}(g\|f)$—this is known as *belief propagation*)
- MCMC is less elegant, but is a controlled approximation (we get better results by increasing the number of iterations)
- MCMC is slower, but on modern computers this isn't usually a problem

Variational versus MCMC

- Variational approximations have an elegance
- They can produce good answers (although not always)
- They can be extended (e.g. by minimising $\mathrm{KL}(f\|g)$ rather than $\mathrm{KL}(g\|f)$—this is known as *belief propagation*)
- MCMC is less elegant, but is a controlled approximation (we get better results by increasing the number of iterations)
- MCMC is slower, but on modern computers this isn't usually a problem

Variational versus MCMC

- Variational approximations have an elegance
- They can produce good answers (although not always)
- They can be extended (e.g. by minimising $\mathrm{KL}(f\|g)$ rather than $\mathrm{KL}(g\|f)$—this is known as *belief propagation*)
- MCMC is less elegant, but is a controlled approximation (we get better results by increasing the number of iterations)
- MCMC is slower, but on modern computers this isn't usually a problem

Variational versus MCMC

- Variational approximations have an elegance
- They can produce good answers (although not always)
- They can be extended (e.g. by minimising $\mathrm{KL}(f\|g)$ rather than $\mathrm{KL}(g\|f)$—this is known as *belief propagation*)
- MCMC is less elegant, but is a controlled approximation (we get better results by increasing the number of iterations)
- MCMC is slower, but on modern computers this isn't usually a problem

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

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

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

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

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