Asymptotically Independent Markov Sampling: a new MCMC scheme for Bayesian Inference

Konstantin Zuev

Department of Computing & Mathematical Sciences
California Institute of Technology

http://www.its.caltech.edu/∼zuev

February 18, 2011
Probability and Statistics Seminar, USC

Joint work with Professor J.L. Beck

Outline

Outline

Main Problem of Computational Bayesian Inference

Problem: To estimate

$$\mathbb{E}_\pi[h] = \int_{\mathcal{X}} h(x)\pi(x)dx$$

- $\mathcal{X} \subseteq \mathbb{R}^d$ parameter space
- $h : \mathcal{X} \to \mathbb{R}$ function of interest
- $\pi(x)$ posterior PDF on $\mathcal{X}$

Main Problem of Computational Bayesian Inference

Problem: To estimate

$$\mathbb{E}_\pi[h] = \int_{\mathcal{X}} h(x)\pi(x)dx$$

- $\mathcal{X} \subseteq \mathbb{R}^d$ parameter space
- $h : \mathcal{X} \to \mathbb{R}$ function of interest
- $\pi(x)$ posterior PDF on $\mathcal{X}$

"Easy" Cases:

- d is small $(d = 1, 2, 3) \Rightarrow$ numerical integration
- $\pi(x)$ is easy to sample from $\Rightarrow$ Monte Carlo method

Main Problem of Computational Bayesian Inference

Problem: To estimate

$$\mathbb{E}_\pi[h] = \int_{\mathcal{X}} h(x)\pi(x)dx$$

- $\mathcal{X} \subseteq \mathbb{R}^d$ parameter space
- $h : \mathcal{X} \to \mathbb{R}$ function of interest
- $\pi(x)$ posterior PDF on $\mathcal{X}$

"Easy" Cases:

- d is small $(d = 1, 2, 3) \Rightarrow$ numerical integration
- $\pi(x)$ is easy to sample from $\Rightarrow$ Monte Carlo method

Typical Case in Applications:

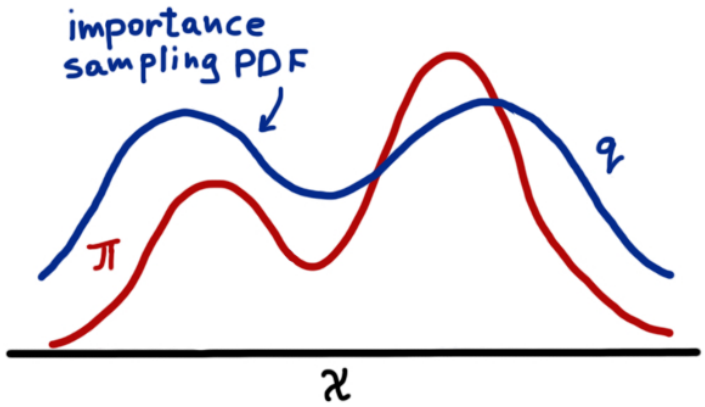
- d is large $(d \sim 10^2 - 10^3)$
- $\pi(x)$ is known only up to a normalizing constant, $\pi(x) = f(x)/Z$

Standard Methods

- Importance Sampling

$$x^{(i)} \sim q(x), \quad w^{(i)} = \frac{\pi(x^{(i)})}{q(x^{(i)})}$$

$$\hat{h}_N = \frac{\sum_{i=1}^{N} w^{(i)} h(x^{(i)})}{\sum_{i=1}^{N} w^{(i)}}$$

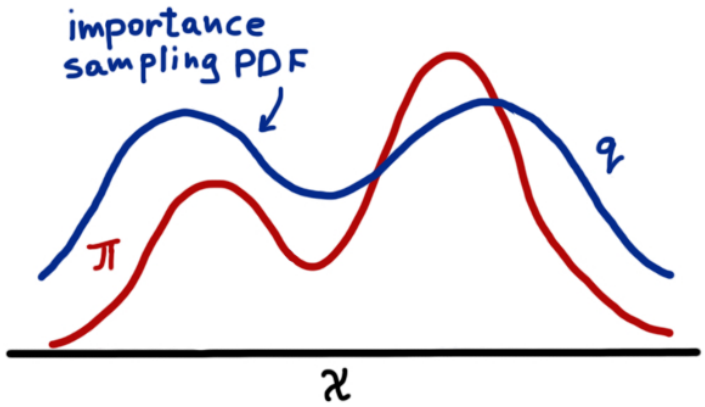


Standard Methods

- Importance Sampling

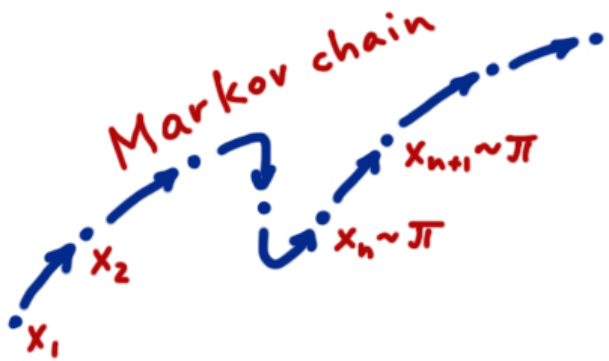
$$x^{(i)} \sim q(x), \quad w^{(i)} = \frac{\pi(x^{(i)})}{q(x^{(i)})}$$

$$\hat{h}_N = \frac{\sum_{i=1}^{N} w^{(i)} h(x^{(i)})}{\sum_{i=1}^{N} w^{(i)}}$$



- Markov chain Monte Carlo

$$\hat{h}_N = \frac{1}{N - N_0} \sum_{i=N_0+1}^{N} h(x^{(i)})$$

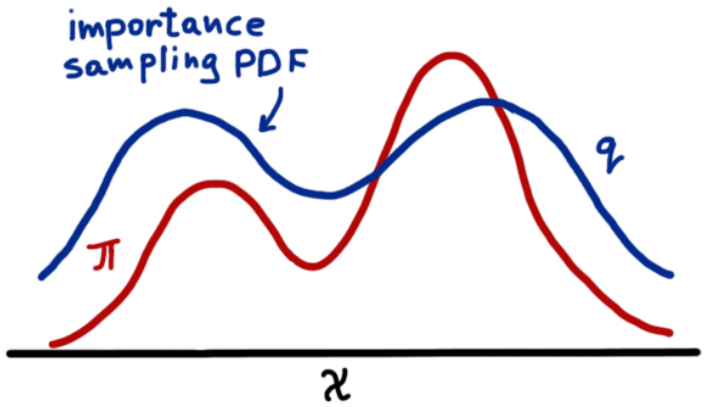


Standard Methods

- Importance Sampling

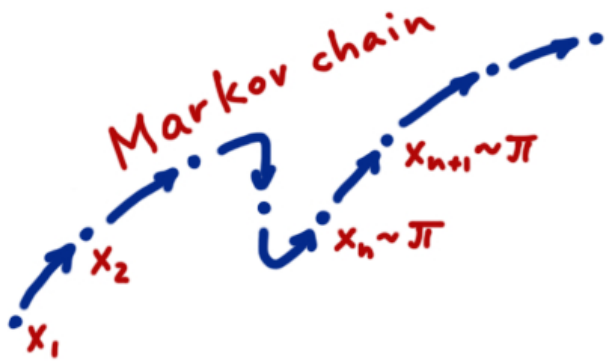
$$x^{(i)} \sim q(x), \quad w^{(i)} = \frac{\pi(x^{(i)})}{q(x^{(i)})}$$

$$\hat{h}_N = \frac{\sum_{i=1}^{N} w^{(i)} h(x^{(i)})}{\sum_{i=1}^{N} w^{(i)}}$$



- Markov chain Monte Carlo

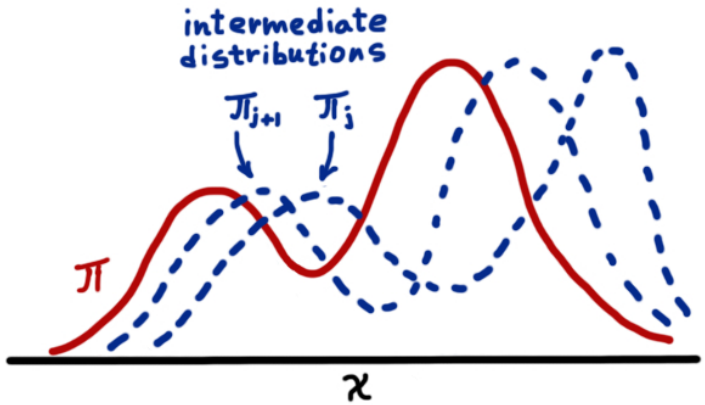
$$\hat{h}_N = \frac{1}{N - N_0} \sum_{i=N_0+1}^{N} h(x^{(i)})$$



- Annealing

$$\pi_0(x), \ldots, \pi_m(x) = \pi(x)$$

$$x_j^{(1)}, \ldots, x_j^{(N)} \sim \pi_j(x)$$



Asymptotically Independent Markov Sampling: Basic Idea

AIMS Scheme

Asymptotically Independent Markov Sampling: Basic Idea

AIMS Scheme

1. Annealing Step: $\pi_0(x), \ldots, \pi_m(x) = \pi(x)$
   - $\pi_0(x)$ is a prior PDF, which is easy to sample from
   - $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$, where $0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$

Asymptotically Independent Markov Sampling: Basic Idea

AIMS Scheme

1. Annealing Step: $\pi_0(x), \ldots, \pi_m(x) = \pi(x)$
   - $\pi_0(x)$ is a prior PDF, which is easy to sample from
   - $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$, where $0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$
2. Importance Sampling Approximation:
   - Based on $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$, construct an approximation $\hat{\pi}_1(x) \approx \pi_1(x)$

Asymptotically Independent Markov Sampling: Basic Idea

AIMS Scheme

1. Annealing Step: $\pi_0(x), \ldots, \pi_m(x) = \pi(x)$
   - $\pi_0(x)$ is a prior PDF, which is easy to sample from
   - $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$, where $0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$
2. Importance Sampling Approximation:
   - Based on $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$, construct an approximation $\hat{\pi}_1(x) \approx \pi_1(x)$
3. MCMC sampling:
   - Sample $x_1^{(1)}, \ldots, x_1^{(N)} \sim \pi_1(x)$ using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_1(x)$

Asymptotically Independent Markov Sampling: Basic Idea

AIMS Scheme

1. Annealing Step: $\pi_0(x), \ldots, \pi_m(x) = \pi(x)$
   - $\pi_0(x)$ is a prior PDF, which is easy to sample from
   - $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$, where $0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$
2. Importance Sampling Approximation:
   - Based on $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$, construct an approximation $\hat{\pi}_1(x) \approx \pi_1(x)$
3. MCMC sampling:
   - Sample $x_1^{(1)}, \ldots, x_1^{(N)} \sim \pi_1(x)$ using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_1(x)$

Iterate Steps 2 and 3 for $j = 1, \ldots, m$:

$$x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \quad \overset{\text{IMH}}{\rightsquigarrow} \quad x_j^{(1)}, \ldots, x_j^{(N)} \sim \pi_j(x)$$

Asymptotically Independent Markov Sampling: Basic Idea

AIMS Scheme

1. Annealing Step: $\pi_0(x), \ldots, \pi_m(x) = \pi(x)$
   - $\pi_0(x)$ is a prior PDF, which is easy to sample from
   - $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$, where $0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$
2. Importance Sampling Approximation:
   - Based on $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$, construct an approximation $\hat{\pi}_1(x) \approx \pi_1(x)$
3. MCMC sampling:
   - Sample $x_1^{(1)}, \ldots, x_1^{(N)} \sim \pi_1(x)$ using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_1(x)$

Iterate Steps 2 and 3 for $j = 1, \ldots, m$:

$$x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \quad \overset{\text{IMH}}{\rightsquigarrow} \quad x_j^{(1)}, \ldots, x_j^{(N)} \sim \pi_j(x)$$

AIMS estimate: $\hat{h}_N = \frac{1}{N} \sum_{i=1}^{N} h(x_m^{(i)}), \qquad x_m^{(1)}, \ldots, x_m^{(N)} \sim \pi_m(x)$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

$$\pi_j(x)dx = \int_{\mathcal{X}} \pi_j(y) T_j(y, dx) dy$$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

$$\begin{aligned}\pi_j(x)dx &= \int_{\mathcal{X}} \pi_j(y) T_j(y, dx) dy \\ &= \int_{\mathcal{X}} \pi_{j-1}(y) \frac{\pi_j(y)}{\pi_{j-1}(y)} T_j(y, dx) dy\end{aligned}$$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

$$\begin{aligned}
\pi_j(x)dx &= \int_{\mathcal{X}} \pi_j(y) T_j(y, dx) dy \\
&= \int_{\mathcal{X}} \pi_{j-1}(y) \frac{\pi_j(y)}{\pi_{j-1}(y)} T_j(y, dx) dy \\
&\approx \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)
\end{aligned}$$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

$$\begin{aligned}\pi_j(x)dx &= \int_{\mathcal{X}} \pi_j(y) T_j(y, dx) dy \\ &= \int_{\mathcal{X}} \pi_{j-1}(y) \frac{\pi_j(y)}{\pi_{j-1}(y)} T_j(y, dx) dy \\ &\approx \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)\end{aligned}$$

$$w_{j-1}^{(i)} = \frac{\pi_j(x_{j-1}^{(i)})}{\pi_{j-1}(x_{j-1}^{(i)})}$$

$$\bar{w}_{j-1}^{(i)} = \frac{w_{j-1}^{(i)}}{\sum_{k=1}^{N} w_{j-1}^{(k)}}$$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

$$\begin{aligned}
\pi_j(x)dx &= \int_{\mathcal{X}} \pi_j(y) T_j(y, dx) dy \\
&= \int_{\mathcal{X}} \pi_{j-1}(y) \frac{\pi_j(y)}{\pi_{j-1}(y)} T_j(y, dx) dy \\
&\approx \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx) \overset{\text{def}}{=} \hat{\pi}_j(dx)
\end{aligned}$$

$$w_{j-1}^{(i)} = \frac{\pi_j(x_{j-1}^{(i)})}{\pi_{j-1}(x_{j-1}^{(i)})}$$

$$\bar{w}_{j-1}^{(i)} = \frac{w_{j-1}^{(i)}}{\sum_{k=1}^{N} w_{j-1}^{(k)}}$$

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

$$\begin{aligned}
\pi_j(x)dx &= \int_{\mathcal{X}} \pi_j(y) T_j(y, dx) dy \\
&= \int_{\mathcal{X}} \pi_{j-1}(y) \frac{\pi_j(y)}{\pi_{j-1}(y)} T_j(y, dx) dy \\
&\approx \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx) \overset{\text{def}}{=} \hat{\pi}_j(dx)
\end{aligned}$$

$$w_{j-1}^{(i)} = \frac{\pi_j(x_{j-1}^{(i)})}{\pi_{j-1}(x_{j-1}^{(i)})}$$

$$\bar{w}_{j-1}^{(i)} = \frac{w_{j-1}^{(i)}}{\sum_{k=1}^{N} w_{j-1}^{(k)}}$$

- $\hat{\pi}_j(dx)$ will usually have both continuous and discrete parts

Importance Sampling Approximation

At annealing level j: $\quad x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)} \sim \pi_{j-1}(x) \quad \overset{\text{IS}}{\rightsquigarrow} \quad \hat{\pi}_j(x) \approx \pi_j(x)$

- $T_j(x, dy)$ is a transition kernel
- $\pi_j(x)$ is stationary w.r.t. $T_j(x, dy)$

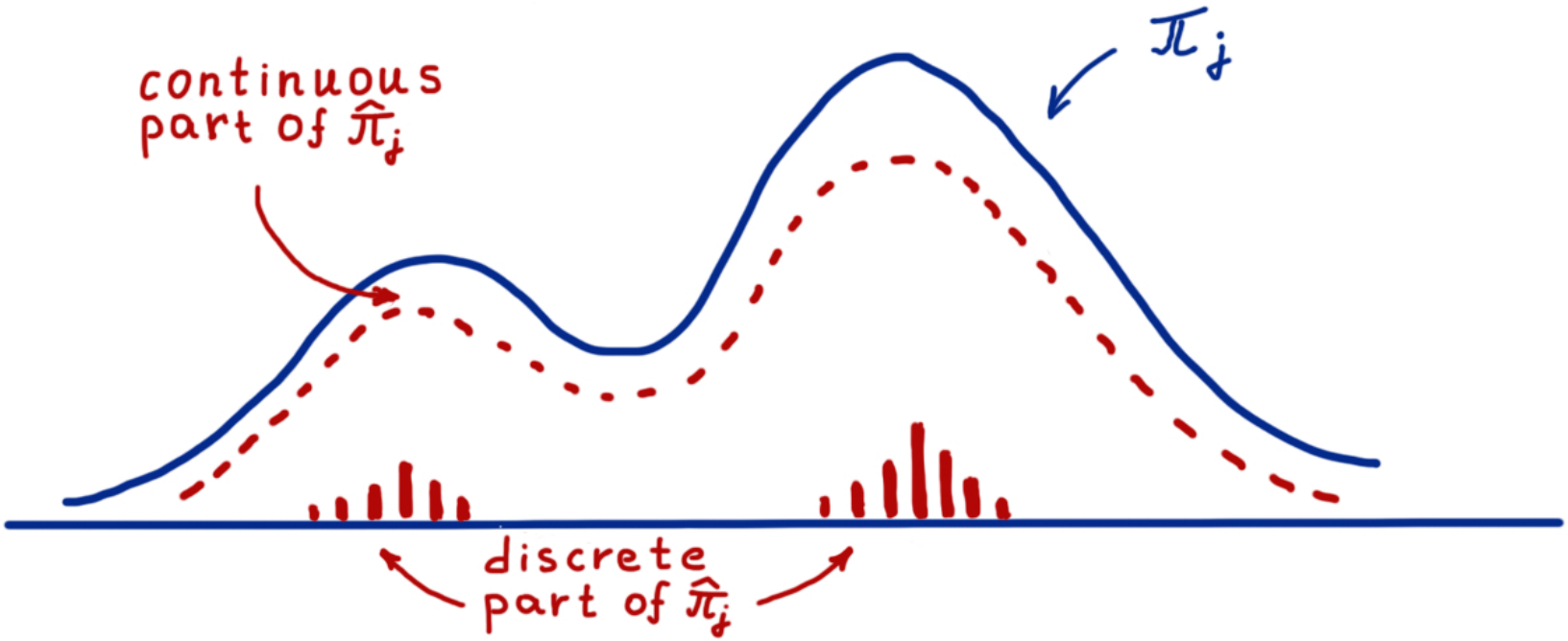
$$\begin{aligned}
\pi_j(x)dx &= \int_{\mathcal{X}} \pi_j(y) T_j(y, dx) dy \\
&= \int_{\mathcal{X}} \pi_{j-1}(y) \frac{\pi_j(y)}{\pi_{j-1}(y)} T_j(y, dx) dy \\
&\approx \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx) \overset{\text{def}}{=} \hat{\pi}_j(dx)
\end{aligned}$$

$$w_{j-1}^{(i)} = \frac{\pi_j(x_{j-1}^{(i)})}{\pi_{j-1}(x_{j-1}^{(i)})}$$

$$\bar{w}_{j-1}^{(i)} = \frac{w_{j-1}^{(i)}}{\sum_{k=1}^{N} w_{j-1}^{(k)}}$$

- $\hat{\pi}_j(dx)$ will usually have both continuous and discrete parts
- $T(x, dy) = k(x, y)dy + r(x)\delta_x(dy)$

Schematic Illustration



$$\pi_j(x)dx \approx \hat{\pi}_j(dx) = \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$$

The Random Walk Metropolis-Hastings transition kernel

$$T_j(x, dy) = q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy + r_j(x)\delta_x(dy)$$

The Random Walk Metropolis-Hastings transition kernel

$$T_j(x, dy) = q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy + r_j(x)\delta_x(dy)$$

- $q_j(x, y)$ is a (symmetric) local proposal PDF

The Random Walk Metropolis-Hastings transition kernel

$$T_j(x, dy) = q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy + r_j(x)\delta_x(dy)$$

- $q_j(x, y)$ is a (symmetric) local proposal PDF
- $r_j(x) = 1 - \int_{\mathcal{X}} q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy$

The Random Walk Metropolis-Hastings transition kernel

$$T_j(x, dy) = q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy + r_j(x)\delta_x(dy)$$

- $q_j(x, y)$ is a (symmetric) local proposal PDF
- $r_j(x) = 1 - \int_{\mathcal{X}} q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy$

Example: $\pi_j(x) = \mathcal{N}_{0,1}(x)$ $\pi_{j-1}(x) = \mathcal{N}_{0,2}(x)$ $q_j(x, y) = \mathcal{N}_{x,1/2}(y)$

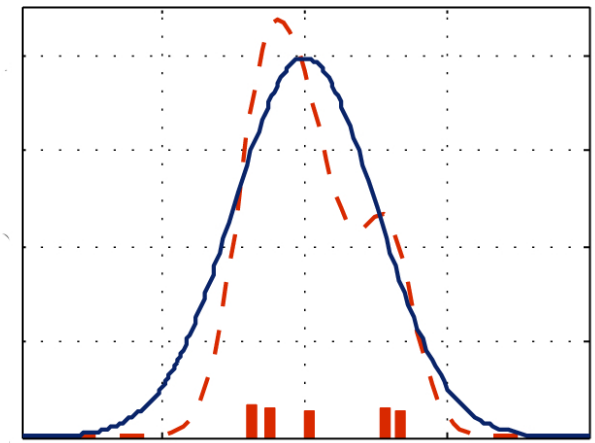
The Random Walk Metropolis-Hastings transition kernel

$$T_j(x, dy) = q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy + r_j(x)\delta_x(dy)$$

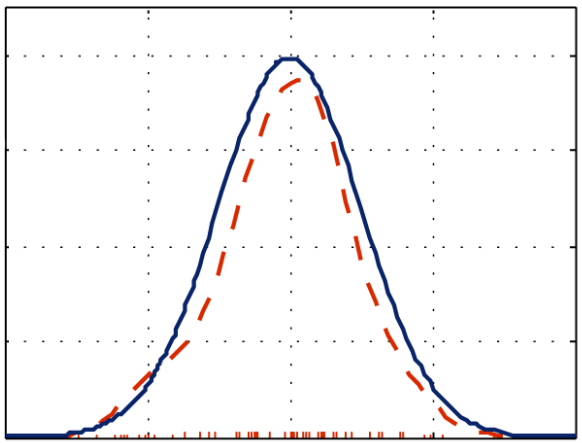
- $q_j(x, y)$ is a (symmetric) local proposal PDF
- $r_j(x) = 1 - \int_{\mathcal{X}} q_j(x, y) \min\left\{1, \frac{\pi_j(y)}{\pi_j(x)}\right\} dy$

Example: $\pi_j(x) = \mathcal{N}_{0,1}(x)$ $\pi_{j-1}(x) = \mathcal{N}_{0,2}(x)$ $q_j(x, y) = \mathcal{N}_{x,1/2}(y)$

$$\hat{\pi}_j(dx) = \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$$



N=5



N=50

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$

2. $$x_j^{(i+1)} = \begin{cases} \xi, & \text{w.p. } \alpha(x_j^{(i)}, \xi) \\ x_j^{(i)}, & \text{w.p. } 1 - \alpha(x_j^{(i)}, \xi) \end{cases} \qquad \alpha(x, \xi) = \min\left\{1, \frac{\pi_j(\xi)\hat{\pi}_j(x)}{\pi_j(x)\hat{\pi}_j(\xi)}\right\}$$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$
   - It is easy to sample from $\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$

2. $$x_j^{(i+1)} = \begin{cases} \xi, & \text{w.p. } \alpha(x_j^{(i)}, \xi) \\ x_j^{(i)}, & \text{w.p. } 1 - \alpha(x_j^{(i)}, \xi) \end{cases} \qquad \alpha(x, \xi) = \min\left\{1, \frac{\pi_j(\xi)\hat{\pi}_j(x)}{\pi_j(x)\hat{\pi}_j(\xi)}\right\}$$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$
   - It is easy to sample from $\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$
     a) Select k from $\{1, \dots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \dots, \bar{w}_{j-1}^{(N)}\}$

2. $$x_j^{(i+1)} = \begin{cases} \xi, & \text{w.p. } \alpha(x_j^{(i)}, \xi) \\ x_j^{(i)}, & \text{w.p. } 1 - \alpha(x_j^{(i)}, \xi) \end{cases} \qquad \alpha(x, \xi) = \min\left\{1, \frac{\pi_j(\xi)\hat{\pi}_j(x)}{\pi_j(x)\hat{\pi}_j(\xi)}\right\}$$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$
   - It is easy to sample from $\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$
     a) Select k from $\{1, \dots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \dots, \bar{w}_{j-1}^{(N)}\}$
     b) Sample $\xi \sim T_j(x_{j-1}^{(k)}, \cdot)$
2. $$x_j^{(i+1)} = \begin{cases} \xi, & \text{w.p. } \alpha(x_j^{(i)}, \xi) \\ x_j^{(i)}, & \text{w.p. } 1 - \alpha(x_j^{(i)}, \xi) \end{cases} \qquad \alpha(x, \xi) = \min\left\{1, \frac{\pi_j(\xi)\hat{\pi}_j(x)}{\pi_j(x)\hat{\pi}_j(\xi)}\right\}$$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$
   - It is easy to sample from $\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$
     a) Select k from $\{1, \dots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \dots, \bar{w}_{j-1}^{(N)}\}$
     b) Sample $\xi \sim T_j(x_{j-1}^{(k)}, \cdot)$
2. $$x_j^{(i+1)} = \begin{cases} \xi, & \text{w.p. } \alpha(x_j^{(i)}, \xi) \\ x_j^{(i)}, & \text{w.p. } 1 - \alpha(x_j^{(i)}, \xi) \end{cases} \qquad \alpha(x, \xi) = \min\left\{1, \frac{\pi_j(\xi)\hat{\pi}_j(x)}{\pi_j(x)\hat{\pi}_j(\xi)}\right\} =?$$
   - It is not clear how to compute $\alpha(x, \xi)$, since $\hat{\pi}_j$ does not have a PDF!

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$
   - It is easy to sample from $\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$
     a) Select k from $\{1, \dots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \dots, \bar{w}_{j-1}^{(N)}\}$
     b) Sample $\xi \sim T_j(x_{j-1}^{(k)}, \cdot)$
2. $$x_j^{(i+1)} = \begin{cases} \xi, & \text{w.p. } \alpha(x_j^{(i)}, \xi) \\ x_j^{(i)}, & \text{w.p. } 1 - \alpha(x_j^{(i)}, \xi) \end{cases} \qquad \alpha(x, \xi) = \min\left\{1, \frac{\pi_j(\xi)\hat{\pi}_j(x)}{\pi_j(x)\hat{\pi}_j(\xi)}\right\} =?$$
   - It is not clear how to compute $\alpha(x, \xi)$, since $\hat{\pi}_j$ does not have a PDF!

$$\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} q_j(x_{j-1}^{(i)}, x) \min\left\{1, \frac{\pi_j(x)}{\pi_j(x_{j-1}^{(i)})}\right\} dx + \sum_{i=1}^N A_i \delta_{x_{j-1}^{(i)}}(dx)$$

MCMC step

Our goal: To sample from π_j using the Independent Metropolis-Hastings algorithm with the global proposal distribution $\hat{\pi}_j$

$$\text{IMH UPDATE } x_j^{(i)} \to x_j^{(i+1)}$$

1. Generate a candidate state $\xi \sim \hat{\pi}_j$
   - It is easy to sample from $\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$
     a) Select k from $\{1, \dots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \dots, \bar{w}_{j-1}^{(N)}\}$
     b) Sample $\xi \sim T_j(x_{j-1}^{(k)}, \cdot)$
2. $$x_j^{(i+1)} = \begin{cases} \xi, & \text{w.p. } \alpha(x_j^{(i)}, \xi) \\ x_j^{(i)}, & \text{w.p. } 1 - \alpha(x_j^{(i)}, \xi) \end{cases} \qquad \alpha(x, \xi) = \min\left\{1, \frac{\pi_j(\xi)\hat{\pi}_j(x)}{\pi_j(x)\hat{\pi}_j(\xi)}\right\} =?$$
   - It is not clear how to compute $\alpha(x, \xi)$, since $\hat{\pi}_j$ does not have a PDF!

$$\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} q_j(x_{j-1}^{(i)}, x) \min\left\{1, \frac{\pi_j(x)}{\pi_j(x_{j-1}^{(i)})}\right\} dx + \sum_{i=1}^N A_i \delta_{x_{j-1}^{(i)}}(dx)$$

Key idea: Reduce the sample space $\mathcal{X} \rightsquigarrow \mathcal{X}_j^* \stackrel{\text{def}}{=} \mathcal{X} \setminus \{x_{j-1}^{(1)}, \dots, x_{j-1}^{(N)}\}$

AIMS at annealing level j

Markov chain $x_j^{(i)} \to x_j^{(i+1)} \sim \pi_j(x)$ on $\mathcal{X}_j^* = \mathcal{X} \setminus \{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\}$

AIMS at annealing level j

Markov chain $x_j^{(i)} \to x_j^{(i+1)} \sim \pi_j(x)$ on $\mathcal{X}_j^* = \mathcal{X} \setminus \{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\}$

1. Generate a global candidate state $\xi_g \sim \hat{\pi}_j$:

AIMS at annealing level j

Markov chain $x_j^{(i)} \to x_j^{(i+1)} \sim \pi_j(x)$ on $\mathcal{X}_j^* = \mathcal{X} \setminus \{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\}$

1. Generate a global candidate state $\xi_g \sim \hat{\pi}_j$:
   a) Select k from $\{1, \ldots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \ldots, \bar{w}_{j-1}^{(N)}\}$

AIMS at annealing level j

Markov chain $x_j^{(i)} \to x_j^{(i+1)} \sim \pi_j(x)$ on $\mathcal{X}_j^* = \mathcal{X} \setminus \{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\}$

1. Generate a global candidate state $\xi_g \sim \hat{\pi}_j$:
   a) Select k from $\{1, \ldots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \ldots, \bar{w}_{j-1}^{(N)}\}$
   b) Generate a local candidate state $\xi_l \sim q_j(x_{j-1}^{(k)}, \xi)$

AIMS at annealing level j

Markov chain $x_j^{(i)} \to x_j^{(i+1)} \sim \pi_j(x)$ on $\mathcal{X}_j^* = \mathcal{X} \setminus \{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\}$

1. Generate a global candidate state $\xi_g \sim \hat{\pi}_j$:
   a) Select k from $\{1, \ldots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \ldots, \bar{w}_{j-1}^{(N)}\}$
   b) Generate a local candidate state $\xi_l \sim q_j(x_{j-1}^{(k)}, \xi)$
   c) Accept/reject ξ_l by setting $\xi_g = \begin{cases} \xi_l, & \text{w.p. } \min\left\{1, \frac{\pi_j(\xi_l)}{\pi_j(x_{j-1}^{(k)})}\right\} \\ x_{j-1}^{(k)}, & \text{w.p. } 1 - \min\left\{1, \frac{\pi_j(\xi_l)}{\pi_j(x_{j-1}^{(k)})}\right\} \end{cases}$

AIMS at annealing level j

Markov chain $x_j^{(i)} \to x_j^{(i+1)} \sim \pi_j(x)$ on $\mathcal{X}_j^* = \mathcal{X} \setminus \{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\}$

1. Generate a global candidate state $\xi_g \sim \hat{\pi}_j$:
   a) Select k from $\{1, \ldots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \ldots, \bar{w}_{j-1}^{(N)}\}$
   b) Generate a local candidate state $\xi_l \sim q_j(x_{j-1}^{(k)}, \xi)$
   c) Accept/reject ξ_l by setting $\xi_g = \begin{cases} \xi_l, & \text{w.p. } \min\left\{1, \frac{\pi_j(\xi_l)}{\pi_j(x_{j-1}^{(k)})}\right\} \\ x_{j-1}^{(k)}, & \text{w.p. } 1 - \min\left\{1, \frac{\pi_j(\xi_l)}{\pi_j(x_{j-1}^{(k)})}\right\} \end{cases}$
2. Accepts/reject ξ_g as follows:

$$x_j^{(i+1)} = x_j^{(i)}, \text{ if } \xi_g = x_{j-1}^{(k)}$$

$$x_j^{(i+1)} = \begin{cases} \xi_g, & \text{w.p. } \min\left\{1, \frac{\pi_j(\xi_g)\hat{\pi}_j^c(x_j^{(i)})}{\pi_j(x_j^{(i)})\hat{\pi}_j^c(\xi_g)}\right\} \\ x_j^{(i)}, & \text{w.p. } 1 - \min\left\{1, \frac{\pi_j(\xi_g)\hat{\pi}_j^c(x_j^{(i)})}{\pi_j(x_j^{(i)})\hat{\pi}_j^c(\xi_g)}\right\} \end{cases}$$

AIMS at annealing level j

Markov chain $x_j^{(i)} \to x_j^{(i+1)} \sim \pi_j(x)$ on $\mathcal{X}_j^* = \mathcal{X} \setminus \{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\}$

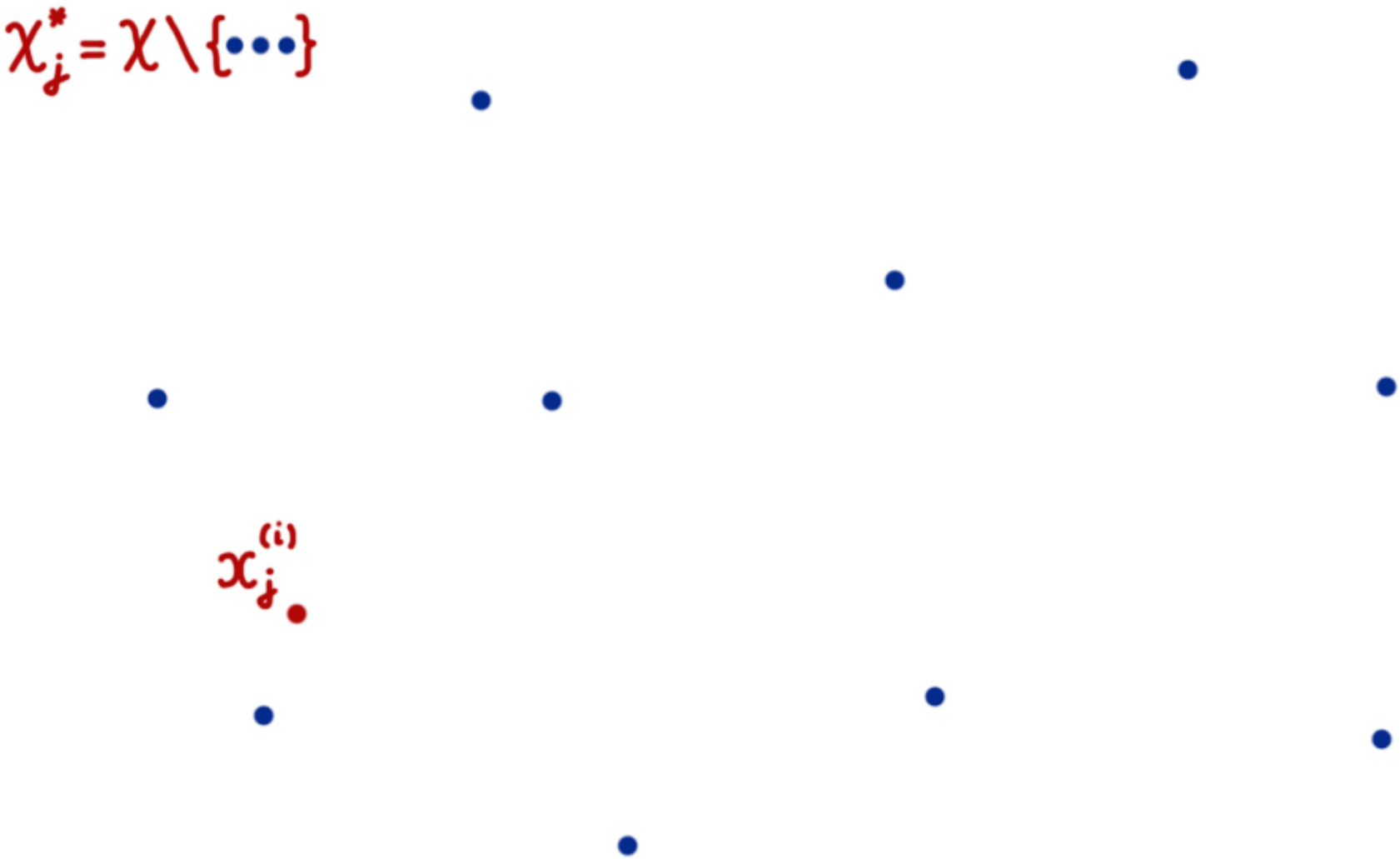
1. Generate a global candidate state $\xi_g \sim \hat{\pi}_j$:
   a) Select k from $\{1, \ldots, N\}$ with probabilities $\{\bar{w}_{j-1}^{(1)}, \ldots, \bar{w}_{j-1}^{(N)}\}$
   b) Generate a local candidate state $\xi_l \sim q_j(x_{j-1}^{(k)}, \xi)$
   c) Accept/reject ξ_l by setting $\xi_g = \begin{cases} \xi_l, & \text{w.p. } \min\left\{1, \frac{\pi_j(\xi_l)}{\pi_j(x_{j-1}^{(k)})}\right\} \\ x_{j-1}^{(k)}, & \text{w.p. } 1 - \min\left\{1, \frac{\pi_j(\xi_l)}{\pi_j(x_{j-1}^{(k)})}\right\} \end{cases}$
2. Accepts/reject ξ_g as follows:

$$x_j^{(i+1)} = x_j^{(i)}, \text{ if } \xi_g = x_{j-1}^{(k)}$$

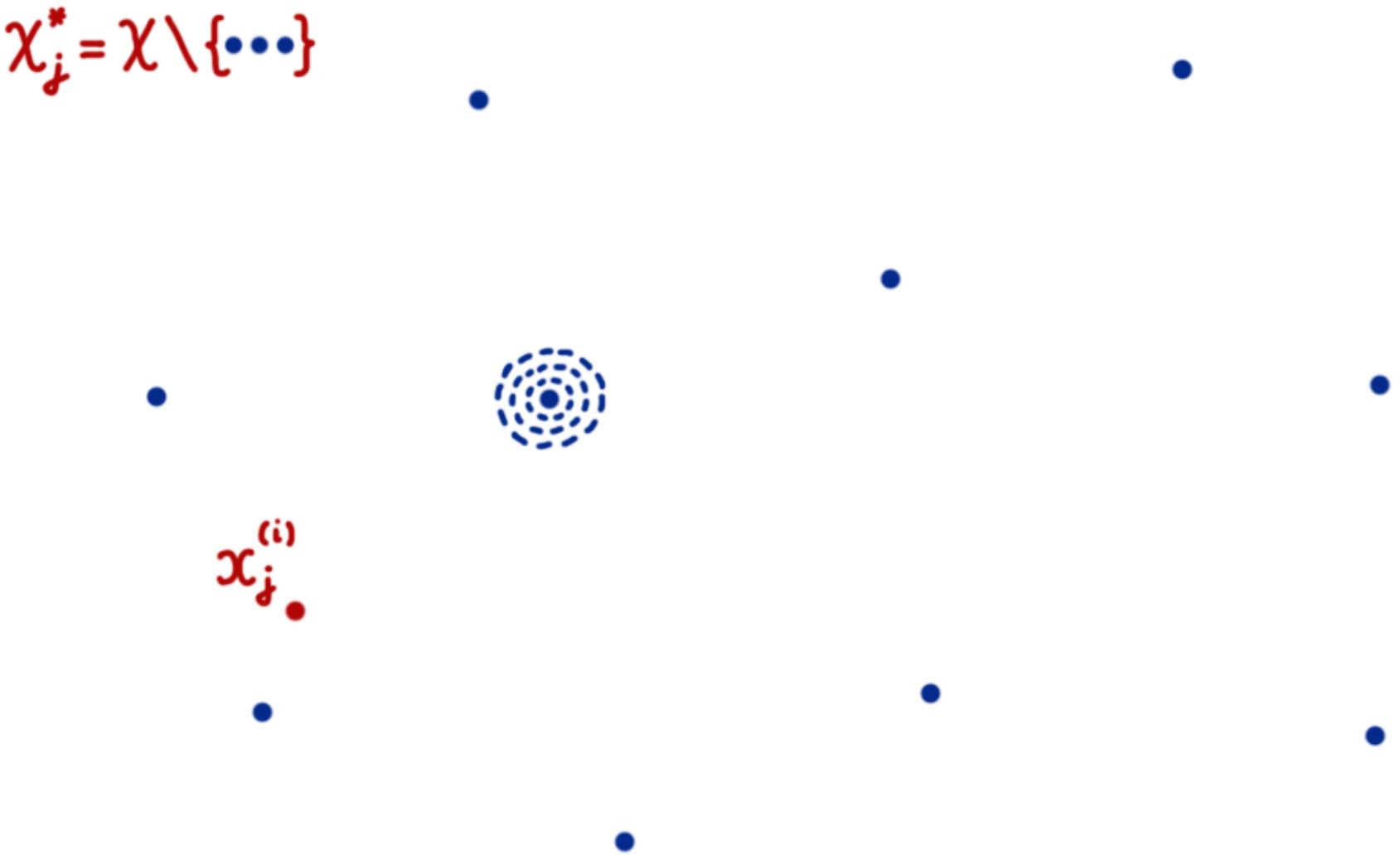
$$x_j^{(i+1)} = \begin{cases} \xi_g, & \text{w.p. } \min\left\{1, \frac{\pi_j(\xi_g)\hat{\pi}_j^c(x_j^{(i)})}{\pi_j(x_j^{(i)})\hat{\pi}_j^c(\xi_g)}\right\} \\ x_j^{(i)}, & \text{w.p. } 1 - \min\left\{1, \frac{\pi_j(\xi_g)\hat{\pi}_j^c(x_j^{(i)})}{\pi_j(x_j^{(i)})\hat{\pi}_j^c(\xi_g)}\right\} \end{cases}$$

$$\hat{\pi}_j^c(x) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} q_j(x_{j-1}^{(i)}, x) \min\left\{1, \frac{\pi_j(x)}{\pi_j(x_{j-1}^{(i)})}\right\}$$

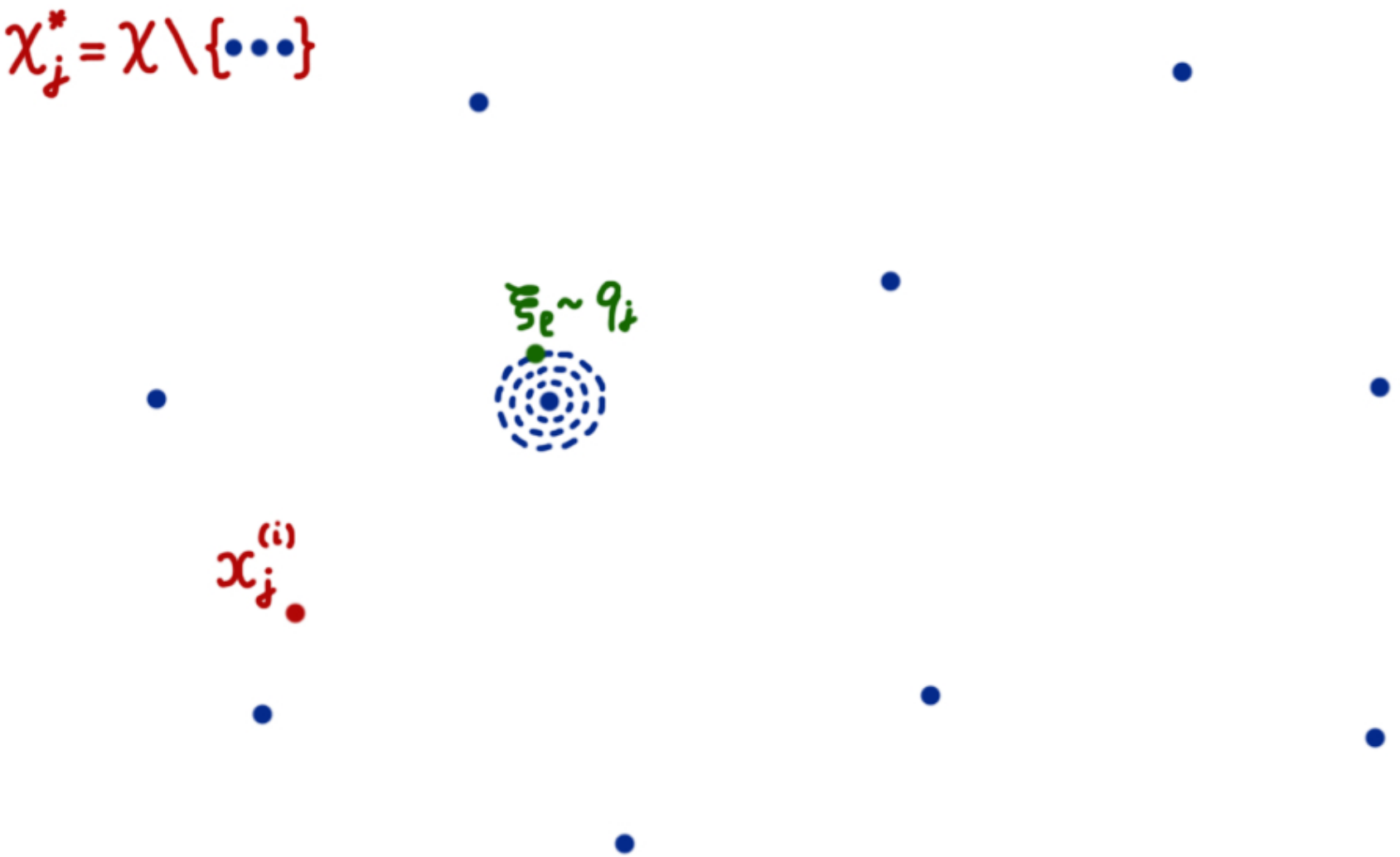
Schematic Illustration



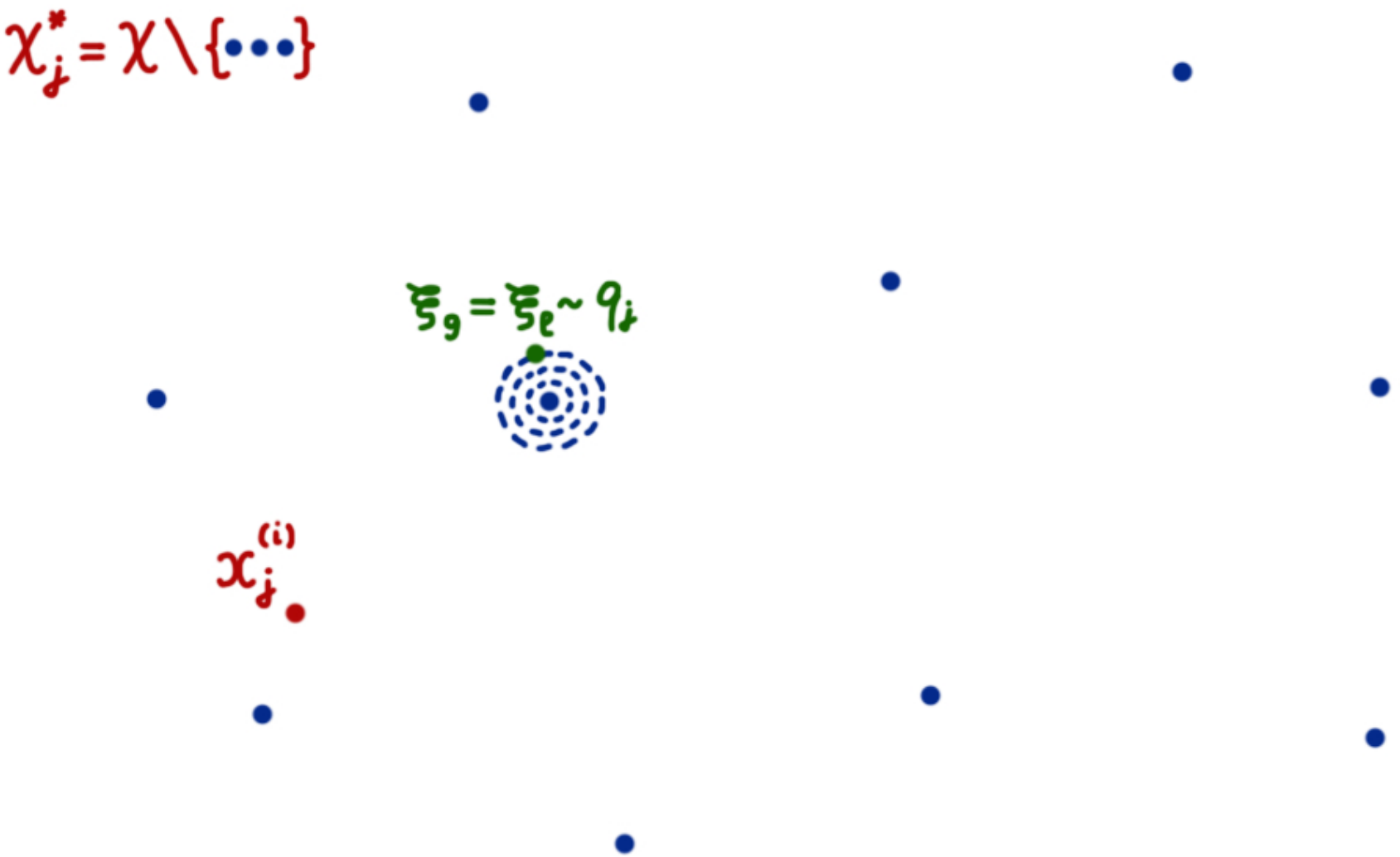
Schematic Illustration



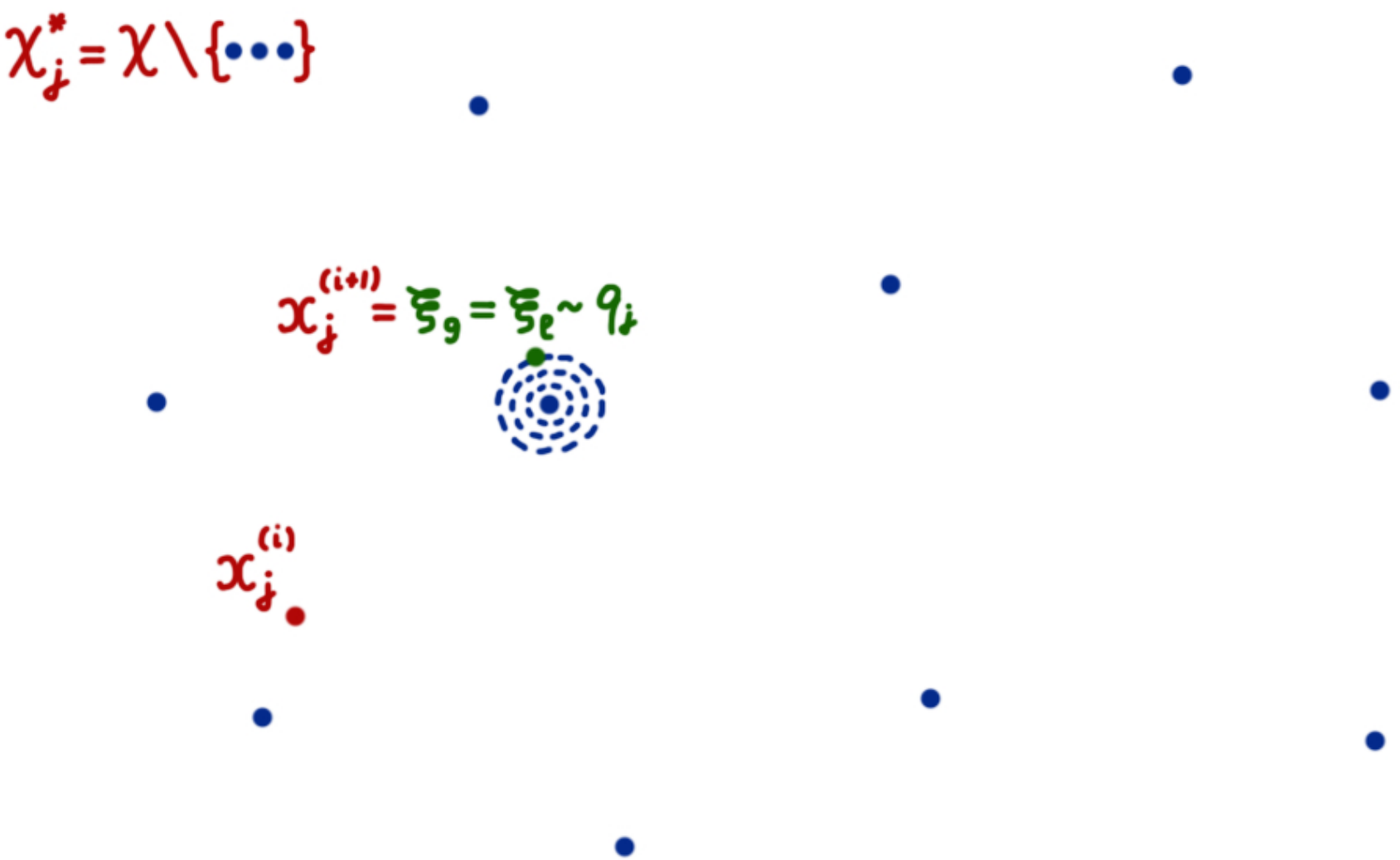
Schematic Illustration



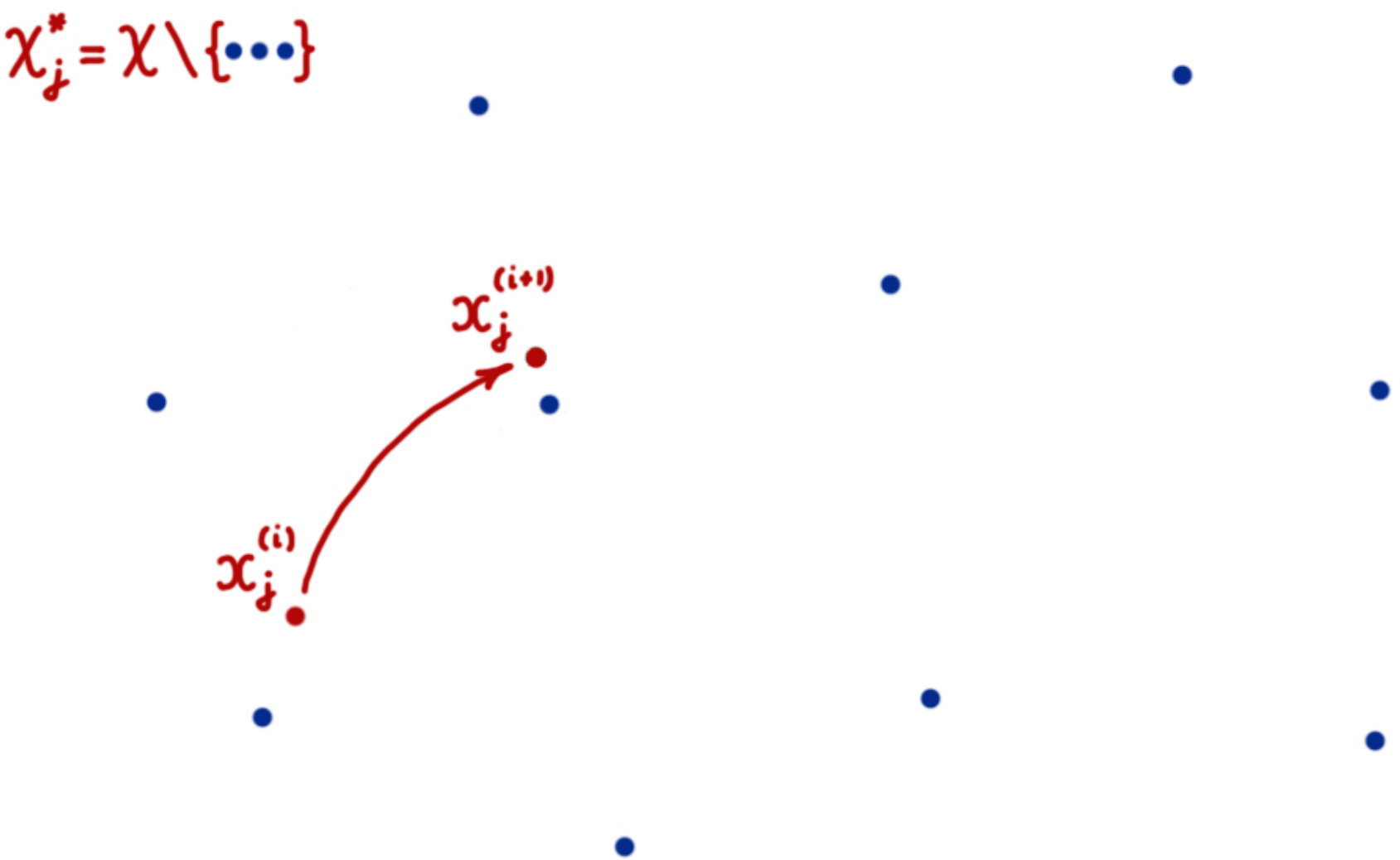
Schematic Illustration



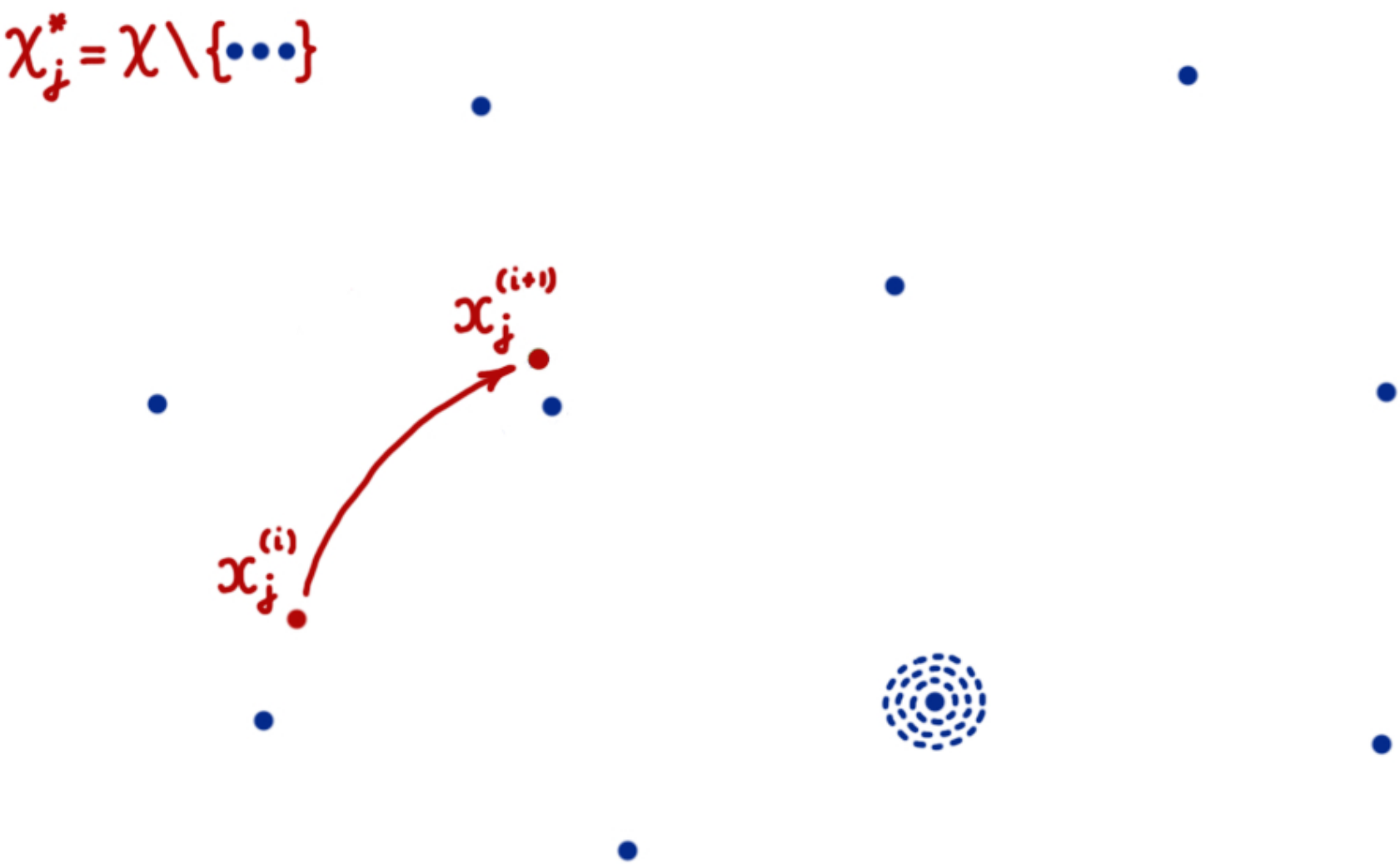
Schematic Illustration



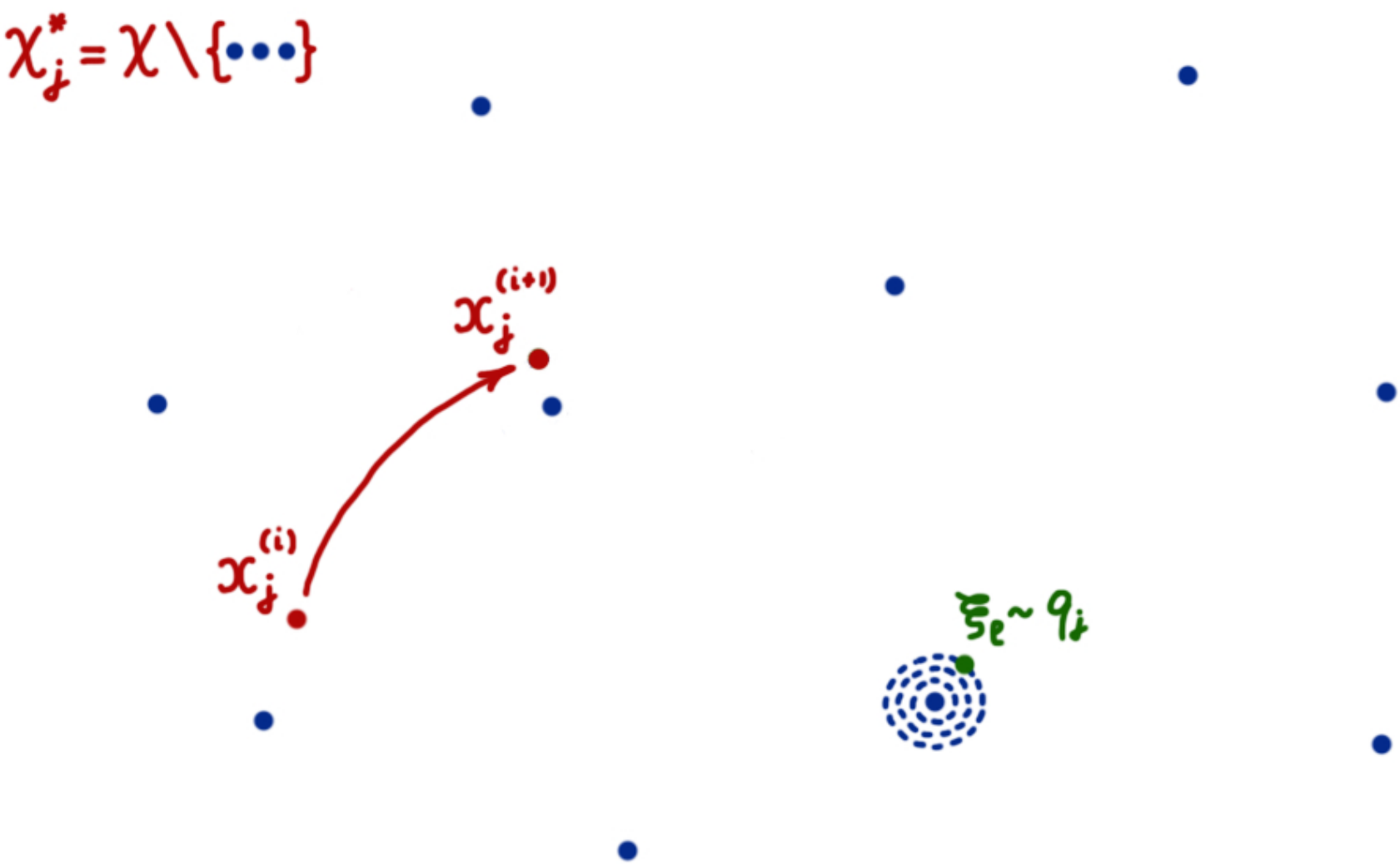
Schematic Illustration



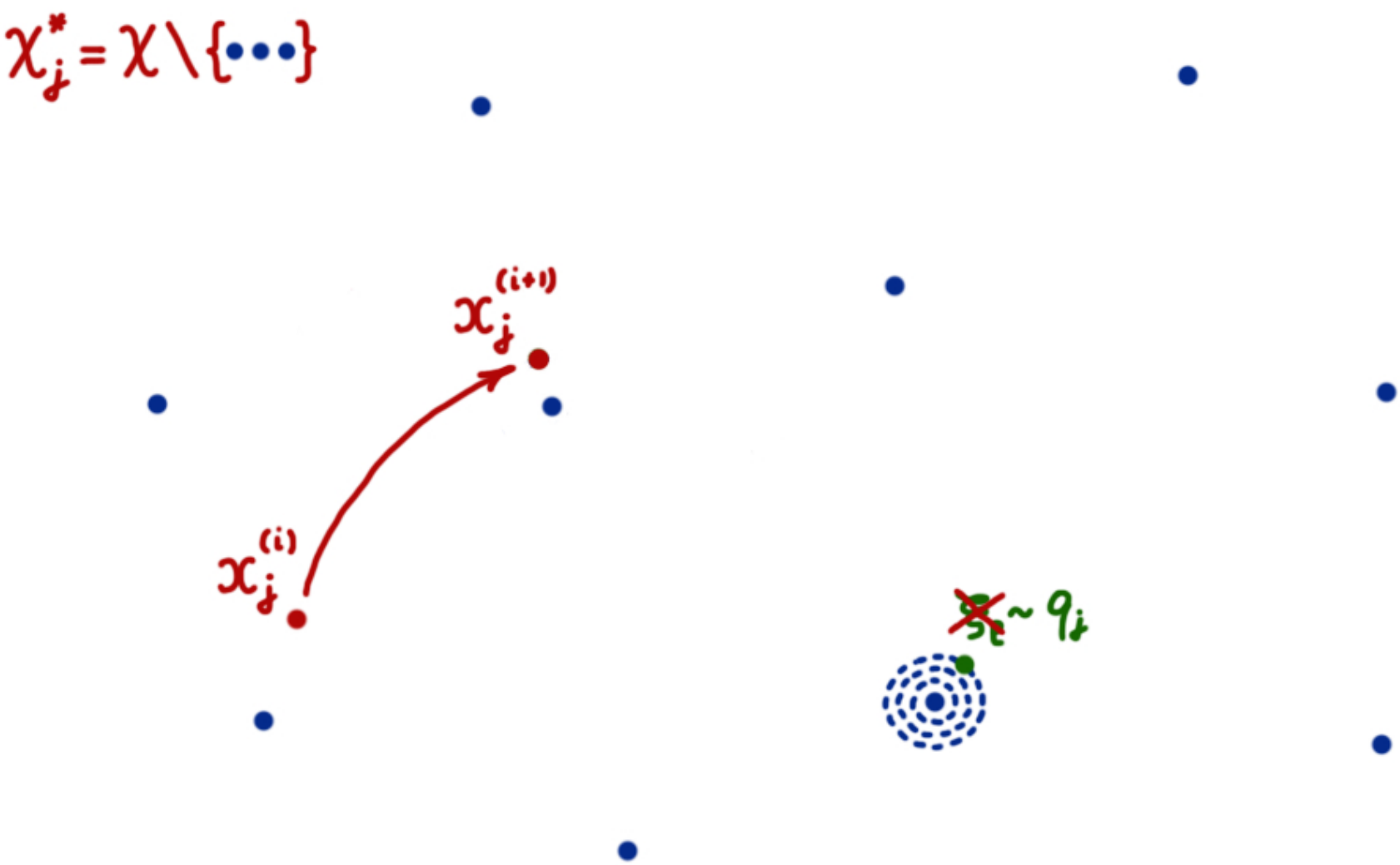
Schematic Illustration



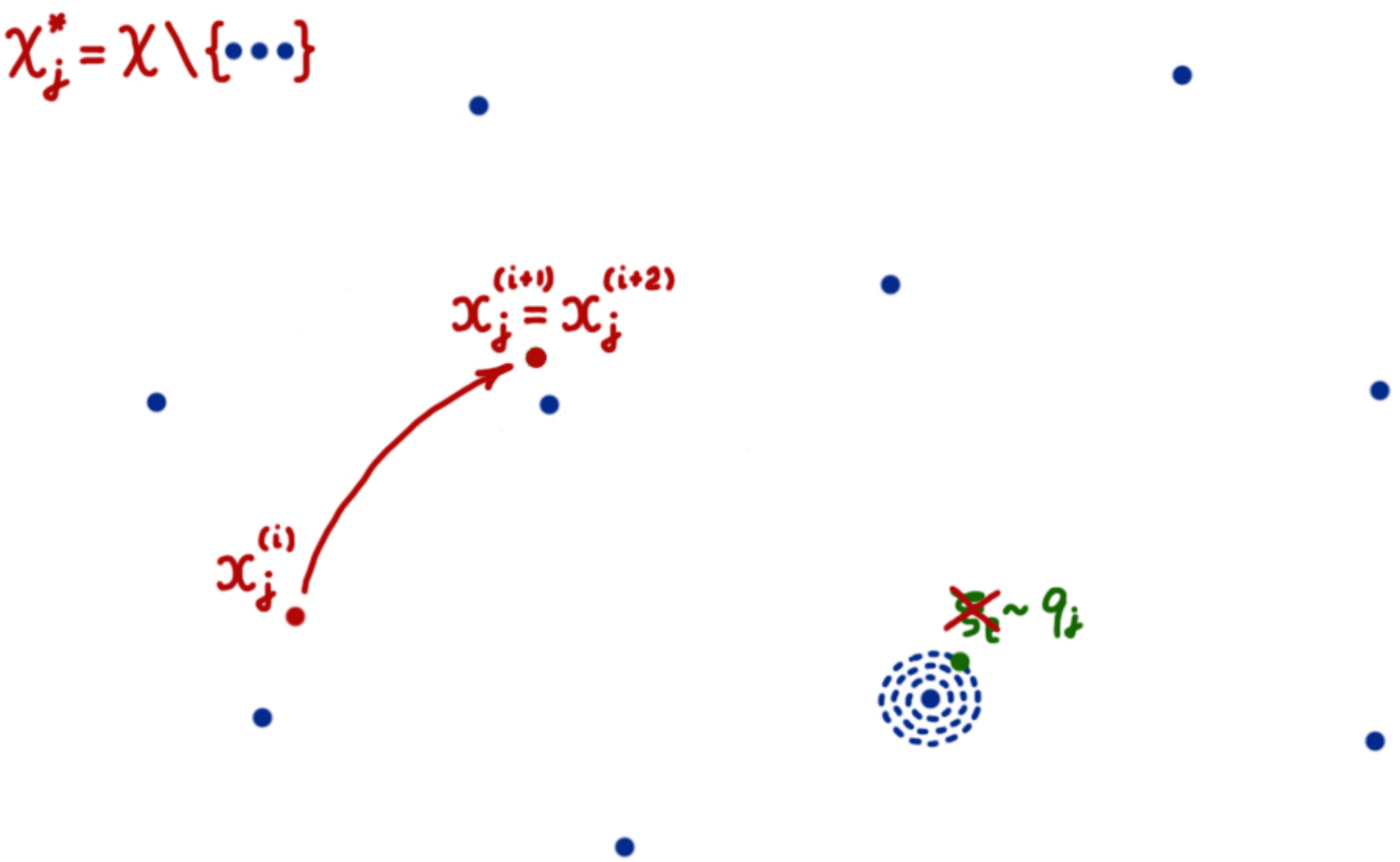
Schematic Illustration



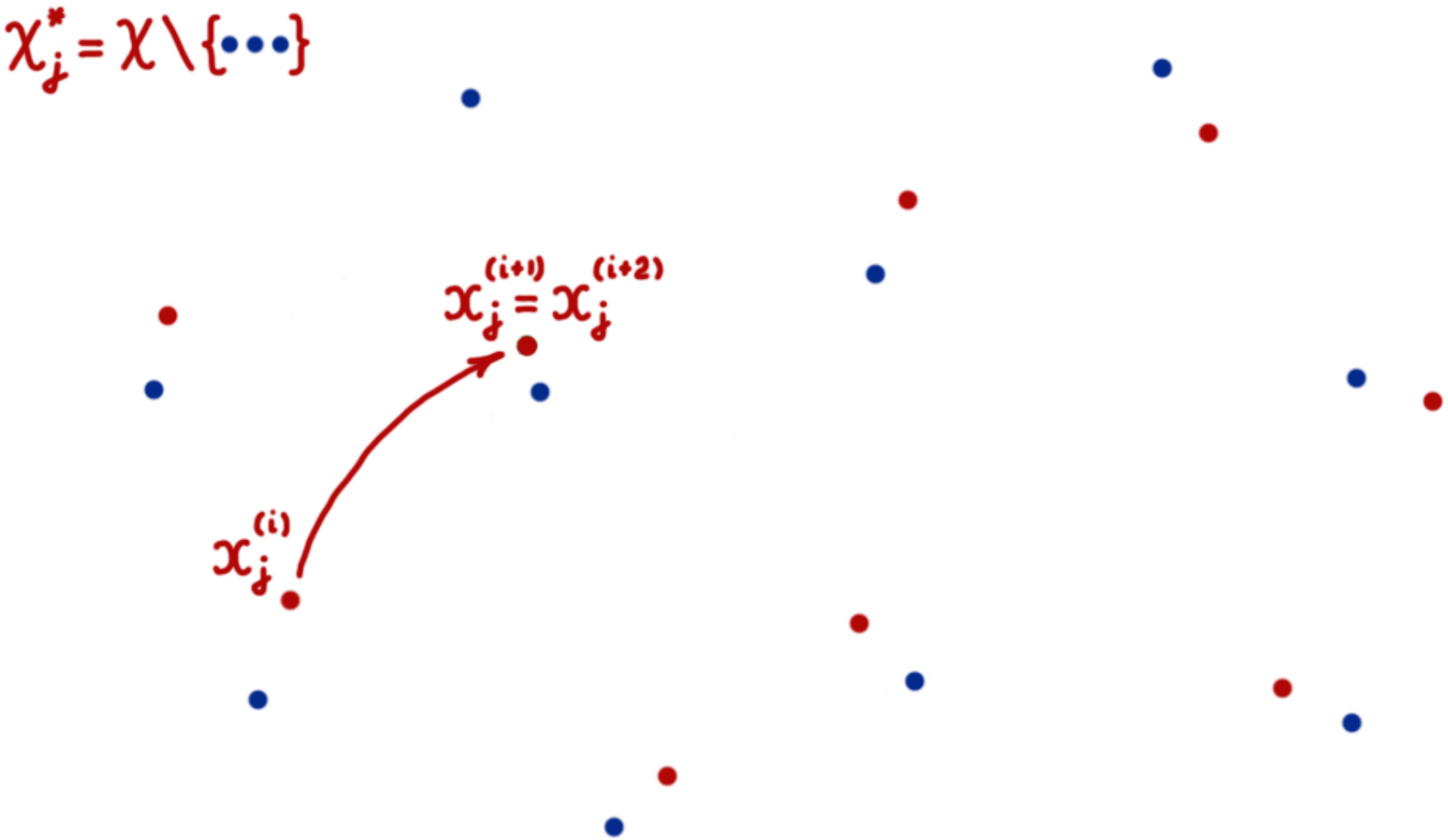
Schematic Illustration



Schematic Illustration



Schematic Illustration



Ergodic Properties of AIMS

Theorem (1)

Let $x_j^{(1)}, x_j^{(2)}, \ldots$ *be the Markov chain on* $\mathcal{X}_j^* = \mathcal{X} \setminus \left\{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\right\}$ *generated by AIMS and let* $\mathcal{K}_j$ *denote its transition kernel. Then* π_j *is the stationary distribution of the Markov chain and*

$$\lim_{n\to\infty} \|\mathcal{K}_j^n(x,\cdot) - \pi_j(\cdot)\|_{\mathrm{TV}} = 0, \qquad \forall x \in \mathcal{X}_j^*$$

Ergodic Properties of AIMS

Theorem (1)

Let $x_j^{(1)}, x_j^{(2)}, \ldots$ *be the Markov chain on* $\mathcal{X}_j^* = \mathcal{X} \setminus \left\{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\right\}$ *generated by AIMS and let* $\mathcal{K}_j$ *denote its transition kernel. Then* π_j *is the stationary distribution of the Markov chain and*

$$\lim_{n\to\infty} \|\mathcal{K}_j^n(x,\cdot) - \pi_j(\cdot)\|_{\mathrm{TV}} = 0, \qquad \forall x \in \mathcal{X}_j^*$$

Sketch of proof:

- Transition kernel

$$\mathcal{K}_j(x, dy) = \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dy) \min\left\{1, \frac{\pi_j(y)\hat{\pi}_j^c(x)}{\pi_j(x)\hat{\pi}_j^c(y)}\right\} dy + R_j(x)\delta_x(dy)$$

Ergodic Properties of AIMS

Theorem (1)

Let $x_j^{(1)}, x_j^{(2)}, \ldots$ *be the Markov chain on* $\mathcal{X}_j^* = \mathcal{X} \setminus \left\{x_{j-1}^{(1)}, \ldots, x_{j-1}^{(N)}\right\}$ *generated by AIMS and let* $\mathcal{K}_j$ *denote its transition kernel. Then* π_j *is the stationary distribution of the Markov chain and*

$$\lim_{n\to\infty} \|\mathcal{K}_j^n(x,\cdot) - \pi_j(\cdot)\|_{\mathrm{TV}} = 0, \qquad \forall x \in \mathcal{X}_j^*$$

Sketch of proof:

- Transition kernel

$$\mathcal{K}_j(x, dy) = \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dy) \min\left\{1, \frac{\pi_j(y)\hat{\pi}_j^c(x)}{\pi_j(x)\hat{\pi}_j^c(y)}\right\} dy + R_j(x)\delta_x(dy)$$

- The Detailed Balance Condition

$$\pi_j(dx)\mathcal{K}_j(x, dy) = \pi_j(dy)\mathcal{K}_j(y, dx)$$

Ergodic Properties of AIMS

Definition

A Markov chain $x^{(1)}, x^{(2)}, \ldots$ with transition kernel T is called uniformly ergodic if

$$\lim_{n\to\infty} \sup_{x\in\mathcal{X}} \|T^n(x,\cdot) - \pi(\cdot)\|_{\mathrm{TV}} = 0$$

Ergodic Properties of AIMS

Definition

A Markov chain $x^{(1)}, x^{(2)}, \ldots$ with transition kernel T is called uniformly ergodic if

$$\lim_{n\to\infty} \sup_{x\in\mathcal{X}} \|T^n(x,\cdot) - \pi(\cdot)\|_{\mathrm{TV}} = 0$$

Theorem (2)

If there exists a constant M such that $\pi_j(x) \leq M\hat{\pi}_j^c(x)$ for all $x \in \mathcal{X}_j^*$, then AIMS produces a uniformly ergodic chain and

$$\|\mathcal{K}_j^n(x,\cdot) - \pi_j(\cdot)\|_{\mathrm{TV}} \leq \left(1 - \frac{1}{M}\right)^n, \qquad \forall x \in \mathcal{X}_j^*$$

Ergodic Properties of AIMS

Definition

A Markov chain $x^{(1)}, x^{(2)}, \ldots$ with transition kernel T is called uniformly ergodic if

$$\lim_{n\to\infty} \sup_{x\in\mathcal{X}} \|T^n(x,\cdot) - \pi(\cdot)\|_{\mathrm{TV}} = 0$$

Theorem (2)

If there exists a constant M such that $\pi_j(x) \leq M\hat{\pi}_j^c(x)$ for all $x \in \mathcal{X}_j^*$, then AIMS produces a uniformly ergodic chain and

$$\|\mathcal{K}_j^n(x,\cdot) - \pi_j(\cdot)\|_{\mathrm{TV}} \leq \left(1 - \frac{1}{M}\right)^n, \qquad \forall x \in \mathcal{X}_j^*$$

Corollary

If $\mathcal{X} \subset \mathbb{R}^d$ is a compact set and q_j is a "good" local proposal PDF, then such a constant M exists.

Ergodic Properties of AIMS

$$\bar{\mathcal{A}}_j = \mathbb{E}_{\pi_j}\left[\mathbb{P}\left(x \xrightarrow{\kappa_j} y \neq x\right)\right] = \text{“acceptance rate”}$$

Ergodic Properties of AIMS

$$\bar{\mathcal{A}}_j = \mathbb{E}_{\pi_j}\left[\mathbb{P}\left(x \xrightarrow{\kappa_j} y \neq x\right)\right] = \text{“acceptance rate”}$$

- RWMH: neither high nor low acceptance rate is good

Ergodic Properties of AIMS

$$\bar{\mathcal{A}}_j = \mathbb{E}_{\pi_j}\left[\mathbb{P}\left(x \xrightarrow{\kappa_j} y \neq x\right)\right] = \text{“acceptance rate”}$$

- RWMH: neither high nor low acceptance rate is good
- IMH: the higher acceptance rate, the better

Ergodic Properties of AIMS

$$\bar{\mathcal{A}}_j = \mathbb{E}_{\pi_j}\left[\mathbb{P}\left(x \xrightarrow{\kappa_j} y \neq x\right)\right] = \text{“acceptance rate”}$$

- RWMH: neither high nor low acceptance rate is good
- IMH: the higher acceptance rate, the better

Theorem (3)

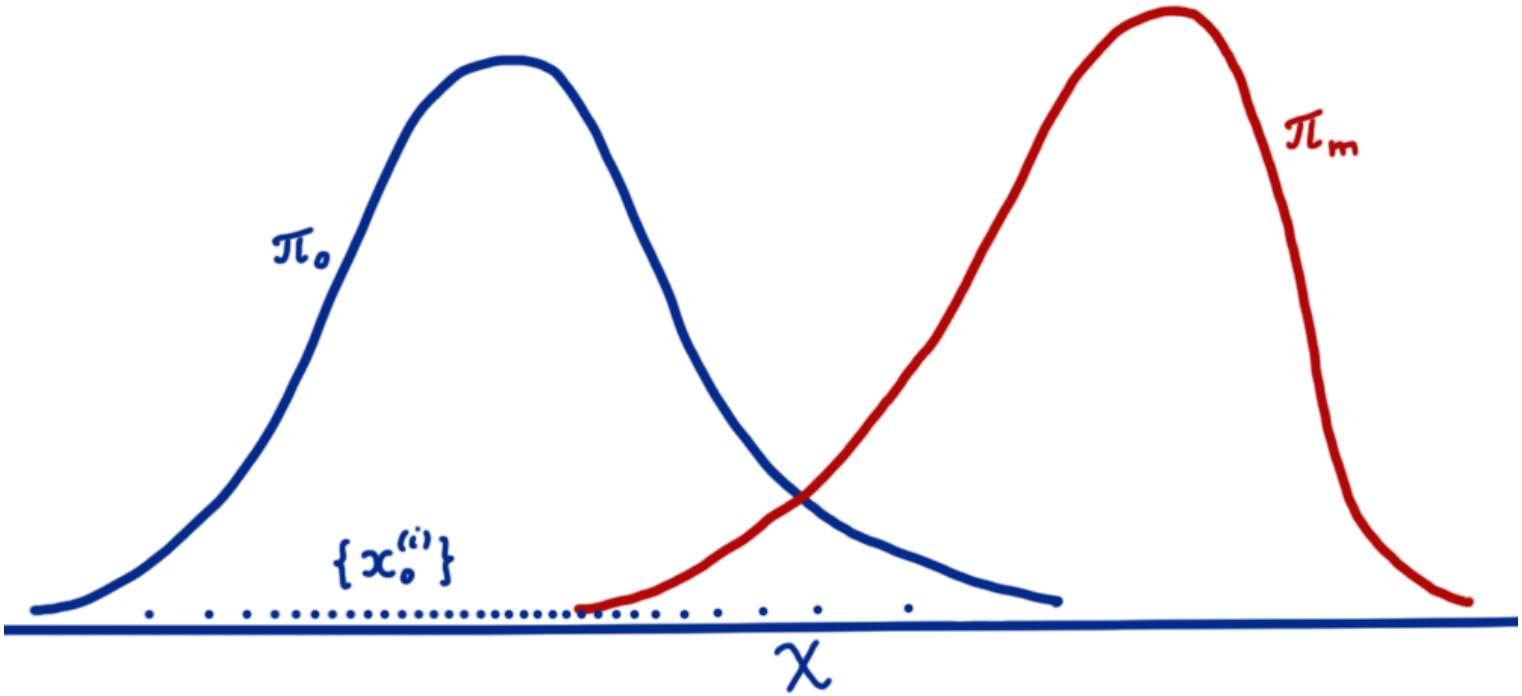
1. $$\bar{\mathcal{A}}_j \leq \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} \mathbb{P}\left(x_{j-1}^{(i)} \xrightarrow{T_j} \xi_g \neq x_{j-1}^{(i)}\right)$$
2. *If there exists a constant* M *such that* $\pi_j(x) \leq M\hat{\pi}_j^c(x)$ *for all* $x \in \mathcal{X}_j^*$, *then*

$$\bar{\mathcal{A}}_j \geq \frac{1}{M}$$

Annealing

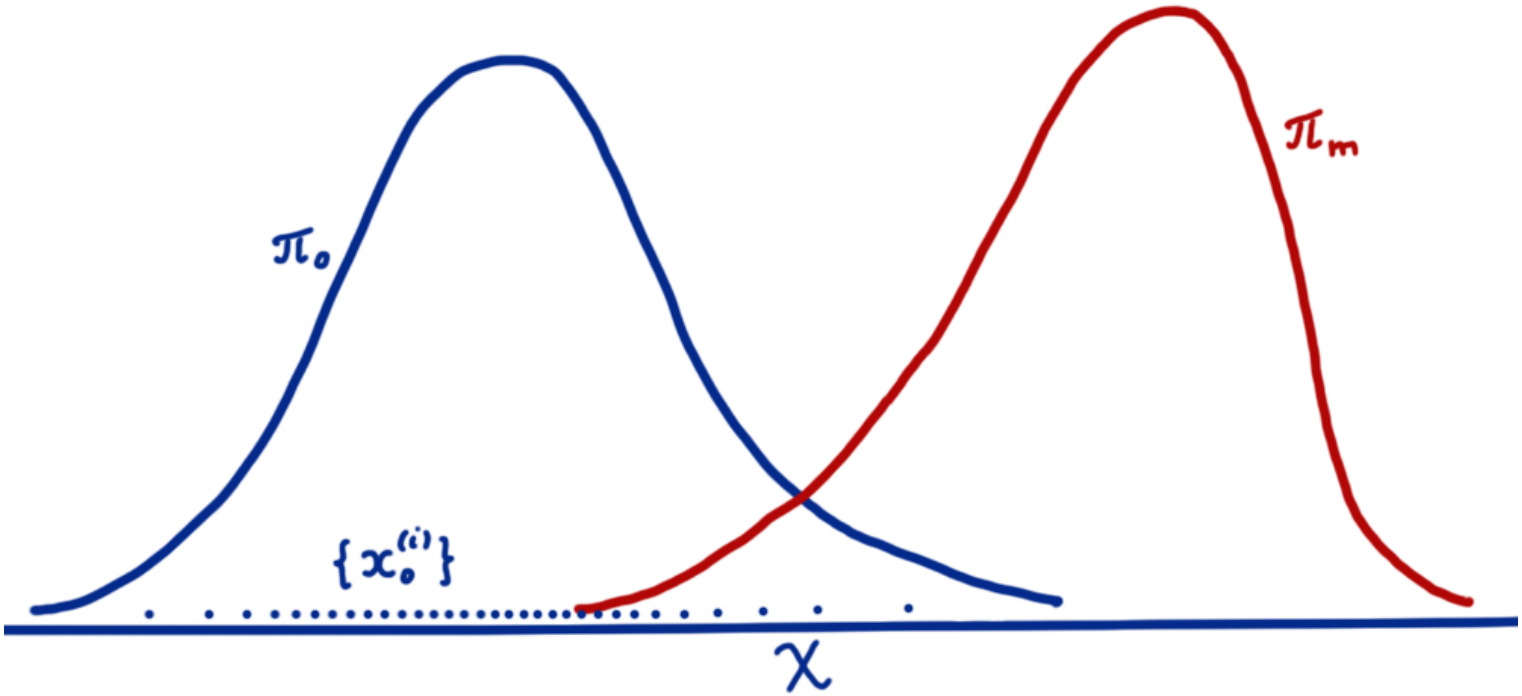
Annealing

- Why do we need intermediate distributions $\pi_0, \pi_1, \ldots, \pi_m = \pi$?



Annealing

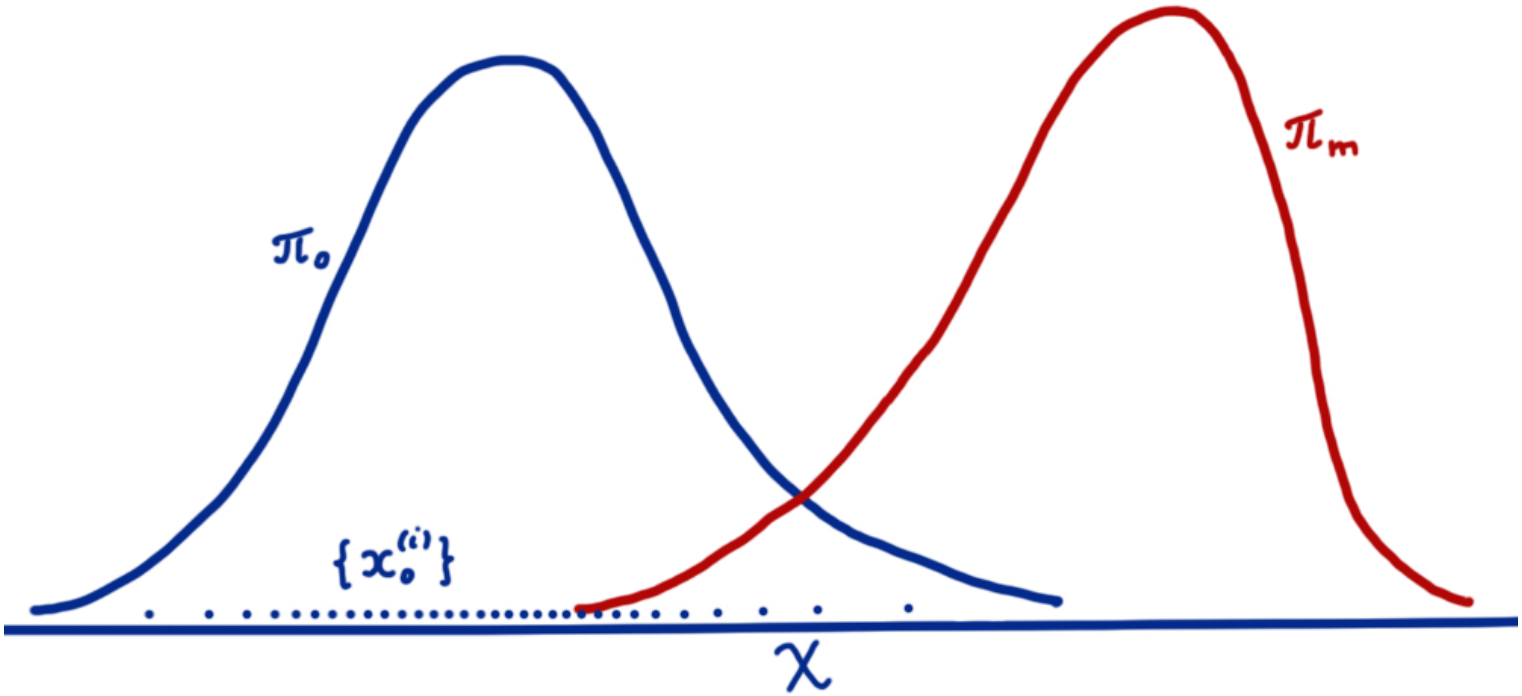
- Why do we need intermediate distributions $\pi_0, \pi_1, \ldots, \pi_m = \pi$?



- IMH: the proposal PDF must be a good approximation of the target PDF

Annealing

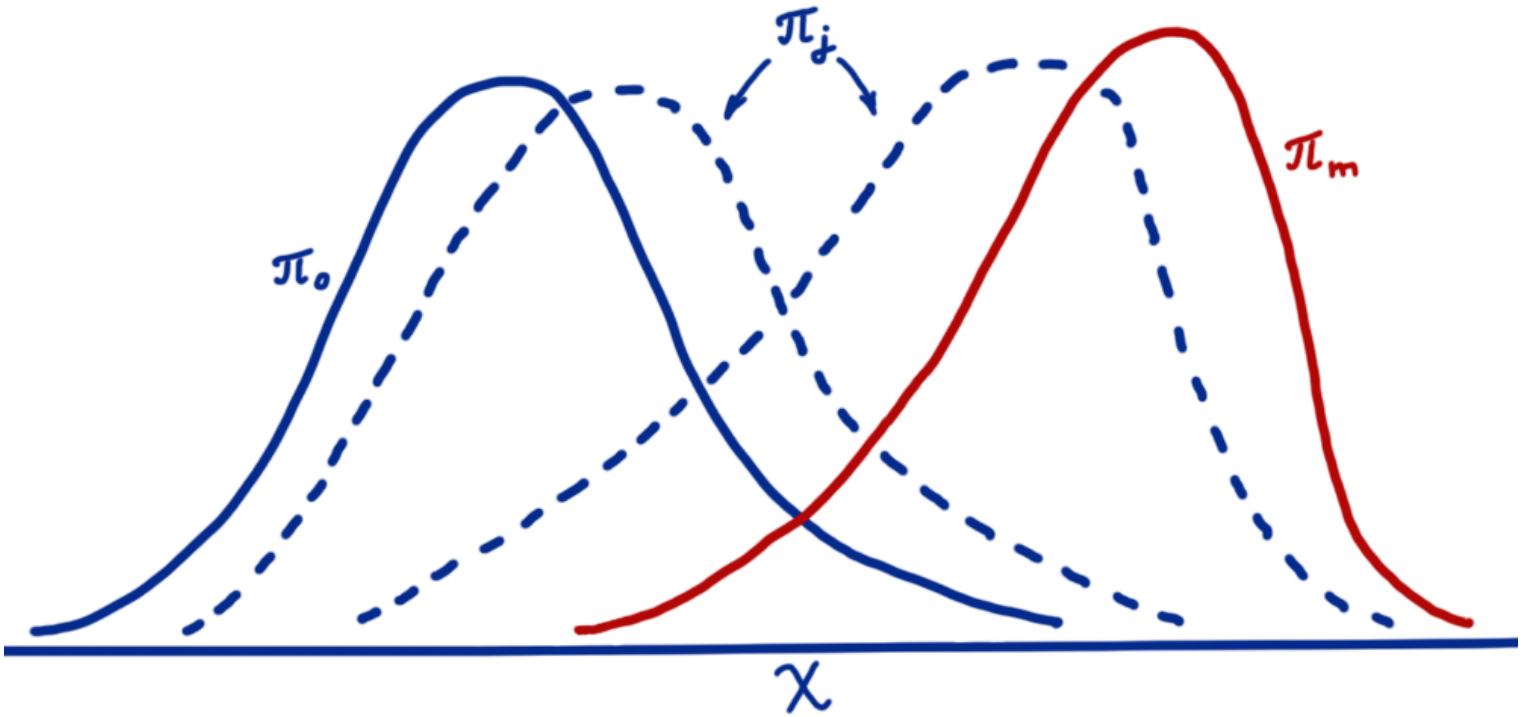
- Why do we need intermediate distributions $\pi_0, \pi_1, \ldots, \pi_m = \pi$?



- IMH: the proposal PDF must be a good approximation of the target PDF
- If π_0 and π_m are too different $\Rightarrow$ $\hat{\pi}_m$ is a very poor approximation of π_m

Annealing

- Why do we need intermediate distributions $\pi_0, \pi_1, \ldots, \pi_m = \pi$?



- IMH: the proposal PDF must be a good approximation of the target PDF
- If π_0 and π_m are too different $\Rightarrow$ $\hat{\pi}_m$ is a very poor approximation of π_m

Annealing scheme

- What annealing scheme should we use?

Annealing scheme

- What annealing scheme should we use?
- In Bayesian framework:

$$\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}, \qquad 0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$$

Annealing scheme

- What annealing scheme should we use?
- In Bayesian framework:

$$\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}, \qquad 0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$$

- Effective Sample Size measures how different π_{j-1} is from π_j

$$N_{\text{eff}} = \frac{N}{1 + \text{var}_{\pi_{j-1}} \left[\frac{\pi_j(x)}{\pi_{j-1}(x)} \right]}$$

Annealing scheme

- What annealing scheme should we use?
- In Bayesian framework:

$$\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}, \qquad 0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$$

- Effective Sample Size measures how different π_{j-1} is from π_j

$$N_{\text{eff}} = \frac{N}{1 + \text{var}_{\pi_{j-1}}\left[\frac{\pi_j(x)}{\pi_{j-1}(x)}\right]} \approx \left[\sum_{i=1}^{N} (\bar{w}_{j-1}^{(i)})^2\right]^{-1}$$

Annealing scheme

- What annealing scheme should we use?
- In Bayesian framework:

$$\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}, \qquad 0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$$

- Effective Sample Size measures how different π_{j-1} is from π_j

$$N_{\text{eff}} = \frac{N}{1 + \text{var}_{\pi_{j-1}}\left[\frac{\pi_j(x)}{\pi_{j-1}(x)}\right]} \approx \left[\sum_{i=1}^{N} (\bar{w}_{j-1}^{(i)})^2\right]^{-1}$$

- Interpretation: N weighted samples $(x_{j-1}^{(1)}, \bar{w}_{j-1}^{(1)}), \ldots, (x_{j-1}^{(N)}, \bar{w}_{j-1}^{(N)}) \dot{\sim} \pi_j$ are worth $N_{\text{eff}} (\leq N)$ i.i.d. samples from π_j

Annealing scheme

- What annealing scheme should we use?
- In Bayesian framework:

$$\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}, \qquad 0 = \beta_0 < \beta_1 < \ldots < \beta_m = 1$$

- Effective Sample Size measures how different π_{j-1} is from π_j

$$N_{\text{eff}} = \frac{N}{1 + \text{var}_{\pi_{j-1}}\left[\frac{\pi_j(x)}{\pi_{j-1}(x)}\right]} \approx \left[\sum_{i=1}^{N} (\bar{w}_{j-1}^{(i)})^2\right]^{-1}$$

- Interpretation: N weighted samples $(x_{j-1}^{(1)}, \bar{w}_{j-1}^{(1)}), \ldots, (x_{j-1}^{(N)}, \bar{w}_{j-1}^{(N)}) \dot{\sim} \pi_j$ are worth $N_{\text{eff}} (\leq N)$ i.i.d. samples from π_j
- Equation on β_j:

$$\frac{N^{\text{eff}}}{N} \approx \left[N \sum_{i=1}^{N} (\bar{w}_{j-1}^{(i)})^2\right]^{-1} = \gamma \in (0, 1)$$

Asymptotically Independent Markov Sampling Procedure

Asymptotically Independent Markov Sampling Procedure

1. Sample $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$

Asymptotically Independent Markov Sampling Procedure

1. Sample $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$

$$\text{At annealing level } j = 1, 2, \ldots:$$

2. Determine the next intermediate distribution $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$

$$\left[N \frac{\sum_{i=1}^{N} L(x_{j-1}^{(i)})^{2(\beta_j - \beta_{j-1})}}{\left(\sum_{i=1}^{N} L(x_{j-1}^{(i)})^{\beta_j - \beta_{j-1}} \right)^2} \right]^{-1} = \gamma$$

Asymptotically Independent Markov Sampling Procedure

1. Sample $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$

$$\text{At annealing level } j = 1, 2, \ldots:$$

2. Determine the next intermediate distribution $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$

$$\left[N \frac{\sum_{i=1}^N L(x_{j-1}^{(i)})^{2(\beta_j - \beta_{j-1})}}{\left(\sum_{i=1}^N L(x_{j-1}^{(i)})^{\beta_j - \beta_{j-1}} \right)^2} \right]^{-1} = \gamma$$

3. Find its Importance Sampling approximation $\hat{\pi}_j \approx \pi_j$

$$\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$$

Asymptotically Independent Markov Sampling Procedure

1. Sample $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$

$$\text{At annealing level } j = 1, 2, \ldots:$$

2. Determine the next intermediate distribution $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$

$$\left[N \frac{\sum_{i=1}^N L(x_{j-1}^{(i)})^{2(\beta_j - \beta_{j-1})}}{\left(\sum_{i=1}^N L(x_{j-1}^{(i)})^{\beta_j - \beta_{j-1}} \right)^2} \right]^{-1} = \gamma$$

3. Find its Importance Sampling approximation $\hat{\pi}_j \approx \pi_j$

$$\hat{\pi}_j(dx) = \sum_{i=1}^N \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$$

4. Sample $x_j^{(1)}, \ldots, x_j^{(N)} \sim \pi_j(x)$ using $\hat{\pi}_j$ as the global proposal distribution

Asymptotically Independent Markov Sampling Procedure

1. Sample $x_0^{(1)}, \ldots, x_0^{(N)} \sim \pi_0(x)$

$$\text{At annealing level } j = 1, 2, \ldots:$$

2. Determine the next intermediate distribution $\pi_j(x) \propto \pi_0(x) L(x)^{\beta_j}$

$$\left[N \frac{\sum_{i=1}^{N} L(x_{j-1}^{(i)})^{2(\beta_j - \beta_{j-1})}}{\left(\sum_{i=1}^{N} L(x_{j-1}^{(i)})^{\beta_j - \beta_{j-1}} \right)^2} \right]^{-1} = \gamma$$

3. Find its Importance Sampling approximation $\hat{\pi}_j \approx \pi_j$

$$\hat{\pi}_j(dx) = \sum_{i=1}^{N} \bar{w}_{j-1}^{(i)} T_j(x_{j-1}^{(i)}, dx)$$

4. Sample $x_j^{(1)}, \ldots, x_j^{(N)} \sim \pi_j(x)$ using $\hat{\pi}_j$ as the global proposal distribution
5. Repeat $2, 3, 4$ until the posterior $\pi(x) \propto \pi_0(x) L(x)$ has been sampled

Example: multi-modal mixture of Gaussians in 2D

- Target PDF:

$$\pi(x) \propto$$

$$\mathcal{U}_{[0,a]^2}(x) \cdot \sum_{i=1}^{M} w_i \mathcal{N}_{(\mu_i, \sigma^2 \mathbb{I}_2)}(x)$$

  - $a = 10,\ M = 10,\ w_i = 0.1$
  - $\mu_i \sim \mathcal{U}_{[0,a]^2},\ \sigma = 0.1$

Example: multi-modal mixture of Gaussians in 2D

- Target PDF:

$$\pi(x) \propto$$

$$\mathcal{U}_{[0,a]^2}(x) \cdot \sum_{i=1}^{M} w_i \mathcal{N}_{(\mu_i, \sigma^2 \mathbb{I}_2)}(x)$$

  - $a = 10,\ M = 10,\ w_i = 0.1$
  - $\mu_i \sim \mathcal{U}_{[0,a]^2},\ \sigma = 0.1$

- AIMS parameters:
  - $\gamma = 1/2$
  - $N = 10^3$ per annealing level
  - $q_j(x, y) = \mathcal{N}_{(x, c^2 \mathbb{I}_2)}(y)$
  - $c = 0.2$

Example: multi-modal mixture of Gaussians in 2D

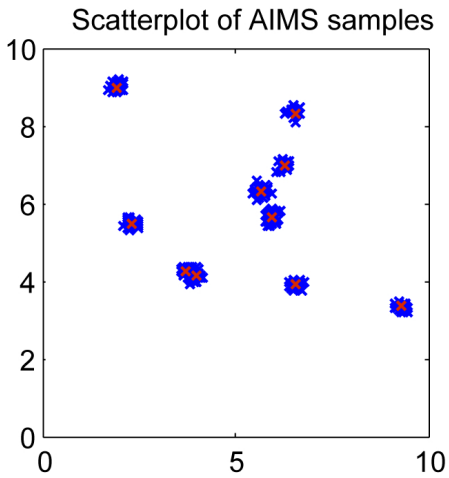
- Target PDF:

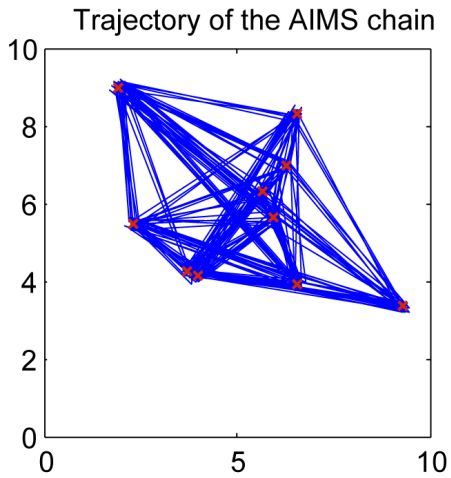
$$\pi(x) \propto$$

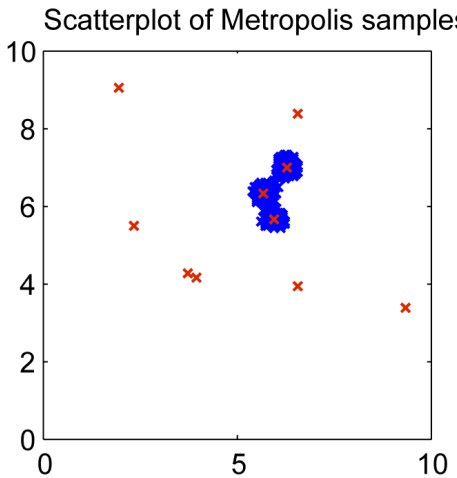
$$\mathcal{U}_{[0,a]^2}(x) \cdot \sum_{i=1}^{M} w_i \mathcal{N}_{(\mu_i, \sigma^2 \mathbb{I}_2)}(x)$$

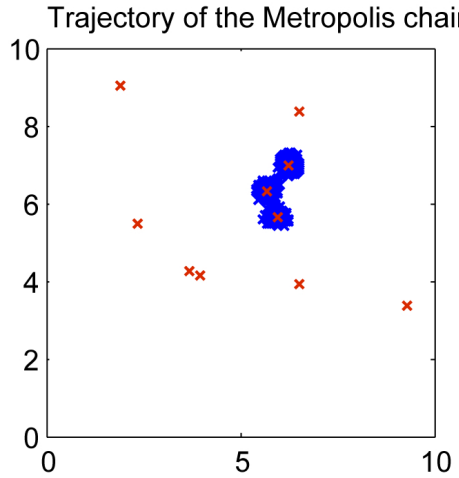
  - $a = 10,\ M = 10,\ w_i = 0.1$
  - $\mu_i \sim \mathcal{U}_{[0,a]^2},\ \sigma = 0.1$

- AIMS parameters:
  - $\gamma = 1/2$
  - $N = 10^3$ per annealing level
  - $q_j(x, y) = \mathcal{N}_{(x, c^2 \mathbb{I}_2)}(y)$
  - $c = 0.2$









Example: multi-modal mixture of Gaussians in 2D

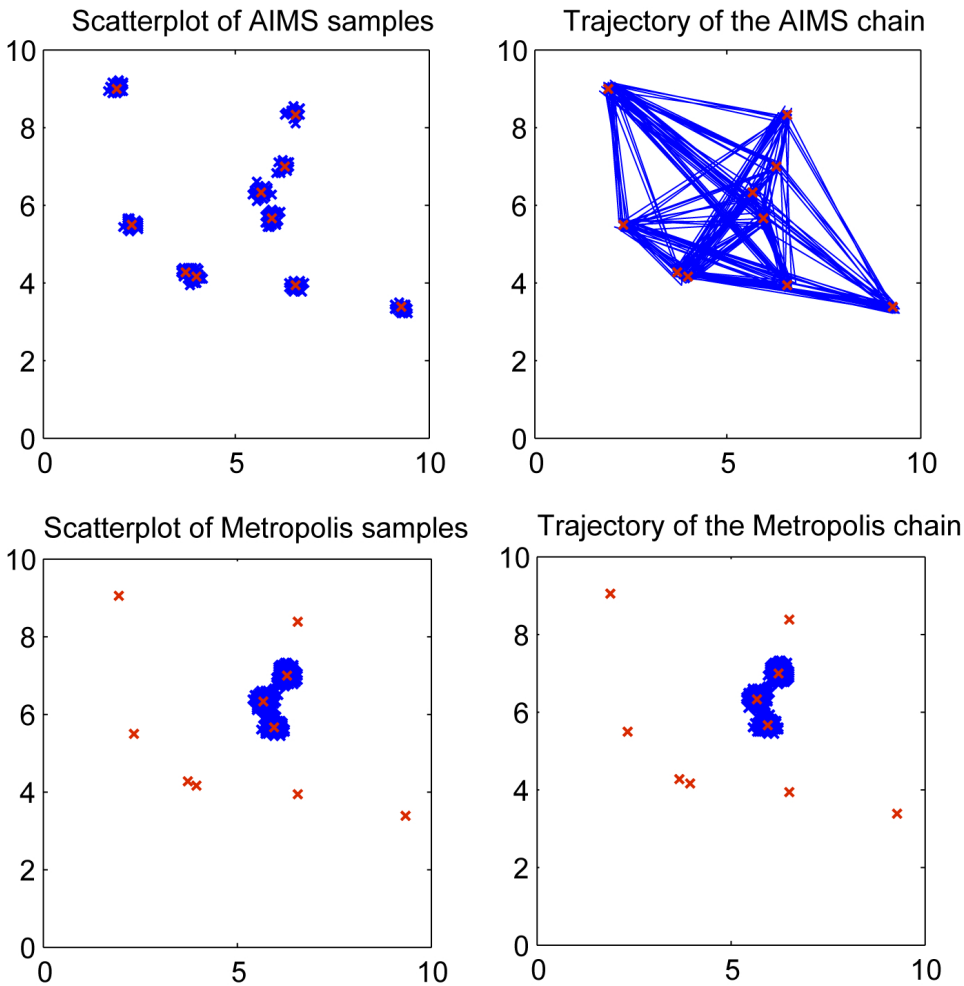
- Target PDF:

$$\pi(x) \propto$$

$$\mathcal{U}_{[0,a]^2}(x) \cdot \sum_{i=1}^{M} w_i \mathcal{N}_{(\mu_i, \sigma^2 \mathbb{I}_2)}(x)$$

  - $a = 10,\ M = 10,\ w_i = 0.1$
  - $\mu_i \sim \mathcal{U}_{[0,a]^2},\ \sigma = 0.1$

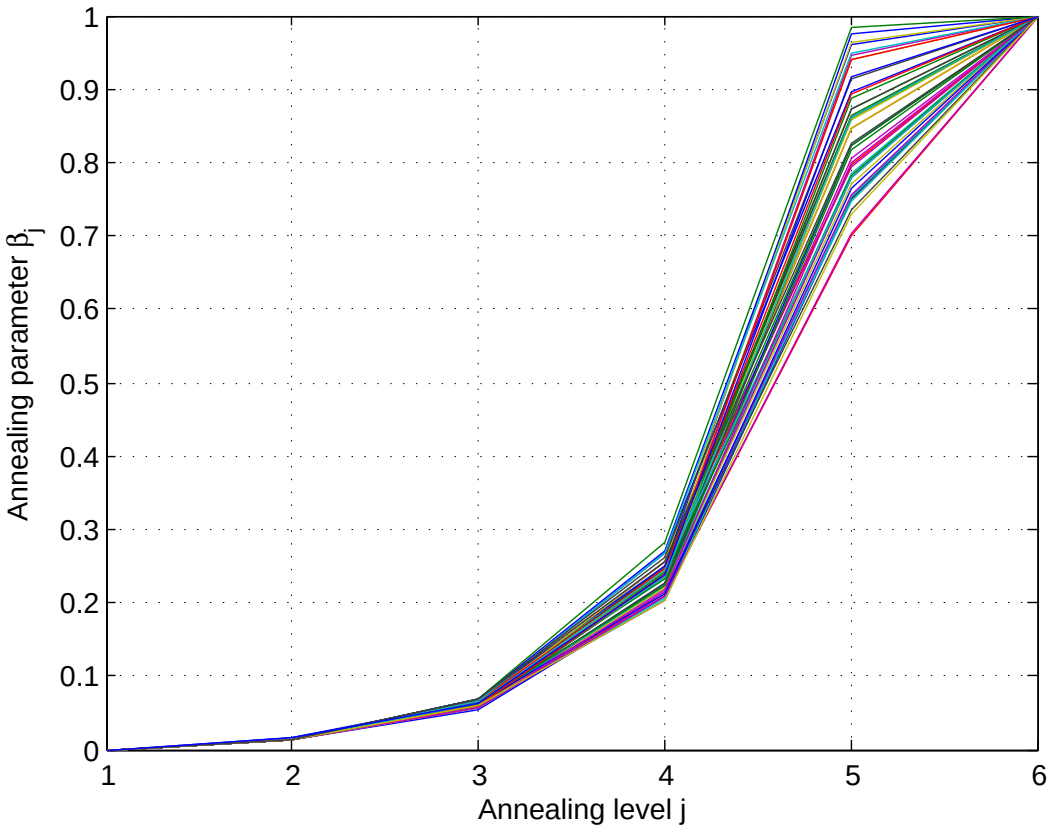
- AIMS parameters:
  - $\gamma = 1/2$
  - $N = 10^3$ per annealing level
  - $q_j(x, y) = \mathcal{N}_{(x, c^2 \mathbb{I}_2)}(y)$
  - $c = 0.2$



- AIMS successfully samples all 10 modes with correct proportion of times
- RWMH completely missed 7 modes

Example: multi-modal mixture of Gaussians in 2D

- A total number of $m = 6$ intermediate distributions in the annealing scheme



- Based on 50 independent runs of the algorithm

Example: Feed-Forward Neural Network

Example: Feed-Forward Neural Network

1) What is a Feed-Forward Neural Network?

Example: Feed-Forward Neural Network

1) What is a Feed-Forward Neural Network?

Definition

Let

$$Y : \mathbb{R}^p \to \mathbb{R}, \qquad Y(X) = \sum_{j=1}^{M} \alpha_j \Psi(X\beta_j)$$

where $\alpha_j \in \mathbb{R}$, $\beta_j \in \mathbb{R}^p$, and Ψ is a sigmoidal function (e.g. $\Psi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$).
This model is called a Feed-Forward Neural Network with

- activation function Ψ
- p input units $(X_1, \ldots X_p) = X$
- one hidden layer with M hidden units
- one output unit Y

The parameters α_j and β_j are called the connection weights.

Example: Feed-Forward Neural Network

2) Why Feed-Forward Neural Networks are important?

Example: Feed-Forward Neural Network

2) Why Feed-Forward Neural Networks are important?

Universal Approximation Property:

A FFNN with sufficient number of hidden units and properly adjusted connection weights can approximate most functions arbitrarily well.

Example: Feed-Forward Neural Network

2) Why Feed-Forward Neural Networks are important?

Universal Approximation Property:

A FFNN with sufficient number of hidden units and properly adjusted connection weights can approximate most functions arbitrarily well.

Theorem (Cybenko 1989, Hornik et al 1989)

Finite sums $Y(X) = \sum_{j=1}^{M} \alpha_j \Psi(X\beta_j)$ *are dense in the set of real continuous functions on the* p*-dimensional unit cube.*

Example: Feed-Forward Neural Network

2) Why Feed-Forward Neural Networks are important?

Universal Approximation Property:

A FFNN with sufficient number of hidden units and properly adjusted connection weights can approximate most functions arbitrarily well.

Theorem (Cybenko 1989, Hornik et al 1989)

Finite sums $Y(X) = \sum_{j=1}^{M} \alpha_j \Psi(X\beta_j)$ *are dense in the set of real continuous functions on the* p*-dimensional unit cube.*

$$Y = f(X) + \varepsilon \qquad \rightsquigarrow \qquad \mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

Example: Feed-Forward Neural Network

2) Why Feed-Forward Neural Networks are important?

Universal Approximation Property:

A FFNN with sufficient number of hidden units and properly adjusted connection weights can approximate most functions arbitrarily well.

Theorem (Cybenko 1989, Hornik et al 1989)

Finite sums $Y(X) = \sum_{j=1}^{M} \alpha_j \Psi(X\beta_j)$ *are dense in the set of real continuous functions on the* p*-dimensional unit cube.*

$$Y = f(X) + \varepsilon \qquad \rightsquigarrow \qquad \mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

$$f(X) \approx \hat{f}(X) = \sum_{j=1}^{M} \alpha_j \Psi(X\beta_j)$$

Example: Feed-Forward Neural Network

- Model:

$$Y_i = \sum_{j=1}^{M} \alpha_j \Psi(X_i \beta_j) + \varepsilon_i$$

  - $M = 2$, $p = 2$,
  - $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

Example: Feed-Forward Neural Network

- Model:

$$Y_i = \sum_{j=1}^{M} \alpha_j \Psi(X_i \beta_j) + \varepsilon_i$$

  - $M = 2$, $p = 2$,
  - $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

- Sample Space

$$x = (\alpha, \beta, \log \sigma^{-2}) \in \mathcal{X} = \mathbb{R}^7$$

Example: Feed-Forward Neural Network

- Model:

$$Y_i = \sum_{j=1}^{M} \alpha_j \Psi(X_i \beta_j) + \varepsilon_i$$

  - $M = 2$, $p = 2$,
  - $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

- Sample Space

$$x = (\alpha, \beta, \log \sigma^{-2}) \in \mathcal{X} = \mathbb{R}^7$$

- Prior Distributions
  - $\alpha_j \sim \mathcal{N}(0, \sigma_0^2)$
  - $\beta_j \sim \mathcal{N}(0, \sigma_0^2 \mathbb{I}_2)$
  - $\log \sigma^{-2} \sim \mathcal{N}(0, \sigma_0^2)$
  - $\sigma_0 = 5$

Example: Feed-Forward Neural Network

- Model:

$$Y_i = \sum_{j=1}^{M} \alpha_j \Psi(X_i \beta_j) + \varepsilon_i$$

  - $M = 2$, $p = 2$,
  - $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

- Sample Space

$$x = (\alpha, \beta, \log \sigma^{-2}) \in \mathcal{X} = \mathbb{R}^7$$

- Prior Distributions
  - $\alpha_j \sim \mathcal{N}(0, \sigma_0^2)$
  - $\beta_j \sim \mathcal{N}(0, \sigma_0^2 \mathbb{I}_2)$
  - $\log \sigma^{-2} \sim \mathcal{N}(0, \sigma_0^2)$
  - $\sigma_0 = 5$

- "Training" Data

$$\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

  - $\alpha_1 = 5$, $\alpha_2 = -5$
  - $\beta_1 = (5, -1)$, $\beta_2 = (2, -3)$
  - $\sigma = 0.1$
  - $X_i = (1, i/10)$, for $i = 1, \ldots, n$
  - $n = 100$

Example: Feed-Forward Neural Network

- Model:

$$Y_i = \sum_{j=1}^{M} \alpha_j \Psi(X_i \beta_j) + \varepsilon_i$$

  - $M = 2$, $p = 2$,
  - $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

- Sample Space

$$x = (\alpha, \beta, \log \sigma^{-2}) \in \mathcal{X} = \mathbb{R}^7$$

- Prior Distributions
  - $\alpha_j \sim \mathcal{N}(0, \sigma_0^2)$
  - $\beta_j \sim \mathcal{N}(0, \sigma_0^2 \mathbb{I}_2)$
  - $\log \sigma^{-2} \sim \mathcal{N}(0, \sigma_0^2)$
  - $\sigma_0 = 5$

- "Training" Data

$$\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

  - $\alpha_1 = 5$, $\alpha_2 = -5$
  - $\beta_1 = (5, -1)$, $\beta_2 = (2, -3)$
  - $\sigma = 0.1$
  - $X_i = (1, i/10)$, for $i = 1, \ldots, n$
  - $n = 100$

- Likelihood Function

$$L(x) = \prod_{i=1}^{n} \mathcal{N}_{(0,\sigma^2)} \left(Y_i - \sum_{j=1}^{M} \alpha_j \Psi(X_i, \beta_j) \right)$$

Example: Feed-Forward Neural Network

- Model:

$$Y_i = \sum_{j=1}^{M} \alpha_j \Psi(X_i \beta_j) + \varepsilon_i$$

  - $M = 2$, $p = 2$,
  - $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

- Sample Space

$$x = (\alpha, \beta, \log \sigma^{-2}) \in \mathcal{X} = \mathbb{R}^7$$

- Prior Distributions
  - $\alpha_j \sim \mathcal{N}(0, \sigma_0^2)$
  - $\beta_j \sim \mathcal{N}(0, \sigma_0^2 \mathbb{I}_2)$
  - $\log \sigma^{-2} \sim \mathcal{N}(0, \sigma_0^2)$
  - $\sigma_0 = 5$

- "Training" Data

$$\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

  - $\alpha_1 = 5$, $\alpha_2 = -5$
  - $\beta_1 = (5, -1)$, $\beta_2 = (2, -3)$
  - $\sigma = 0.1$
  - $X_i = (1, i/10)$, for $i = 1, \ldots, n$
  - $n = 100$

- Likelihood Function

$$L(x) = \prod_{i=1}^{n} \mathcal{N}_{(0,\sigma^2)} \left(Y_i - \sum_{j=1}^{M} \alpha_j \Psi(X_i, \beta_j) \right)$$

- Point Prediction $X \rightsquigarrow Y = f(X)$

$$\mathbb{E}_\pi[Y|X] = \int_{\mathcal{X}} \hat{f}(X, x) \pi(x) dx$$

Example: Feed-Forward Neural Network

$$\mathbb{E}_\pi[Y|X] = \int_{\mathcal{X}} \hat{f}(X,x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \hat{f}(X, x_m^{(i)}), \qquad x_m^{(i)} \sim \pi_m = \pi$$

Example: Feed-Forward Neural Network

$$\mathbb{E}_\pi[Y|X] = \int_{\mathcal{X}} \hat{f}(X,x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \hat{f}(X, x_m^{(i)}), \qquad x_m^{(i)} \sim \pi_m = \pi$$

- AIMS parameters:
  - $\gamma = 1/2$
  - $N = 3 \times 10^3$ per level
  - $q_j(x,y) = \mathcal{N}_{(x,c^2\mathbb{I}_7)}(y)$
  - $c = 0.5$

Example: Feed-Forward Neural Network

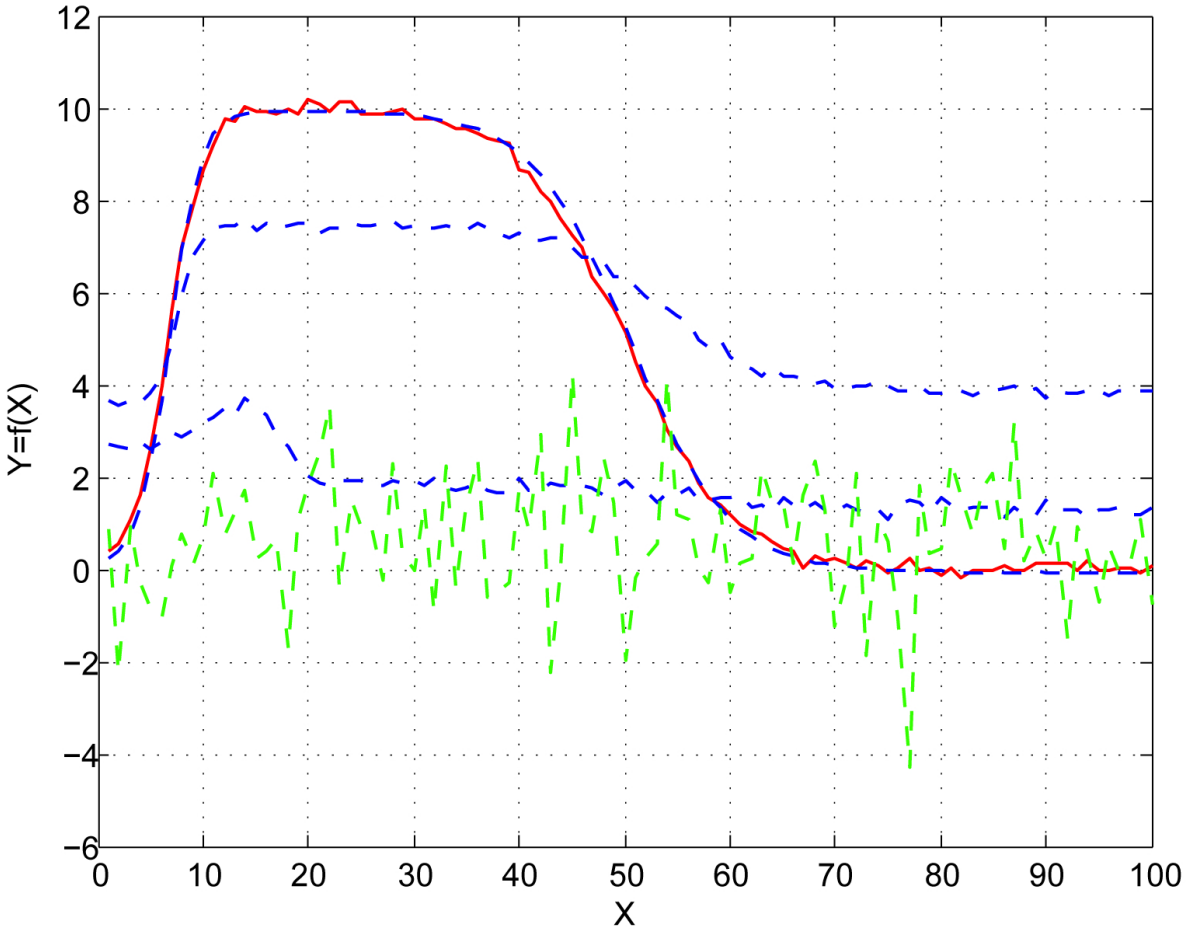
$$\mathbb{E}_\pi[Y|X] = \int_{\mathcal{X}} \hat{f}(X,x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \hat{f}(X, x_m^{(i)}), \qquad x_m^{(i)} \sim \pi_m = \pi$$

- AIMS parameters:
  - $\gamma = 1/2$
  - $N = 3 \times 10^3$ per level
  - $q_j(x,y) = \mathcal{N}_{(x,c^2\mathbb{I}_7)}(y)$
  - $c = 0.5$
- Total number of PDF's
  - $m = 10$

Example: Feed-Forward Neural Network

$$\mathbb{E}_\pi[Y|X] = \int_{\mathcal{X}} \hat{f}(X,x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \hat{f}(X, x_m^{(i)}), \qquad x_m^{(i)} \sim \pi_m = \pi$$

- AIMS parameters:
  - $\gamma = 1/2$
  - $N = 3 \times 10^3$ per level
  - $q_j(x,y) = \mathcal{N}_{(x,c^2\mathbb{I}_7)}(y)$
  - $c = 0.5$
- Total number of PDF's
  - $m = 10$
- Red: $Y = f(X)$, $(X,Y) \in \mathcal{D}$
- Green: $\frac{1}{N}\sum_{i=1}^{N} \hat{f}(X, x_0^{(i)})$
- Blue: $\frac{1}{N}\sum_{i=1}^{N} \hat{f}(X, x_j^{(i)})$



Summary

- Asymptotically Independent Markov Sampling
  - Importance sampling
  - MCMC
  - Annealing

Summary

- Asymptotically Independent Markov Sampling
  - Importance sampling
  - MCMC
  - Annealing
- Key idea:
  1. use importance sampling with π_{j-1} as the instrumental density to construct an approximation $\hat{\pi}_j \approx \pi_j$
  2. employ $\hat{\pi}_j$ as the independent proposal distribution for sampling from π_j by the independent Metropolis-Hastings algorithm

Summary

- Asymptotically Independent Markov Sampling
  - Importance sampling
  - MCMC
  - Annealing
- Key idea:
  1. use importance sampling with π_{j-1} as the instrumental density to construct an approximation $\hat{\pi}_j \approx \pi_j$
  2. employ $\hat{\pi}_j$ as the independent proposal distribution for sampling from π_j by the independent Metropolis-Hastings algorithm
- Ergodic properties:
  - under certain conditions, AIMS produces a uniformly ergodic Markov chain

Summary

- Asymptotically Independent Markov Sampling
  - Importance sampling
  - MCMC
  - Annealing
- Key idea:
  1. use importance sampling with π_{j-1} as the instrumental density to construct an approximation $\hat{\pi}_j \approx \pi_j$
  2. employ $\hat{\pi}_j$ as the independent proposal distribution for sampling from π_j by the independent Metropolis-Hastings algorithm
- Ergodic properties:
  - under certain conditions, AIMS produces a uniformly ergodic Markov chain
- Examples:
  - Multi-modal mixture of Gaussians in 2D
  - Feed-Forward Neural Network

Thank You for attention!