Theoretical Guarantees for Markov Chain Monte Carlo (MCMC) Algorithms

Raaz Dwivedi
Advised by Prof Martin Wainwright and Prof Bin Yu
Department of EECS
Random Sampling

• We consider the problem of drawing random samples from a given density (known up-to proportionality)

$$X_1, X_2, \ldots, X_m \sim \pi$$
Sampling: A fundamental task

\[ \mathbb{E}[g(X)] = \int g(x)\pi(x)dx \approx \frac{1}{m}\sum_{i=1}^{m}g(X_i) \]

\[ X_1, X_2, \ldots, X_m \sim \pi \]

Monte Carlo Approximations

Rare event simulations

Bayesian inference

Sampling

Integration

Optimization

\[ \min_x g(x) \iff \text{sample from } e^{-g(x)/T} \]

Zeroth order optimization

Escaping saddle points

Simulated annealing
Starting point: The reverse direction!  
From optimization to sampling

- Find the global minimum (or a stationary point)
  \[
  \min_{x \in \mathbb{R}^d} f(x)
  \]

- Gradient descent:
  \[
  x_{k+1} = x_k - h \nabla f(x_k)
  \]

- Stochastic Gradient Algorithm:
  \[
  X_{k+1} = X_k - h \nabla f(X_k) + h \xi_{k+1}
  \]

- Sampling: draw samples from the density
  \[
  \pi(x) \propto e^{-f(x)}
  \]

- Unadjusted Langevin algorithm (ULA):
  \[
  X_{k+1} = X_k - h \nabla f(X_k) + \sqrt{2h} \xi_{k+1}
  \]

  \[\xi_k \overset{i.i.d.}{\sim} \mathcal{N}(0, I_d \times d)\]

Starting point: The reverse direction!
From optimization to sampling

Interior point method for linear programming

Sampling from polytopes

[Dikin 1967, Nemivroski 1990]

[Kannan and Narayanan 2012]
Motivation for current work:
Better understanding of sampling for continuous spaces

- Metropolis Hastings Algorithms [1953, 1970] literature rich with numerous algorithms
- Good understanding for sampling on discrete state space in literature
- Theoretical understanding for sampling from continuous spaces: an active area of research
- Explicit theoretical guarantees gain us
  - Provably correct benchmark for comparison, sometime further insight into the pros and cons of the algorithm,
  - Breadcrumbs for designing better algorithms
Today’s talk:

- Optimization subject to linear constraints
- Sampling from Polytopes
- Convex Optimization
- Log-Concave Sampling
Part I: Uniform Sampling on Polytopes

Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

\[
\mathcal{X} = \left\{ x \in \mathbb{R}^d \mid Ax \leq b \right\}
\]

- \( n \) linear constraints
- \( d \) dimensions
- \( n > d \)
- \( A \) and \( b \) are known

\( \text{tetrahedron} \)
\( n=4 \)

\( \text{cube} \)
\( n=6 \)

\( \text{dodecahedron} \)
\( n=12 \)
Part I: Uniform Sampling on Polytopes
Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

\[ \mathcal{X} = \left\{ x \in \mathbb{R}^d \mid Ax \leq b \right\} \]

- Applications in
  - Statistical physics: Hard disk simulations
  - Sampling contingency tables
  - Mixed integer convex programming

- \( n \) linear constraints
- \( d \) dimensions
- \( n > d \)
- \( A \) and \( b \) are known
Uniform sampling: Existing methods

• Sampling on convex sets:
  • **Ball Walk** [Lovász and Simonovits 1990, 1992, 1993]

• Sampling on polytopes:
  • **Dikin Walk** [Kannan and Narayanan 2012, Narayanan 2015, Sachdeva and Vishnoi 2016]
  • **Geodesic Walk** [Lee and Vempala 2016], **Riemannian Hamiltonian Monte Carlo** [Lee and Vempala 2017]
Ball Walk [Lovász and Simonovits 1990]

- Propose a uniform point in a ball around $x$
- Reject if outside the polytope, else move to it
- In case of rejection, define next state as $x$

$$z \sim \text{Unif} \left[ \mathbb{B} \left( x, \frac{c}{\sqrt{d}} \right) \right]$$
Ball walk mixes slowly for sharp sets

- Many rejections near sharp corners
Ball walk mixes slowly for sharp sets

- Mixing time depends on conditioning of the set

\[ \left\| P(x_k) - \pi \right\|_{TV} \leq \delta \]

- Number of steps \( k \geq \mathcal{O} \left( \frac{d^2}{\delta^2} \frac{R_{out}^2}{R_{in}^2} \right) \)

- Per step cost = \( \mathcal{O} (nd) \)
Ball walk mixes slowly for sharp sets

- Mixing time depends on conditioning of the set

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

- Number of steps \( k \geq \mathcal{O} \left( \frac{d^2}{\delta^2} \frac{R_{out}^2}{R_{in}^2} \right) \)

- Per step cost = \( \mathcal{O} (nd) \)

Conditioning ratio:
Unknown
Can be exponential in d
Improving Ball Walk: Adaptive ellipsoids?
Dikin Walk [Kannan and Narayanan 2012]

• Based on log barrier for polytope used in interior point methods [Dikin 1967, Nemirovski 1990]
Dikin Walk [Kannan and Narayanan 2012]

- Propose $z \sim \mathcal{N} \left( x, \frac{1}{d} D_x^{-1} \right)$

- The inverse covariance defined by the Hessian of the log barrier

$$D_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$A = \begin{bmatrix} -a_1^\top \\ -a_2^\top \\ \vdots \\ -a_n^\top \end{bmatrix}$$
Dikin Walk [Kannan and Narayanan 2012]

- Propose \( z \sim \mathcal{N}(x, \frac{1}{d} D^{-1}_x) \)

- Reject \( z \) if it is outside the set

- Otherwise, accept \( z \) with probability

\[
P(\text{accept } z) = \min \left\{ 1, \frac{P(z \to x)}{P(x \to z)} \right\}
\]

- In case of rejection, define next state as \( x \)
Upper bounds on mixing times

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th>#steps (k)</th>
<th>Ball Walk</th>
<th>Dikin Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{d^2}{\delta^2} \frac{R_{out}^2}{R_{in}^2} )</td>
<td>( nd \log \frac{1}{\delta} )</td>
<td>( n = # \text{linear constraints} )</td>
</tr>
<tr>
<td>( n &gt; d )</td>
<td></td>
<td>( d = # \text{dimensions} )</td>
</tr>
</tbody>
</table>

| cost/step | | n = \#linear constraints |
|-----------| | \( \delta = \text{accuracy} \) |
| \( nd \) | | \( \frac{R_{out}}{R_{in}} = \text{conditioning} \) |
Upper bounds on mixing times

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th>#steps (k)</th>
<th>Ball Walk</th>
<th>Dikin Walk</th>
<th>?</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{d^2}{\delta^2} \frac{R_{out}^2}{R_{in}^2} )</td>
<td>( nd \log \frac{1}{\delta} )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cost/step</th>
<th>Ball Walk</th>
<th>Dikin Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>( nd )</td>
<td>( nd^2 )</td>
<td></td>
</tr>
</tbody>
</table>

What if \( n \gg d \)?
A closer look at Dikin walk: Proposals shrink with # constraints

Square, 4 constraints

Square, overparameterized

[Similar argument holds even when the set is not overparameterized.]
How to improve the Dikin walk?:
Even better ellipsoids?

Put weights on constraints

\[ D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]
\[ \nabla_x = \sum_{i=1}^{n} w_i(x) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

Hessians of weighted barriers in optimization
Our work: 
Exploiting improved barriers for sampling

[Dikin Proposal]

\[ z \sim \mathcal{N} \left( x, \frac{1}{d} D_x^{-1} \right) \]

\[ D_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

[Chen, D., Wainwright and Yu 2017]

[Vaidya Proposal]

\[ z \sim \mathcal{N} \left( x, \frac{1}{\sqrt{nd}} V_x^{-1} \right) \]

\[ V_x \propto \sum_{i=1}^{n} \left( \sigma_{x,i} + \frac{d}{n} \right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

\[ \sigma_{x,i} = \frac{a_i^\top D_x^{-1} a_i}{(b_i - a_i^\top x)^2} \]
Our work: Exploiting improved barriers for sampling

[Dikin Proposal]

\[ z \sim \mathcal{N}\left(x, \frac{1}{d} D_x^{-1}\right) \]

\[ D_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

[Vaidya Proposal]

\[ z \sim \mathcal{N}\left(x, \frac{1}{\sqrt{nd}} V_x^{-1}\right) \]

\[ V_x \propto \sum_{i=1}^{n} \left(\sigma_{x,i} + \frac{d}{n}\right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

\[ \sigma_{x,i} = \frac{a_i^\top D_x^{-1} a_i}{(b_i - a_i^\top x)^2} \]

Inspiration from Optimization:

Log Barrier Method
[Dikin 1967, Nemirovski 1990]

Volumetric Barrier Method
[Vaidya 1993]
Our work: Exploiting improved barriers for sampling

[Kannan and Narayanan 2012]

**Dikin Proposal**

\[ z \sim \mathcal{N}\left( x, \frac{1}{d} \mathbf{D}_x^{-1} \right) \]

\[ \mathbf{D}_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

Unit weight, sums to \( n \)

[Chen, D., Wainwright and Yu 2017]

**Vaidya Proposal**

\[ z \sim \mathcal{N}\left( x, \frac{1}{\sqrt{nd}} \mathbf{V}_x^{-1} \right) \]

\[ \mathbf{V}_x \propto \sum_{i=1}^{n} \left( \sigma_{x,i} + \frac{d}{n} \right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

\[ \sigma_{x,i} = \frac{a_i^\top \mathbf{D}_x^{-1} a_i}{(b_i - a_i^\top x)^2} \]

[0, 1] valued, sums to \( d \)
Vaidya vs Dikin proposals

Square, **4 constraints**

Dikin
Vaidya

Square, **overparameterized**

Dikin
Vaidya
### Upper bounds: Vaidya walk mixes in fewer steps!

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th></th>
<th>Ball Walk</th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#Steps</strong></td>
<td>( \frac{d^2}{\delta^2} \frac{R_{\text{max}}}{R_{\text{min}}} )</td>
<td>( nd \log \frac{1}{\delta} )</td>
<td>( n^{0.5} d^{1.5} \log \frac{1}{\delta} )</td>
</tr>
<tr>
<td><strong>Per Step Cost</strong></td>
<td>( nd )</td>
<td>( nd^2 )</td>
<td>( nd^2 )</td>
</tr>
</tbody>
</table>

- Similar cost/step as Dikin walk
- \( n \) constraints
- \( d \) dimensions
- \( n > d \)
Upper bounds: Vaidya walk mixes in fewer steps!

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th>#Steps</th>
<th>Ball Walk</th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
<th>What if ( n \gg d )?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \frac{d^2}{\delta^2} ) ( \frac{R_{max}^2}{R_{min}^2} )</td>
<td>( nd \log \frac{1}{\delta} )</td>
<td>( n^{0.5} d^{1.5} \log \frac{1}{\delta} )</td>
<td>similar cost/step as Dikin walk</td>
</tr>
<tr>
<td>Per Step Cost</td>
<td>( nd )</td>
<td>( nd^2 )</td>
<td>( nd^2 )</td>
<td></td>
</tr>
</tbody>
</table>

n constraints d dimensions \( n > d \)
Simulation: Dikin Walk vs Vaidya Walk

#dimensions = 2
k = #iterations
#experiments = 200

initial

$\begin{array}{c}
\text{initial} \\
\text{$k = 0$} \\
\end{array}$

target

$\begin{array}{c}
\text{target} \\
\text{$k = \infty$} \\
\end{array}$
Small #constraints: No Winner!

#constraints = 4

k = #iterations

#experiments = 200

Dikin Walk

$k=10$

$k=100$

$k=500$

$k=1000$

Vaidya Walk

$k=10$

$k=100$

$k=500$

$k=1000$
What if $n \gg d$? Vaidya walk wins!

- #constraints = 2048
- $k = \#\text{iterations}$
- #experiments = 200

$k=10$ | $k=100$ | $k=500$ | $k=1000$
---|---|---|---
Dikin Walk | ![Dikin Walk $k=10$](image1) | ![Dikin Walk $k=100$](image2) | ![Dikin Walk $k=500$](image3) | ![Dikin Walk $k=1000$](image4)
Vaidya Walk | ![Vaidya Walk $k=10$](image5) | ![Vaidya Walk $k=100$](image6) | ![Vaidya Walk $k=500$](image7) | ![Vaidya Walk $k=1000$](image8)
Scaling with #constraints

Approx. Mixing Time

$\alpha n^{0.9}$

$\alpha n^{0.45}$
Can we improve further?

**Dikin Proposal**

\[ z \sim \mathcal{N}\left( x, \frac{1}{d} D_x^{-1} \right) \]

\[ D_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

**Vaidya Proposal**

\[ z \sim \mathcal{N}\left( x, \frac{1}{\sqrt{nd}} \nu_x^{-1} \right) \]

\[ \nu_x \propto \sum_{i=1}^{n} \left( \sigma_{x,i} + \frac{d}{n} \right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

\[ \sigma_{x,i} = \frac{a_i^\top D_x^{-1} a_i}{(b_i - a_i^\top x)^2} \]

**Inspiration from Optimization:**

- Log Barrier Method
  - [Dikin 1967, Nemirovski 1990]
- Volumetric Barrier Method
  - [Vaidya 1993]
Yes.. via the John Walk!

**Dikin Proposal**

\[ z \sim \mathcal{N}\left(x, \frac{1}{d} D_x^{-1}\right) \]

\[ D_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

**Vaidya Proposal**

\[ z \sim \mathcal{N}\left(x, \frac{1}{\sqrt{nd}} V_x^{-1}\right) \]

\[ V_x \propto \sum_{i=1}^{n} \left( \sigma_{x,i} + \frac{d}{n} \right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

\[ \sigma_{x,i} = \frac{a_i^\top D_x^{-1} a_i}{(b_i - a_i^\top x)^2} \]

**John Proposal**

\[ z \sim \mathcal{N}\left(x, \frac{1}{d^{1.5}} J_x^{-1}\right) \]

\[ J_x \propto \sum_{i=1}^{n} J_{x,i} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

\[ J_{x,i} = \text{convex program} \]

**Inspiration from Optimization:**

- **Log Barrier Method**
  - [Dikin 1967, Nemirovski 1990]
- **Volumetric Barrier Method**
  - [Vaidya 1993]
- **John's Ellipsoidal Algorithm**
  - [John 1948, Lee and Sidford 2015]
John walk is “faster” for large #constraints (n)

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th></th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
<th>John Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Steps</td>
<td>( nd \log \frac{1}{\delta} )</td>
<td>( n^{0.5} d^{1.5} \log \frac{1}{\delta} )</td>
<td>( d^{2.5} \log^4 \frac{n}{d} \log \frac{1}{\delta} )</td>
</tr>
<tr>
<td>Per Step Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

n = #constraints  
d = #dimensions  
n > d
John walk is “faster” for large #constraints (n)

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th></th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
<th>John Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Steps</td>
<td>( nd \log \frac{1}{\delta} )</td>
<td>( n^{0.5} d^{1.5} \log \frac{1}{\delta} )</td>
<td>( d^{2.5} \log^4 \frac{n}{d} \log \frac{1}{\delta} )</td>
</tr>
<tr>
<td>Per Step Cost</td>
<td>( nd^2 )</td>
<td>( nd^2 )</td>
<td>( nd^2 \log^2 n )</td>
</tr>
</tbody>
</table>
John walk is “faster” for large #constraints (n)

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th>#Steps</th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
<th>John Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Steps</td>
<td>(nd \log \frac{1}{\delta})</td>
<td>(n^{0.5}d^{1.5} \log \frac{1}{\delta})</td>
<td>(d^{2.5} \log^4 \frac{n}{d} \log \frac{1}{\delta})</td>
</tr>
<tr>
<td>Per Step Cost</td>
<td>(nd^2)</td>
<td>(nd^2)</td>
<td>(nd^2 \log^2 n)</td>
</tr>
</tbody>
</table>

What if \(n \gg d\)?
Conjecture: Faster mixing for John walk

\[ \|P(x_k) - \pi\|_{TV} \leq \delta \]

<table>
<thead>
<tr>
<th></th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
<th>John Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Steps</td>
<td>( nd \log \frac{1}{\delta} )</td>
<td>( n^{0.5}d^{1.5} \log \frac{1}{\delta} )</td>
<td>( d^2 \log^c \frac{n}{d} \log \frac{1}{\delta} )</td>
</tr>
<tr>
<td>Per Step Cost</td>
<td>( nd^2 )</td>
<td>( nd^2 )</td>
<td>( nd^2 \log^2 n )</td>
</tr>
</tbody>
</table>
Proof Outline

Transition distributions

\[ \mathcal{T}_x \]

\[ \mathcal{T}_y \]
Proof Outline

Transition distributions

$\mathcal{T}_x$ $\mathcal{T}_y$

$\rho$

$x$ $y$

$\Delta$
Proof Outline

Transition distributions

Isoperimetry + Conductance bounds for spectral gap

\[ \text{spectral gap} \geq 1 - \frac{\rho^2 \Delta^2}{2} \]
Proof Outline

Transition distributions

Isoperimetry + Conductance bounds for spectral gap

spectral gap \geq 1 - \frac{\rho^2 \Delta^2}{2}

\| P(x_k) - \pi \|_{TV} \leq \delta \text{ for } k \geq O\left(\frac{\log(1/\delta)}{\Delta^2 \rho^2}\right)
Proof Outline

$\rho = \frac{1}{2}$

$\|\mathcal{T}_x - \mathcal{T}_y\|_{TV} \leq \frac{1}{2}$ whenever $d(x, y) \leq \Delta$

$\|\mathcal{T}_x - \mathcal{T}_y\|_{TV} \leq \|\mathcal{T}_x - \mathcal{P}_x\|_{TV} + \|\mathcal{T}_y - \mathcal{P}_y\|_{TV}$

$\quad + \|\mathcal{P}_x - \mathcal{P}_y\|_{TV}$
Proof Outline

Transition distributions

\[ \rho = \frac{1}{2} \]

\[ \| \mathcal{T}_x - \mathcal{T}_y \|_{TV} \leq \| \mathcal{T}_x - \mathcal{P}_x \|_{TV} + \| \mathcal{T}_y - \mathcal{P}_y \|_{TV} + \| \mathcal{P}_x - \mathcal{P}_y \|_{TV} \]

Difference in proposal and transition distribution due to accept-reject step

Difference in proposal distributions at two points
Easy part: Analyzing difference in the proposal distributions

\[ \mathcal{P}_x = \mathcal{N} \left( x, \frac{c}{\sqrt{nd}} \mathcal{V}_x^{-1} \right) \]

\[ \mathcal{V}_x = \sum_{i=1}^{n} \left( \sigma_{x,i} + \frac{d}{n} \right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

\[ \| \mathcal{P}_x - \mathcal{P}_y \|_{TV} \text{ is small, if} \]

\[ x \approx y \]

\[ \mathcal{V}_x \approx \mathcal{V}_y \]

Smoothness of weights
Hard part: Analyzing the accept-reject step

Difference caused by accept-reject step at each point

\[ \| \mathcal{T}_x - \mathcal{P}_x \|_{TV} \leq 2 \mathbb{P}(z \notin \mathcal{X}) + \mathbb{E} \left[ \min \left\{ 1, \frac{P(z \to x)}{P(x \to z)} \right\} \right] \]
Hard part: Analyzing the accept-reject step

Difference caused by accept-reject step at each point

\[ \| T_x - P_x \|_{TV} \leq 2P(z \notin X) + \mathbb{E} \left[ \min \left\{ 1, \frac{P(z \rightarrow x)}{P(x \rightarrow z)} \right\} \right] \]
Hard part: Analyzing the accept-reject step

Difference caused by accept-reject step at each point

\[ \| \mathcal{T}_x - \mathcal{P}_x \|_{TV} \leq 2 \mathbb{P}(z \notin \mathcal{X}) + \mathbb{E} \left[ \min \left\{ 1, \frac{P(z \rightarrow x)}{P(x \rightarrow z)} \right\} \right] \]

Randomness in \( z \) + Smoothness of weights

Taylor Series + Gaussian polynomial tail bounds

\((z - x)^T \mathcal{V}_x (z - x) \approx (z - x)^T \mathcal{V}_x (z - x)\) 
\[ \log \det \mathcal{V}_x \approx \log \det \mathcal{V}_x \]
for random \( z \sim \mathcal{P}_x \)
Part I Summary: Sampling meets optimization

Optimization on Polytopes

- Log Barrier Method [1967, 1990s]
- Volumetric Barrier Method [1993]
- John Ellipsoidal Algorithm [1948, 2015]

Sampling on Polytopes

- Dikin Walk [2012]
- Vaidya Walk [2017]
- John Walk [2017]

Fast MCMC algorithms on polytopes
https://arxiv.org/abs/1710.08165
Future Directions

- Improving dependency on $d$
  [Lee and Vempala 2016, 2017]

- Sampling on sketched polytopes

- Non-uniform sampling
  [Rakhlin et al. 2015, Bubeck et al. 2015]
Part II: Log-Concave Sampling

Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

\[ \pi(x) \propto e^{-f(x)} \quad \text{where } f : \mathbb{R}^d \to \mathbb{R} \text{ is convex} \]
Part II: Log-Concave Sampling

Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

\[ \pi(x) \propto e^{-f(x)} \quad \text{where } f: \mathbb{R}^d \to \mathbb{R} \text{ is convex} \]

• Examples include Gaussian distributions, Laplace distributions, exponential and logistic distributions

• Frequentist set ups: form confidence intervals around the MLE

• Bayesian inference and inverse problems: MAP and credible interval estimation

• Large scale stochastic/Bayesian optimization
From optimization to sampling

- **Optimization**: find the global minimum (or a stationary point)
  \[
  \min_{x \in \mathbb{R}^d} f(x)
  \]

- **Gradient descent**:
  \[
  x_{k+1} = x_k - h \nabla f(x_k)
  \]

- **Stochastic Gradient Algorithm**:
  \[
  X_{k+1} = X_k - h \nabla f(X_k) + h\xi_{k+1}
  \]

- **Sampling**: draw samples from the density
  \[
  \pi(x) \propto e^{-f(x)}
  \]

- **Unadjusted Langevin algorithm (ULA)**:
  \[
  X_{k+1} = X_k - h \nabla f(X_k) + \sqrt{2h}\xi_{k+1}
  \]

  \[
  \xi_k \overset{i.i.d.}{\sim} \mathcal{N}(0, I_{d \times d})
  \]

Langevin algorithms: Origins?

• Classical Langevin stochastic differential equation

\[ dX_t = -\nabla f(X_t)dt + \sqrt{2}dB_t \quad \text{where } B_t \text{ is standard Brownian motion} \]

• Under mind regularity conditions: as \( t \to \infty \), distribution of \( X_t \) converges to \( \pi(x) \propto e^{-f(x)} \)

\[ \|P(X_t) - \pi\|_{TV} \xrightarrow{t \to \infty} 0 \]

• ULA updates: forward discretization of the Langevin SDE

\[ X_{k+1} - X_k = -h\nabla f(X_k) + \sqrt{2h}\xi_{k+1} \]

(no accept-reject step)
ULA performance:
Large step size leads to large bias!

Histogram (multiple runs) upon convergence

Trace-plot for one run
ULA performance:
Small step mixes slowly!

Histogram (multiple runs) upon convergence

Trace-plot for one run
ULA: Step-size and speed/bias tradeoff

L1 distance between histograms

target accuracy

Iteration

ULa large
ULA opt
ULA small
How does one remove the asymptotic bias?

- Via the **classical** Metropolis-Hastings correction step

- Metropolis adjusted Langevin algorithm (MALA):
  1. Use ULA updates as proposals
     \[
     z = x - h\nabla f(x) + \sqrt{2h}\xi
     \]
  2. Accept \( z \) with probability
     \[
     \min \left\{ 1, \frac{e^{-f(z)}}{e^{-f(x)}} \frac{P(z \to x)}{P(x \to z)} \right\}
     \]
  3. In case of rejection, stay at \( x \)
MALA: Fast convergence with no bias

L1 distance between histograms

Iteration
Langevin algorithms: Traditional wisdom

- Rich body of work for Langevin algorithms
- ULA and MALA first suggested by Parisi in 1981 and formally introduced by Grenander & Miller in 1994
- Sufficient conditions for convergence first established by Roberts and Tweedie in 1996
Langevin algorithms: Prior work

<table>
<thead>
<tr>
<th>Type of results</th>
<th>Existing Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discretization &amp; integration errors, Ergodicity, Asymptotic convergence</td>
<td>[Talay &amp; Tubaro ‘90], [Meyn &amp; Tweedie ‘95], [Roberts &amp; Rosenthal ‘96, ‘01, ‘02]</td>
</tr>
</tbody>
</table>

- Find conditions on $\pi$ and a Lyapunov function $V$ that contracts outside a ball

$$E[V(X_1)|X_0 = x] \leq \lambda V(x) + C\|_{B(0,R)}(x), \quad \lambda < 1$$

sufficient to establish geometric ergodicity

$$\|P(x_k|x_0 = x) - \pi\|_{TV} \leq V(x)R\rho^k \quad \text{for some } \rho < 1 \text{ and } R > 0$$

For limited class of distributions, non-explicit rates, hard to track dependency on problem parameters
### Langevin algorithms: Related work

<table>
<thead>
<tr>
<th>Type of results</th>
<th>Existing Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discretization &amp; integration errors, Ergodicity, Asymptotic convergence</td>
<td>[Talay &amp; Tubaro ‘90], [Meyn &amp; Tweedie ‘95], [Roberts &amp; Rosenthal ‘96, ‘01, ‘02]</td>
</tr>
<tr>
<td>Revived interest for non-asymptotic results</td>
<td>[Bou-Rabee &amp; Hairer ‘09], [Roberts &amp; Rosenthal ‘14]</td>
</tr>
<tr>
<td>Explicit non-asymptotic bounds</td>
<td>[Dalalyan ‘15, ‘17], [Durmus &amp; Moulines ‘15, ‘16], [Cheng &amp; Bartlett ‘17]</td>
</tr>
</tbody>
</table>

Recent work uses coupling arguments for diffusions
Mixing time bounds: Strongly log-concave

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

- \( \pi(x) \propto e^{-f(x)} \)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ULA [Dalalyan 2016]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f ) is ( L )-smooth and ( m )-strongly-convex</td>
<td>( d \left( \frac{L}{m} \right)^2 \frac{1}{\delta^2} )</td>
</tr>
</tbody>
</table>
Mixing time bounds: Strongly log-concave

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

\[ \pi(x) \propto e^{-f(x)} \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ULA [Dalalyan 2016]</th>
<th>MALA [D., Chen, Wainwright, Yu 2018]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f ) is ( L )-smooth and ( m )-strongly-convex</td>
<td>( d \left( \frac{L}{m} \right)^2 \frac{1}{\delta^2} )</td>
<td>( d \left( \frac{L}{m} \right) \log \frac{1}{\delta} )</td>
</tr>
</tbody>
</table>
Mixing time bounds: Strongly log-concave

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

\[ \pi(x) \propto e^{-f(x)} \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ULA [Dalalyan 2016]</th>
<th>MALA [D., Chen, Wainwright, Yu 2018]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f ) is ( L )-smooth and ( m )-strongly-convex</td>
<td>( d \left( \frac{L}{m} \right)^2 \frac{1}{\delta^2} )</td>
<td>( d \left( \frac{L}{m} \right) \log \frac{1}{\delta} )</td>
</tr>
</tbody>
</table>

- Mixing time of MALA has
  - exponentially better dependence on accuracy \( \delta \)
  - better dependence on conditioning \( L/m \)
Mixing time bounds: Strongly and weakly log-concave

$\|P(x_k) - \pi\|_{TV} \leq \delta$

$\pi(x) \propto e^{-f(x)}$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ULA [Dalalyan 2016]</th>
<th>MALA [D., Chen, Wainwright, Yu 2018]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$ is $L$-smooth and $m$-strongly-convex</td>
<td>$d \left( \frac{L}{m} \right)^2 \frac{1}{\delta^2}$</td>
<td>$d \left( \frac{L}{m} \right) \log \frac{1}{\delta}$</td>
</tr>
<tr>
<td>$f$ is convex and $L$-smooth</td>
<td>$d^3 L^2 \frac{1}{\delta^4}$</td>
<td>$d^2 L^{1.5} \frac{1}{\delta^{1.5}}$</td>
</tr>
</tbody>
</table>
Mixing time bounds:
Strongly and weakly log-concave

\[ \| P(x_k) - \pi \|_{TV} \leq \delta \]

\( \pi(x) \propto e^{-f(x)} \)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ULA [Dalalyan 2016]</th>
<th>MALA [D., Chen, Wainwright, Yu 2018]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f ) is ( L )-smooth and ( m )-strongly-convex</td>
<td>[ d \left( \frac{L}{m} \right)^2 \frac{1}{\delta^2} ]</td>
<td>[ d \left( \frac{L}{m} \right) \log \frac{1}{\delta} ]</td>
</tr>
<tr>
<td>( f ) is convex and ( L )-smooth</td>
<td>[ d^3 L^2 \frac{1}{\delta^4} ]</td>
<td>[ d^2 L^{1.5} \frac{1}{\delta^{1.5}} ]</td>
</tr>
</tbody>
</table>

Faster!
The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region
The difference between MALA and ULA: An informal proof

• Both algorithms have a good spectral gap in a high probability region

• ULA has a biased stationary distribution

\[ \|P(x_k) - \pi\|_{TV} \leq \|P(x_k) - \pi_{ULA}\|_{TV} + \|\pi_{ULA} - \pi\|_{TV} \]
The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region

- ULA has a biased stationary distribution

\[ \| P(x_k) - \pi \|_{TV} \leq \| P(x_k) - \pi_{ULA} \|_{TV} + \| \pi_{ULA} - \pi \|_{TV} \]

\[ O(e^{-kh}) \quad \mathcal{O}(\sqrt{h}) \]

Bias
The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region

- ULA has a biased stationary distribution

\[ \| P( x_k ) - \pi \|_{TV} \leq \| P( x_k ) - \pi_{ULA} \|_{TV} + \| \pi_{ULA} - \pi \|_{TV} \]

\[ \mathcal{O}( e^{-kh} ) \leq \delta / 2 \]

\[ \mathcal{O}( \sqrt{h} ) \leq \delta / 2 \]

\[ k \geq \mathcal{O} \left( \frac{1}{h} \log \frac{1}{\delta} \right) = \mathcal{O} \left( \frac{1}{\delta^2} \right) \]
The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region.

- ULA has a biased stationary distribution.

\[
\| P(x_k) - \pi \|_{TV} \leq \| P(x_k) - \pi_{ULA} \|_{TV} + \| \pi_{ULA} - \pi \|_{TV} \\
O(e^{-kh}) \leq \delta/2 \quad O(\sqrt{h}) \leq \delta/2
\]

\[ k \geq O \left( \frac{1}{h} \log \frac{1}{\delta} \right) = O \left( \frac{1}{\delta^2} \right) \]

- MALA is unbiased: larger step size implies faster mixing.
Part II: Summary

Convex Optimization → Log-concave Sampling

Gradient Descent + noise variance $h$ → Unadjusted Langevin Algorithm + noise variance $h^2$ → Stochastic Gradient Algorithm → Unadjusted Langevin Algorithm

Unadjusted Langevin Algorithm + Accept Reject Step → Metropolis-Adjusted Langevin Algorithm

Provides exponential gain in mixing time
Future Directions

- No gradient: Metropolis random walk
  \[ O(d) \text{ slower!} \]
  [D., Chen, Wainwright, Yu 2018]

- With Hessian:
  Can we have a faster algorithm?

- Higher order methods:
  Hamiltonian Monte Carlo
  Underdamped Langevin
  [Cheng et al. 2017, Smith et al. 2018]

- Framework for lower bounds on mixing times?

- General/Mixture distributions:
  Non-log concave sampling
  (Simulated Tempering)
Summary: Connections

- Optimization
- Sampling

- Interior Point Methods
  - Randomized Interior Point Methods

- Gradient Methods
  - Langevin Algorithms
Summary: Findings

- **Faster** Interior Point Methods → **Faster** Randomized Interior Point Methods
- **Faster** Gradient Methods + accept-reject step → **Faster** Langevin Algorithms
So far...

- **Mixing times**
- **Function specific mixing times:** Estimating mean and covariance
Looking forward..

- Mixing times
- Function specific mixing times: Estimating mean and covariance
- Learning from data
Looking forward..

Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

- Learning from data

- Mixing times

- Function specific mixing times: Estimating mean and covariance
Looking forward...

Data driven manifold learning:
Low dimensional structure in deep networks

• Mixing times

• Function specific mixing times:
  Estimating mean and covariance

Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

• Learning from data
Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

Will the model generalize or not?
Choice of kernel matters!

Data driven manifold learning:
Low dimensional structure in deep networks

Learning from data

Mixing times

Function specific mixing times:
Estimating mean and covariance

Looking forward..

Distributions

Samples
Looking forward..

Data driven manifold learning: Low dimensional structure in deep networks

Will the model generalize or not? Choice of kernel matters!

Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

Samples

Learning from data

Improving sample quality:
From Monte Carlo to Quasi Monte Carlo to reduce discrepancy

• Mixing times

• Function specific mixing times: Estimating mean and covariance

Distributions
Fast MCMC algorithms on polytopes
https://arxiv.org/abs/1710.08165

Log-concave sampling: Metropolis Hastings
Algorithms are fast!
http://arxiv.org/abs/1801.02309