Fast MCMC Algorithms on Polytopes

Raaz Dwivedi, Department of EECS

Joint work with

Yuansi Chen          Martin Wainwright          Bin Yu
Random Sampling

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

\[ X_1, X_2, \ldots, X_m \sim \pi^* \]
Applications

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

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

- Probabilities of Events
- Rare Event Simulations
- Bayesian Posterior Mean
- Volume Computation (polynomial time)
Applications

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

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

- Probabilities of Events
- Rare Event Simulations
- Bayesian Posterior Mean
- Volume Computation (polynomial time)
Applications

\[
\min_{x \in \mathcal{K}} g(x)
\]

• **Zeroth order optimization**: Polynomial time algorithms based on Random Walk

• **Convex optimization**: Bertsimas and Vempala 2004, Kalai and Vempala 2006, Kannan and Narayanan 2012, Hazan et al. 2015

• **Non-convex optimization, Simulated Annealing**: Aarts and Korst 1989, Rakhlin et al. 2015
Uniform Sampling on Polytopes

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

- \( n \) linear constraints
- \( d \) dimensions
- \( n > d \)
Uniform Sampling on Polytopes

- Integration of arbitrary functions under linear constraints
- Mixed Integer Programming
- Sampling non-negative integer matrices with specified row and column sums (contingency tables)
- Connections between optimization and sampling algorithms
Goal

Given $A$ and $b$, and a starting distribution $\mu_0$,
design an MCMC algorithm

that generates a random sample from uniform distribution on

$$\mathcal{X} = \left\{ x \in \mathbb{R}^d \ \bigg| \ Ax \leq b \right\}$$

in as few steps as possible!

Convergence Rate: **Mixing time for total variation**

$$\| \mu_0 P^k - \pi^* \|_{\text{TV}} \leq \epsilon$$
Markov Chain Monte Carlo

- **Design** a Markov Chain which can converge to the desired distribution
  - Metropolis Hastings Algorithms (1950s), Gibbs Sampling (1980s)

- **Simulate** the Markov chain for **several steps to get a sample**
Markov Chain Monte Carlo

• Sampling on convex sets: **Ball Walk** (Lovász et al. 1990), **Hit-and-run** (Smith et al. 1993, Lovász 1999),

• Sampling on polytopes: **Dikin Walk** (Kannan and Hariharan 2012, Hariharan 2015, Sachdeva and Vishnoi 2016), **Geodesic Walk** (Lee and Vempala 2016)
**Ball Walk** [Lovász and Simonovits 1990]

- Propose a uniform point in a ball around $x$
  
  - reject if outside the polytope, else move to it

\[
z \sim U \left[ \mathbb{B} \left( x, \frac{c}{\sqrt{d}} \right) \right]
\]
Ball Walk [Lovász and Simonovits 1990]

• Many rejections near sharp corners
Ball Walk [Lovász and Simonovits 1990]

- Mixing time depends on *conditioning* of the set

\[
\#\text{steps} = \mathcal{O}\left(d^2 \frac{R_{\text{max}}^2}{R_{\text{min}}^2}\right)
\]

per step cost = \(n d\)

Can be exponential in \(d\)
May be a variable shape ellipsoid?
Dikin Walk [Kannan and Narayanan 2012]

- Proposal: \( z \sim \mathcal{N} \left( x, \frac{r^2}{d} D_x^{-1} \right) \)

- Another variant: \( z \sim \mathbb{U} [D_x(r)] \)

- Accept Reject:

\[
P(\text{accept } z) = \min \left\{ 1, \frac{P(z \rightarrow x)}{P(x \rightarrow z)} \right\}
\]
**Dikin Walk** [Kannan and Narayanan 2012]

- **Proposal**
  \[ z \sim \mathcal{N} \left( x, \frac{r^2}{d} D_x^{-1} \right) \]

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

  \[ A = \begin{bmatrix} a_1^\top \\ a_2^\top \\ \vdots \\ a_n^\top \end{bmatrix} \quad \mathcal{K} = \{ x \in \mathbb{R}^d | Ax \leq b \} \]

**Log Barrier Method**

(Optimization)

[Dikin 1967, Nemirovski 1990]
## Upper bounds

<table>
<thead>
<tr>
<th>#Steps</th>
<th>Ball Walk</th>
<th>Dikin Walk</th>
<th>( \frac{d^2}{R_n^2} )</th>
<th>( \frac{d^2}{R_m^2} )</th>
<th>( nd )</th>
<th>( nd^2 )</th>
</tr>
</thead>
</table>

- \( n = \#\text{constraints} \)
- \( d = \#\text{dimensions} \)
- \( n > d \)
Slow mixing of Dikin Walk

#constraints = 4

#constraints = 128
"If any two points that are $\Delta$ apart have $\rho$ overlap in their transition regions, then the chain mixes in $\mathcal{O}\left(\frac{1}{\Delta^2 \rho^2}\right)$ steps."

–Lovász’s Lemma

(Distance and overlap measured in appropriately)
“If any two points that are \( \Delta \) apart have \( \rho \) overlap in their transition regions, then the chain mixes in \( O \left( \frac{1}{\Delta^2 \rho^2} \right) \) steps.”

–Lovász’s Lemma

For any fixed overlap \( \rho \), we want far away points to have \( \rho \) overlapping regions, and hence large ellipsoids (contained within the polytope) are useful.
Improving Dikin Walk

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

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

Importance weighting of constraints
Improving Dikin Walk

Dikin Proposal

\[ z \sim \mathcal{N} \left( x, \frac{r^2}{d} D_x^{-1} \right) \]

\[ D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]
Sampling meets optimization (again!!)

[Dikin Proposal]
\[ z \sim \mathcal{N} \left( x, \frac{r^2}{d} D_x^{-1} \right) \]
\[ D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]

[Vaidya Proposal]
\[ z \sim \mathcal{N} \left( x, \frac{r^2}{\sqrt{nd}} V_x^{-1} \right) \]
\[ V_x = \sum_{i=1}^{n} \left( \sigma_{x,i} + \frac{d}{n} \right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \]
\[ \sigma_{x,i} = \frac{a_i^\top D_x^{-1} a_i}{(b_i - a_i^\top x)^2} \]

[Log Barrier Method]
[Dikin 1967, Nemirovskii 1990]

[Volumetric Barrier Method]
[Vaidya 1993]

[Kannan and Narayanan 2012]

[Chen, D., Wainwright and Yu 2017]
Vaidya Walk [Chen, D., Wainwright, Yu 2017]

#constraints = 4

#constraints = 128
# Convergence Rates

<table>
<thead>
<tr>
<th>#Steps</th>
<th>Ball Walk</th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per Step Cost</td>
<td>$d^2 \frac{R^2_{\text{max}}}{R^2_{\text{min}}}$</td>
<td>$nd$</td>
<td>$n^{0.5}d^{1.5}$</td>
</tr>
</tbody>
</table>

$n$ constraints
$d$ dimensions
$n > d$
# Convergence Rates

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

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

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

$k = 0$

$k = \infty$
Dikin Walk vs Vaidya Walk

$k = \#\text{iterations}

\#\text{constraints} = 64
\#\text{experiments} = 200

$k=10$
Dikin Walk vs Vaidya Walk

#constraints = 64  #experiments = 200  k = #iterations

$k = 10$  $k = 100$

Dikin Walk

Vaidya Walk
Dikin Walk vs Vaidya Walk

#constraints = 64
#experiments = 200
k = #iterations

$k=10$

$k=100$

$k=500$

Dikin Walk

Vaidya Walk
Small number of constraints: No Winner!

- #constraints = 64
- #experiments = 200
- $k = \#\text{iterations}$

Dikin Walk

Vaidya Walk

$k = 10$

$k = 100$

$k = 500$

$k = 1000$
Dikin Walk vs Vaidya Walk

#constraints = 2048
#experiments = 200
k = #iterations

$k=10$

Dikin Walk

Vaidya Walk
Dikin Walk vs Vaidya Walk

#constraints = 2048
#experiments = 200
k = #iterations

$k=10$

$k=100$

Dikin Walk

Vaidya Walk
Dikin Walk vs Vaidya Walk

#constraints = 2048
#experiments = 200
k = #iterations

$k=10$

Dikin Walk

$k=100$

$k=500$

Vaidya Walk
Vaidya walk wins!

#constraints = 2048
#experiments = 200

$k = \#iterations$

$k = 10$

$k = 100$

$k = 500$

$k = 1000$
Dikin Walk vs Vaidya Walk

$O(nd)$ vs $O(n^{0.5}d^{1.5})$

Approx. Mixing Time

$\propto n^{0.9}$

$\propto n^{0.45}$
Polytope approximation to Circle

#constraints = 5

#constraints = 8
#constraints = 64

**Dikin Walk**

**Vaidya Walk**
Dikin Walk

#constraints = 64

Vaidya Walk

#constraints = 2048
Can we improve further?

Dikin Proposal

\[ z \sim \mathcal{N}\left( x, \frac{r^2}{d} D^{-1}_x \right) \]

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

Vaidya Proposal

\[ z \sim \mathcal{N}\left( x, \frac{r^2}{\sqrt{nd}} V_x^{-1} \right) \]

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

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

Log Barrier Method
[Dikin 1967, Nemeirovski 1990]

Vaidya’s Volumetric Barrier Method
[Vaidya 1993]
John Walk

Dikin Proposal

\[ z \sim \mathcal{N}\left(x, \frac{r^2}{d}D_x^{-1}\right) \]

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

Vaidya Proposal

\[ z \sim \mathcal{N}\left(x, \frac{r^2}{\sqrt{nd}}V_x^{-1}\right) \]

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

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

John Proposal

\[ z \sim \mathcal{N}\left(x, \frac{r^2}{d^{1.5}}J_x^{-1}\right) \]

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

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

Log Barrier Method

[Dikin 1967, Nemirovski 1990]

Vaidya’s Volumetric Barrier Method

[Vaidya 1993]

John’s Ellipsoidal Algorithm

[Fritz John 1948, Lee and Sidford 2015]
## Mixing Times

<table>
<thead>
<tr>
<th></th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
<th>John Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#Steps</strong></td>
<td>$nd$</td>
<td>$n^{0.5}d^{1.5}$</td>
<td>$d^{2.5} \log^4 \frac{n}{d}$</td>
</tr>
<tr>
<td><strong>Per Step Cost</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Mixing Times

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

$n = \#\text{constraints}$

$d = \#\text{dimensions}$

$n > d$
**Conjecture**

<table>
<thead>
<tr>
<th></th>
<th>Dikin Walk</th>
<th>Vaidya Walk</th>
<th>John Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#Steps</strong></td>
<td>$nd$</td>
<td>$n^{0.5}d^{1.5}$</td>
<td>$d^2 \log^c_c \left( \frac{n}{d} \right)$</td>
</tr>
<tr>
<td><strong>Per Step Cost</strong></td>
<td>$nd^2$</td>
<td>$nd^2$</td>
<td>$nd^2 \log^2 n$</td>
</tr>
</tbody>
</table>
“For the John walk, the log factors are bottleneck in practice.”

– Numerical Experiments
Proof Idea

- Proof relies on Lovasz’s Lemma
- Need to establish that near by points have similar transition distributions
- Have to show that the weighted matrices are sufficiently smooth — *use of weights makes it involved*
Summary

Optimization

Log Barrier Method 1967, 1980s
Volumetric Barrier Method 1993
John Ellipsoidal Algorithm 1948, 2015

Sampling

Dikin Walk 2012
Vaidya Walk 2017
John Walk 2017

faster