Approximating $\ell_{q \to p}$ Norms of Non-Negative Matrices in Nearly-Linear Time

Etienne Objois, Adrian Vladu

Institut de Recherche en Informatique Fondamentale (IRIF)

March 10, 2026

INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

Université
Paris Cité

Spectral Norm

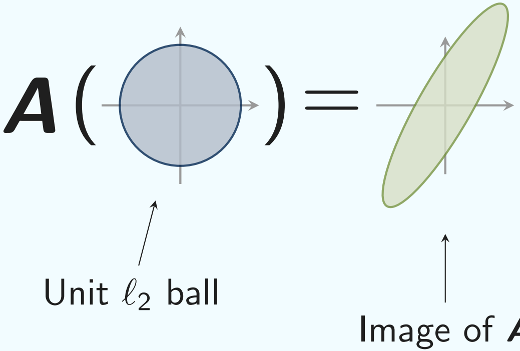
Spectral Norm

We want to approximate

$$\|\boldsymbol{A}\|_2 = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2 \leq 1} \|\boldsymbol{A}\boldsymbol{x}\|_2 = \max_{\boldsymbol{x} \neq \boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$

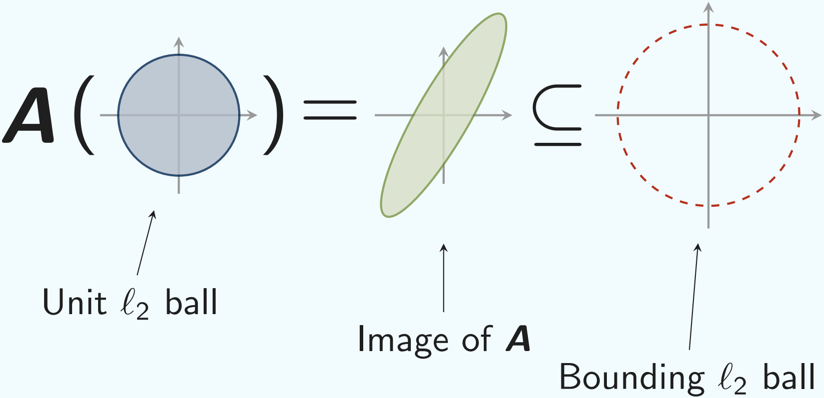
Size of a Matrix

What is the "size" of a matrix $\boldsymbol{A}$?



Size of a Matrix

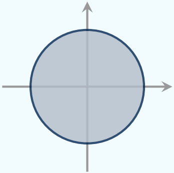
What is the "size" of a matrix $\boldsymbol{A}$?



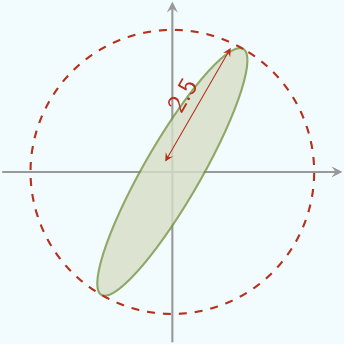
Spectral Norm: Visual Representation

Unit ℓ_2 ball

Image of $\boldsymbol{A}$



$$\longrightarrow \boldsymbol{A} = \begin{pmatrix} 1.3 & -1.2 \\ 2.2 & 0.8 \end{pmatrix} \longrightarrow$$



$$\max \|\boldsymbol{Ax}\|_2 = 2.5$$

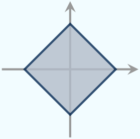
Other Unit Balls

$\|\boldsymbol{x}\|_p = (\sum_i |\boldsymbol{x}_i|^p)^{\frac{1}{p}}$

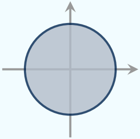
$p = 2$: euclidean norm

$p \geq 1$: norm

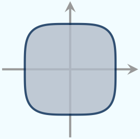
Unit ℓ_1 ball
$\{\|\boldsymbol{x}\|_1 \leq 1\}$



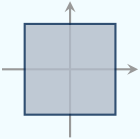
Unit ℓ_2 ball
$\{\|\boldsymbol{x}\|_2 \leq 1\}$



Unit ℓ_4 ball
$\{\|\boldsymbol{x}\|_4 \leq 1\}$

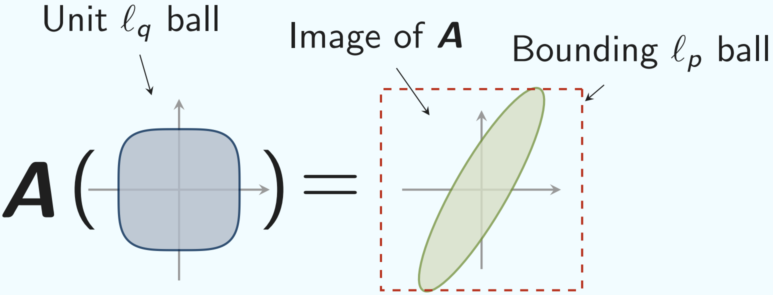


Unit ℓ_∞ ball
$\{\|\boldsymbol{x}\|_\infty \leq 1\}$



Generalizing to $\ell_{q\to p}$ Spaces

$$\|\boldsymbol{A}\|_{q\to p} = \max_{\boldsymbol{x}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_q}$$



Hardness

General Version of our Problem

For $q, p \geq 1$, and a matrix $\boldsymbol{A}$, find

$$\left\|\boldsymbol{A}\right\|_{q \to p} = \max_{\left\|\boldsymbol{x}\right\|_q = 1} \left\|\boldsymbol{A}\boldsymbol{x}\right\|_p .$$

Hardness

General Version of our Problem

For $q, p \geq 1$, and a matrix $\boldsymbol{A}$, find

$$\left\|\boldsymbol{A}\right\|_{q\to p} = \max_{\left\|\boldsymbol{x}\right\|_q = 1} \left\|\boldsymbol{A}\boldsymbol{x}\right\|_p .$$

- MaxCut: $\left\|\boldsymbol{A}\right\|_{\infty\to 1}$.

Hardness

General Version of our Problem

For $q, p \geq 1$, and a matrix $\boldsymbol{A}$, find

$$\|\boldsymbol{A}\|_{q \to p} = \max_{\|\boldsymbol{x}\|_q = 1} \|\boldsymbol{A}\boldsymbol{x}\|_p .$$

- MaxCut: $\|\boldsymbol{A}\|_{\infty \to 1}$.
- Small-Set Expansion Hypothesis (a variation of Unique Game Conjecture).

$\Longrightarrow$ the problem is **hard** under some conjectures.

Our Motivation and Assumptions

Linear Oblivious Routings

- **For a routing matrix $\boldsymbol{A}$:**
  - $\|\boldsymbol{A}\|_p$ is the **quality** of the routing.
  - **Goal:** Compute high quality $\boldsymbol{A}$.
  - **Certificate:** Compute the quality of the routing.

Our Problem

For $q = p \geq 1$, an entry wise non-negative matrix $\boldsymbol{A}$, and $\varepsilon > 0$ find $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ such that

$$\frac{\|\boldsymbol{Ax}\|_p}{\|\boldsymbol{x}\|_p} \geq (1 - \varepsilon) \cdot \|\boldsymbol{A}\|_p .$$

Our Motivation and Assumptions

Linear Oblivious Routings

- **For a routing matrix $\boldsymbol{A}$:**
  - $\|\boldsymbol{A}\|_p$ is the **quality** of the routing.
  - **Goal:** Compute high quality $\boldsymbol{A}$.
  - **Certificate:** Compute the quality of the routing.
- **Properties of a routing matrix:**
  1. $\boldsymbol{A}$ is entry wise non-negative.
  2. $q = p$.

Our Problem

For $q = p \geq 1$, an entry wise non-negative matrix $\boldsymbol{A}$, and $\varepsilon > 0$ find $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ such that

$$\frac{\|\boldsymbol{A}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p} \geq (1 - \varepsilon) \cdot \|\boldsymbol{A}\|_p .$$

Potentials: Definition for the ℓ_2 Case

$$\boldsymbol{x}_{t+1} = \frac{\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_t}{\left\|\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_t\right\|_2}, \qquad \boldsymbol{\Phi}(\boldsymbol{x}_t) = \frac{\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_t}{\boldsymbol{x}_t}$$

vector operations
are coordinate wise

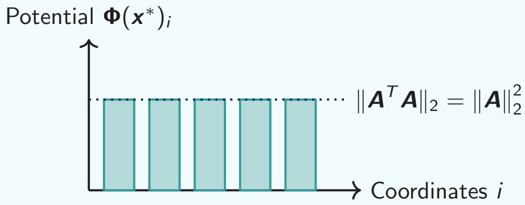
- Potential $=$ a vector of $\boldsymbol{x}$.
- $\boldsymbol{\Phi}(\boldsymbol{x}) \approx$ gradient $+$ something.

Potentials: Definition for the ℓ_2 Case

$$\boldsymbol{x}_{t+1} = \frac{\boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x}_t}{\left\| \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x}_t \right\|_2}, \qquad \boldsymbol{\Phi}(\boldsymbol{x}_t) = \frac{\boldsymbol{A}^T \boldsymbol{A} \boldsymbol{x}_t}{\boldsymbol{x}_t}$$

vector operations
are coordinate wise

- Potential = a vector of $\boldsymbol{x}$.
- $\boldsymbol{\Phi}(\boldsymbol{x}) \approx$ gradient + something.

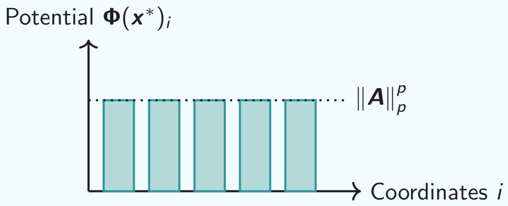


Potentials: Definition for the ℓ_p Case

$$\boldsymbol{x}_{t+1} = \frac{\boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{x}_t)^{p-1}}{\left\| \boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{x}_t)^{p-1} \right\|_p}, \qquad \boldsymbol{\Phi}(\boldsymbol{x}_t) = \frac{\boldsymbol{A}^T (\boldsymbol{A}\boldsymbol{x}_t)^{p-1}}{\boldsymbol{x}_t^{p-1}}$$

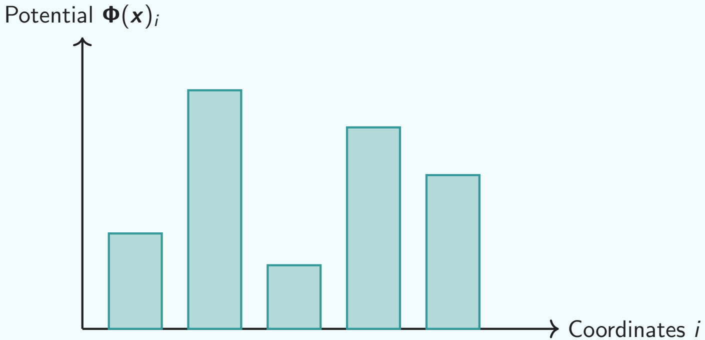
vector operations are coordinate wise

- Potential = a vector of $\boldsymbol{x}$.
- $\boldsymbol{\Phi}(\boldsymbol{x}) \approx$ gradient + something.



Potential Vector: Properties

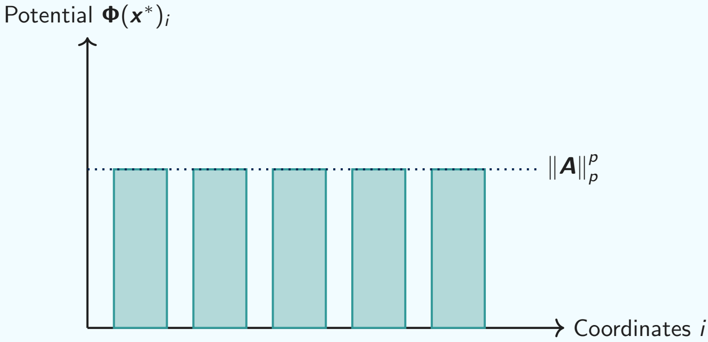
Three Properties:



Potential Vector: Properties

Three Properties:

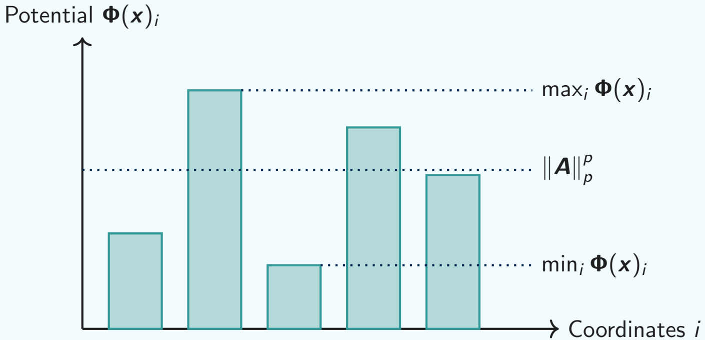
1. At optimum, $\mathbf{\Phi}(\boldsymbol{x}^*)_i = \|\boldsymbol{A}\|_p^p$ for all i.



Potential Vector: Properties

Three Properties:

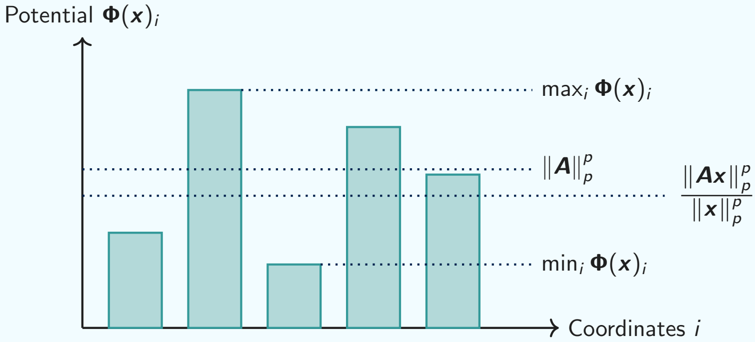
1. At optimum, $\boldsymbol{\Phi}(\boldsymbol{x}^*)_i = \|\boldsymbol{A}\|_p^p$ for all i.
2. Entries of $\boldsymbol{\Phi}(\boldsymbol{x})$ sandwich $\|\boldsymbol{A}\|_p^p$.



Potential Vector: Properties

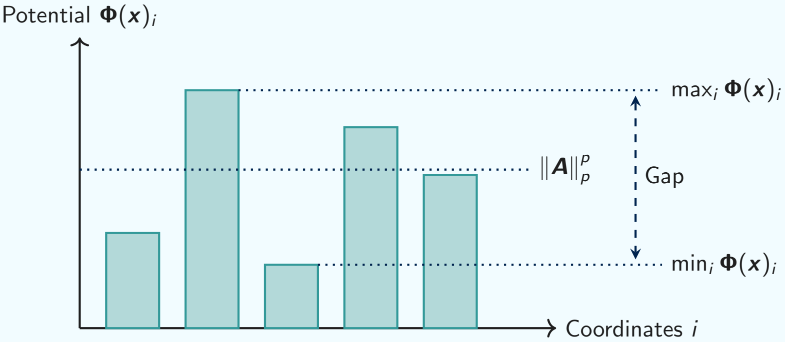
Three Properties:

1. At optimum, $\boldsymbol{\Phi}(\boldsymbol{x}^*)_i = \|\boldsymbol{A}\|_p^p$ for all i.
2. Entries of $\boldsymbol{\Phi}(\boldsymbol{x})$ sandwich $\|\boldsymbol{A}\|_p^p$.
3. Entries of $\boldsymbol{\Phi}(\boldsymbol{x})$ sandwich $\|\boldsymbol{A}\boldsymbol{x}\|_p^p \,/\, \|\boldsymbol{x}\|_p^p$.



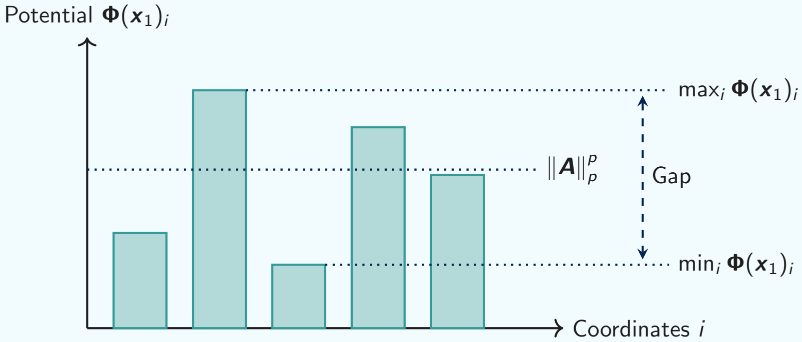
Visual Representation of Power Iteration

1 **Goal:** Find $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\max_i \boldsymbol{\Phi}(\boldsymbol{x})_i / \min_i \boldsymbol{\Phi}(\boldsymbol{x})_i \leq 1 + \varepsilon$.



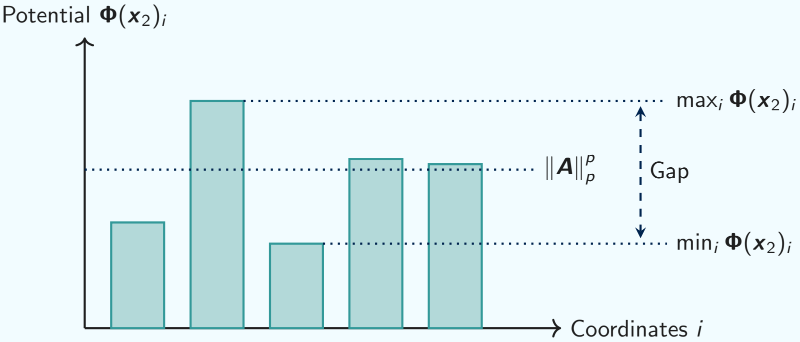
Visual Representation of Power Iteration

1. **Goal:** Find $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\max_i \boldsymbol{\Phi}(\boldsymbol{x})_i / \min_i \boldsymbol{\Phi}(\boldsymbol{x})_i \leq 1 + \varepsilon$.
2. [Boyd, '74]: With Power Iteration, this ratio decreases.



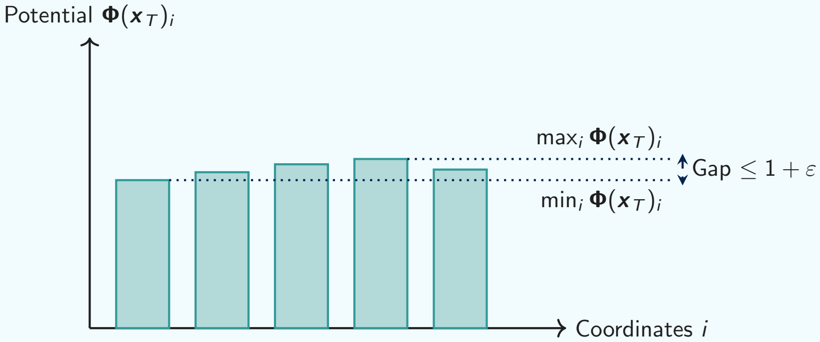
Visual Representation of Power Iteration

1. **Goal:** Find $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\max_i \boldsymbol{\Phi}(\boldsymbol{x})_i / \min_i \boldsymbol{\Phi}(\boldsymbol{x})_i \leq 1 + \varepsilon$.
2. [Boyd, '74]: With Power Iteration, this ratio decreases.



Visual Representation of Power Iteration

1. **Goal:** Find $\boldsymbol{x} \in \mathbb{R}^n_{>0}$ such that $\max_i \boldsymbol{\Phi}(\boldsymbol{x})_i / \min_i \boldsymbol{\Phi}(\boldsymbol{x})_i \leq 1 + \varepsilon$.
2. [Boyd, '74]: With Power Iteration, this ratio decreases.
3. [Bhaskara and Vijayaraghavan, 2011]: $\widetilde{\mathcal{O}}(n^3/\varepsilon)$ iterations suffice.



Power Iterations Downsides

$$\textbf{Goal: } \text{Find } \boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \text{ such that } \frac{\max_i \boldsymbol{\Phi}(\boldsymbol{x})_i}{\min_i \boldsymbol{\Phi}(\boldsymbol{x})_i} \leq 1 + \varepsilon.$$

- Power Iteration converges exponentially fast on matrices that are well-conditioned.
- Sparse matrices are **ill-conditioned**.

Power Iterations Downsides

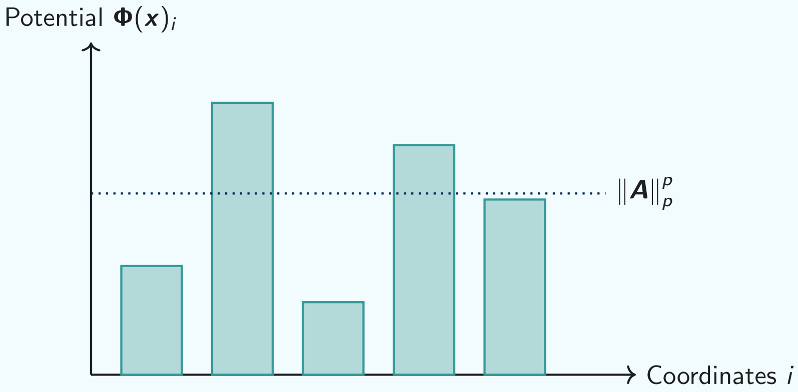
$$\textbf{Goal: } \text{Find } \boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \text{ such that } \frac{\max_i \boldsymbol{\Phi}(\boldsymbol{x})_i}{\min_i \boldsymbol{\Phi}(\boldsymbol{x})_i} \leq 1 + \varepsilon.$$

- Power Iteration converges exponentially fast on matrices that are well-conditioned.
- Sparse matrices are **ill-conditioned**.

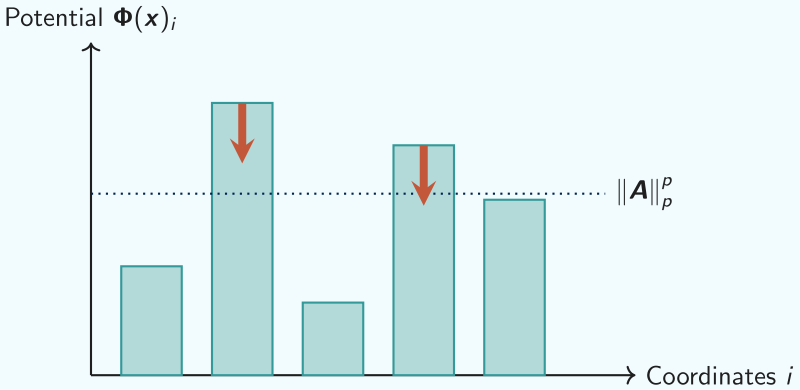
Somehow:

- We want to develop a method that does not require any conditioning on $\boldsymbol{A}$.
- Use potential $\boldsymbol{\Phi}(\boldsymbol{x}) \approx$ gradient $+ \|\boldsymbol{A}\|_p^p$

Intuition of the New Algorithm

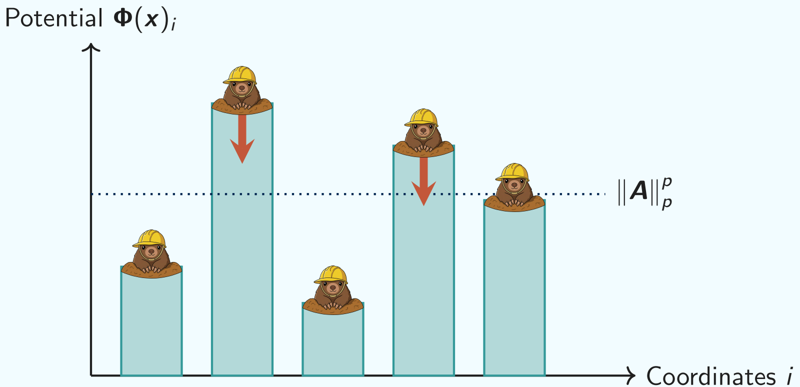


Intuition of the New Algorithm

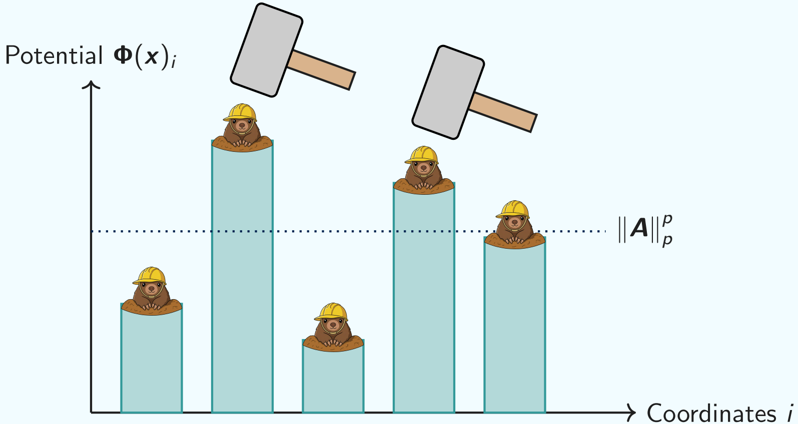


First Idea: Whac-A-Mole

Intuition of the Whac-A-Mole Algorithm



Intuition of the Whac-A-Mole Algorithm



Formalizing the Hammer

Coordinate-Scaling (i.e. the Hammer)

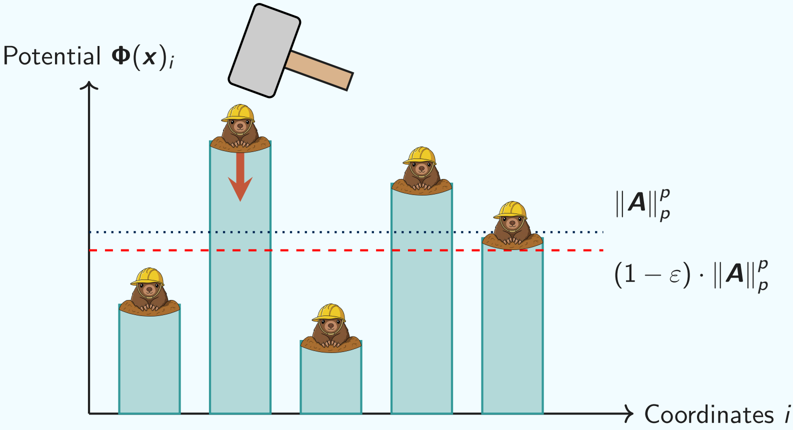
For some $\alpha > 0$, and a set $I \subseteq [n]$:

$$\boldsymbol{x}'_i = \begin{cases} (1+\alpha)\cdot\boldsymbol{x}_i & \text{if } i \in I; \\ \boldsymbol{x}_i & \text{otherwise.} \end{cases}$$

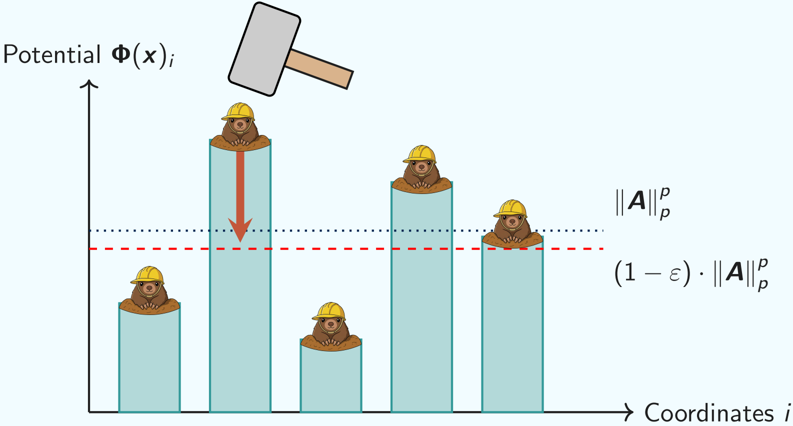
The Hammer

The value α controls the extent to which each entry of the potential vector varies between iterations.

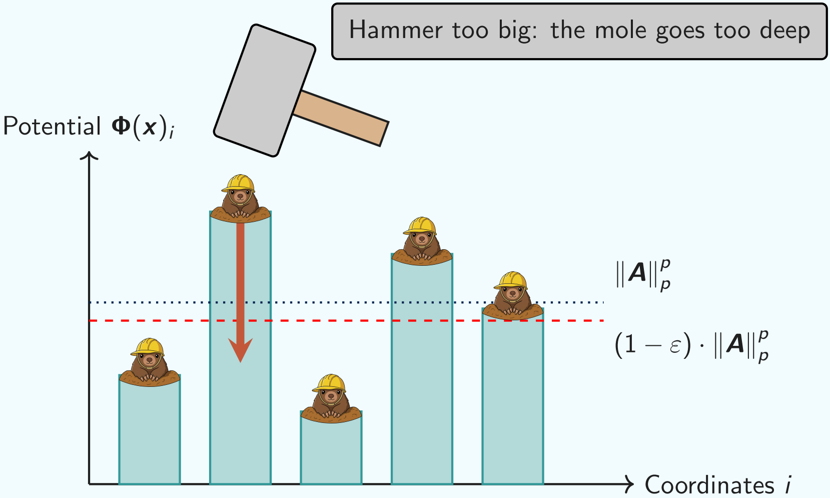
Calibrating the Hammer



Calibrating the Hammer



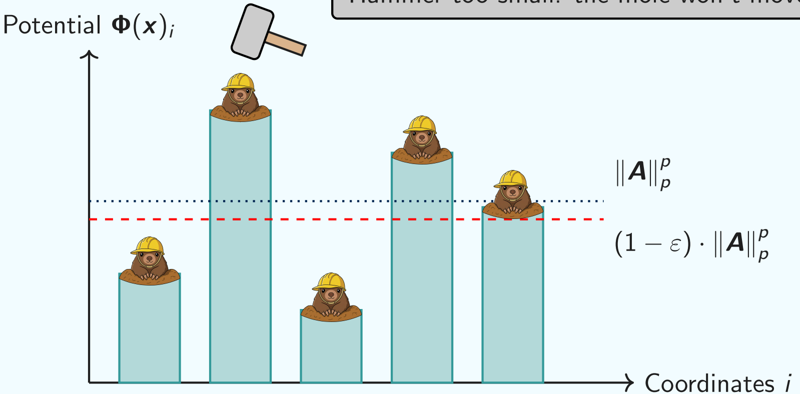
Calibrating the Hammer



Calibrating the Hammer

Hammer too big: the mole goes too deep

Hammer too small: the mole won't move



Main Theorem

Theorem

If:

- The Hammer is **calibrated**.
- The Hammer **hits moles with large potential**.

Then:

- One mole being hit $\approx \varepsilon^{-1}$ times shows convergence.

Rules of the Game

Rules

- We are only allowed to hit moles with **potential larger than** $\|\boldsymbol{A}\|_p^p$.
- Once there is **a mole** that is hit $\approx \varepsilon^{-1}$ times, we win !

Problems

- Is there always a mole with potential larger than $\|\boldsymbol{A}\|_p^p$?

Rules of the Game

Rules

- We are only allowed to hit moles with **potential larger than** $\|\boldsymbol{A}\|_p^p$.
- Once there is **a mole** that is hit $\approx \varepsilon^{-1}$ times, we win !

Problems

- Is there always a mole with potential larger than $\|\boldsymbol{A}\|_p^p$?
  - Yes ! The potentials **sandwich** $\|\boldsymbol{A}\|_p^p$.

First strategy: Hit the mole with the largest potential.

First Try: The Width-Dependent Trap

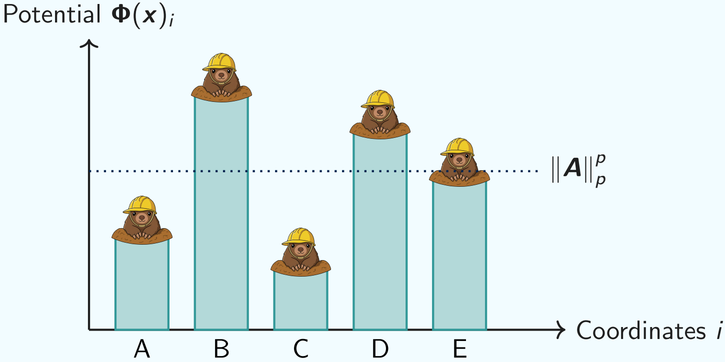
History

Mole	A	B	C	D	E
#Hit	0	0	0	0	0

Running time

End: One mole is hit ε^{-1} times.

Strategy: Hit the highest!



First Try: The Width-Dependent Trap

History

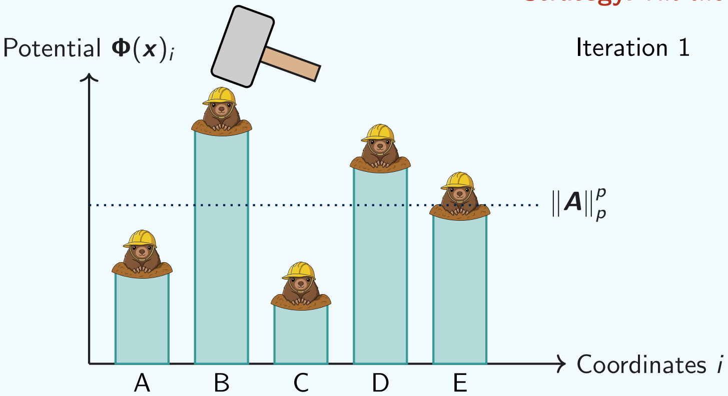
Mole	A	B	C	D	E
#Hit	0	1	0	0	0

Running time

End: One mole is hit ε^{-1} times.

Strategy: Hit the highest!

Iteration 1



First Try: The Width-Dependent Trap

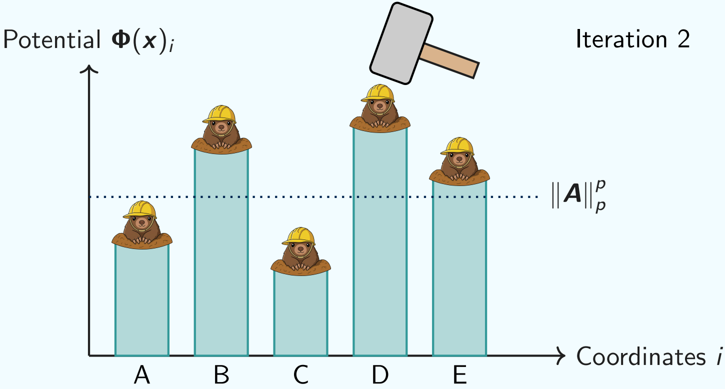
History

Mole	A	B	C	D	E
#Hit	0	1	0	1	0

Running time

End: One mole is hit ε^{-1} times.

Strategy: Hit the highest!



First Try: The Width-Dependent Trap

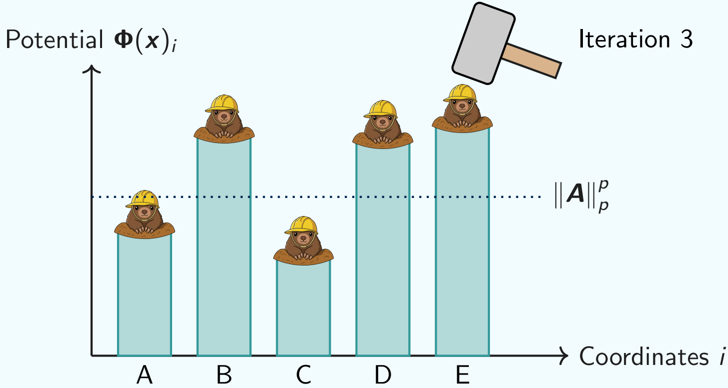
History

Mole	A	B	C	D	E
#Hit	0	1	0	1	1

Running time

End: One mole is hit ε^{-1} times.

Strategy: Hit the highest!



First Try: The Width-Dependent Trap

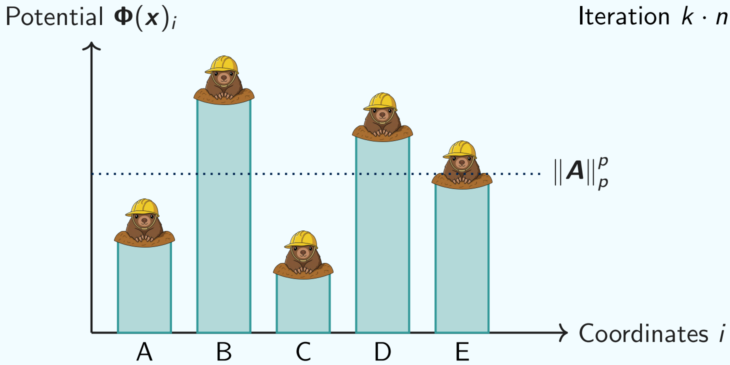
History

Mole	A	B	C	D	E
#Hit	k	k	k	k	k

Running time

End: One mole is hit ε^{-1} times.
After $k \cdot n$ iterations, one mole is hit k times.

Strategy: Hit the highest!



First Try: The Width-Dependent Trap

History

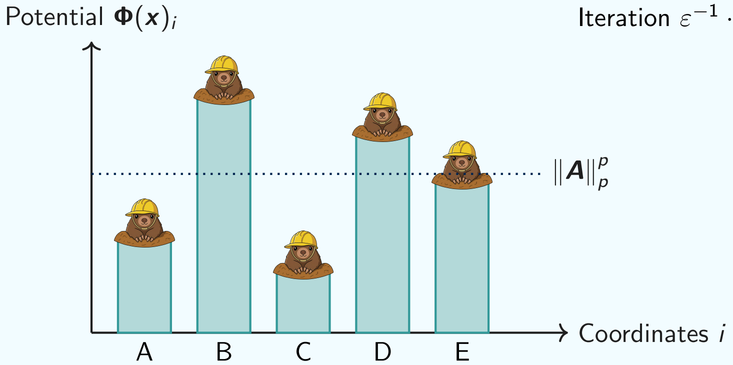
Mole	A	B	C	D	E
#Hit	$\frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$

Running time

End: One mole is hit ε^{-1} times.
After $\frac{n}{\varepsilon}$ iterations, one mole is hit $\frac{1}{\varepsilon}$ times.

Strategy: Hit the highest!

Iteration $\varepsilon^{-1} \cdot n$



First Try: The Width-Dependent Trap

Running Time

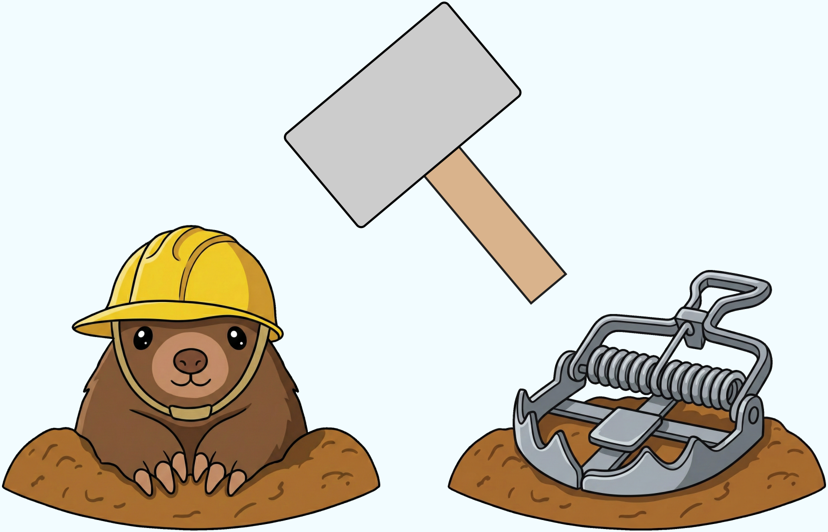
Convergence after $\widetilde{\mathcal{O}}(\frac{n}{\varepsilon})$ iterations.

- Faster than power iteration.
- Not *width-independent* (i.e. dependency on n).

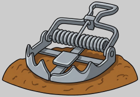
Solution

- We want to ensure the **same mole is hit** at each iteration.
- We will **trap moles** with potential slightly smaller than $\|\boldsymbol{A}\|_p^p$.

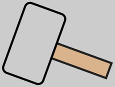
Second Idea: Whac-A-Mole + Trap the Moles



Second Try: One Slide Proof



A mole without large potential will never have large potential.

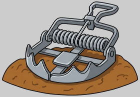


At each iteration, moles with large potential are hit.

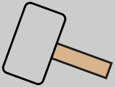
Properties of $\boldsymbol{\Phi}(\boldsymbol{x})$

At each iteration, at least one mole has large potential.

Second Try: One Slide Proof



A mole without large potential will never have large potential.



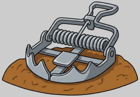
At each iteration, moles with large potential are hit.

Large potential moles at time T $=$ hit T times.

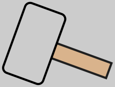
Properties of $\boldsymbol{\Phi}(\boldsymbol{x})$

At each iteration, at least one mole has large potential.

Second Try: One Slide Proof



A mole without large potential will never have large potential.



At each iteration, moles with large potential are hit.

Properties of $\boldsymbol{\Phi}(\boldsymbol{x})$

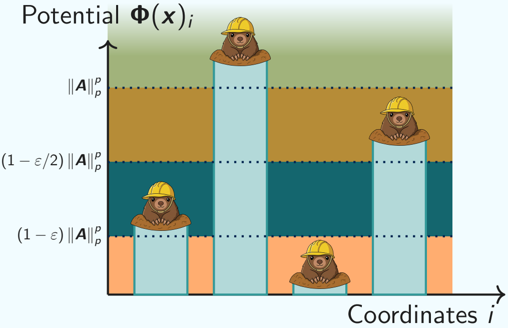
At each iteration, at least one mole has large potential.

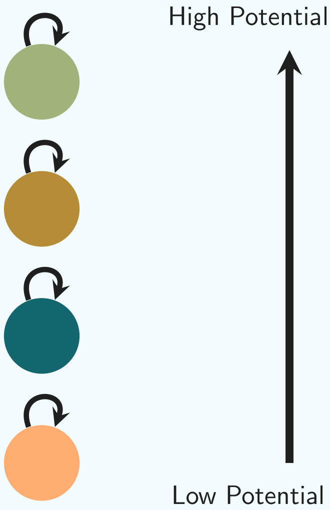
Large potential moles at time T $=$ hit T times.

At time T, at least one mole has been hit T times.

Second Try: Trap the Moles

Objective: No new moles in green.

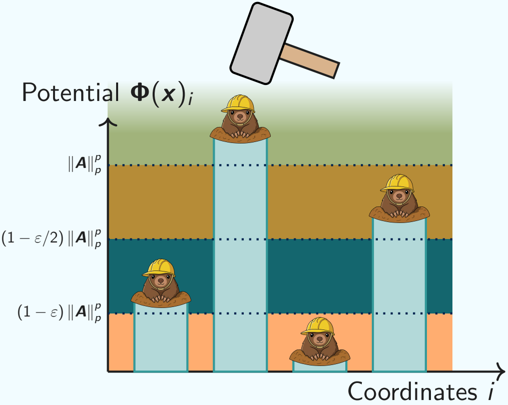


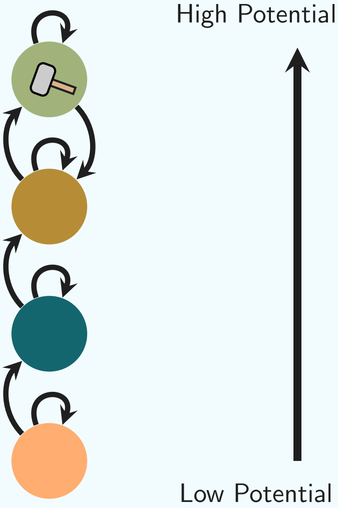


Second Try: Trap the Moles

Objective: No new moles in green.

- Hit moles in green.

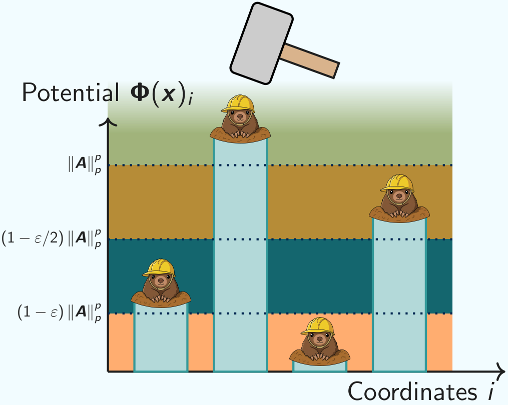


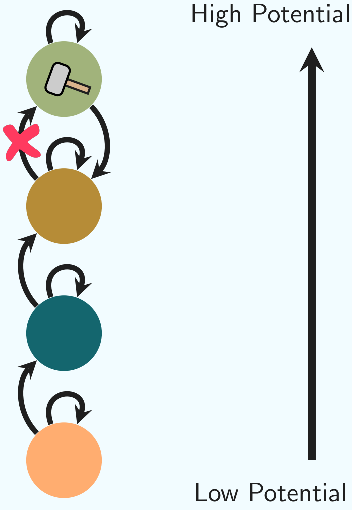


Second Try: Trap the Moles

Objective: No new moles in green.

- Hit moles in green.

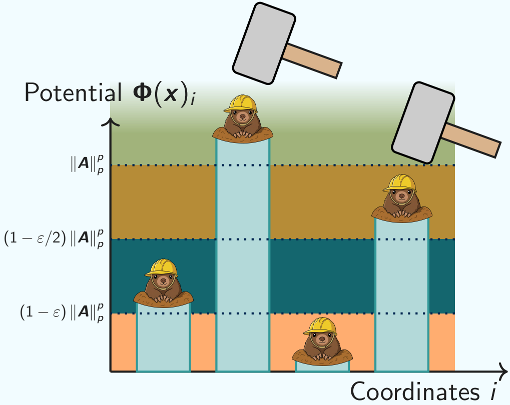


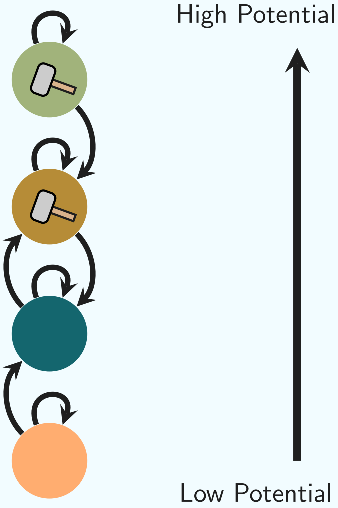


Second Try: Trap the Moles

Objective: No new moles in green.

- Hit moles in green.
- Hit moles in brown.

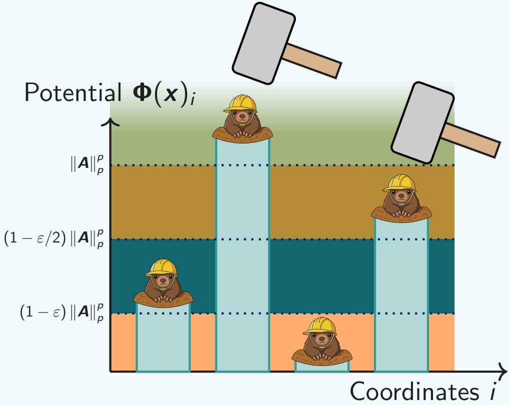


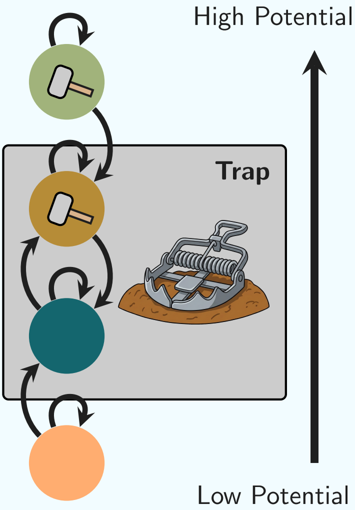


Second Try: Trap the Moles

Objective: No new moles in green.

- Hit moles in green.
- Hit moles in brown.





Second Try: Running Time and Remarks

Running Time

- Convergence after $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$ iterations.
- Each iteration requires computing the potentials, hence each iteration runs in $\textsc{nnz}(\boldsymbol{A})$.

Second Try: Running Time and Remarks

Running Time

- Convergence after $\widetilde{\mathcal{O}}(\frac{1}{\varepsilon})$ iterations.
- Each iteration requires computing the potentials, hence each iteration runs in $\textsc{nnz}(\boldsymbol{A})$.

Remarks

- **Knowledge of $\|\boldsymbol{A}\|_p$:** Same algorithm with a guess and binary search.
- **Tall inputs:** Precondition the algorithm with Lewis weight sampling.

Conclusion

Main Elements

- $\widetilde{\mathcal{O}}(\frac{\mathrm{NNZ}(\boldsymbol{A})}{q\varepsilon})$ time algorithm to find a $1-\varepsilon$ approximation to $\|\boldsymbol{A}\|_{q\to p}$ when $\boldsymbol{A}$ has non-negative entries, $q \geq p \geq 1$.
- Assumptions correspond to instances where PTAS exists.

Conclusion

Main Elements

- $\widetilde{\mathcal{O}}(\frac{\mathrm{NNZ}(\boldsymbol{A})}{q\varepsilon})$ time algorithm to find a $1-\varepsilon$ approximation to $\|\boldsymbol{A}\|_{q\to p}$ when $\boldsymbol{A}$ has non-negative entries, $q \geq p \geq 1$.
- Assumptions correspond to instances where PTAS exists.

Future Works

- Use this algorithm to compute competitive tree-based linear oblivious routings in time $\widetilde{\mathcal{O}}(m^3)$?

Conclusion

Main Elements

- $\widetilde{\mathcal{O}}(\frac{\mathrm{NNZ}(\boldsymbol{A})}{q\varepsilon})$ time algorithm to find a $1-\varepsilon$ approximation to $\|\boldsymbol{A}\|_{q\to p}$ when $\boldsymbol{A}$ has non-negative entries, $q \geq p \geq 1$.
- Assumptions correspond to instances where PTAS exists.

Future Works

- Use this algorithm to compute competitive tree-based linear oblivious routings in time $\widetilde{\mathcal{O}}(m^3)$?
- Generalize the method to approximate other operator norms ?
- Hardness when $p > q > 1$?

Conclusion

Main Elements

- $\widetilde{\mathcal{O}}(\frac{\mathrm{NNZ}(\boldsymbol{A})}{q\varepsilon})$ time algorithm to find a $1-\varepsilon$ approximation to $\|\boldsymbol{A}\|_{q\to p}$ when $\boldsymbol{A}$ has non-negative entries, $q \geq p \geq 1$.
- Assumptions correspond to instances where PTAS exists.

Future Works

- Use this algorithm to compute competitive tree-based linear oblivious routings in time $\widetilde{\mathcal{O}}(m^3)$?
- Generalize the method to approximate other operator norms ?
- Hardness when $p > q > 1$?

My website

Thank You !