Adaptive Matrix Sparsification and Applications to Empirical Risk Minimization

Albert Weng
Georgia Tech

Joint work with:
Yang P. Liu, Richard Peng, Colin Tang, Junzhao Yang (CMU)

Problem Formulation

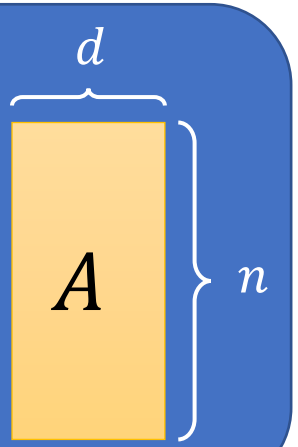
$$\min_{y \in \mathbb{R}^d} \sum_{i=1}^{m} f_i(A_i y - c_i)$$

Linear Program

$$\min_{x \in \mathbb{R}^n} c^\top x$$
$$A^\top x = b$$
$$x_i \geq 0$$

Generalize to $l_i \leq x_i \leq u_i$

Focus on tall & dense:
for $A \in \mathbb{R}^{n \times d}$
- $n \gg d$
- $\Omega(nd)$ nonzero entries



Empirical Risk Minimization

$$\min c^\top x$$
$$A^\top x = b$$
$$x \in K_1 \times \cdots \times K_m$$

- $K_i \subseteq \mathbb{R}^{n_i}$ compact, convex, $n_i \leq C_K$
- $\sum_i n_i = n$

Applications: linear regression, l_p regression, LASSO, logistic regression, support vector machines, quantile regression, AdaBoost

Minimum Enclosing Ball and Ellipsoid (MEB, MVEE)

Complexity Overview

	Linear Programming	
$n = O(d)$ regime	[CLS21, JSWZ21]	$n^{\max\{\omega, 2+1/18\}}$
$n \gg d$ regime	[BLSS20, BLL+21]	$nd + d^{2.5}$

Lower Bound
$\Omega(nd + d^{\omega})$
(conditional on solving linear systems)

d^{ω} is time to multiply two $d \times d$ matrices; currently $\omega \approx 2.37$

Complexity Overview

	Linear Programming		Empirical Risk Minimization	
$n = O(d)$ regime	[CLS21, JSWZ21]	$n^{\max\{\omega, 2+1/18\}}$	[LSZ19]	$n^{\max\{\omega, 2+1/6\}}$
$n \gg d$ regime	[BLSS20, BLL+21]	$nd + d^{2.5}$		

Lower Bound
$\Omega(nd + d^{\omega})$
(conditional on solving linear systems)

Complexity Overview

Standard interior point method with $\sqrt{n}$ iterations

	Linear Programming		Empirical Risk Minimization	
$n = O(d)$ regime	[CLS21, JSWZ21]	$n^{\max\{\omega, 2+1/18\}}$	[LSZ19]	$n^{\max\{\omega, 2+1/6\}}$
$n \gg d$ regime	[BLSS20, BLL+21]	$nd + d^{2.5}$		

Lower Bound
$\Omega(nd + d^{\omega})$
(conditional on solving linear systems)

Complexity Overview

Standard interior point method with $\sqrt{n}$ iterations

	Linear Programming		Empirical Risk Minimization	
$n = O(d)$ regime	[CLS21, JSWZ21]	$n^{\max\{\omega, 2+1/18\}}$	[LSZ19]	$n^{\max\{\omega, 2+1/6\}}$
$n \gg d$ regime	[BLSS20, BLL+21]	$nd + d^{2.5}$		

Lee-Sidford barrier: more complicated IPM, only needs $\sqrt{d}$ iterations. Unknown how to extend!

Lower Bound
$\Omega(nd + d^{\omega})$
(conditional on solving linear systems)

Complexity Overview

Standard interior point method with $\sqrt{n}$ iterations

	Linear Programming		Empirical Risk Minimization	
$n = O(d)$ regime	[CLS21, JSWZ21]	$n^{\max\{\omega, 2+1/18\}}$	[LSZ19]	$n^{\max\{\omega, 2+1/6\}}$
$n \gg d$ regime	[BLSS20, BLL+21]	$nd + d^{2.5}$	Our paper	$nd + d^{11}$

Lee-Sidford barrier: more complicated IPM, only needs $\sqrt{d}$ iterations. Unknown how to extend.

Combinatorial matrix sparsification data structure. (still $\sqrt{n}$ iterations!)

Lower Bound
$\Omega(nd + d^{\omega})$
(conditional on solving linear systems)

Central Path Method

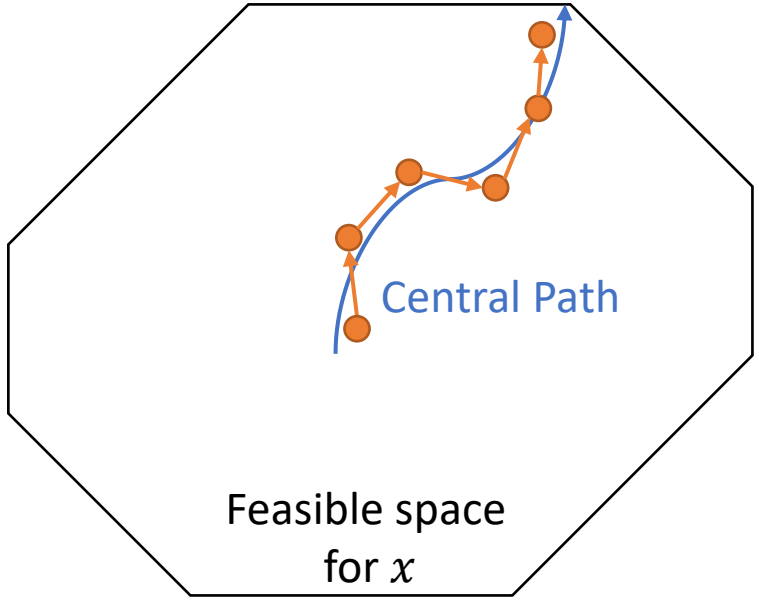
$$\min c^\top x$$
$$A^\top x = b$$
$$x \geq 0$$

Parameterized by t: x moves towards optimal solution as $t \to 0$.

$-\log(x_i)$ enforces $x_i \geq 0$ (blows up as $x_i \to 0$)

Central Path (log barrier):

$$x^{(t)} = \arg\min_{A^\top x = b} c^\top x + t \sum_{i=1}^{m} -\log(x_i)$$



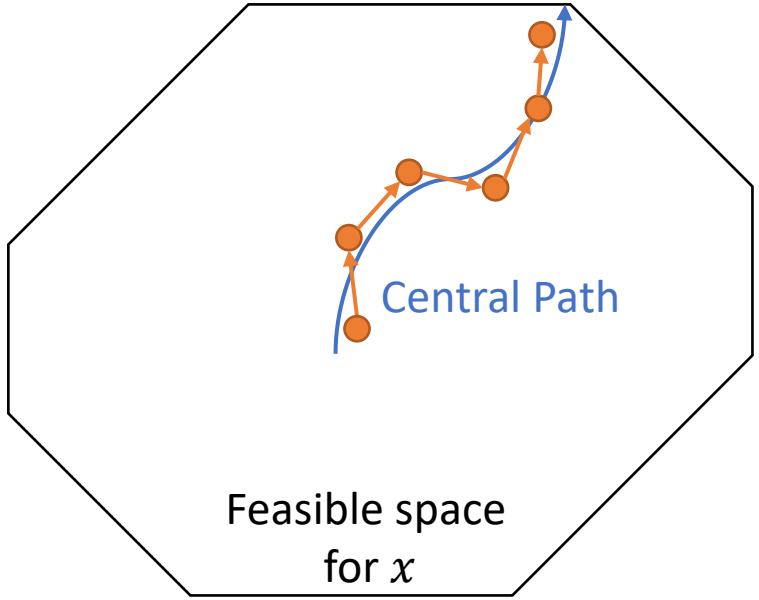
ERM Central Path Setup

$$\min c^\top x$$
$$A^\top x = b$$
$$x \in K_1 \times \cdots \times K_m$$

- We assume access to $K_i \subseteq \mathbb{R}^{n_i}$ through ϕ_i, a barrier for K_i.
- Let $x_i \in \mathbb{R}^{n_i}$ be the coordinates that correspond to K_i.

Central Path:

$$x^{(t)} = \arg\min_{A^\top x = b} c^\top x + t \underbrace{\sum_{i=1}^{m} \phi_i(x_i)}_{\Phi(x)}$$



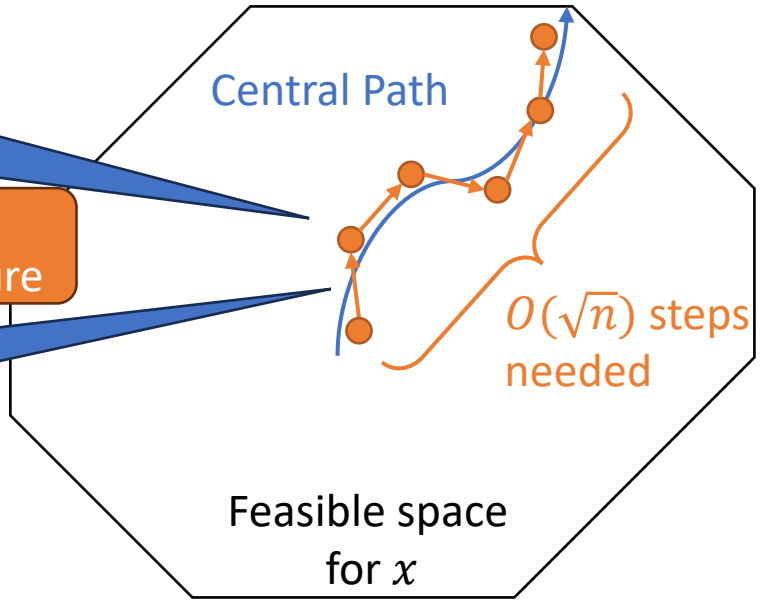
IPM Analysis

IPM Overview:
- start at point close to central path
- take a step: close to the central path, smaller t.

Take a step:
solve an approximate linear system in the matrix $A^\top \nabla^2 \Phi(x)^{-1} A$.

Too expensive to compute!
Use matrix sparsification data structure

x changes slowly:
only $\tilde{O}(n)$ coordinates change throughout the IPM



Central Path:

$$x^{(t)} = \arg\min_{A^\top x = b} c^\top x + t \underbrace{\sum_{i=1}^{m} \phi_i(x_i)}_{\Phi(x)}$$

High Level Idea

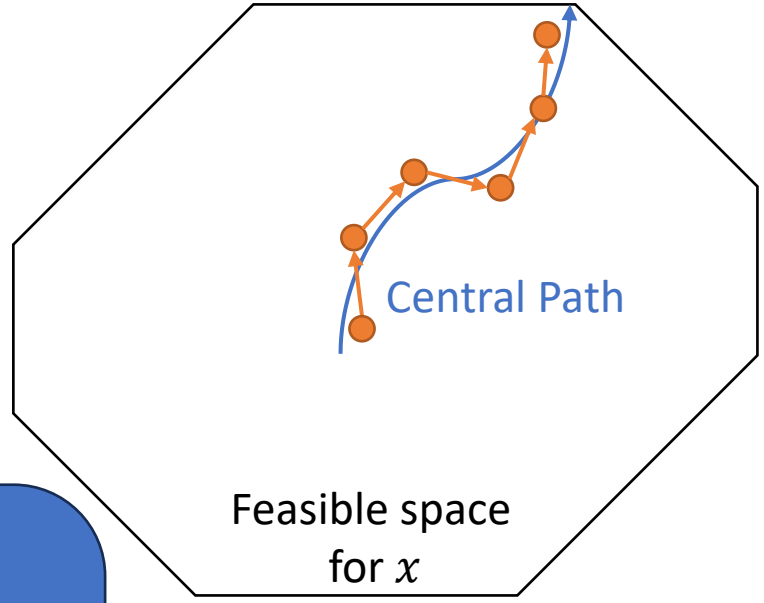
Solve $\min_{A^\top x = b, x_i \in K_i \forall i} c^\top x$

↓ Central path method:
solve a system of form $A^\top \nabla^2 \Phi(x)^{-1} A$

Approximately maintain $A^\top \nabla^2 \Phi(x)^{-1} A$
as x changes slowly

↓ Leverage score sampling:
sample by leverage scores

Maintain leverage score overestimates
of rows of $\nabla^2 \Phi(x)^{-1/2} A$



Leverage Scores:
For matrix $M \in \mathbb{R}^{n \times d}$,
leverage score τ_i of row m_i
is

$$\tau_i := m_i^\top (M^\top M)^{-1} m_i$$

- importance of a row
- between 0 and 1
- sum to d

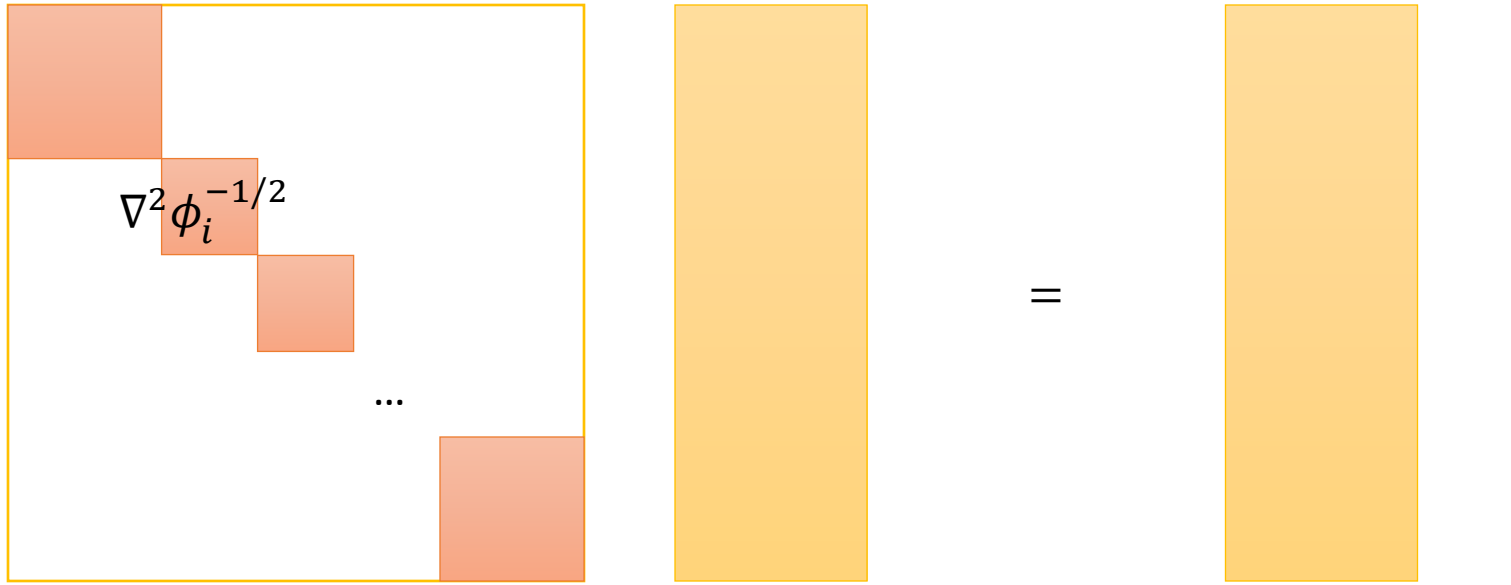
Sampling:
Subsample m_i w.p. $\geq \tau_i$
to get $\widetilde{M}$, then

$$\widetilde{M}^\top \widetilde{M} \approx M^\top M$$

Maintaining $\nabla^2\Phi(x)^{-1/2}A$

Leverage Scores
$\tau_i(M) := m_i^\top (M^\top M)^{-1} m_i$

$$\Phi(x) := \sum_{i=1}^{m} \phi_i(x_i)$$

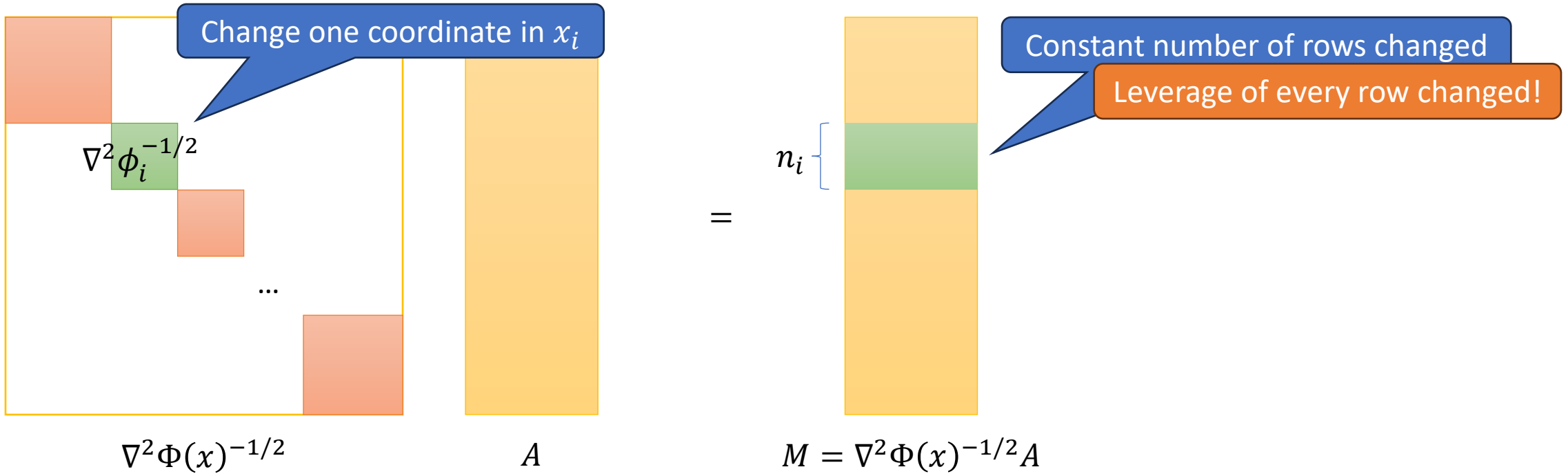


$\nabla^2\Phi(x)^{-1/2}$ A $M = \nabla^2\Phi(x)^{-1/2}A$

Data structure problem: $\widetilde{O}(n)$ coordinates of x change throughout the IPM. Maintain leverage scores of $\nabla^2\Phi(x)^{-1/2}A$.

Maintaining $\nabla^2\Phi(x)^{-1/2}A$

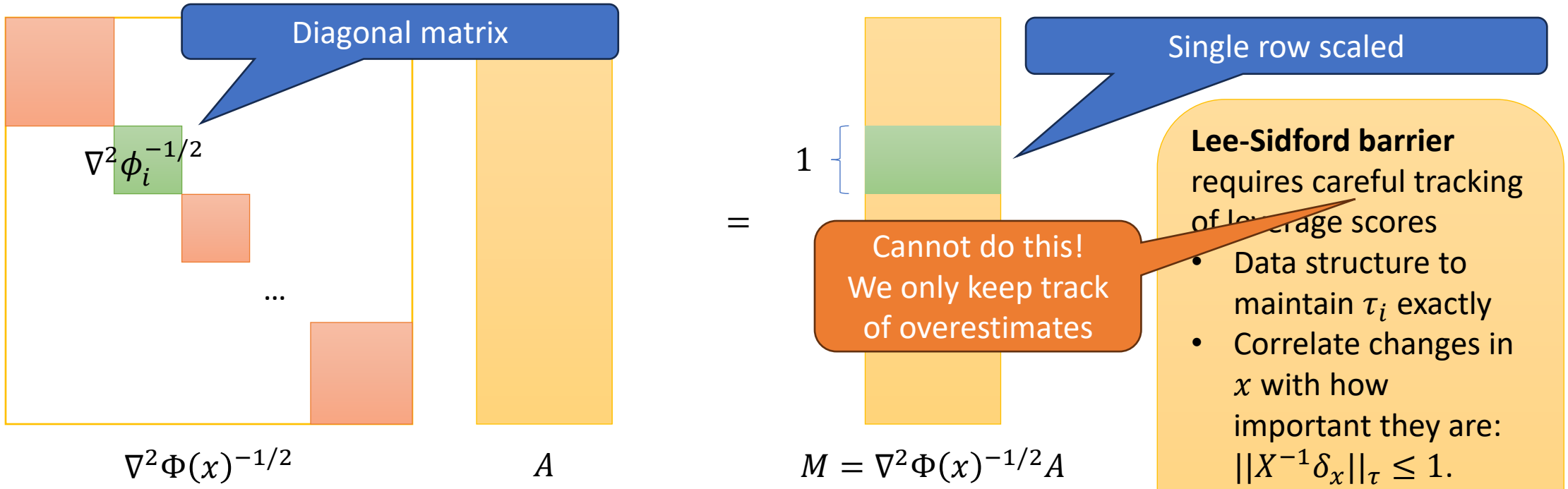
Leverage Scores
$\tau_i(M) := m_i^\top (M^\top M)^{-1} m_i$



Data structure problem: $\widetilde{O}(n)$ coordinates of x change throughout the IPM.
Maintain leverage scores of $\nabla^2\Phi(x)^{-1/2}A$.

Linear Programming Approach

Leverage Scores
$\tau_i(M) := m_i^\top (M^\top M)^{-1} m_i$



Lee-Sidford barrier
requires careful tracking of leverage scores

- Data structure to maintain τ_i exactly
- Correlate changes in x with how important they are: $||X^{-1}\delta_x||_\tau \leq 1$.
- Need leverage to not change much: $||T^{-1}\delta_\tau||_\tau \leq 1$

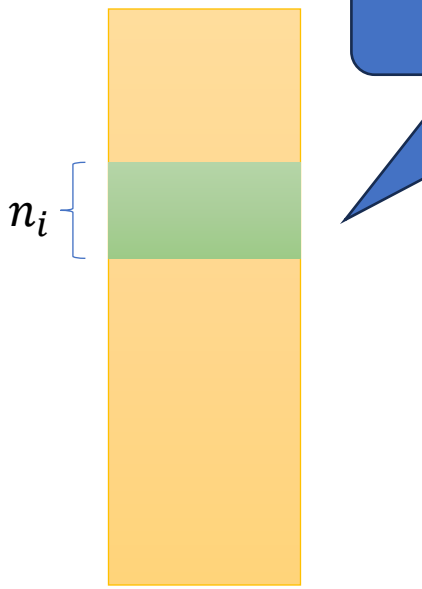
Data structure problem: $\widetilde{O}(n)$ coordinates of x change throughout the IPM. Maintain leverage scores of $\nabla^2 \Phi(x)^{-1/2} A$.

Data Structure Approach

Leverage Scores
$\tau_i(M) := m_i^\top (M^\top M)^{-1} m_i$

Data Structure Problem
- $\tilde{O}(n)$ arbitrary row changes
- Comes in $O(\sqrt{n})$ groups (iterations)
- Only need to maintain overestimates $\tilde{\tau}_i \geq \tau_i$
- Cannot overestimate by too much:

$$\sum_i \tilde{\tau}_i \leq \tilde{O}(d)$$



Entire block changed
(not just scaling)

$$M = \nabla^2 \Phi(x)^{-1/2} A$$

Data structure problem: $\widetilde{O}(n)$ coordinates of x change throughout the IPM. Maintain leverage scores of $\nabla^2 \Phi(x)^{-1/2} A$.

Leverage Score Maintenance

Leverage Score Maintenance

Data Structure Problem
Given $M \in \mathbb{R}^{n \times d}$
undergoing Q iterations
of insertions/deletions of
total size $\tilde{O}(n)$,
maintain $\tilde{\tau}_i$:

- $\tilde{\tau}_i \geq m_i^\top (M^\top M)^{-1} m_i$
- $\sum_i \tilde{\tau}_i \leq \tilde{O}(d)$

in time $O(nd + Qd^6)$

Obstacle: rows with too high leverage

Leverage Score Maintenance

Data Structure Problem
Given $M \in \mathbb{R}^{n \times d}$ undergoing Q iterations of insertions/deletions of total size $\tilde{O}(n)$, maintain $\tilde{\tau}_i$:

- $\tilde{\tau}_i \geq m_i^\top (M^\top M)^{-1} m_i$
- $\sum_i \tilde{\tau}_i \leq \tilde{O}(d)$

in time $O(nd + Qd^6)$

Obstacle: rows with too high leverage

- deleting a row with leverage score 1 will break our approximations

Only perform row halvings!

Leverage Score Maintenance

Data Structure Problem
Given $M \in \mathbb{R}^{n \times d}$
undergoing Q iterations
of insertions/deletions of
total size $\tilde{O}(n)$,
maintain $\tilde{\tau}_i$:

- $\tilde{\tau}_i \geq m_i^\top (M^\top M)^{-1} m_i$
- $\sum_i \tilde{\tau}_i \leq \tilde{O}(d)$

in time $O(nd + Qd^6)$

Obstacle: rows with too high leverage

- deleting a row with leverage score 1 will break our approximations

Only perform row halvings!

- "remove" at most leverage 0.5 in a step: leverage scores are preserved multiplicatively across each step

Decremental Halving Reduction

Fully Dynamic Problem
Given $M \in \mathbb{R}^{n \times d}$
undergoing Q iterations
of insertions/deletions of
total size $\tilde{O}(n)$,
maintain $\tilde{\tau}_i$:

- $\tilde{\tau}_i \geq m_i^\top (M^\top M)^{-1} m_i$
- $\sum_i \tilde{\tau}_i \leq \tilde{O}(d)$

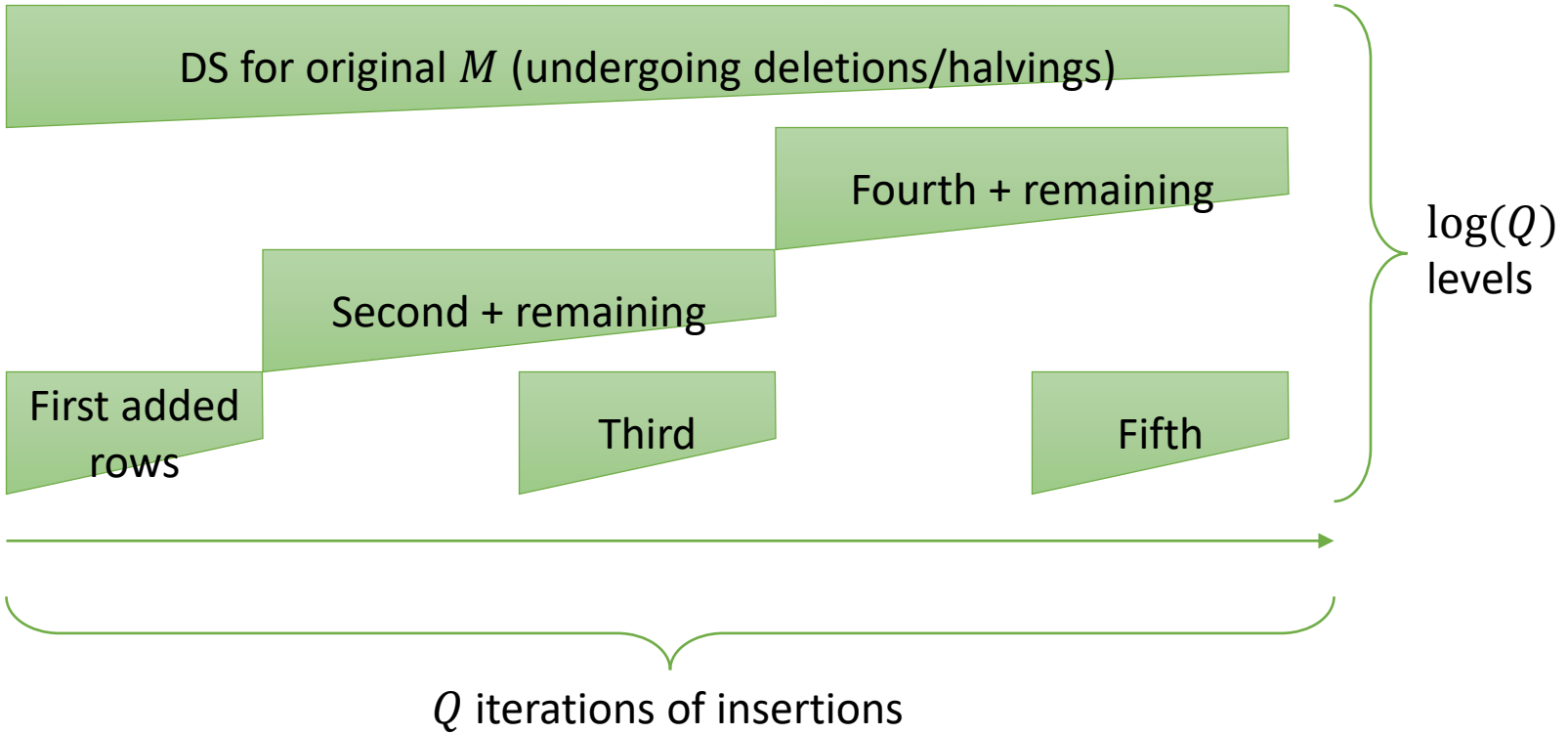
in time $O(nd + Qd^6)$

Decremental Problem
Given $M \in \mathbb{R}^{n \times d}$
undergoing $\tilde{O}(n)$ row
halvings, maintain $\tilde{\tau}_i$
in time $O(nd + d^6)$

Deletion: $\tilde{O}(\log(n))$ row halvings

Insertion: $\tilde{O}(\log(Q))$ levels

Gram matrix sums to total gram matrix, so overestimates!



Bounding Leverage Score Changes

Data Structure Problem
Given $M \in \mathbb{R}^{n \times d}$ undergoing $\tilde{O}(n)$ row halvings, maintain $\tilde{\tau}_i$:

- $\tilde{\tau}_i \geq m_i^\top (M^\top M)^{-1} m_i$
- $\sum_i \tilde{\tau}_i \leq \tilde{O}(d)$

in time $O(nd + d^6)$

detect row leverage score increase of d/n

Fact 1: If we remove leverage of ≤ 0.5, then no leverage score of the remaining rows has more than doubled.

We can wait and group row halvings into batches of constant leverage decrease.

Fact 2: Removing leverage σ decreases $\det(M^\top M)$ by a factor of $(1 - \sigma)$.
(IPM maintains polynomially bounded singular values of $M^\top M$.)

Sum of deleted leverage is at most $O(d \log n)$.

Therefore, at most $\tilde{O}(d)$ batches.

Bounding Leverage Score Changes

Data Structure Problem
Given $M \in \mathbb{R}^{n \times d}$ undergoing $\tilde{O}(n)$ row halvings, maintain $\tilde{\tau}_i$:

- $\tilde{\tau}_i \geq m_i^\top (M^\top M)^{-1} m_i$
- $\sum_i \tilde{\tau}_i \leq \tilde{O}(d)$

in time $O(nd + d^6)$

detect row leverage score increase of d/n

Fact 1: If we remove leverage of ≤ 0.5, then no leverage score of the remaining rows has more than doubled.

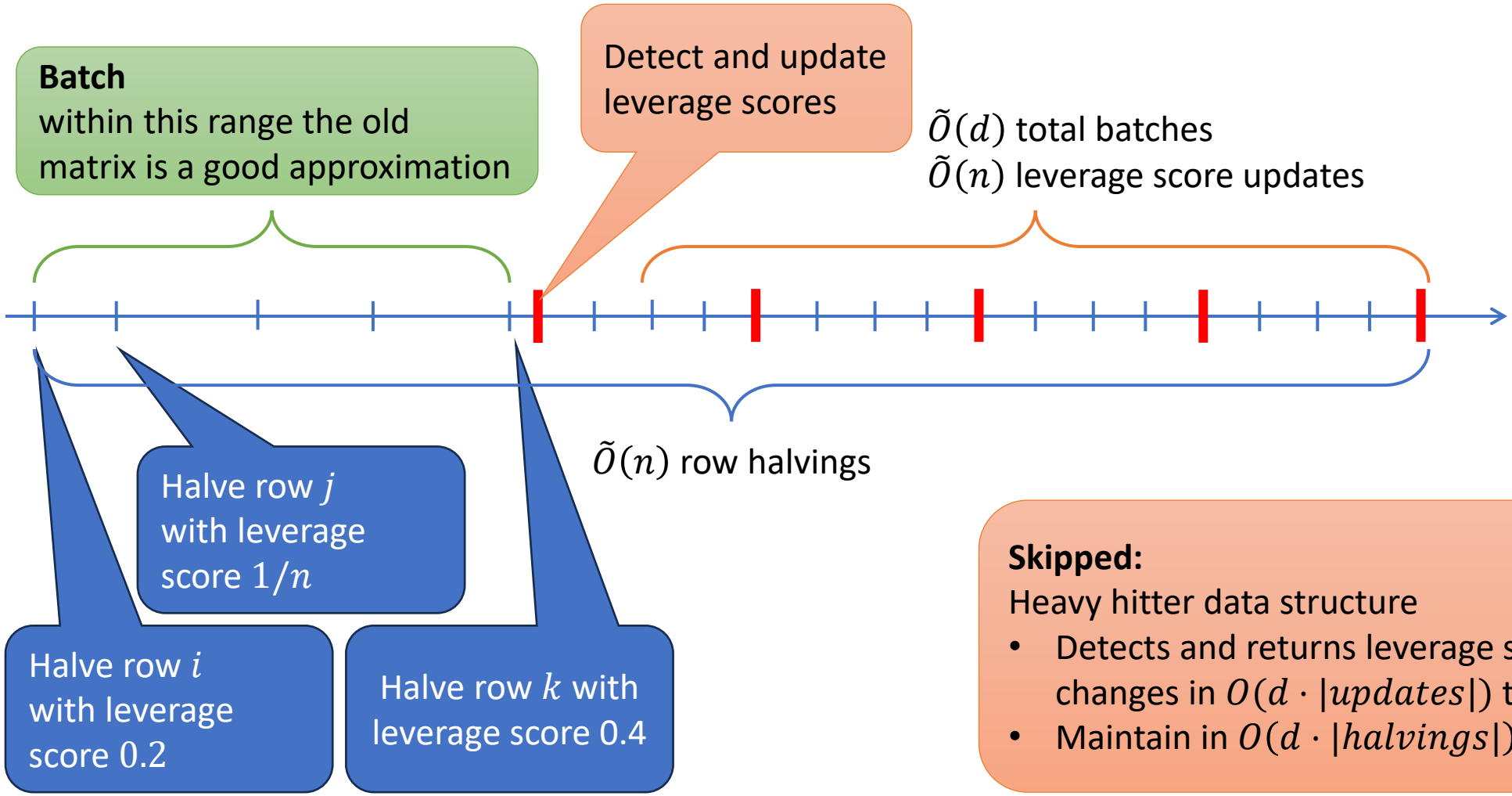
We can wait and group row halvings into batches of constant leverage decrease.

Fact 2: Removing leverage σ decreases $\det(M^\top M)$ by a factor of $(1 - \sigma)$.
(IPM maintains polynomially bounded singular values of $M^\top M$.)

Sum of deleted leverage is at most $O(d \log n)$.
Increased leverage also at most $O(d \log n)$.

Therefore, at most $\tilde{O}(d)$ batches, $\tilde{O}(n)$ leverage score updates.

Batching Implementation



Skipped:
Heavy hitter data structure

- Detects and returns leverage score changes in $O(d \cdot |updates|)$ time
- Maintain in $O(d \cdot |halvings|)$ time

Conclusion

Summary

- New data structure for matrix sparsification via batching updates
- First tall dense ERM in near linear time
- Lee-Sidford machinery not required

Open

- Reduce the poly d term
- Apply Lee-Sidford barrier to ERM (or other generalizations of linear programming)