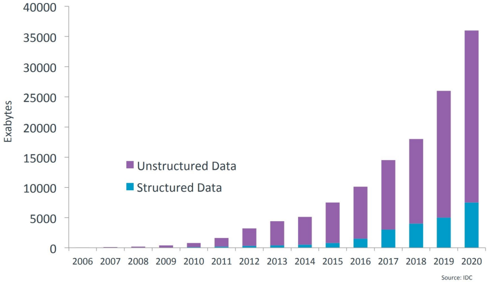
Randomized Sketching for Convex and Non-Convex Optimization

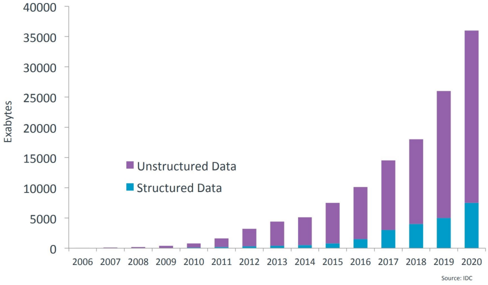
Mert Pilanci
Department of Electrical Engineering
Stanford University

December 6, 2018

Scale of data

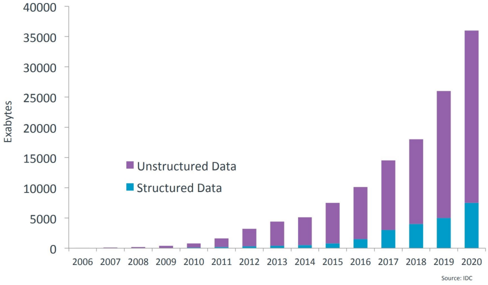


Scale of data



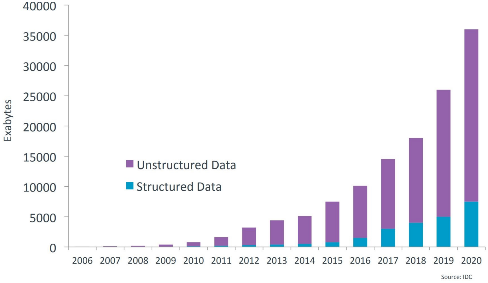
- Every day, we create 2.5 billion gigabytes of data

Scale of data

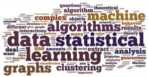


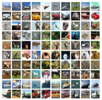
- Every day, we create 2.5 billion gigabytes of data
- Data stored grows 4x faster than world economy (Mayer-Schonberger)

Scale of data

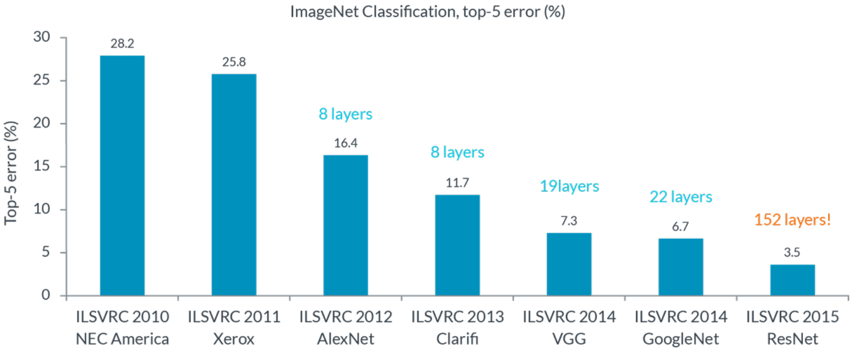


- Every day, we create 2.5 billion gigabytes of data
- Data stored grows 4x faster than world economy (Mayer-Schonberger)



Deep learning revolution



- Machine learning
- Statistical estimation and data analysis
- Signal processing and control theory
- Computational imaging
- Design and manufacturing
- Decision making

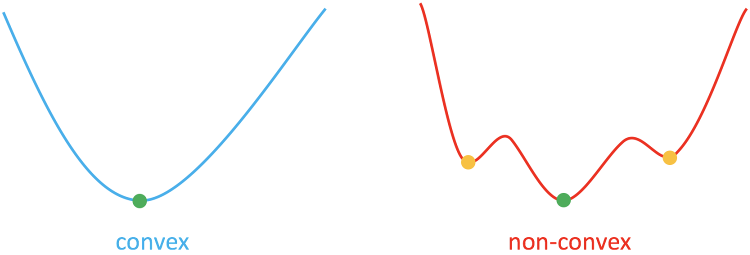
$$\text{minimize} \quad g(x) \quad \textsf{subject to} \quad \text{constraints}$$

Machine learning and statistics

- More data points reduce sampling error, higher significance
  $\rightarrow$ Large scale optimization problems

Machine learning and statistics

- More data points reduce sampling error, higher significance
  $\rightarrow$ Large scale optimization problems
- Complex models can improve accuracy
  $\rightarrow$ Non-convex optimization problems harder to solve as dimensions grow



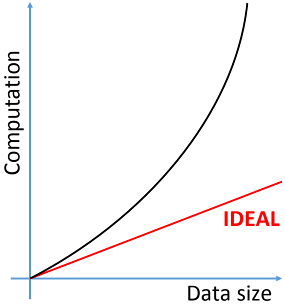
Machine learning and statistics

- More data points reduce sampling error, higher significance
  $\rightarrow$ Large scale optimization problems
- Complex models can improve accuracy
  $\rightarrow$ Non-convex optimization problems harder to solve as dimensions grow

What if we could reduce the data volume without losing any significant information ?

Machine learning and statistics

- More data points reduce sampling error, higher significance
  → Large scale optimization problems
- Complex models can improve accuracy
  → Non-convex optimization problems harder to solve as dimensions grow



Goals:

1. Find optimal trade-offs between **computation** and **accuracy**
2. Leverage **distributed computation**
3. Tackle **non-convexity**

Outline

Optimization and Big Data

Sketching

Distributed Sketching

Non-convex Problems

Ongoing work

Convex optimization and big data

$$\min\text{imize} \quad f(Ax) \quad \text{subject to } x \in \mathcal{C}$$

- Data matrix $A \in \mathbb{R}^{n \times d}$ is extremely large

Convex optimization and big data

$$\text{minimize} \quad f(Ax) \quad \text{subject to } x \in \mathcal{C}$$

- Data matrix $A \in \mathbb{R}^{n \times d}$ is extremely large

Examples:

- Airline dataset (120GB) $n = 120 \times 10^6,\ d = 28$
  Flight arrival and departure details from 1987 to 2008

Convex optimization and big data

$$\text{minimize} \quad f(Ax) \quad \text{subject to } x \in \mathcal{C}$$

- Data matrix $A \in \mathbb{R}^{n \times d}$ is extremely large

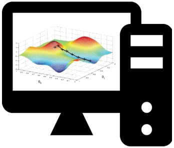
Examples:

- Airline dataset (120GB) $n = 120 \times 10^6, \; d = 28$
  Flight arrival and departure details from 1987 to 2008
- Imagenet dataset (1.31TB) $n = 14 \times 10^6$, $d = 2 \times 10^5$
  14 Million images for visual recognition

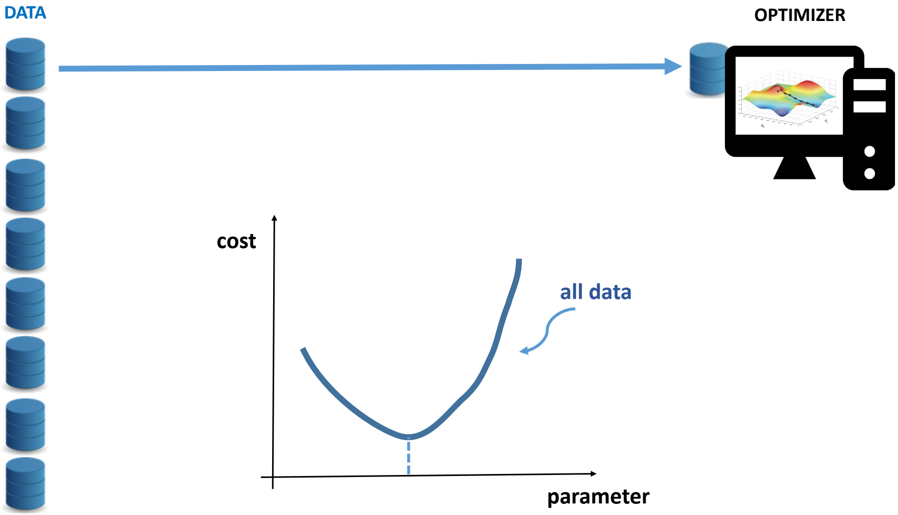
[US Department of Transportation]
[Deng et al. 2009]

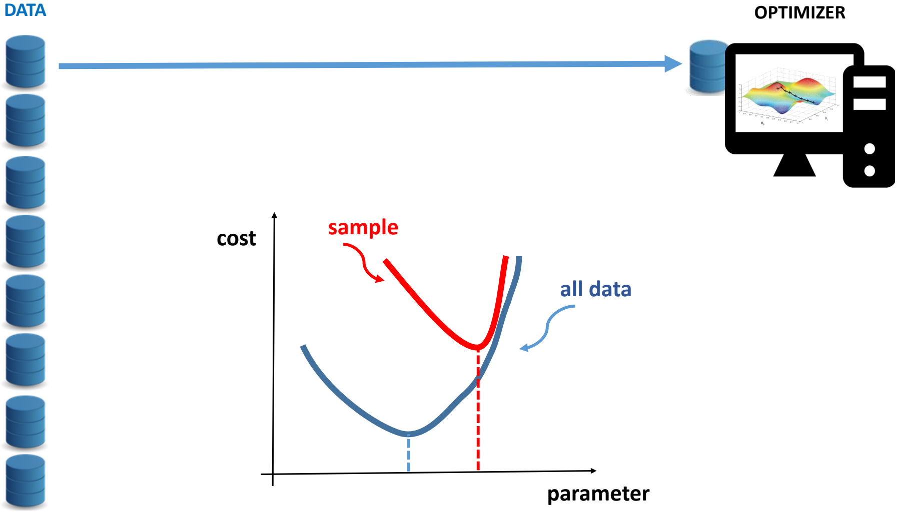
DATA

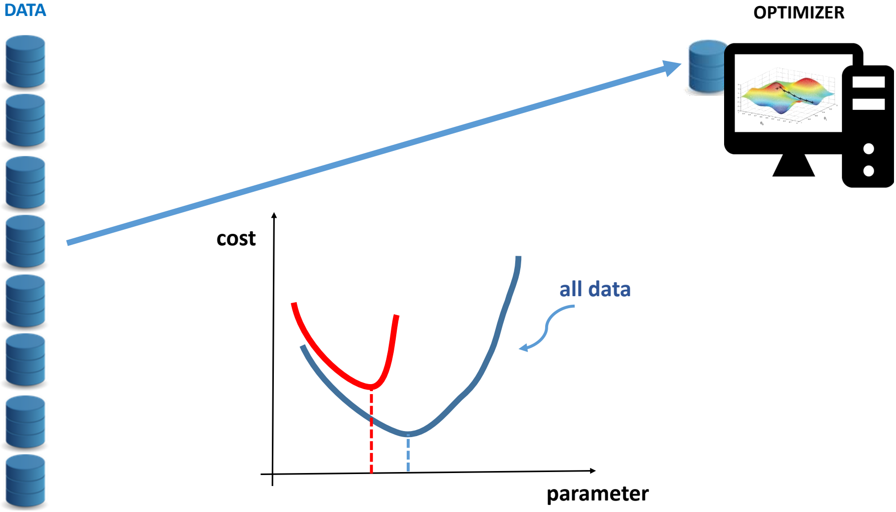
OPTIMIZER

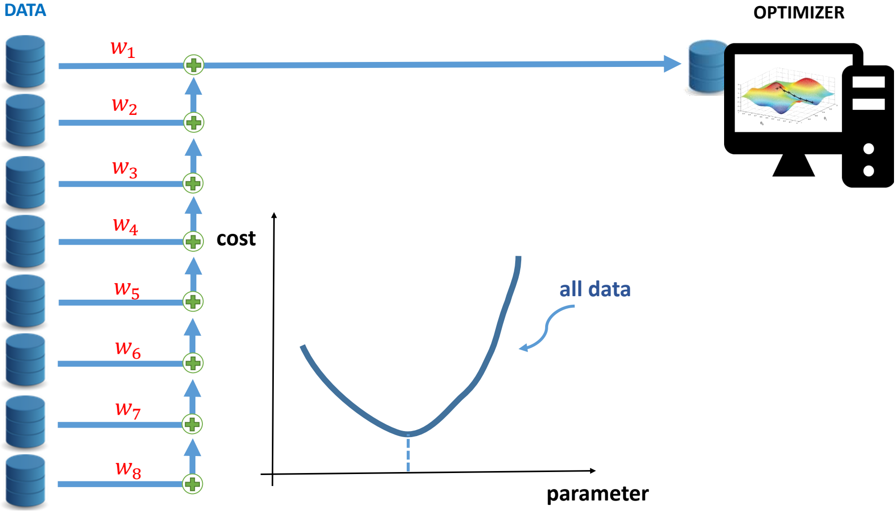


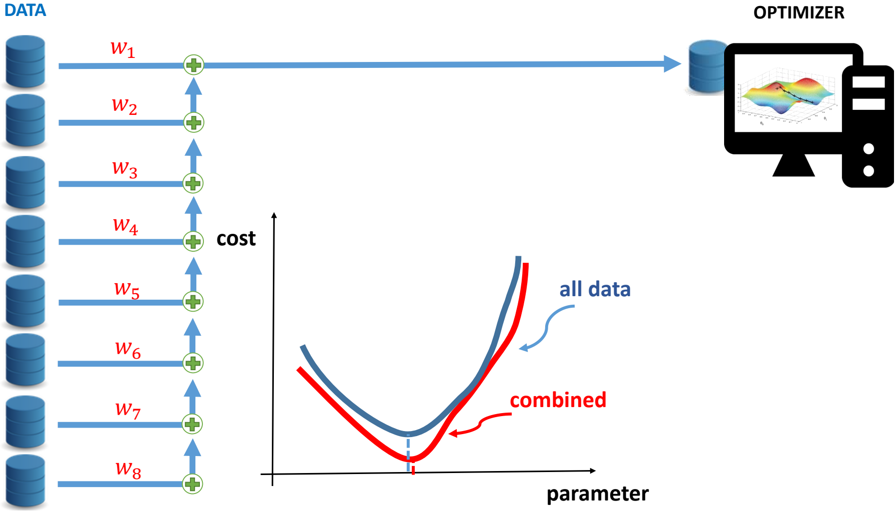

DATA
OPTIMIZER


DATA
OPTIMIZER
cost
all data
parameter


DATA
OPTIMIZER
cost
sample
all data
parameter


DATA
OPTIMIZER
cost
all data
parameter


DATA
w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
OPTIMIZER
cost
all data
parameter

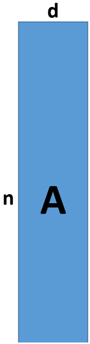

DATA
OPTIMIZER
w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
cost
all data
combined
parameter

Example : Least squares prediction

- A : $n \times d$ feature matrix, and y : $n \times 1$ response vector
- Original $\mathbf{OPT} = \min_{x \in \mathcal{C}} \underbrace{\|Ax - y\|^2}_{f(Ax)}$

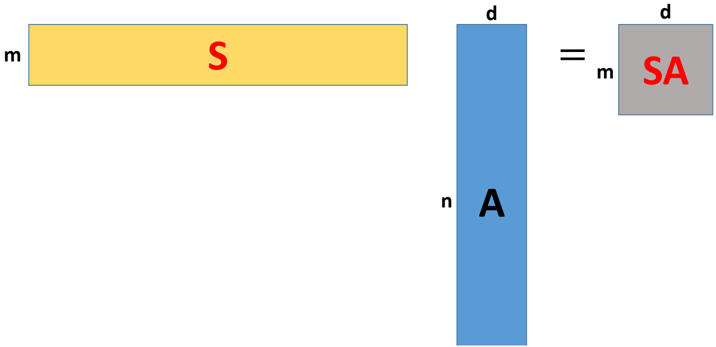
Example : Least squares prediction

- A : $n \times d$ feature matrix, and y : $n \times 1$ response vector
- Original $\qquad \mathbf{OPT} = \min_{x \in \mathcal{C}} \underbrace{\|Ax - y\|^2}_{f(Ax)}$



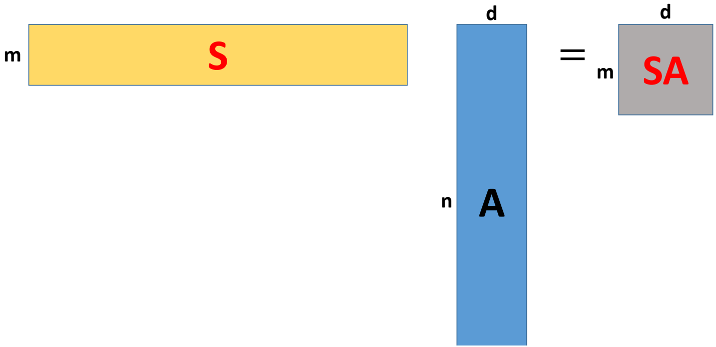
Example : Least squares prediction

- A : $n \times d$ feature matrix, and y : $n \times 1$ response vector
- Original $\quad \mathbf{OPT} = \min_{x \in \mathcal{C}} \underbrace{\|Ax - y\|^2}_{f(Ax)}$



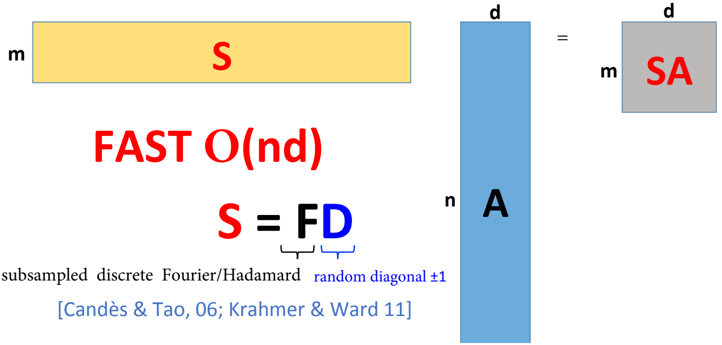
Example : Least squares prediction

- $A : n \times d$ feature matrix, and $y : n \times 1$ response vector
- Original $\quad \mathbf{OPT} = \min_{x \in \mathcal{C}} \underbrace{\|Ax - y\|^2}_{f(Ax)}$
- Approximate $\quad \widehat{x} = \arg\min_{x \in \mathcal{C}} \|S(Ax - y)\|^2$
- $S : m \times n$ *sketching* matrix (e.g., i.i.d. ± 1 random matrix)



Example : Least squares prediction

- A : $n \times d$ feature matrix, and y : $n \times 1$ response vector
- Original $\quad \mathbf{OPT} = \min_{x \in \mathcal{C}} \underbrace{\|Ax - y\|^2}_{f(Ax)}$
- Approximate $\quad \widehat{x} = \arg\min_{x \in \mathcal{C}} \|S(Ax - y)\|^2$
- S : $m \times n$ *sketching* matrix (e.g., i.i.d. ± 1 random matrix)



[Candès & Tao, 06; Krahmer & Ward 11]

Example : Least squares prediction

- $A : n \times d$ feature matrix, and $y : n \times 1$ response vector
- Original $\quad \mathbf{OPT} = \min_{x \in \mathcal{C}} \underbrace{\|Ax - y\|^2}_{f(Ax)}$
- Approximate $\quad \widehat{x} = \arg\min_{x \in \mathcal{C}} \|S(Ax - y)\|^2$
- $S : m \times n$ *sketching* matrix (e.g., i.i.d. ± 1 random matrix)

Theorem : Cost approximation

If $m \geq 2\,\mathrm{rank}^*(A)/\epsilon$, then

$\mathbf{OPT} \leq f(A\widehat{x}) \leq (1+\epsilon)\mathbf{OPT}$

with high probability

Sarlós 06; Rokhlin and Tygert 08; **P.** and Wainwright, IEEE Trans. Info. Theory 2015 $\quad *$effective rank

Example : Least squares prediction

- $A : n \times d$ feature matrix, and $y : n \times 1$ response vector
- Original $\quad \mathbf{OPT} = \min_{x \in \mathcal{C}} \underbrace{\|Ax - y\|^2}_{f(Ax)}$
- Approximate $\quad \widehat{x} = \arg\min_{x \in \mathcal{C}} \|S(Ax - y)\|^2$
- $S : m \times n$ *sketching* matrix (e.g., i.i.d. ± 1 random matrix)

Theorem : Converse

If $m \leq c_0 \operatorname{rank}^*(A)/\epsilon$, then

$f(A\widehat{x}) \geq (1 + \epsilon)\mathbf{OPT}$

with probability $> \frac{1}{2}$

P. and Wainwright, IEEE Trans. Info. Theory 2015

Example : Least squares prediction

Practical use

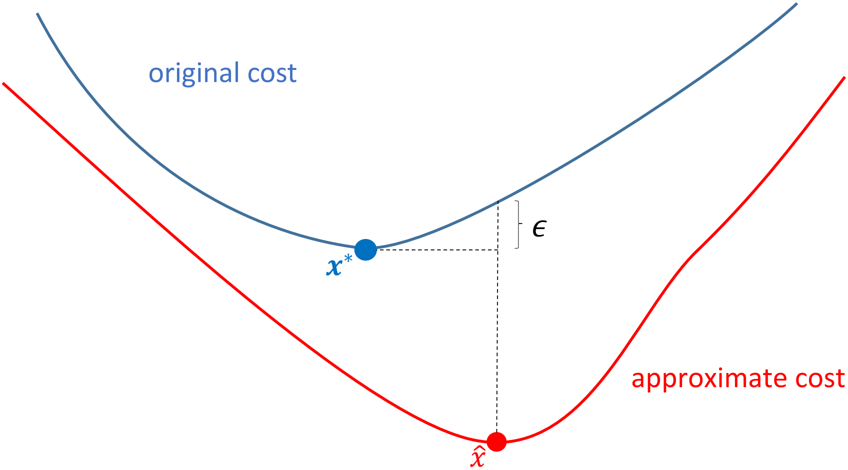
Airline dataset $n = 120{,}000{,}000, \; d = 28$

$m = 500$ gives 1.1-approximation

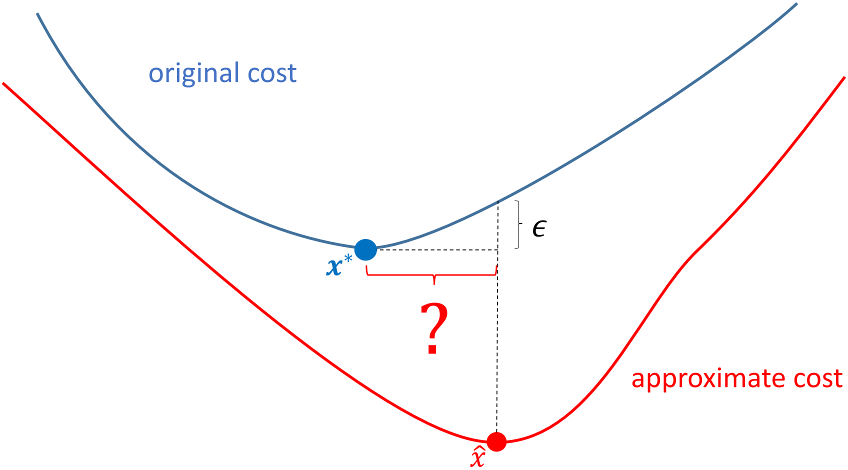
$m = 5000$ gives 1.01-approximation

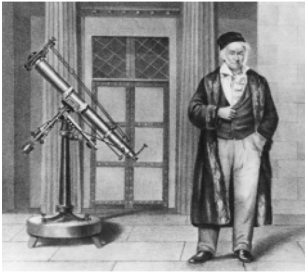
[**P** and Wainwright, IEEE Trans. Info. Theory 2015]

$$\min_x \|Ax - y\|_2^2 \qquad \text{and} \qquad \min_x \|S(Ax - y)\|_2^2$$



$$\min_x \|Ax - y\|_2^2 \quad \text{and} \quad \min_x \|S(Ax - y)\|_2^2$$

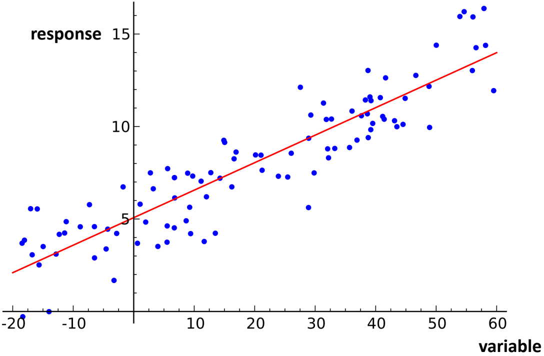




Least squares

$$\min_x \|Ax - y\|^2$$

[Gauss, 1795]



Statistical error

- Consider the noisy observation model

$$y = Ax^* + w, \qquad \text{where} \quad w_1, w_2 ... \sim N(0, \sigma^2)$$

Statistical error

- Consider the noisy observation model

$$y = Ax^* + w, \qquad \text{where} \quad w_1, w_2... \sim N(0, \sigma^2)$$

- estimation error:

$$\mathbb{E}\|x_{LS} - x^*\|_2^2 = O\left(\frac{d}{n}\sigma^2\right)$$

Statistical error

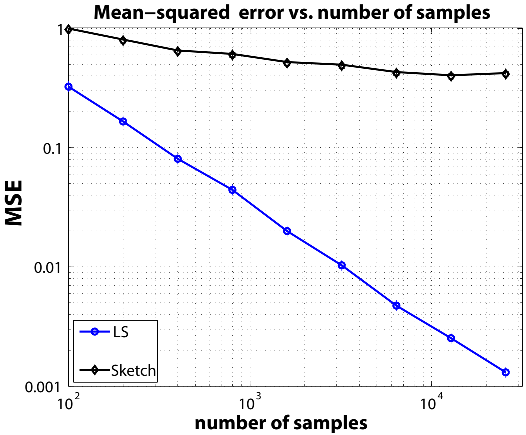
- Observation model $y = Ax^* + w$, where $w \sim N(0, \sigma^2 I_n)$
- Is the sketched solution $\hat{x}$ statistically optimal?

$$\hat{x} = \arg\min_x \|SAx - Sy\|^2 \quad \text{where } S \in \mathbb{R}^{m \times n}$$

Statistical error

- Observation model $y = Ax^* + w$, where $w \sim N(0, \sigma^2 I_n)$
- Is the sketched solution $\hat{x}$ statistically optimal?

$$\hat{x} = \arg\min_x \|SAx - Sy\|^2 \quad \text{where } S \in \mathbb{R}^{m \times n}$$



Suboptimality of the Classical Sketch

- Is the sketched solution $\hat{x}$ statistically optimal?

Suboptimality of the Classical Sketch

- Is the sketched solution $\hat{x}$ statistically optimal?
- **No**, information-theoretically impossible!

Suboptimality of the Classical Sketch

- Is the sketched solution $\hat{x}$ statistically optimal?
- **No**, information-theoretically impossible!
- For Gaussian, i.i.d. ± 1 or DFT/Hadamard based S and $m =$ constant $\times\ \mathrm{rank}(A)$

Theorem

Any estimator that is a function of (SA, Sy) obeys

$$\mathbb{E}_{S,w}\left[\|\hat{x} - x^*\|^2\right] \gtrsim \ \sigma^2$$

[**P.** and Wainwright, Journal of Machine Learning Research 2016]

Suboptimality of the Classical Sketch

- Is the sketched solution $\hat{x}$ statistically optimal?
- **No**, information-theoretically impossible!
- For Gaussian, i.i.d. ± 1 or DFT/Hadamard based S and $m =$ constant $\times \operatorname{rank}(A)$

Theorem

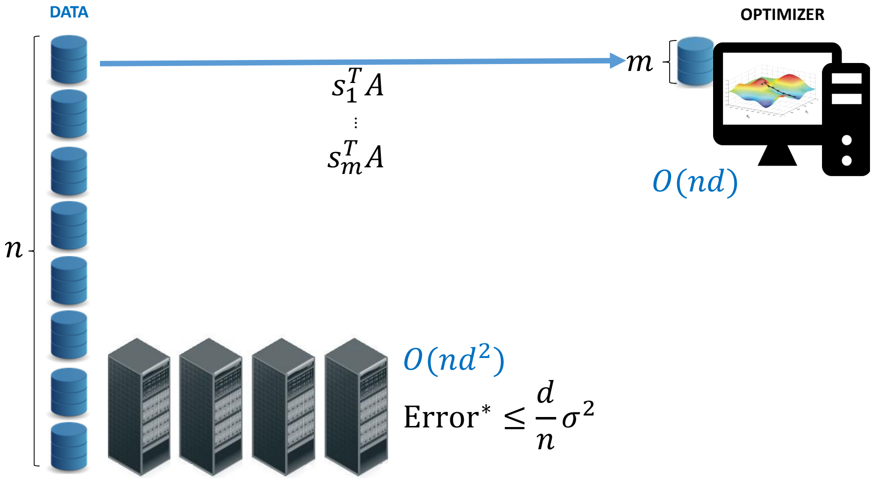
Any estimator that is a function of (SA, Sy) obeys

$$\mathbb{E}_{S,w}\left[\|\hat{x} - x^*\|^2\right] \gtrsim \ \sigma^2$$

	statistical error	computation
sketch	**suboptimal**	$\mathbf{O(nd)}$
original	**optimal**	$\mathbf{O(nd^2)}$

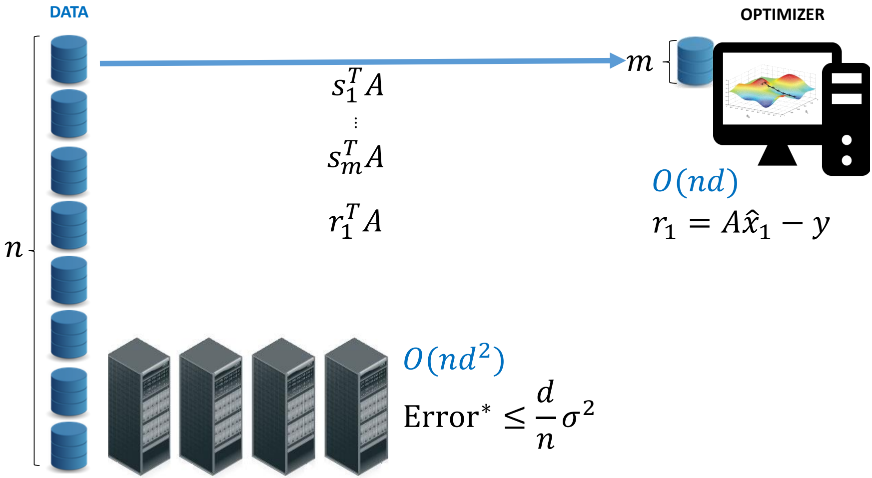
[**P.** and Wainwright, Journal of Machine Learning Research 2016]

Is there an **optimal** algorithm with complexity $\mathbf{O(nd)}$?



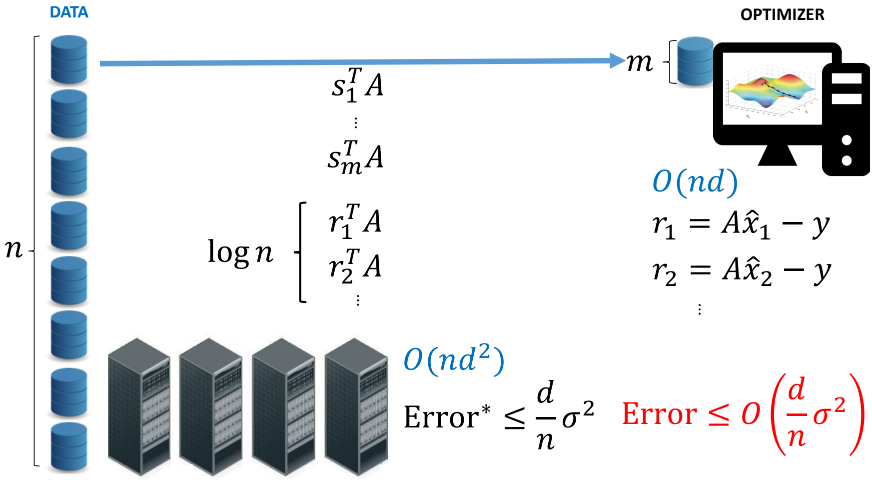
Is there an **optimal** algorithm with complexity $\mathbf{O(nd)}$?

YES!

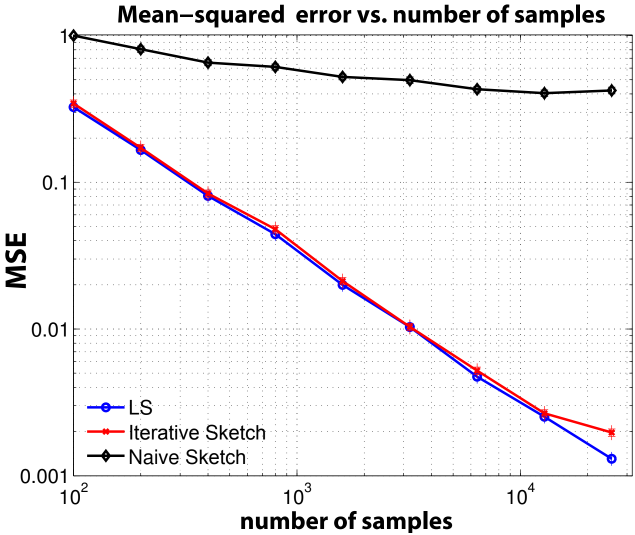


Is there an **optimal** algorithm with complexity $\mathbf{O(nd)}$?

YES!



Numerical Simulation: Estimation Error



Iterative Sketch

$$\boxed{\begin{array}{c} S A \\ r_1^T A \\ \vdots \\ r_t^T A \end{array}}$$

Iterative Sketch

$$\boxed{\begin{matrix} SA \\ r_1^T A \\ \vdots \\ r_t^T A \end{matrix}}$$

- Statistical model : $y = Ax^* + w$ where $x^* \in \mathcal{C}$

Theorem (Optimality)

Iterative Sketch achieves optimal prediction error with $\log(n/d)$ iterations for any convex set $\mathcal{C}$

	statistical error	computation
sketch	**suboptimal**	$\mathbf{O(nd)}$
original	**optimal**	$\mathbf{O(nd^2)}$
iterative sketch	**optimal**	$\mathbf{O(nd)}$

[**P.** and Wainwright, Journal of Machine Learning Research 2016]

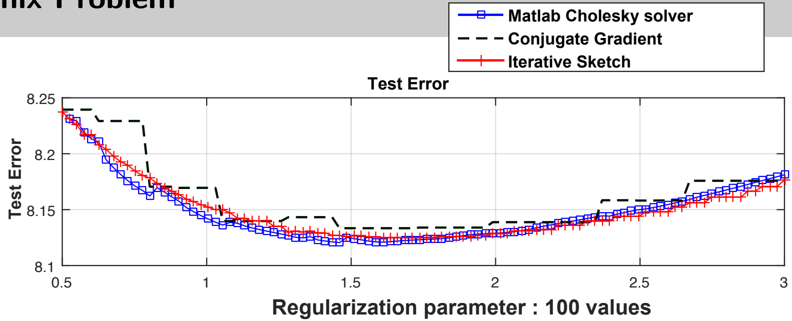
Netflix Problem

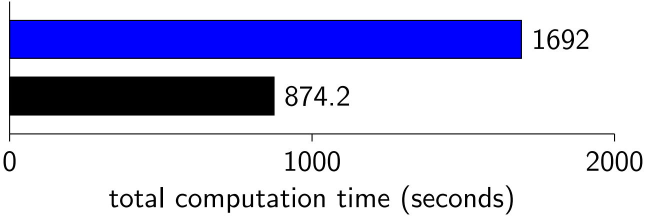
- 500000×17000 matrix A of ratings (users $\times$ movies)
- Predict the ratings of a particular movie
- Least-squares regression with ℓ_2 regularization

$$\min_x \|Ax - y\|^2 + \lambda \|x\|_2^2$$

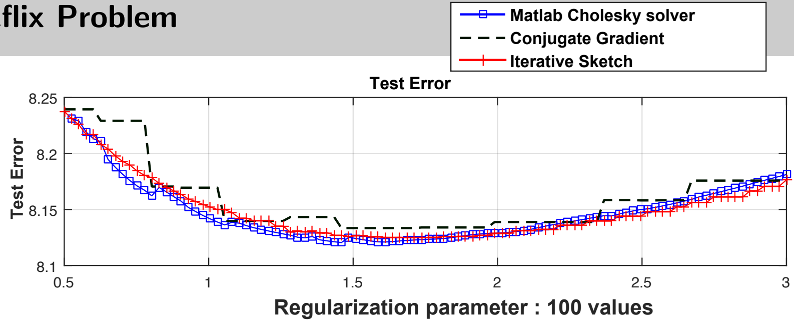
- Partition into test and training sets, solve for all values of $\lambda \in \{1, 2, ..., 100\}$.

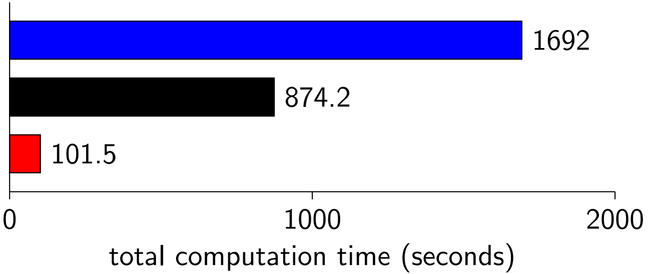
Netflix Problem





Netflix Problem



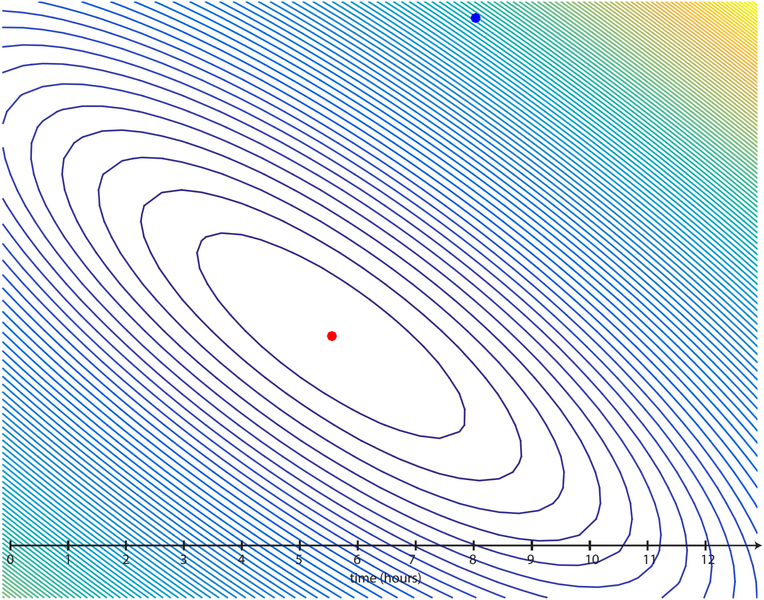


How to generalize to arbitrary functions ?

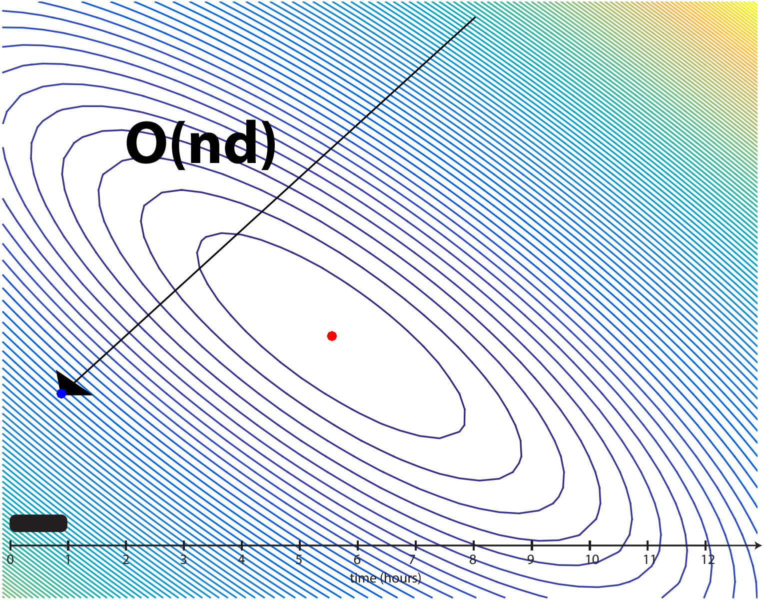
Convex objective, where $A \in \mathbb{R}^{n \times d}$ is a large data matrix

$$x^* = \arg\min_{x \in \mathcal{C}} \; f(Ax)$$

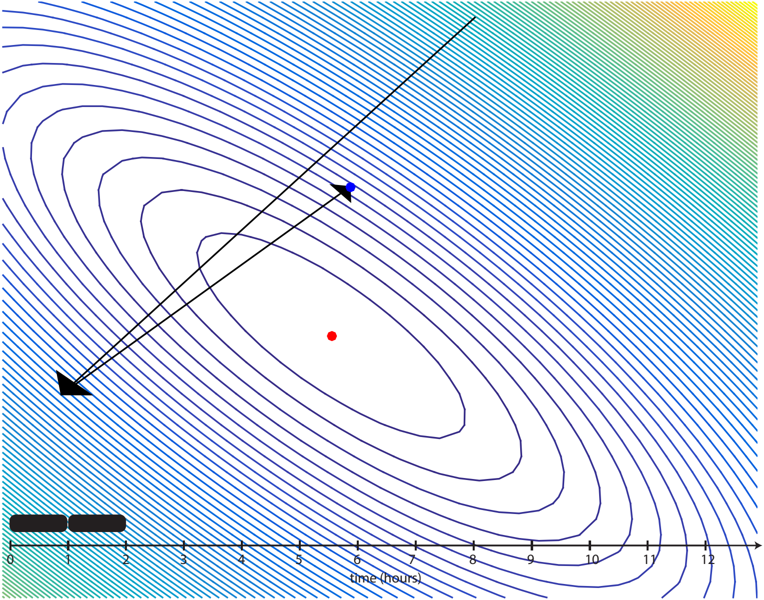
Gradient Descent



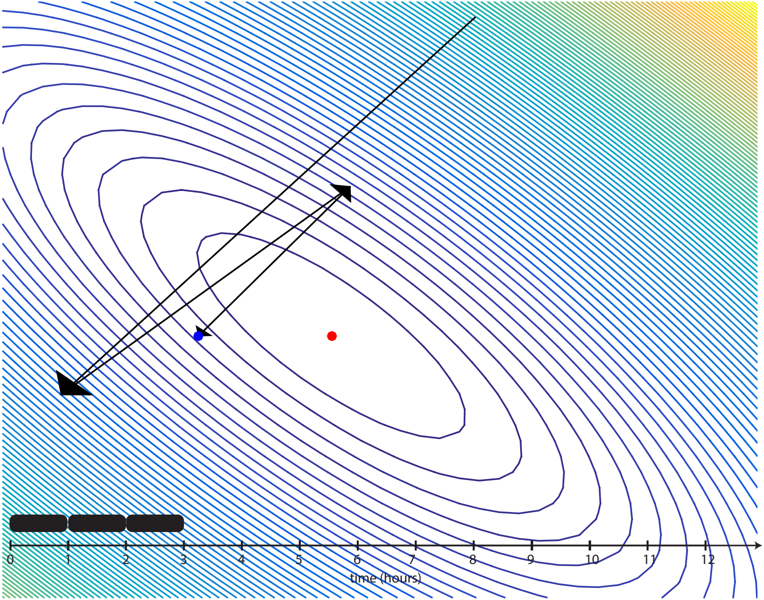
Gradient Descent



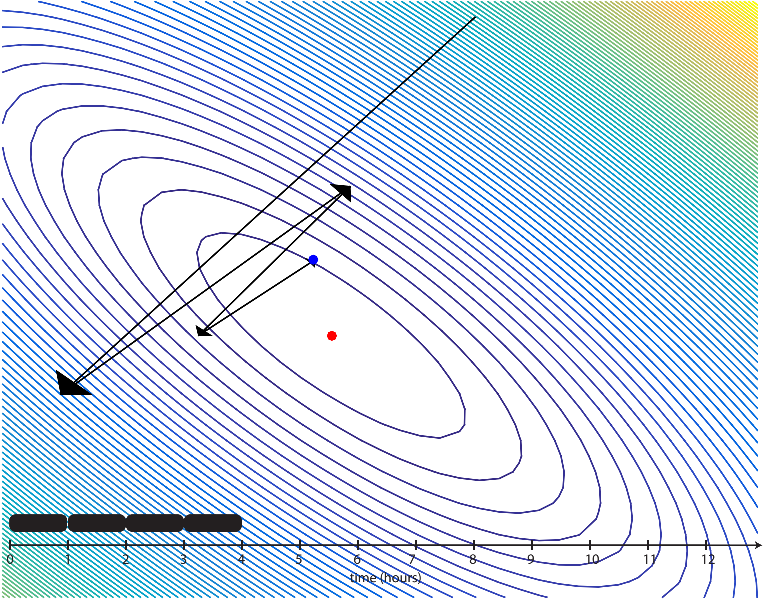
Gradient Descent



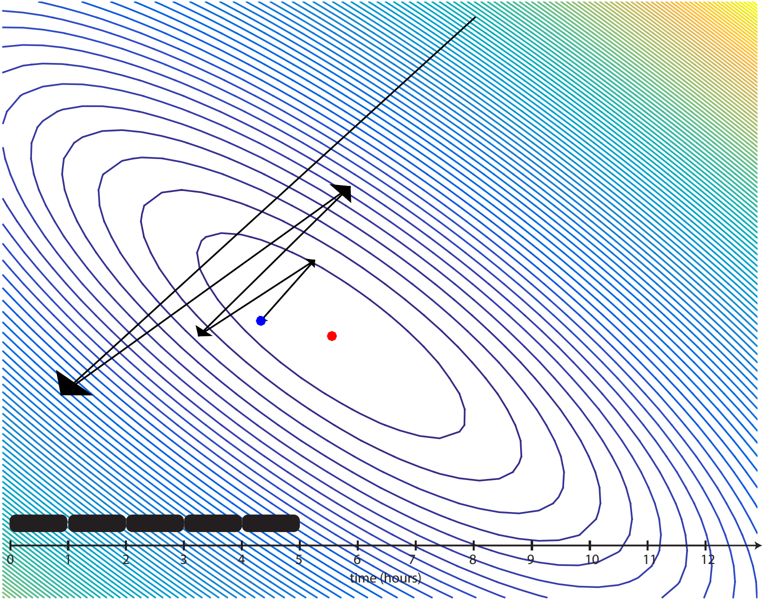
Gradient Descent



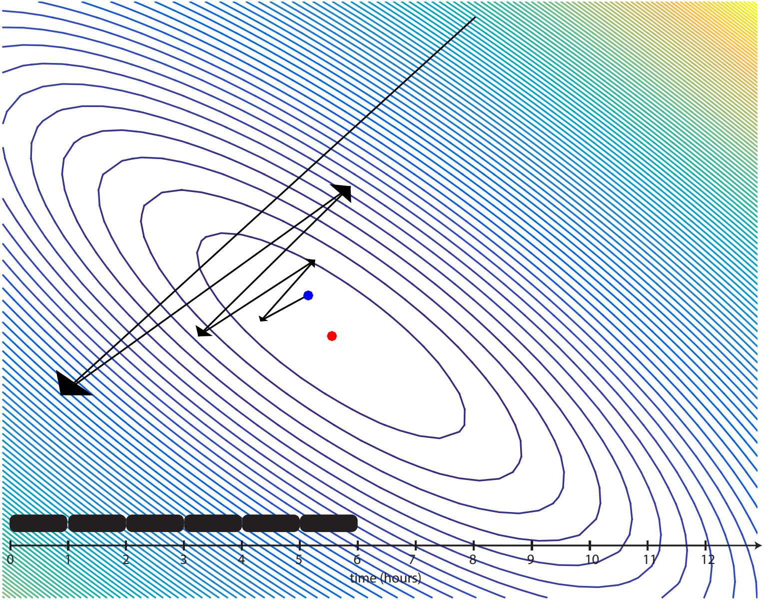
Gradient Descent



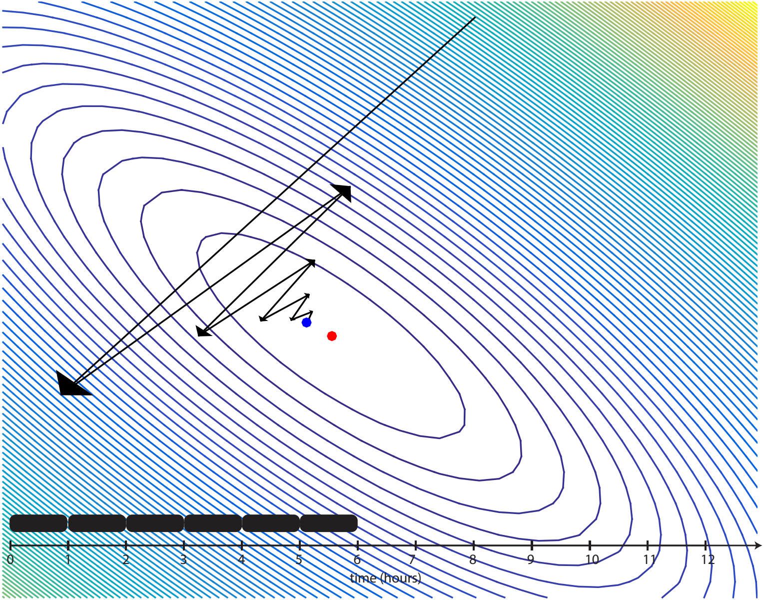
Gradient Descent



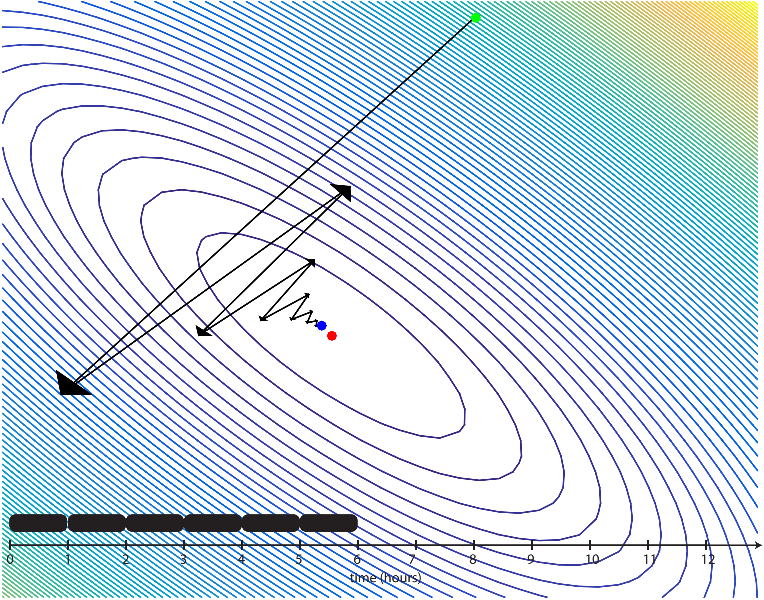
Gradient Descent



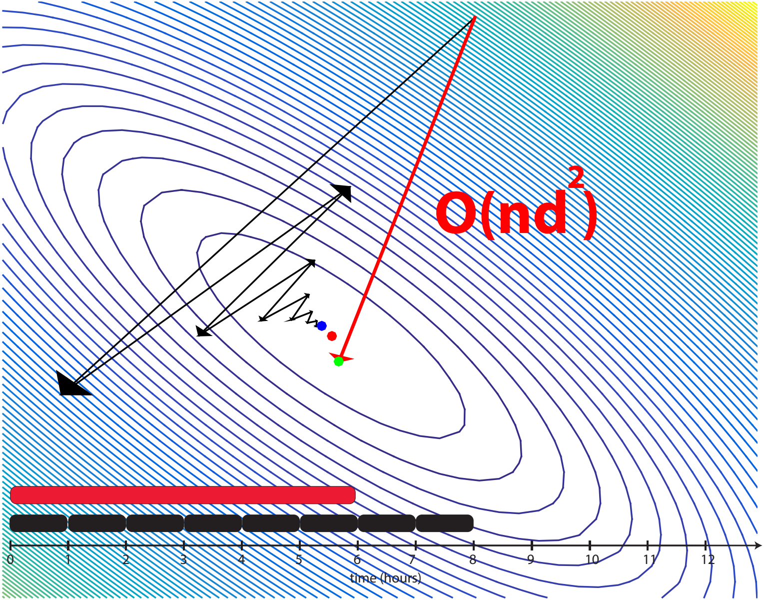
Gradient Descent vs



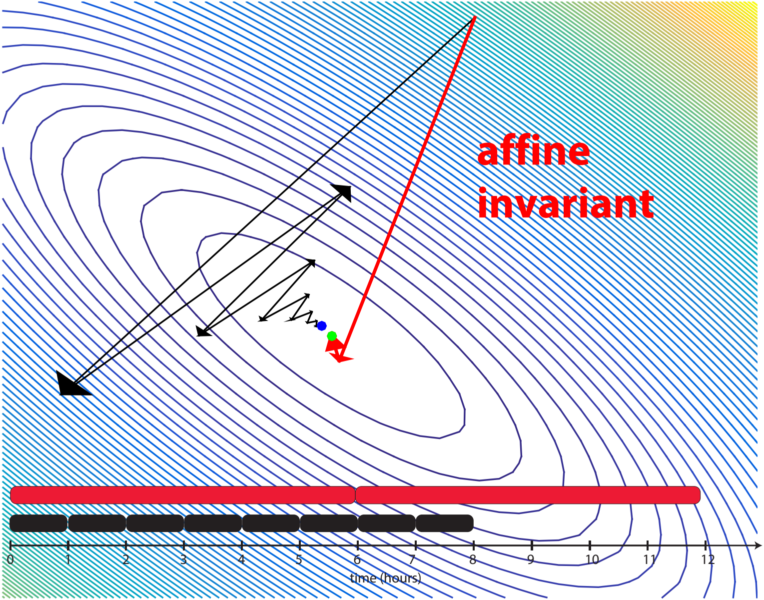
Gradient Descent vs Newton's Method



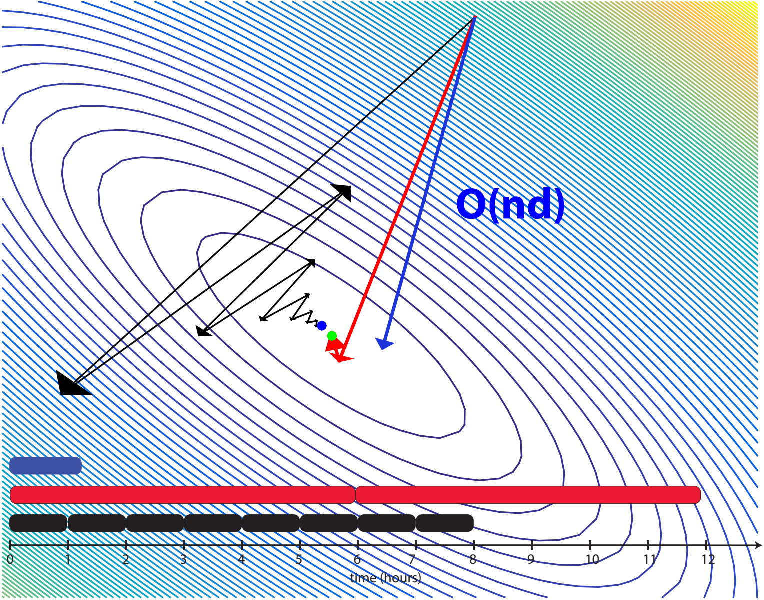
Gradient Descent vs Newton's Method



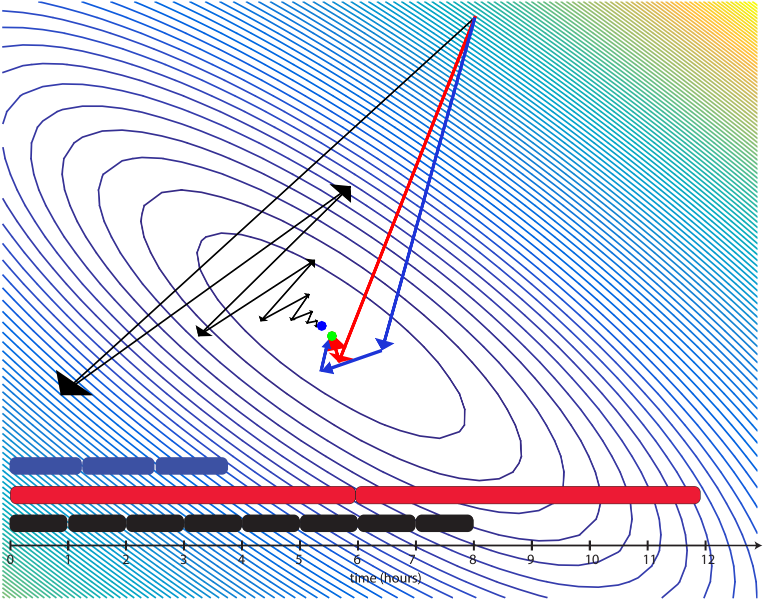
Gradient Descent vs Newton's Method



Gradient Descent vs Newton's Method



Gradient Descent vs Newton's Method



Introducing Newton Sketch

- Newton's Method

$$x^{t+1} = \arg\min_{x \in \mathcal{C}} \; \langle \nabla g(x^t), \, x - x^t \rangle + \frac{1}{2} \| \nabla^2 g(x^t)^{1/2} (x - x^t) \|^2$$

Introducing Newton Sketch

- Newton's Method

$$x^{t+1} = \arg\min_{x \in \mathcal{C}} \ \langle \nabla g(x^t),\, x - x^t \rangle + \frac{1}{2} \|\nabla^2 g(x^t)^{1/2}(x - x^t)\|^2$$

Definition (Newton Sketch)

$$x^{t+1} = \arg\min_{x \in \mathcal{C}} \ \langle \nabla g(x^t),\, x - x^t \rangle + \frac{1}{2} \|S^t \nabla^2 g(x^t)^{1/2}(x - x^t)\|^2$$

Introducing Newton Sketch

- Newton's Method

$$x^{t+1} = \arg\min_{x \in \mathcal{C}} \ \langle \nabla g(x^t),\, x - x^t \rangle + \frac{1}{2} \|\nabla^2 g(x^t)^{1/2}(x - x^t)\|^2$$

Definition (Newton Sketch)

$$x^{t+1} = \arg\min_{x \in \mathcal{C}} \ \langle \nabla g(x^t),\, x - x^t \rangle + \frac{1}{2} \|S^t \nabla^2 g(x^t)^{1/2}(x - x^t)\|^2$$

- Iterative Sketch is a special case $g(x) = \|Ax - y\|^2$

Convergence of Newton Sketch

Theorem

Newton Sketch is **affine invariant** *in distribution and the number of iterations for* ϵ *accuracy is less than*

$$C \log(1/\epsilon)$$

C *is a constant* **independent** *of* f *& data (same assumptions).*

[**P**. and Wainwright. SIAM Journal on Optimization, 2017]

Convergence of Newton Sketch

Theorem

Newton Sketch is **affine invariant** *in distribution and the number of iterations for ϵ accuracy is less than*

$$C \log(1/\epsilon)$$

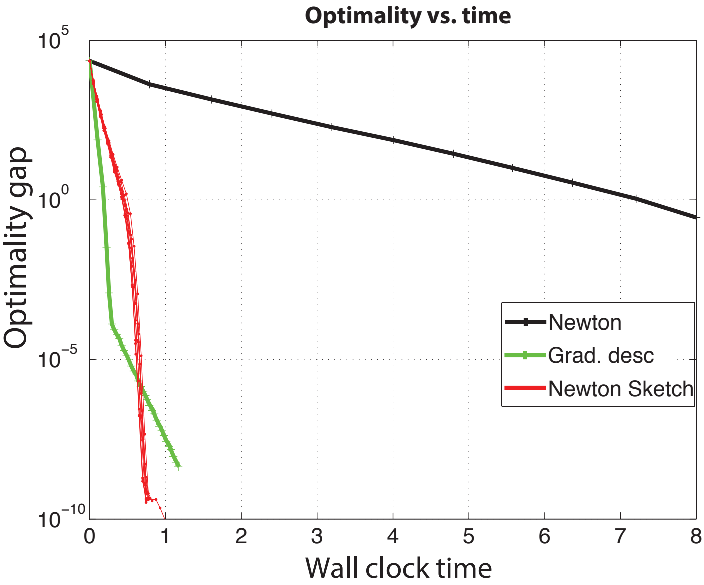
C *is a constant* **independent** *of f & data (same assumptions).*

[**P**. and Wainwright. SIAM Journal on Optimization, 2017]

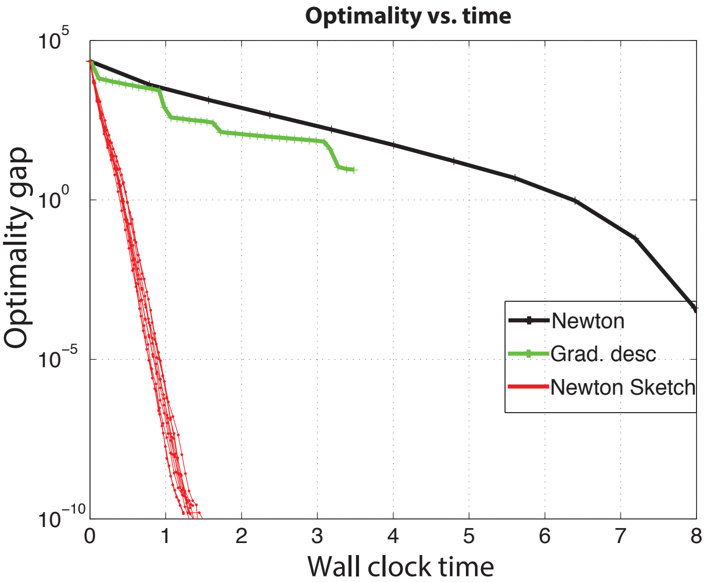
	computation
Gradient Descent	$O(\kappa\, nd \log(1/\epsilon))$
Newton's Method	$O(nd^2 \log\log(1/\epsilon))$
Newton Sketch	$O(nd \log(1/\epsilon))$

Dependence on curvature κ is unavoidable among first order methods [Nesterov, 04]

Logistic Regression ($n = 500,000$, $d = 5,000$ uncorrelated)

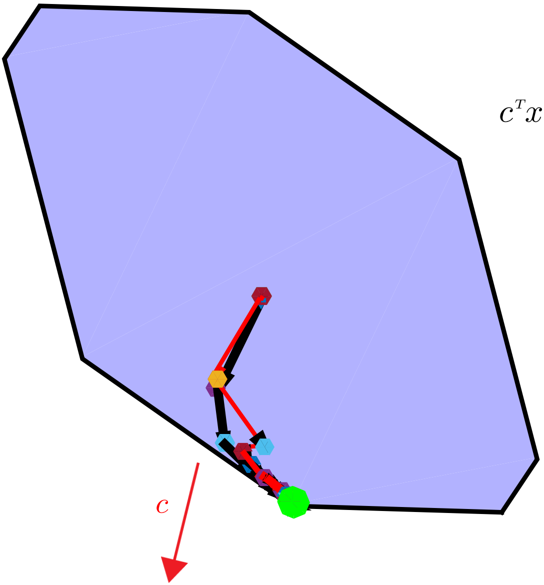


Logistic Regression ($n = 500,000$, $d = 5,000$ correlation 0.1)



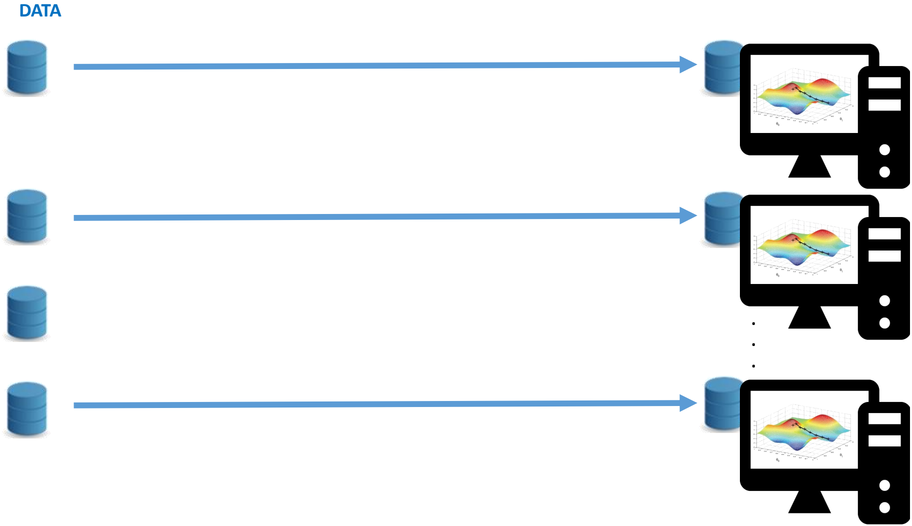
$$\min_{Ax \leq b} \; c^T x$$

$$c^T x - \mu \sum_{i=1}^{n} \log(b_i - a_i^T x)$$

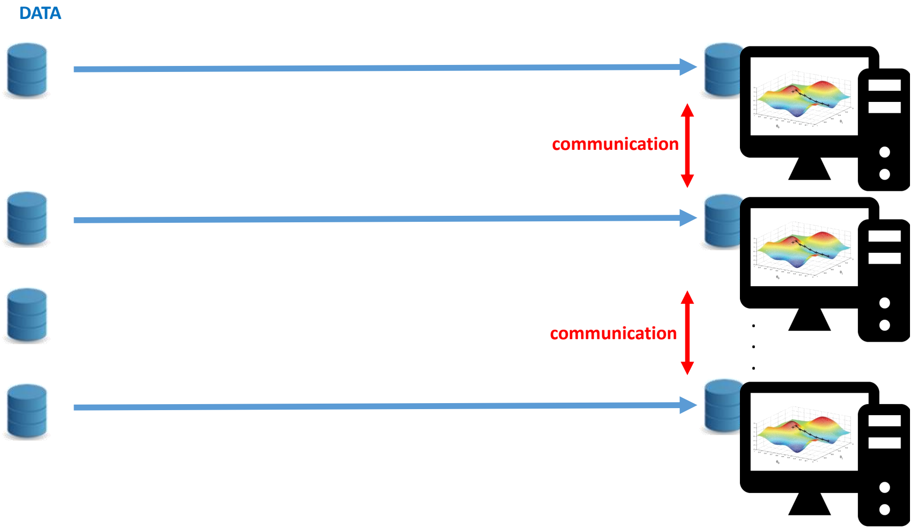


Distributed Optimization

Distributed Optimization



Distributed Optimization



Distributed optimization

- $A : n \times d$ feature matrix, and $y : n \times 1$ response vector
- Partition data $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \;\; y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
- Least squares cost

$$\min_x \|Ax - y\|^2 = \min_x \|A_1 x - y_1\|^2 + \|A_2 x - y_2\|^2$$

Distributed optimization

- A : $n \times d$ feature matrix, and y : $n \times 1$ response vector
- Partition data $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
- Least squares cost

$$\min_x \|Ax - y\|^2 = \min_x \|A_1 x - y_1\|^2 + \|A_2 x - y_2\|^2$$

Alternating Directions Method of Multipliers (ADMM)

(Hestenes, Powell 1969, Gabay et al. 1976, Boyd et al. 2011)

$$\min_x \|Ax - y\|^2 = \min_{x_1 = x_2} \|A_1 x_1 - y_1\|^2 + \|A_2 x_2 - y_2\|^2$$

Distributed optimization

- $A : n \times d$ feature matrix, and $y : n \times 1$ response vector
- Partition data $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \;\; y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
- Least squares cost

$$\min_x \|Ax - y\|^2 = \min_x \|A_1 x - y_1\|^2 + \|A_2 x - y_2\|^2$$

Alternating Directions Method of Multipliers (ADMM)

(Hestenes, Powell 1969, Gabay et al. 1976, Boyd et al. 2011)

$$\min_x \|Ax - y\|^2 = \min_{x_1 = x_2} \|A_1 x_1 - y_1\|^2 + \|A_2 x_2 - y_2\|^2$$

$$= \min_{x_1, x_2} \|A_1 x_1 - y_1\|^2 + \|A_2 x_2 - y_2\|^2 + \lambda^T (x_1 - x_2)$$

Alternating Directions Method of Multipliers

$\rightarrow$ x_1 update on machine 1

$$\min_{x_1} \|A_1x_1 - y_1\|^2 + \underbrace{\|A_2x_2 - y_2\|^2}_{constant} + \lambda^T(x_1 - x_2) + \rho\|x_1 - x_2\|^2$$

involves only A_1, y_1, x_2

Alternating Directions Method of Multipliers

$\rightarrow$ x_1 update on machine 1

$$\min_{x_1} \|A_1x_1 - y_1\|^2 + \underbrace{\|A_2x_2 - y_2\|^2}_{constant} + \lambda^T(x_1 - x_2) + \rho\|x_1 - x_2\|^2$$

involves only A_1, y_1, x_2

$\rightarrow$ x_2 update on machine 2

$$\min_{x_2} \underbrace{\|A_1x_1 - y_1\|^2}_{constant} + \|A_2x_2 - y_2\|^2 + \lambda^T(x_1 - x_2) + \rho\|x_1 - x_2\|^2$$

involves only A_2, y_2, x_1

Alternating Directions Method of Multipliers

→ x_1 update on machine 1

$$\min_{x_1} \|A_1 x_1 - y_1\|^2 + \underbrace{\|A_2 x_2 - y_2\|^2}_{constant} + \lambda^T(x_1 - x_2) + \rho\|x_1 - x_2\|^2$$

involves only A_1, y_1, x_2

→ x_2 update on machine 2

$$\min_{x_2} \underbrace{\|A_1 x_1 - y_1\|^2}_{constant} + \|A_2 x_2 - y_2\|^2 + \lambda^T(x_1 - x_2) + \rho\|x_1 - x_2\|^2$$

involves only A_2, y_2, x_1

→ communicate $x_1 \Longleftrightarrow x_2$, update $\lambda \leftarrow \lambda + \rho(x_1 - x_2)$

Alternating Directions Method of Multipliers

- (Informal) Under some assumptions, ADMM converges in $O(\kappa_A \log(1/\epsilon))$ iterations, where κ_A is a conditioning parameter
- $\#$ iterations $=$ rounds of communication

Distributed Sketching and ADMM

A_1 and A_2 are $n \times d$

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

Define multiple sketching matrices

$$S_1 = \begin{bmatrix} I_{n\times n}, \ 0_{n\times n} \end{bmatrix}, \quad S_2 = \begin{bmatrix} 0_{n\times n}, \ I_{n\times n} \end{bmatrix}$$

Distributed Sketching and ADMM

A_1 and A_2 are $n \times d$

$$A = \left[\begin{array}{c} A_1 \\ A_2 \end{array} \right]$$

Define multiple sketching matrices

$$S_1 = \left[\begin{array}{cc} I_{n\times n}, & 0_{n\times n} \end{array} \right], \quad S_2 = \left[\begin{array}{cc} 0_{n\times n}, & I_{n\times n} \end{array} \right]$$

$$S_1 A = A_1$$

$$S_2 A = A_2$$

Distributed Sketching and ADMM

A_1 and A_2 are $n \times d$

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

Define multiple sketching matrices

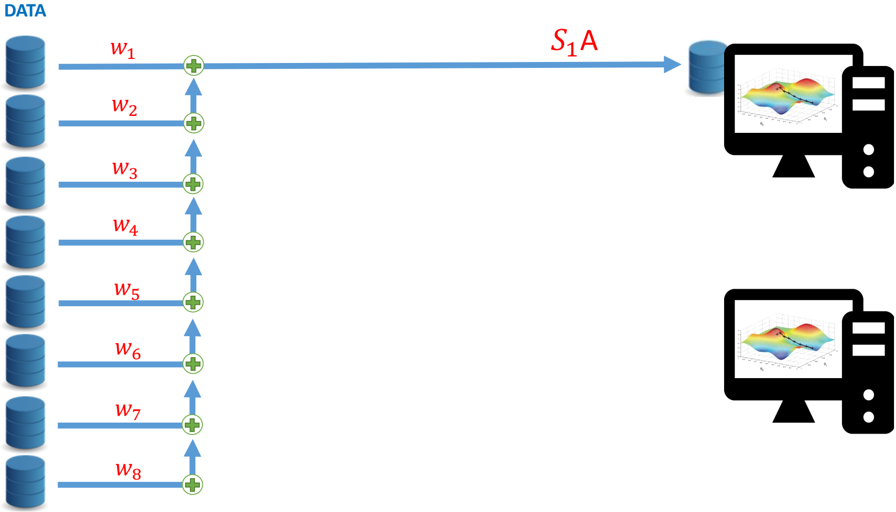
$$S_1 = \begin{bmatrix} I_{n \times n}, & 0_{n \times n} \end{bmatrix}, \quad S_2 = \begin{bmatrix} 0_{n \times n}, & I_{n \times n} \end{bmatrix}$$

$$S_1 A = A_1$$

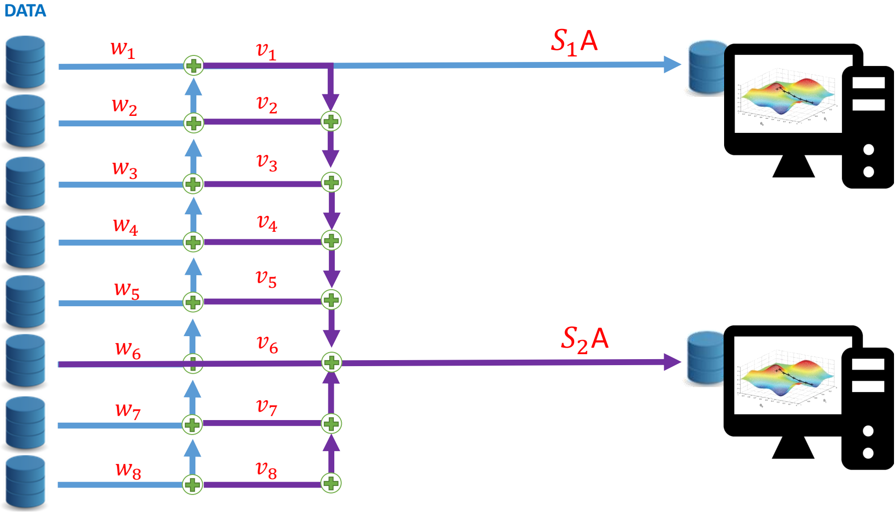
$$S_2 A = A_2$$

ADMM is operating on (naive) sketches!

Distributed Sketching



Distributed Sketching



Let S be a random orthogonal matrix, $S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$, $S^T S = I$

e.g., DFT matrix with randomly permuted rows

Randomized Direction Method of Multipliers

Let S be a random orthogonal matrix, $S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$, $S^T S = I$

e.g., DFT matrix with randomly permuted rows

$\rightarrow$ x_1 update on machine 1

$$\min_{x_1} \|S_1(Ax_1 - y)\|^2 + \underbrace{\|S_2(Ax_2 - y)\|^2}_{constant} + \lambda^\top (x_1 - x_2) + \frac{\rho}{2}\|A(x_1 - x_2)\|^2$$

$\rightarrow$ x_2 update on machine 2

$$\min_{x_2} \underbrace{\|S_1(Ax_1 - y)\|^2}_{constant} + \|S_2(Ax_2 - y)\|^2 + \lambda^\top (x_1 - x_2) + \frac{\rho}{2}\|A(x_1 - x_2)\|^2$$

$\rightarrow$ communicate $\quad x_1 \Longleftrightarrow x_2$ and update λ

Randomized Direction Method of Multipliers

Let S be a random orthogonal matrix, $S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$, $S^T S = I$

e.g., DFT matrix with randomly permuted rows

$\rightarrow$ x_1 update on machine 1

$$\min_{x_1} \|S_1(Ax_1 - y)\|^2 + \underbrace{\|S_2(Ax_2 - y)\|^2}_{constant} + \lambda^\top(x_1 - x_2) + \frac{1}{2}\|A(x_1 - x_2)\|^2$$

$\rightarrow$ x_2 update on machine 2

$$\min_{x_2} \underbrace{\|S_2(Ax_2 - y)\|^2}_{constant} + \|S_2(Ax_2 - y)\|^2 + \lambda^\top(x_1 - x_2) + \frac{1}{2}\|A(x_1 - x_2)\|^2$$

$\rightarrow$ communicate $x_1 \Longleftrightarrow x_2$ and update λ

Randomized Direction Method of Multipliers

Theorem

Local solutions converge, and number of iterations for ϵ accuracy is less than

$$C \log(1/\epsilon)\,,$$

where C is a constant **independent** *of data.*

[**P**. and Candès, 2018]

Randomized Direction Method of Multipliers

Theorem

Local solutions converge, and number of iterations for ϵ accuracy is less than

$$C \log(1/\epsilon)\,,$$

where C is a constant **independent** *of data.*

[**P**. and Candès, 2018]

- $O(\log(1/\epsilon))$ rounds of communication. No condition number dependency.

Randomized Direction Method of Multipliers

Theorem

Local solutions converge, and number of iterations for ϵ accuracy is less than

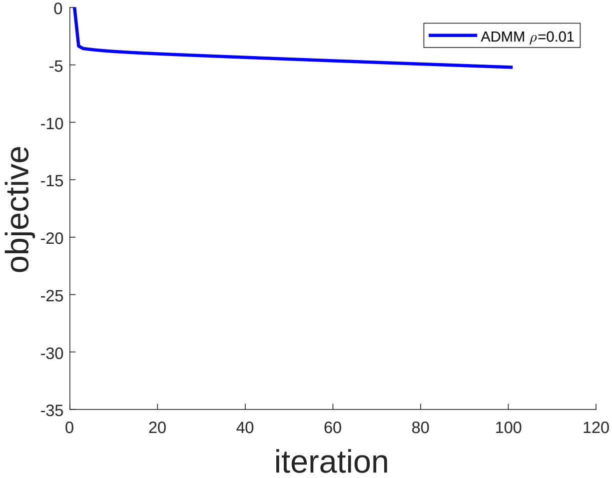
$$C \log(1/\epsilon)\,,$$

where C is a constant **independent** *of data.*

[**P**. and Candès, 2018]

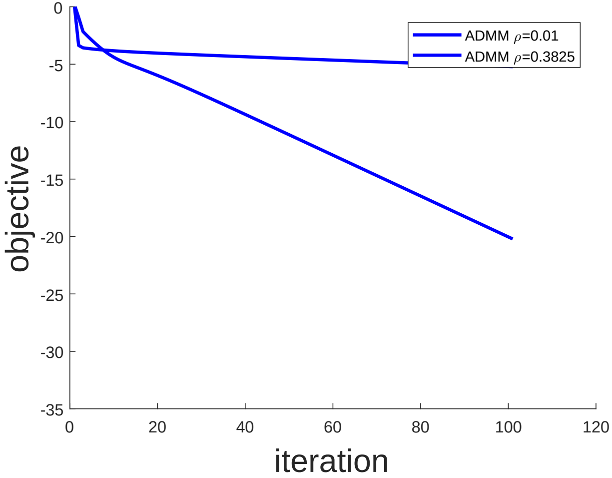
- $O(\log(1/\epsilon))$ rounds of communication. No condition number dependency.
- Exchanging $O(d \log(1/\epsilon))$ bits to communicate $x_1, x_2, ..., x_M$ is information theoretically **optimal**

Random i.i.d. heavy tailed data



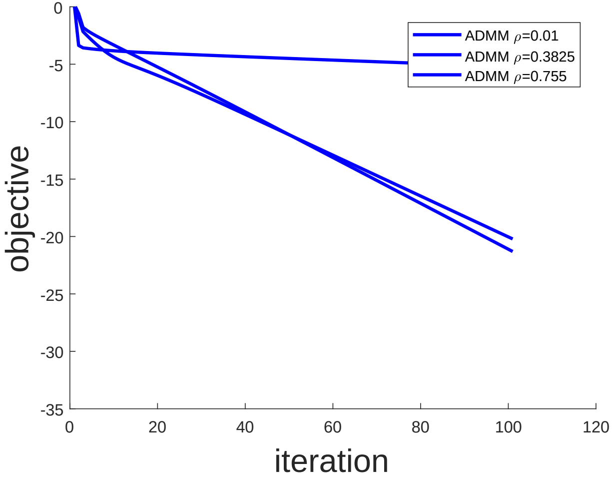
Randomized Direction Method of Multipliers

Random i.i.d. data

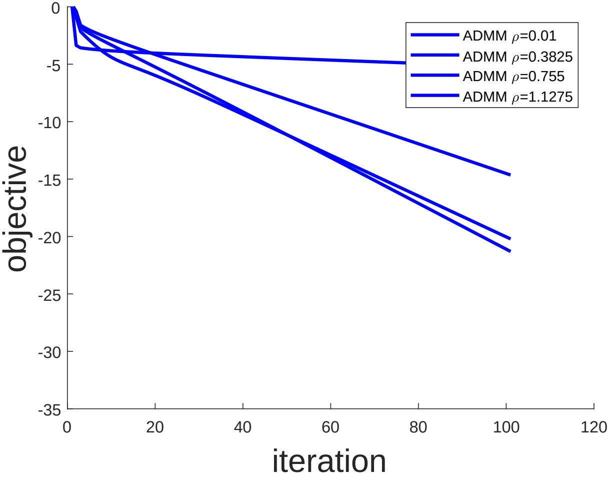


Randomized Direction Method of Multipliers

Random i.i.d. data

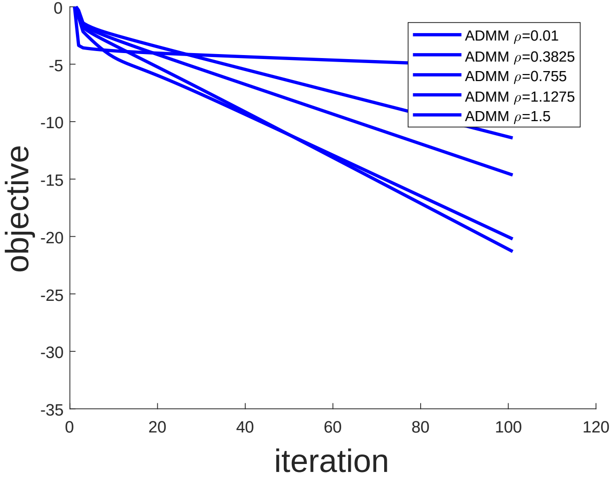


Random i.i.d. data

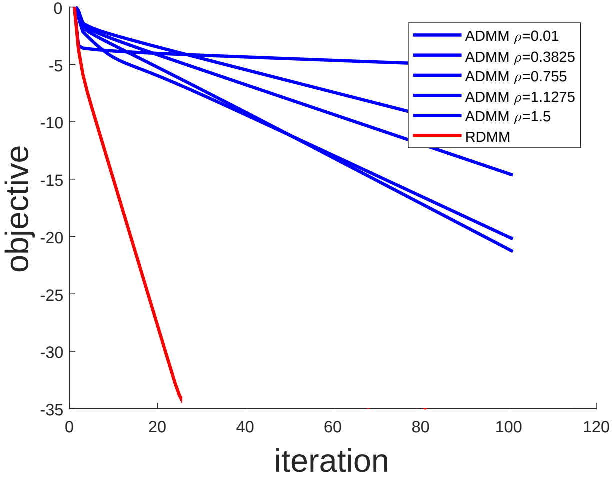


Randomized Direction Method of Multipliers

Random i.i.d. data

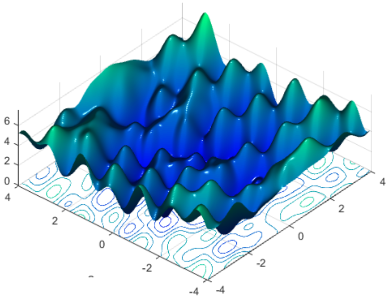


Random i.i.d. data



Non-convex Optimization Problems

- In general, very difficult to solve globally
- Need to make further assumptions



Non-convex Optimization Problems

$$\min_{x} \sum_{i=1}^{n} (f_x(a_i) - y_i)^2$$

Non-convex Optimization Problems

$$\min_x \sum_{i=1}^{n} (f_x(a_i) - y_i)^2$$

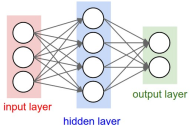
$\rightarrow$ Heuristic: Gauss-Newton method

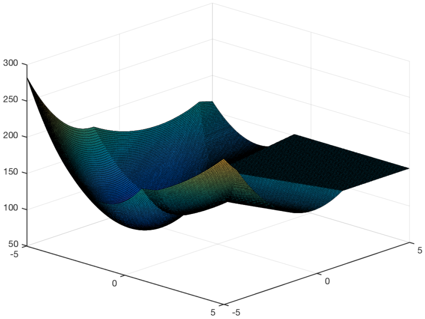
$$x_{t+1} = \arg\min_x \quad \| \underbrace{f_{x_t}(A) + J_t x}_{\text{Taylor's approx for } f_x} - y\|_2^2$$

where $(J_t)_{ij} = \frac{\partial}{\partial x_j} f_x(a_i)$ is the Jacobian matrix

Non-convex Optimization Problems

Deep learning, nonlinear least squares...





2 layer ReLU neural network training loss

$\rightarrow$ Randomized Gauss-Newton method

$$x_{t+1} = \arg\min_{x} \quad \|S_t(f_{x_t}(A) + J_t x - y)\|_2^2.$$

- $S_t J_t$ backpropagation (Pearlmutter, 1994)

→ Randomized Gauss-Newton method

$$x_{t+1} = \arg\min_{x} \quad \|S_t(f_{x_t}(A) + J_t x - y)\|_2^2.$$

- $S_t J_t$ backpropagation (Pearlmutter, 1994)

Theorem

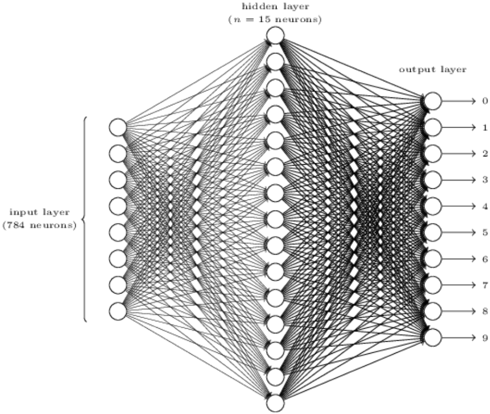
(informal) Consider a **single hidden layer** *neural network,* **Gaussian** *input data A. Randomized Gauss-Newton method converges to a* **global minimum** *in*

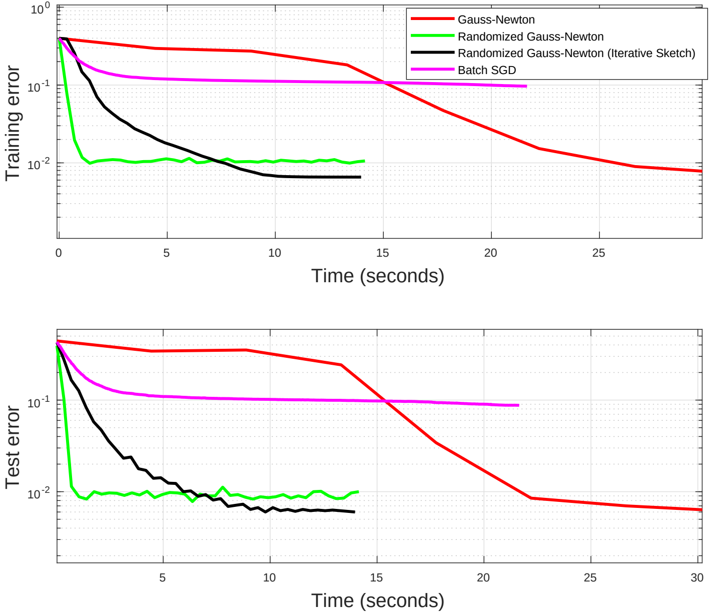
$$C \log(1/\epsilon)\,,$$

iterations.

[**P**. and Candès, 2018]

2 layer neural network on MNIST dataset





Gauss-Newton
Randomized Gauss-Newton
Randomized Gauss-Newton (Iterative Sketch)
Batch SGD
Training error
Time (seconds)
Test error
Time (seconds)

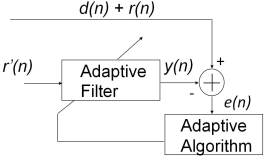
Summary

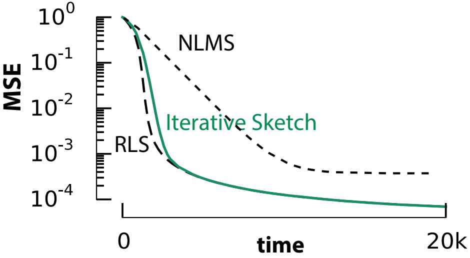
1. Information theoretic lower bounds for sketching
2. Iterative sketching with statistical optimality
3. Distributed sketching
4. Non-convex problems

Extensions: Streaming optimization

- Sketch can be updated by $[S \;\; s_+] \begin{bmatrix} A \\ a_+^T \end{bmatrix} = SA + s_+ a_+^T$

Example: Adaptive filtering



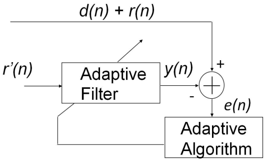


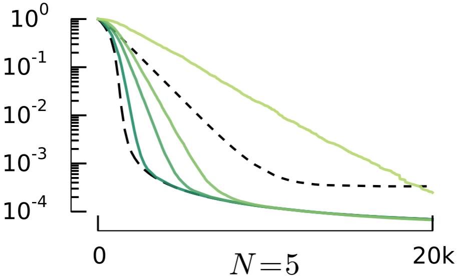
[Scheibler and Vetterli, 16]

Extensions: Streaming optimization

- Sketch can be updated by $[S \;\; s_+] \begin{bmatrix} A \\ a_+^T \end{bmatrix} = SA + s_+ a_+^T$

Example: Adaptive filtering





[Scheibler and Vetterli, 16]

Extensions

- Privacy preserving optimization

Extensions

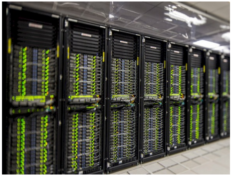
- Privacy preserving optimization

- SA provides privacy
- Mutual information constraint $I(SA; A) \leq \epsilon$

Yang, **P.**, Wainwright, Annals of Statistics, 2015, P. (book chapter) 2018,

Extensions

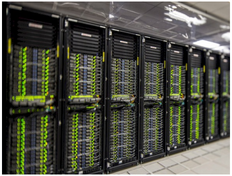
- Distributed and fault tolerant computing



- $S_1 A$, $S_2 A$ ..., $S_m A$ can be lost due to point failures

Extensions

- Distributed and fault tolerant computing



- $S_1 A$, $S_2 A$ ..., $S_m A$ can be lost due to point failures
- Generate another sketch $S_{m+1} A$ i.i.d.

Thank you!

Questions ?