Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 6: Time-data tradeoffs and variance reduction

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-556 (Fall 2020)

License Information for Mathematics of Data Slides

- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.
- Full Text of the License

Outline

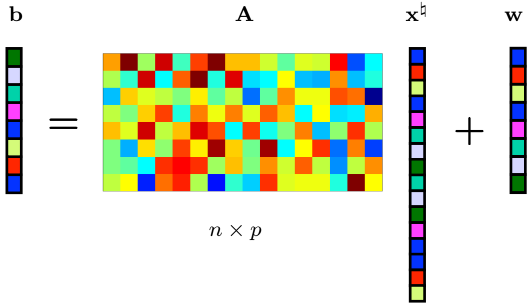
- This class
  1. Time-data trade-offs
  2. Rate iteration-cost trade-offs
  3. Variance reduction
- Next class
  1. Deep learning introduction

A simple *regression* model

$$\boxed{b_i = h_{\mathbf{x}^\natural}(\mathbf{a}_i)}$$

$\mathbf{x}^\natural$: unknown function parameters
$\mathbf{a}_i$: input
$\mathbf{b}_i$: response / output

Linear model:



$$\boxed{\mathbf{b}_i = \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle + \mathbf{w}_i}$$

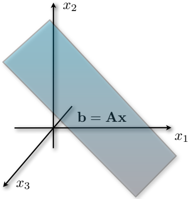
Applications: **Compressive sensing, machine learning, theoretical computer science...**

A simple *regression* model and many *practical* questions

$$\boxed{\mathbf{b}_i = \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle + \mathbf{w}_i}$$

$\mathbf{x}^\natural$: unknown function parameters
$\mathbf{a}_i$: input
$\mathbf{b}_i$: response / output
$\mathbf{w}_i$: perturbations / noise

- ○ Estimation: find $\mathbf{x}^\star$ to minimize $\|\mathbf{x}^\star - \mathbf{x}^\natural\|$
- ○ Prediction: find $\mathbf{x}^\star$ to minimize $L\left(\langle \mathbf{a}_i, \mathbf{x}^\star \rangle, \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle\right)$
- ○ Decision: choose $\mathbf{a}_i$ for estimation or prediction



A difficult estimation challenge when $n < p$:

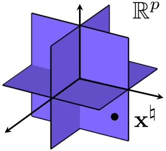
Nullspace (null) of $\mathbf{A}$: $\mathbf{x}^\natural + \boldsymbol{v} \rightarrow \mathbf{b}, \quad \forall \boldsymbol{v} \in \text{null}(\mathbf{A})$

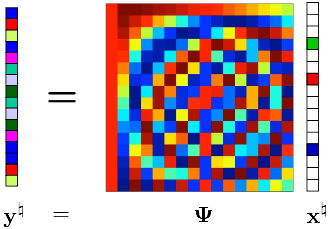
- ○ **Needle in a haystack:** ***We need additional information on*** $\mathbf{x}^\natural$***!***

A natural signal model

Definition (s-sparse vector)

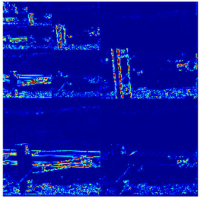
A vector $\mathbf{x} \in \mathbb{R}^p$ is s-sparse if it has at most s non-zero entries.



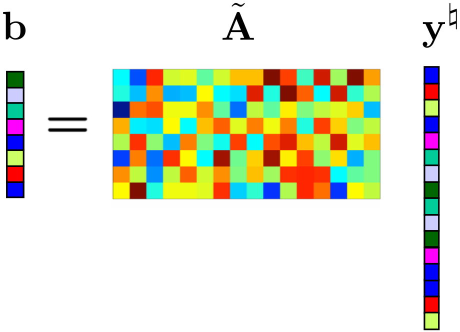


Sparse representations

- $\mathbf{x}^\natural$: *sparse* transform coefficients
- Basis representations $\Psi \in \mathbb{R}^{p \times p}$
  - *Wavelets*, DCT, ...
- Frame representations $\Psi \in \mathbb{R}^{m \times p}$, $m > p$
  - Gabor, curvelets, shearlets, ...
- Other *dictionary* representations...

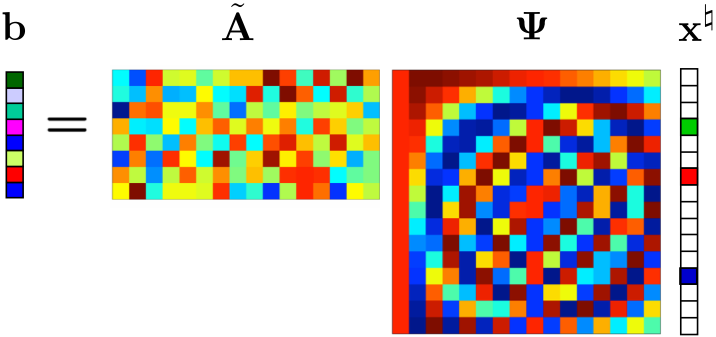


Sparse representations strike back!



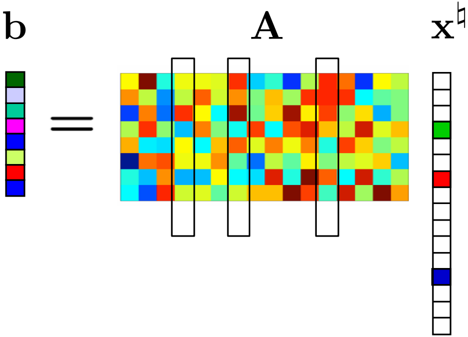
- ○ $\mathbf{b} \in \mathbb{R}^n$, $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times p}$, and $n < p$

Sparse representations strike back!



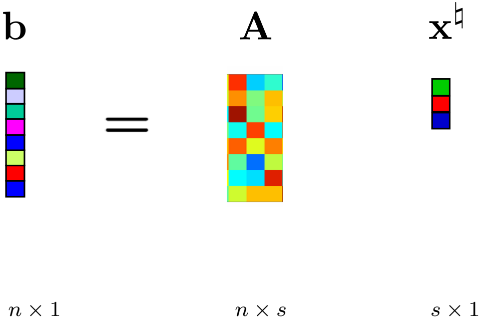
- $\mathbf{b} \in \mathbb{R}^n$, $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times p}$, and $n < p$
- $\boldsymbol{\Psi} \in \mathbb{R}^{p \times p}$, $\mathbf{x}^\natural \in \mathbb{R}^p$, and $\|\mathbf{x}^\natural\|_0 \leq s < n$

Sparse representations strike back!



- $\circ$ $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times p}$, and $\mathbf{x}^\natural \in \mathbb{R}^p$, and $\|\mathbf{x}^\natural\|_0 \leq s < n < p$

Sparse representations strike back!



Observations:

- The matrix $\mathbf{A}$ effectively becomes *overcomplete*.
- We could solve for $\mathbf{x}^\natural$ if we knew *the location of the non-zero entries of $\mathbf{x}^\natural$*.

Enter sparsity

A combinatorial approach for estimating $\mathbf{x}^\natural$ from $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$

We may consider the estimator with the least number of non-zero entries. That is,

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \|\mathbf{x}\|_0 : \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \kappa \right\} \qquad (\mathcal{P}_0)$$

with some $\kappa \geq 0$. If $\kappa = \|\mathbf{w}\|_2$, then $\mathbf{x}^\natural$ is a feasible solution.

Enter sparsity

A combinatorial approach for estimating $\mathbf{x}^\natural$ from $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$

We may consider the estimator with the least number of non-zero entries. That is,

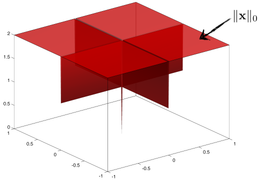
$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \|\mathbf{x}\|_0 : \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \kappa \right\} \qquad (\mathcal{P}_0)$$

with some $\kappa \geq 0$. If $\kappa = \|\mathbf{w}\|_2$, then $\mathbf{x}^\natural$ is a feasible solution.

○ **$\mathcal{P}_0$ has the following characteristics:**

- sample complexity: $\mathcal{O}(s)$
- computational effort: NP-Hard
- stability: No

$\|\mathbf{x}\|_0$ over the unit ℓ_∞-ball



Enter sparsity

A combinatorial approach for estimating $\mathbf{x}^\natural$ from $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$

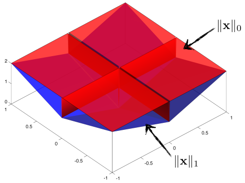
We may consider the estimator with the least number of non-zero entries. That is,

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{\|\mathbf{x}\|_0 : \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \kappa\right\} \qquad (\mathcal{P}_0)$$

with some $\kappa \geq 0$. If $\kappa = \|\mathbf{w}\|_2$, then $\mathbf{x}^\natural$ is a feasible solution.

- **$\mathcal{P}_0$ has the following characteristics:**
  - sample complexity: $\mathcal{O}(s)$
  - computational effort: NP-Hard
  - stability: No
- **Tightest convex relaxation:**
  - $\|\mathbf{x}\|_0^{**}$ is the biconjugate
  - i.e., Fenchel conjugate of Fenchel conjugate
- **Fenchel conjugate:**
  - $f^*(\mathbf{y}) := \sup_{\mathbf{x}:\text{dom}(f)} \mathbf{x}^T\mathbf{y} - f(\mathbf{x})$.

$\|\mathbf{x}\|_1$ is the convex envelope of $\|\mathbf{x}\|_0$



A technicality: Restrict $\mathbf{x}^\natural \in [-1, 1]^p$.

The role of convexity

A convex candidate solution for $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \|\mathbf{x}\|_1 : \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \le \|\mathbf{w}\|_2, \|\mathbf{x}\|_\infty \le 1 \right\}. \qquad \text{(SOCP)}$$

Theorem (A **model** recovery guarantee [11])

Let $\mathbf{A} \in \mathbb{R}^{n\times p}$ *be a matrix of i.i.d. Gaussian random variables with zero mean and variances* $1/n$. *For any* $t > 0$ *with probability at least* $1 - 6\exp\left(-t^2/26\right)$, *we have*

$$\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2 \le \left[\frac{2\sqrt{2s\log(\frac{p}{s}) + \frac{5}{4}s}}{\sqrt{n} - \sqrt{2s\log(\frac{p}{s}) + \frac{5}{4}s} - t}\right] \|\mathbf{w}\|_2 := \varepsilon, \quad \textit{when } \|\mathbf{x}^\natural\|_0 \le s.$$

Observations:

○ perfect recovery (i.e., $\varepsilon = 0$) with $n \ge 2s\log(\frac{p}{s}) + \frac{5}{4}s$ whp when $\mathbf{w} = 0$.

○ ϵ-accurate solution in $k = \mathcal{O}\left(\sqrt{2p+1}\log(\frac{1}{\epsilon})\right)$ iterations via IPM with a total complexity of $\mathcal{O}(n^2p^{1.5}\log(\frac{1}{\epsilon}))$ with each iteration requiring the solution of a structured $n \times 2p$ linear system.

○ robust to noise.

A Time-Data conundrum — I

A computational dogma

Running time of a learning algorithm increases with the size of the data.

A Time-Data conundrum — I

A computational dogma

Running time of a learning algorithm increases with the size of the data.

- Misaligned goals in the statistical and optimization disciplines

Discipline	Goal	Metric
Optimization	reaching numerical ϵ-accuracy	$\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \epsilon$
Statistics	learning ε-accurate model	$\|\mathbf{x}^\star - \mathbf{x}^\natural\| \leq \varepsilon$

- Main issue: **ϵ and ε are NOT the same but should be treated jointly!**

A Time-Data conundrum — II

A stylized formalization of the time-data tradeoff

The goals of optimization and statistical modeling are tightly connected:

$$\underbrace{\|\mathbf{x}^k - \mathbf{x}^\natural\|}_{\text{learning quality}} \leq \underbrace{\|\mathbf{x}^k - \mathbf{x}^\star\|}_{\epsilon\text{: needs “time” } t(k)} + \underbrace{\|\mathbf{x}^\star - \mathbf{x}^\natural\|}_{\varepsilon\text{: needs “data”} n},$$

$\mathbf{x}^\natural$: true model in $\mathbb{R}^p$

$\mathbf{x}^\star$: statistical model estimate

$\mathbf{x}^k$: numerical solution at iteration k

○ As the number of data samples n increases with a fixed optimization formulation,

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \|\mathbf{x}\|_1 : \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \|\mathbf{w}\|_2, \|\mathbf{x}\|_\infty \leq 1 \right\}$$

- numerical methods take longer time t to reach ϵ-accuracy
  - e.g., per-iteration time to solve an $n \times 2p$ linear system
- statistical model estimates ε become more precise when $\|\mathbf{w}\|_2 = \mathcal{O}(\sqrt{n})$
  - $\varepsilon = \dfrac{2\sqrt{2s\log(\frac{p}{s}) + \frac{5}{4}s}}{\sqrt{n} - \sqrt{2s\log(\frac{p}{s}) + \frac{5}{4}s} - t}\|\mathbf{w}\|_2$, with probability $1 - 6\exp(-t^2/26)$.

A Time-Data conundrum — II

A stylized formalization of the time-data tradeoff

The goals of optimization and statistical modeling are tightly connected:

$$\underbrace{\left\|\mathbf{x}^k - \mathbf{x}^\natural\right\|}_{\leq \bar{\varepsilon}(t(k),n)} \leq \underbrace{\left\|\mathbf{x}^k - \mathbf{x}^\star\right\|}_{\epsilon\text{: needs "time" } t(k)} + \underbrace{\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|}_{\varepsilon\text{: needs "data"} n},$$

$\mathbf{x}^\natural$:	true model in $\mathbb{R}^p$
$\mathbf{x}^\star$:	statistical model estimate
$\mathbf{x}^k$:	numerical solution at iteration k
$\bar{\varepsilon}(t(k), n)$:	actual learning quality at time $t(k)$ with n samples

○ As the number of data samples n increases with a fixed optimization formulation,

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \|\mathbf{x}\|_1 : \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \|\mathbf{w}\|_2, \|\mathbf{x}\|_\infty \leq 1 \right\}$$

- numerical methods take longer time t to reach ϵ-accuracy
  - e.g., per-iteration time to solve an $n \times 2p$ linear system
- statistical model estimates ε become more precise when $\|\mathbf{w}\|_2 = \mathcal{O}(\sqrt{n})$
  - $\varepsilon = \dfrac{2\sqrt{2s\log(\frac{p}{s})+\frac{5}{4}s}}{\sqrt{n}-\sqrt{2s\log(\frac{p}{s})+\frac{5}{4}s}-t}\|\mathbf{w}\|_2$, with probability $1 - 6\exp(-t^2/26)$.

"Time" effort has significant diminishing returns on ε in the underdetermined case* (cf., [6, 3, 12, 5, 4])

* "Data" effort also exhibits a similar behavior in the overdetermined case when a signal prior is used due to noise!

Data as a computational resource

A stylized formalization of the time-data tradeoff

The goals of optimization and statistical modeling are tightly connected:

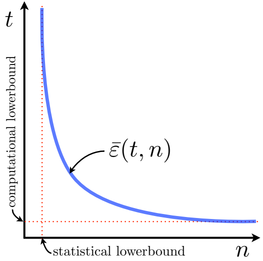
$$\underbrace{\|\mathbf{x}^{k(t)} - \mathbf{x}^\natural\|}_{\leq \bar{\varepsilon}(t,n)} \leq \underbrace{\|\mathbf{x}^{k(t)} - \mathbf{x}^\star\|}_{\epsilon\text{: needs "time" } t} + \underbrace{\|\mathbf{x}^\star - \mathbf{x}^\natural\|}_{\varepsilon\text{: needs "data"} n},$$

$\mathbf{x}^\natural$: true model in $\mathbb{R}^p$

$\bar{\varepsilon}(t,n)$: actual model precision at time t with n samples

Rest of the lecture:

- ○ estimator formulation and sample complexity
- ○ a "continuous" time-data tradeoff
- ○ a different, algorithmic tradeoff with SGD



Sample complexity analysis

Convex optimization formulation for the estimator

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) : \mathbf{b} = \mathbf{A}\mathbf{x} \right\},$$

where $f : \mathbb{R}^p \to \mathbb{R} \cup \{-\infty, \infty\}$ is a convex function.

Sample complexity

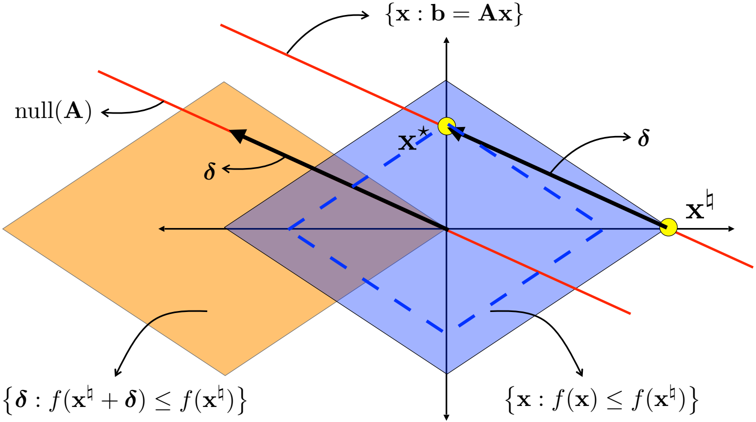
Assume that $A \in \mathbb{R}^{n\times p}$ is a matrix of independent identically distributed (i.i.d.) Gaussian random variables.

What is the minimum number of samples n such that $\mathbf{x}^\star = \mathbf{x}^\natural$ with high probability?

Characterization of the error vector

$$\boxed{\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{f(\mathbf{x}) : \mathbf{b} = \mathbf{A}\mathbf{x}\}}$$

Define the error vector $\boldsymbol{\delta} := \mathbf{x}^\star - \mathbf{x}^\natural$.

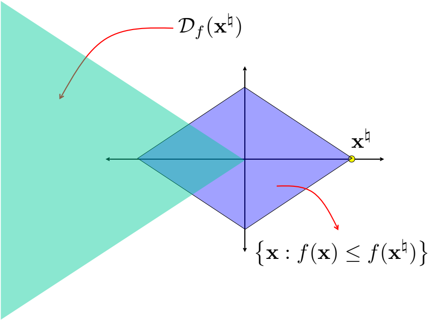

$\{\mathbf{x} : \mathbf{b} = \mathbf{A}\mathbf{x}\}$
$\mathrm{null}(\mathbf{A})$
$\mathbf{x}^\star$
$\boldsymbol{\delta}$
$\boldsymbol{\delta}$
$\mathbf{x}^\natural$
$\{\boldsymbol{\delta} : f(\mathbf{x}^\natural + \boldsymbol{\delta}) \leq f(\mathbf{x}^\natural)\}$
$\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^\natural)\}$

Descent cone

Definition (Descent cone)

Let $f : \mathbb{R}^p \to \mathbb{R} \cup \{-\infty, \infty\}$ be a proper lower-semicontinuous function. The descent cone of f at $\mathbf{x}^\natural$ is defined as

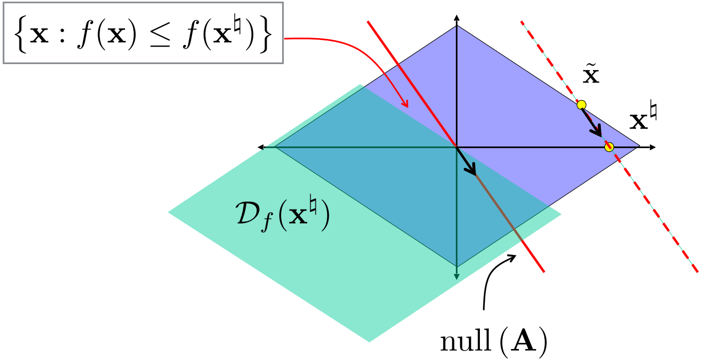
$$\mathcal{D}_f(\mathbf{x}^\natural) := \operatorname{cone}\left(\left\{\mathbf{x} : f(\mathbf{x}^\natural + \mathbf{x}) \le f(\mathbf{x}^\natural)\right\}\right).$$


$\mathcal{D}_f(\mathbf{x}^\natural)$
$\mathbf{x}^\natural$
$\{\mathbf{x} : f(\mathbf{x}) \le f(\mathbf{x}^\natural)\}$

Condition for exact recovery in the *noiseless* case

Proposition (Condition for exact recovery)

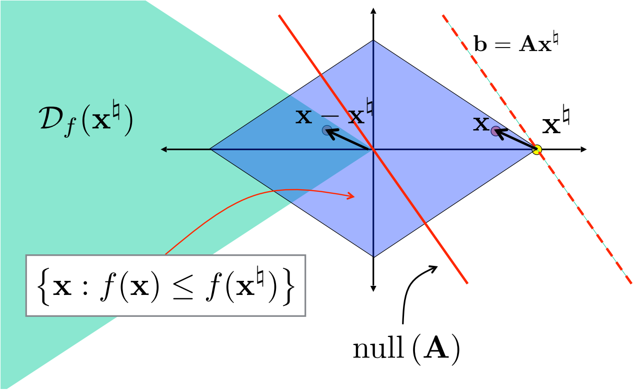
We have successful recovery, i.e., $\boldsymbol{\delta} := \mathbf{x}^\star - \mathbf{x}^\natural = 0$ *with* $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{f(\mathbf{x}) : \mathbf{b} = \mathbf{A}\mathbf{x}\}$*, if and only if* $\mathrm{null}(\mathbf{A}) \cap \mathcal{D}_f(\mathbf{x}^\natural) = \{0\}$.


$\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^\natural)\}$
$\tilde{\mathbf{x}}$
$\mathbf{x}^\natural$
$\mathcal{D}_f(\mathbf{x}^\natural)$
$\mathrm{null}(\mathbf{A})$

Condition for exact recovery in the *noiseless* case

Proposition (Condition for exact recovery)

We have successful recovery, i.e., $\boldsymbol{\delta} := \mathbf{x}^\star - \mathbf{x}^\natural = 0$ *with* $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{f(\mathbf{x}) : \mathbf{b} = \mathbf{A}\mathbf{x}\}$*, if and only if* $\mathrm{null}(\mathbf{A}) \cap \mathcal{D}_f(\mathbf{x}^\natural) = \{0\}$.


$\mathcal{D}_f(\mathbf{x}^\natural)$
$\mathbf{b} = \mathbf{A}\mathbf{x}^\natural$
$\mathbf{x} - \mathbf{x}^\natural$
$\mathbf{x}$
$\mathbf{x}^\natural$
$\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^\natural)\}$
$\mathrm{null}(\mathbf{A})$

Statistical dimension and approximate kinematic formula

Now we have

$$\mathbb{P}\left\{\mathbf{x}^\star = \mathbf{x}^\natural\right\} = \mathbb{P}\left\{\mathrm{null}(\mathbf{A}) \cap \mathcal{D}_f(\mathbf{x}^\natural) = \{0\}\right\}.$$

Definition (Statistical dimension [1][1])

Let $\mathcal{C} \subseteq \mathbb{R}^p$ be a closed convex cone. The statistical dimension of $\mathcal{C}$ is defined as

$$d(\mathcal{C}) := \mathbb{E}\left[\|\mathrm{proj}_{\mathcal{C}}(\mathbf{g})\|_2^2\right].$$

Theorem (Approximate kinematic formula [1])

Let $A \in \mathbb{R}^{n \times p}$, $n < p$, be a matrix of i.i.d. standard Gaussian random variables, and let $\mathcal{C} \subseteq \mathbb{R}^p$ be a closed convex cone. Let $\eta \in (0,1)$ Then

$$\begin{aligned} n \geq d(\mathcal{C}) + c_\eta\sqrt{p} \quad &\Rightarrow \quad \mathbb{P}\{\mathrm{null}(\mathbf{A}) \cap \mathcal{C} = \{0\}\} \geq 1-\eta; \\ n \leq d(\mathcal{C}) - c_\eta\sqrt{p} \quad &\Rightarrow \quad \mathbb{P}\{\mathrm{null}(\mathbf{A}) \cap \mathcal{C} = \{0\}\} \leq \eta, \end{aligned}$$

where $c_\eta := \sqrt{8\log(4/\eta)}$.

[1] The statistical dimension is closely related to the Gaussian complexity [2], Gaussian width [7], and Gaussian squared complexity [6].

Probability of exact recovery

Corollary

For any $\eta \in (0,1)$,

$$\begin{aligned} n \geq d(\mathcal{D}_f(\mathbf{x}^\natural)) + c_\eta \sqrt{p} \quad &\Rightarrow \quad \mathbb{P}\left\{\mathbf{x}^\star = \mathbf{x}^\natural\right\} \geq 1 - \eta; \\ n \leq d(\mathcal{D}_f(\mathbf{x}^\natural)) - c_\eta \sqrt{p} \quad &\Rightarrow \quad \mathbb{P}\left\{\mathbf{x}^\star = \mathbf{x}^\natural\right\} \leq \eta, \end{aligned}$$

where $c_\eta := \sqrt{8\log(4/\eta)}$.

- There is a *phase transition* at $n \approx d(\mathcal{D}_f(\mathbf{x}^\natural))$.

Examples ([1])

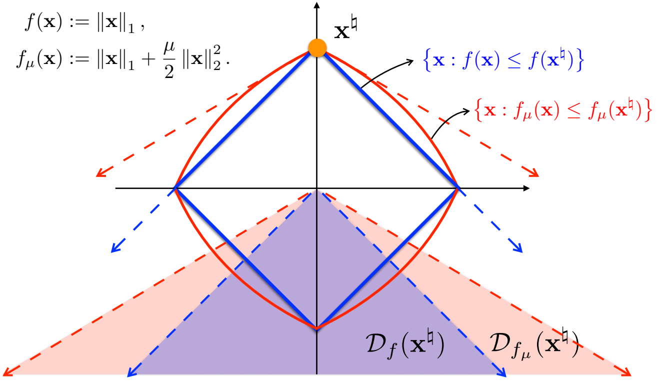
- Let $f(\mathbf{x}) := \|\mathbf{x}\|_1$, and let $\mathbf{x}^\natural \in \mathbb{R}^p$ be s-sparse. Then $d(\mathcal{D}_f(\mathbf{x}^\natural)) \leq 2s\log(p/s) + (5/4)s$.
- Let $f(\mathbf{x}) := \|\mathbf{X}\|_*$, and let $\mathbf{X}^\natural \in \mathbb{R}^{p\times p}$ of rank r. Then $d(\mathcal{D}_f(\mathbf{x}^\natural)) \leq 3r(2p - r)$.

Smoothing increases the statistical dimension

Key properties of the statistical dimension [1]

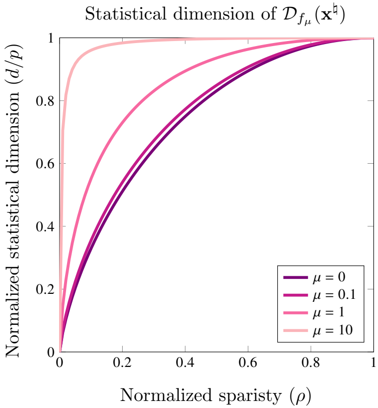
- ○ The statistical dimension is invariant under unitary transformations (rotations).
- ○ Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be closed convex cones. If $\mathcal{C}_1 \subseteq \mathcal{C}_2$, then $d(\mathcal{C}_1) \leq d(\mathcal{C}_2)$.

The larger the statistical dimension is, the more number of observations is required.

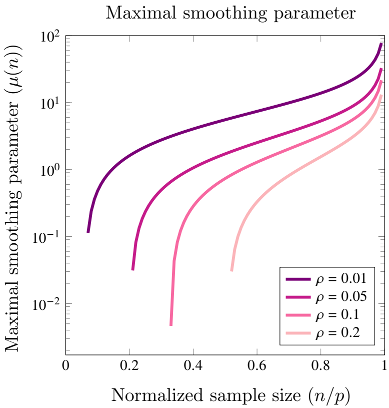


$f(\mathbf{x}) := \|\mathbf{x}\|_1$,
$f_\mu(\mathbf{x}) := \|\mathbf{x}\|_1 + \frac{\mu}{2}\|\mathbf{x}\|_2^2$.
$\mathbf{x}^\natural$
$\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^\natural)\}$
$\{\mathbf{x} : f_\mu(\mathbf{x}) \leq f_\mu(\mathbf{x}^\natural)\}$
$\mathcal{D}_f(\mathbf{x}^\natural)$
$\mathcal{D}_{f_\mu}(\mathbf{x}^\natural)$

Numerical results for the statistical dimension and $\mu(n)$



Statistical dimension of $\mathcal{D}_{f_\mu}(\mathbf{x}^\natural)$



Maximal smoothing parameter

Smoothing decreases the computational cost

∘ Consider the estimator,

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f_\mu(\mathbf{x}) : \mathbf{b} = \mathbf{A}\mathbf{x}, \|\mathbf{x}\|_\infty \le \|\mathbf{x}^\natural\|_\infty \right\}, \quad \mu \in [0,\infty).$$

Proposition

Let $\mu > 0$ and $f(\mathbf{x}) = \|\mathbf{x}\|_1$. Consider solving (1) with a primal-dual method as in [4, 5]. The output after the k-th iteration, $\mathbf{x}^k$, satisfies

$$\left\|\mathbf{x}^\star - \mathbf{x}^k\right\|_2 \le \frac{4p\kappa(\mathbf{A})\left[\rho(1+\mu\left\|\mathbf{x}^\star\right\|_\infty)^2 + (1-\rho)\right]}{\mu k} \propto \frac{1}{\mu k}\bigg|_{\rho\ll 1},$$

where $\rho := s/p$, s being the number of non-zero entries in $\mathbf{x}^\star$, and $\kappa(\mathbf{A})$ denotes the restricted condition number of $\mathbf{A}$.

Observations: ∘ When $\rho \ll 1$, the number of iterations k to achieve the required precision decreases.

∘ In fact, we need $1/(\mu\varepsilon)$ iterations to have an error bound $\left\|\mathbf{x}^\star - \mathbf{x}^k\right\|_2 \le \varepsilon$ for a fixed $\epsilon > 0$.

Time-data tradeoff

- Define the maximal smoothing parameter

$$\mu(n) := \arg\max_{\mu > 0} \left\{ \mu : d\left(\mathcal{D}_{f_\mu}(\mathbf{x}^\natural)\right) \leq n \right\}.$$

- Consider the "conservative" estimator in probability,

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ f_\mu(\mathbf{x})\big|_{\mu = \frac{1}{4}\mu(n)} : \mathbf{b} = \mathbf{A}\mathbf{x} \right\}.$$

Corollary

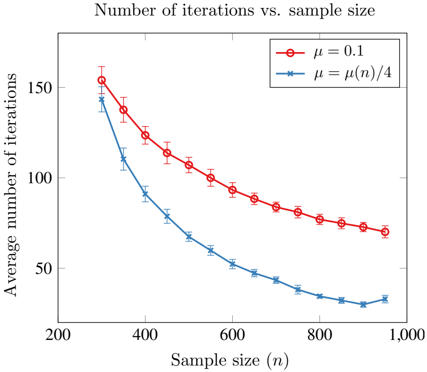
Let $\rho := s/p \ll 1$*. Then we have, with high probability,* $\mathbf{x}^\star = \mathbf{x}^\natural$*, and*

$$\left\|\mathbf{x}^\natural - \mathbf{x}^k\right\|_2 \propto \frac{1}{\mu(n)k}.$$

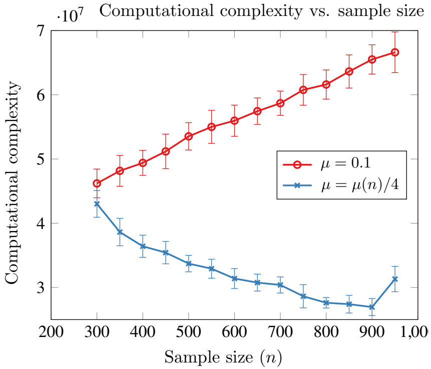
Therefore, to achieve the error bound, $\left\|\mathbf{x}^\natural - \mathbf{x}^k\right\|_2 \leq \varepsilon$ *for a fixed* $\varepsilon > 0$*, it suffices to choose*

$$k = O\left(\frac{1}{\mu(n)}\right).$$

A numerical result for the time-data tradeoff



Number of iterations vs. sample size



Computational complexity vs. sample size

Another trade-off in optimization

- Statistics vs Optimization:

Discipline	Goal	Metric
Optimization	reaching numerical ϵ-accuracy	$\\|\mathbf{x}^k - \mathbf{x}^\star\\| \leq \epsilon$
Statistics	learning ε-accurate model	$\\|\mathbf{x}^\star - \mathbf{x}^\natural\\| \leq \varepsilon$

Remarks: *As data sample size gets larger we have seen that:*

- Algorithms take longer to reach ϵ accuracy.
- However, statistical error ε decreases as the estimation is more precise.

Another trade-off in optimization

- Statistics vs Optimization:

Discipline	Goal	Metric
Optimization	reaching numerical ϵ-accuracy	$\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \epsilon$
Statistics	learning ε-accurate model	$\|\mathbf{x}^\star - \mathbf{x}^\natural\| \leq \varepsilon$

Remarks: *As data sample size gets larger we have seen that:*

- Algorithms take longer to reach ϵ accuracy.
- However, statistical error ε decreases as the estimation is more precise.

Similar analogy exists between per-iteration cost and convergence rate for optimization algorithms

Another trade-off in optimization

- Statistics vs Optimization:

Discipline	Goal	Metric
Optimization	reaching numerical ϵ-accuracy	$\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \epsilon$
Statistics	learning ε-accurate model	$\|\mathbf{x}^\star - \mathbf{x}^\natural\| \leq \varepsilon$

Remarks: *As data sample size gets larger we have seen that:*

- Algorithms take longer to reach ϵ accuracy.
- However, statistical error ε decreases as the estimation is more precise.

Similar analogy exists between per-iteration cost and convergence rate for optimization algorithms

Understanding this trade-off helps us reduce total complexity!

Recall: GD vs. SGD

Problem (Unconstrained convex minimization)

Deterministic setting

$$f^\star = \min_{\mathbf{x}\in\mathbb{R}^p} f(x)$$

- ○ $f(\mathbf{x})$ *is a proper, closed, convex and smooth.*
- ○ *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}\,(f) : f(\mathbf{x}^\star) = f^\star\} \neq \emptyset.$

Stochastic programming

$$f^\star = \min_{\mathbf{x}\in\mathbb{R}^p} \{f(\mathbf{x}) := \mathbb{E}[f(\mathbf{x},\theta)]\}$$

- ○ $f(\mathbf{x})$ *is proper, closed, convex and smooth.*
- ○ *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}\,(f) : f(\mathbf{x}^\star) = f^\star\} \neq \emptyset.$
- ○ θ *is a random vector, supported on set* Θ.

Algorithms

Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)$$

- ○ $\alpha_k < 2/L.$

Stochastic Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k G(\mathbf{x}^k, \theta_k)$$

- ○ $\alpha_k = \mathcal{O}(1/\sqrt{k})$
- ○ $\mathbb{E}[G(\mathbf{x}^k, \theta_k)] = \nabla f(\mathbf{x}^k)$

Example: Convex optimization with finite sum

- Consider the finite sum (e.g., ERM) setting

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x}) \right\}.}$$

Algorithms in the finite sum setting

Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)$$

- $\nabla f(\mathbf{x}^k) = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\mathbf{x}^k)$

Stochastic Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k G(\mathbf{x}^k, \theta_k)$$

- $G(\mathbf{x}^k, \theta_k) = \nabla f_j(\mathbf{x}^k)$, $j \sim \mathrm{Uniform}(\{1, \cdots, n\})$

Example: Convex optimization with finite sum

- Consider the finite sum (e.g., ERM) setting

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x}) \right\}.}$$

Algorithms in the finite sum setting

Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)$$

- $\nabla f(\mathbf{x}^k) = \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\mathbf{x}^k)$

Stochastic Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k G(\mathbf{x}^k, \theta_k)$$

- $G(\mathbf{x}^k, \theta_k) = \nabla f_j(\mathbf{x}^k)$, $j \sim \mathrm{Uniform}(\{1, \cdots, n\})$

- $f(\mathbf{x})$: convex and L-Lipschitz gradient

	rate	cost per iteration	iteration complexity	total complexity
GD	$1/k$	n	$1/\epsilon$	n/ϵ
SGD	$1/\sqrt{k}$	1	$1/\epsilon^2$	$1/\epsilon^2$

Example: Convex optimization with finite sum

- Consider the finite sum (e.g., ERM) setting

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x}) \right\}.}$$

Algorithms in the finite sum setting

Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla f(\mathbf{x}^k)$$

- $\nabla f(\mathbf{x}^k) = \frac{1}{n}\sum_{j=1}^n \nabla f_j(\mathbf{x}^k)$

Stochastic Gradient Descent

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k G(\mathbf{x}^k, \theta_k)$$

- $G(\mathbf{x}^k, \theta_k) = \nabla f_j(\mathbf{x}^k)$, $j \sim \mathrm{Uniform}(\{1, \cdots, n\})$

- $f(\mathbf{x})$: μ-strongly convex and L-Lipschitz gradient

	rate	cost per iteration	iteration complexity	total complexity
GD	ρ^k	n	$\log(1/\epsilon)$	$n\log(1/\epsilon)$
SGD	$1/k$	1	$1/\epsilon$	$1/\epsilon$

When f is μ-strongly convex and L-Lipschitz gradient

Finite sums

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\left\{f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x})\right\}.}$$

	rate	cost per iteration	iteration complexity	total complexity
GD	ρ^k	n	$\log(1/\epsilon)$	$n\log(1/\epsilon)$
SGD	$1/k$	1	$1/\epsilon$	$1/\epsilon$

Remarks:

- ∘ SGD trades off convergence rate with low per-iteration cost.
- ∘ When n is large, SGD proves to be effective.
- ∘ To control variance of the stochastic gradient estimate, SGD decreases step size at a certain rate.
- ∘ In turn, convergence deteriorates from linear to sublinear.

An observation of GD vs. SGD step

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k \nabla f(\mathbf{x}^k) \quad \text{(GD)}$$

Lemma

Assume f is Lipschitz smooth with constant L. Then,

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) \leq (\gamma_k^2 L - \gamma_k)\|\nabla f(\mathbf{x}^k)\|^2.$$

An observation of GD vs. SGD step

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k G(\mathbf{x}^k, \theta_k) \quad \text{(SGD)}$$

Lemma

Assume f is Lipschitz smooth with constant L. Then,

$$\mathbb{E}[f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)] \leq (\gamma_k^2 L - \gamma_k)\mathbb{E}[\|\nabla f(\mathbf{x}^k)\|^2] + L\gamma_k^2\mathbb{E}[\|G(\mathbf{x}^k, \theta_k) - \nabla f(\mathbf{x}^k)\|^2]$$

An observation of GD vs. SGD step

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k G(\mathbf{x}^k, \theta_k) \quad \text{(SGD)}$$

Lemma

Assume f is Lipschitz smooth with constant L. Then,

$$\mathbb{E}[f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)] \leq (\gamma_k^2 L - \gamma_k)\mathbb{E}[\|\nabla f(\mathbf{x}^k)\|^2] + L\gamma_k^2 \mathbb{E}[\|G(\mathbf{x}^k, \theta_k) - \nabla f(\mathbf{x}^k)\|^2]$$

- The variance of gradient estimate dominates as $\nabla f(\mathbf{x}^k) \to 0$.
- To ensure convergence we need to control variance.

$$\gamma_k \to 0 \Longrightarrow \text{Slow convergence!}$$

Can we decrease the variance while using a constant step-size?

An observation of GD vs. SGD step

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k G(\mathbf{x}^k, \theta_k) \quad \text{(SGD)}$$

Lemma

Assume f is Lipschitz smooth with constant L. Then,

$$\mathbb{E}[f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)] \leq (\gamma_k^2 L - \gamma_k)\mathbb{E}[\|\nabla f(\mathbf{x}^k)\|^2] + L\gamma_k^2 \mathbb{E}[\|G(\mathbf{x}^k, \theta_k) - \nabla f(\mathbf{x}^k)\|^2]$$

- The variance of gradient estimate dominates as $\nabla f(\mathbf{x}^k) \to 0$.
- To ensure convergence we need to control variance.

$$\gamma_k \to 0 \Longrightarrow \text{Slow convergence!}$$

Can we decrease the variance while using a constant step-size?

Choose a stochastic gradient, s.t. $\mathbb{E}\left[\|G(\mathbf{x}^k; \theta_k)\|^2\right] \to 0$.

A simple approach: Mini-batch SGD

○ More samples → better estimate for full gradient.

SGD with mini batches

Let $G(\mathbf{x},\theta)$ be an unbiased gradient estimate ($\mathbb{E}[G(\mathbf{x},\theta)] = \nabla f(\mathbf{x})$) and B_k be the batch size. Then,

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \frac{1}{B_k} \sum_{j=1}^{B_k} G(\mathbf{x}^k, \theta_{k,j})$$

Theorem

Let $B_k > 0$ be the batch size and $G(\mathbf{x},\theta)$ be an unbiased gradient estimate with bounded variance, i.e., $\mathbb{E}[\|G(\mathbf{x},\theta) - \nabla f(\mathbf{x})\|^2 \mid \mathbf{x}] \leq \sigma^2$. Then, the mini-batch estimate has the following properties:

$$\mathbb{E}\left[\frac{1}{B_k}\sum_{j=1}^{B_k} G(\mathbf{x},\theta_{k,j})\right] = \nabla f(\mathbf{x}) \qquad \text{and} \qquad \mathbb{E}\left[\left\|\frac{1}{B_k}\sum_{j=1}^{B_k} G(\mathbf{x},\theta_{k,j}) - \nabla f(\mathbf{x})\right\|^2 \mid \mathbf{x}\right] \leq \frac{\sigma^2}{B_k}$$

Remarks: ○ We might need to increase the batch size over time to take variance to 0.

○ We can come up with a “smarter” estimate for $\nabla f(\mathbf{x})$.

How to construct a new estimate $G(\mathbf{x}^k; \theta_k)$? [8]

Finite sum structure:	SGD update rule:
$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^n f_j(\mathbf{x}) \right\}$	$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k \nabla f_j(\mathbf{x}^k)$

- ○ Let $X = \nabla f_j(\mathbf{x}^k)$ be a random variable (due to $j \sim$ Uniform$(\{1, \cdots, n\})$).
- ○ Let $Y = \nabla f_j(\tilde{\mathbf{x}})$ be another random variable, and $\tilde{\mathbf{x}}$ is a particularly selected point.

Remarks:

- ○ We want X and Y to be correlated (we will see why!).
- ○ Given Y, we should be able to estimate $\mathbb{E}[X]$ with more confidence.

Observations:

- ○ Choice of $\tilde{\mathbf{x}}$ affects how correlated X and Y are.
- ○ We can compute $\mathbb{E}[Y] = \frac{1}{n}\sum_{j=1}^n \nabla f_j(\tilde{\mathbf{x}}) = \nabla f(\tilde{\mathbf{x}})$.

Goal:

- ○ Find a **good** estimate of $\mathbb{E}[X] = \frac{1}{n}\sum_{j=1}^n \nabla f_j(\mathbf{x}^k) = \nabla f(\mathbf{x}^k)$.

How to construct a new estimate $G(\mathbf{x}^k; \theta_k)$? [8]

Finite sum structure:	SGD update rule:
$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^n f_j(\mathbf{x}) \right\}$	$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k \nabla f_j(\mathbf{x}^k)$

- ○ Let $X = \nabla f_j(\mathbf{x}^k)$ be a random variable (due to $j \sim$ Uniform$(\{1, \cdots, n\})$).
- ○ Let $Y = \nabla f_j(\tilde{\mathbf{x}})$ be another random variable, and $\tilde{\mathbf{x}}$ is a particularly selected point.

A generalized estimator: $R_\alpha = \alpha(X - Y) + \mathbb{E}[Y]$

- ○ $\mathbb{E}[R_\alpha] = \alpha\mathbb{E}[X] + (1-\alpha)\mathbb{E}[Y]$
- ○ $\mathrm{Var}(R_\alpha) = \alpha^2(\mathrm{Var}(X) + \mathrm{Var}(Y) - 2\mathrm{Cov}(X, Y))$

Observations:

- ○ When $\alpha = 1$, R_α becomes unbiased, i.e., $\mathbb{E}[R_\alpha] = \mathbb{E}[X]$.
- ○ If $\mathrm{Cov}(X, Y)$ is large enough (X and Y are correlated enough), $\mathrm{Var}(R_\alpha) \leq \mathrm{Var}(X)$.

How could we use this information to construct our estimate?

Variance reduction techniques: SVRG

∘ Select the stochastic gradient ∇f_{i_k}, and compute a gradient estimate

$$\mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}).$$

∘ As $\tilde{\mathbf{x}} \to \mathbf{x}^\star$ and $\mathbf{x}^k \to \mathbf{x}^\star$,

$$\nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}) \to 0.$$

∘ Therefore,

$$\mathbb{E}\left[\|\nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}})\|^2\right] \to 0.$$

Remarks:

∘ Remember the generalized estimator: $R_\alpha = \alpha(X - Y) + \mathbb{E}[Y]$.

∘ For SVRG, $\alpha = 1$, $X = \nabla f_{i_k}(\mathbf{x}^k)$ and $Y = \nabla f_{i_k}(\tilde{\mathbf{x}})$.

∘ We will see how $\tilde{\mathbf{x}}$ is computed!

Stochastic gradient algorithm with variance reduction

Stochastic gradient with variance reduction (SVRG) [9, 13]

```
1. Choose $\widetilde{\mathbf{x}}^0 \in \mathbb{R}^p$ as a starting point and $\gamma > 0$ and $q \in \mathbb{N}_+$.
2. For $s = 0, 1, 2 \cdots$, perform:
    2a. $\widetilde{\mathbf{x}} = \widetilde{\mathbf{x}}^s, \quad \widetilde{\mathbf{v}} = \nabla f(\widetilde{\mathbf{x}}), \quad \mathbf{x}^0 = \widetilde{\mathbf{x}}$.
    2b. For $k = 0, 1, \cdots q - 1$, perform:
        $$\begin{cases} \text{Pick } i_k \in \{1, \ldots, n\} \text{ uniformly at random} \\ \mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\widetilde{\mathbf{x}}) + \widetilde{\mathbf{v}} \\ \mathbf{x}^{k+1} := \mathbf{x}^k - \gamma \mathbf{r}_k, \end{cases} \tag{1}$$
    2c. Update $\widetilde{\mathbf{x}}^{s+1} = \frac{1}{m} \sum_{j=0}^{q-1} \mathbf{x}^j$.
```

Features

- ○ The SVRG method uses a multistage scheme to reduce the variance of the stochastic gradient $\mathbf{r}_k$.
- ○ Learning rate γ does not necessarily tend to 0 while $\mathbf{x}^k$ and $\widetilde{\mathbf{x}}^s$ tend to $\mathbf{x}_\star$.
- ○ Each stage, SVRG uses $n + 2q$ component gradient evaluations.
- ○ n for the full gradient at the beginning of each stage, and $2q$ for each of the q stochastic gradient steps.

Convergence analysis

Assumption A5.

(i) f is μ-strongly convex

(ii) The learning rate $0 < \gamma < 1/(4L_{\max})$, where $L_{\max} = \max_{1\leq j\leq n} L_j$.

(iii) q is large enough such that

$$\kappa = \frac{1}{\mu\gamma(1-4\gamma L_{\max})q} + \frac{4\gamma L_{\max}(q+1)}{(1-4\gamma L_{\max})q} < 1.$$

Theorem

Assumptions:

○ *The sequence $\{\widetilde{\mathbf{x}}^s\}_{k\geq 0}$ is generated by SVRG.*

○ *Assumption A5 is satisfied.*

Conclusion: *Linear convergence is obtained:*

$$\mathbb{E}f(\widetilde{\mathbf{x}}^s) - f(\mathbf{x}^\star) \leq \kappa^s(f(\widetilde{\mathbf{x}}^0) - f(\mathbf{x}^\star)).$$

Choice of γ and q, and complexity

Chose γ and q such that $\kappa \in (0,1)$:

For example

$$\gamma = 0.1/L_{\max}, q = 100(L_{\max}/\mu) \Longrightarrow \kappa \approx 5/6.$$

Complexity

$$\mathbb{E}f(\widetilde{\mathbf{x}}^s) - f(\mathbf{x}^\star) \leq \varepsilon, \quad \text{when } s \geq \log((f(\widetilde{\mathbf{x}}^0) - f(\mathbf{x}^\star))/\epsilon)/\log(\kappa^{-1})$$

- Each stage needs $n + 2q$ component gradient evaluations
- With $q = \mathcal{O}(L_{\max}/\mu)$, we obtain an overall complexity of

$$\mathcal{O}\Big((n + L_{\max}/\mu)\log(1/\epsilon)\Big).$$

Comparison: GD vs. SGD vs. SVRG

- GD update:

$$\left\{ \ \mathbf{x}^{k+1} := \mathbf{x}^k - \gamma \nabla f(\mathbf{x}^k), \right.$$

- SGD update:

$$\left\{ \ \mathbf{x}^{k+1} := \mathbf{x}^k - \gamma \nabla f_{i_k}(\mathbf{x}^k), \right.$$

- SVRG update:

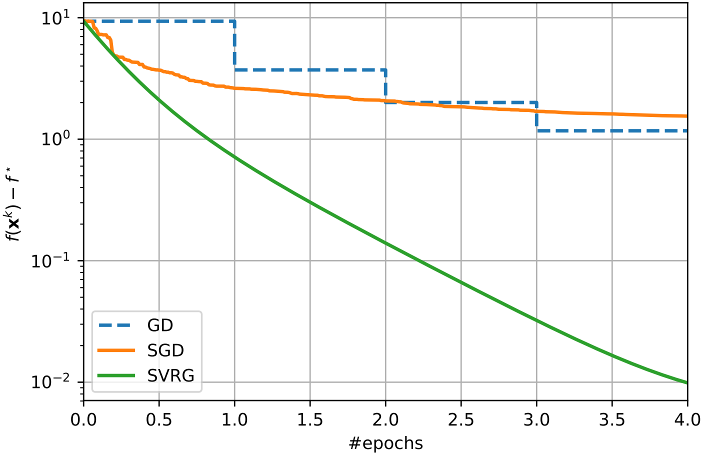
$$\left\{ \begin{array}{l} \mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\widetilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}) \\ \mathbf{x}^{k+1} := \mathbf{x}^k - \gamma \mathbf{r}_k, \end{array} \right.$$

	SGD	SVRG	GD
Requires gradient storage?	no	no	no
Epoch-based	no	yes	no
Parameters	stepsize	stepsize & epoch length	stepsize
Gradient evaluations	1 per iteration	$n + 2q$ per epoch	n per iteration

Table: Comparisons of SGD, SVRG and GD [8]

- Recall that $q = \mathcal{O}(L_{\text{max}}/\mu)$ is the epoch length for SVRG.

Example: ℓ_2-regularized least squares with synthetic data



Taxonomy of algorithms

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x}) \right\}.}$$

- ○ $f(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x})$: μ-strongly convex with L-Lipschitz continuous gradient.

SVRG	GD	SGD
Linear	Linear	Sublinear

Table: Rate of convergence.

- ○ $\kappa = L/\mu$.

SVRG	AGD	SGD
$\mathcal{O}((n+\kappa)\log(1/\varepsilon))$	$\mathcal{O}((n\kappa)\log(1/\varepsilon))$	$1/\varepsilon$

Table: Complexity to obtain ε-solution.

Wrap up!

- Please check Homework 1 on Friday!
- Logistics on remote “lab hours” will be announced on moodle.

*Calculation of $d\left(\mathcal{D}_f\left(\mathbf{x}^\natural\right)\right)$ and $d\left(\mathcal{D}_{f_\mu}\left(\mathbf{x}^\natural\right)\right)$

Lemma ([1])

Let f be a proper lower-semicontinuous convex function, and let $\mathbf{x} \in \operatorname{dom}(f)$. We have

$$d\left(\mathcal{D}_f(\mathbf{x})\right) \leq \inf_{\tau > 0} \mathbb{E}\left[\operatorname{dist}^2\left(\mathbf{g}, \tau \partial f(\mathbf{x})\right)\right],$$

where $\mathbf{g}$ is a vector of i.i.d. standard Gaussian random variables.

The upper bounds on $d\left(\mathcal{D}_f\left(\mathbf{x}^\natural\right)\right)$ and $d\left(\mathcal{D}_{f_\mu}\left(\mathbf{x}^\natural\right)\right)$ can be derived based on above.

Proposition

Let $\mathbf{x}^\natural$ be an s-sparse vector. We have

$$d\left(\mathcal{D}_{f_\mu}\left(\mathbf{x}^\natural\right)\right) \leq \inf_{\tau > 0} \left\{ s(1+\tau^2) + 2\mu f_\mu(\mathbf{x}^\natural)\tau^2 \right.$$
$$\left. + (p-s)\sqrt{\frac{2}{\pi}} \int_\tau^\infty (u-\tau)^2 e^{-u^2/2}\, du \right\}.$$

Note that $f = f_\mu|_{\mu=0}$.

*Variance reduction techniques: SAGA

- Select the stochastic gradient $\mathbf{r}_k$ as

$$\mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}^k_{i_k}) + \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\tilde{\mathbf{x}}^k_j),$$

where, at each iteration, $\tilde{\mathbf{x}}$ is updated as $\tilde{\mathbf{x}}^k_{i_k} = \mathbf{x}^k$ and $\tilde{\mathbf{x}}^k_j$ stays the same for $j \neq i_k$.

- As $\tilde{\mathbf{x}}^k_j \to \mathbf{x}^\star$ and $\mathbf{x}^k \to \mathbf{x}^\star$,

$$\nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}^k_{i_k}) + \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\tilde{\mathbf{x}}^k_j) \to 0.$$

- Therefore,

$$\mathbb{E}\left[\|\nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}^k_{i_k}) + \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\tilde{\mathbf{x}}^k_j)\|^2\right] \to 0.$$

*Variance reduction techniques: SAGA

Stochastic Average Gradient (SAGA) [8]

```
1a. Choose $\tilde{\mathbf{x}}_i^0 = \mathbf{x}^0 \in \mathbb{R}^p, \forall i$, $q \in \mathbb{N}_+$ and stepsize $\gamma > 0$.
1b. Store $\nabla f_i(\tilde{\mathbf{x}}_i^0)$ in a table data-structure with length $n$.
2.  For $k = 0, 1 \ldots$ perform:
2a.     Pick $i_k \in \{1, \ldots, n\}$ uniformly at random
2b.     Take $\tilde{\mathbf{x}}_{i_k}^{k+1} = \mathbf{x}^k$, store $\nabla f_{i_k}(\tilde{\mathbf{x}}_{i_k}^{k+1})$ in the table and leave other entries the same.
2c.     $\mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}_{i_k}^k) + \frac{1}{n}\sum_{j=1}^n \nabla f_j(\tilde{\mathbf{x}}_j^k)$
3.  $\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma \mathbf{r}_k$
```

Recipe:

In each iteration:

- Store last gradient evaluated at each datapoint.
- Previous gradient for datapoint j is $\nabla f_j(\tilde{\mathbf{x}}_j^k)$.
- Perform SG-iterations with the following stochastic gradient

$$\mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}_{i_k}^k) + \frac{1}{n}\sum_{j=1}^n \nabla f_j(\tilde{\mathbf{x}}_j^k).$$

*Convergence of SAGA

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^n f_j(\mathbf{x}) \right\}.}$$

Theorem (Convergence of SAGA [8])

Suppose that f is μ-strongly convex and that the stepsize is $\gamma = \frac{1}{2(\mu n+L)}$ with

$$\rho = 1 - \frac{\mu}{2(\mu n + L)} < 1,$$

$$C = \|\mathbf{x}^0 - \mathbf{x}^\star\|^2 + \frac{n}{\mu n + L}[f(\mathbf{x}^0) - \langle \nabla f(\mathbf{x}^\star), \mathbf{x}^0 - \mathbf{x}^\star\rangle - f(\mathbf{x}^\star)]$$

Then

$$\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}^\star\|^2] \leq \rho^k C.$$

- ○ Allows the constant step-size.
- ○ Obtains linear rate convergence.

*Variance reduction techniques: SARAH

◦ Select the stochastic gradient $\mathbf{r}_k$

$$\mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\mathbf{x}^{k-1}) + \mathbf{r}_{k-1},$$

◦ The variance reduction in SARAH can be characterized as

$$\mathbb{E}[\|\mathbf{r}_k\|^2] \leq \left[1 - \left(\frac{2}{\gamma L} - 1\right)\mu^2\gamma^2\right]^k \mathbb{E}[\|\nabla f(\mathbf{x}^0)\|^2].$$

*Variance reduction techniques: SARAH

Stochastic Recursive Gradient Algorithm (SARAH) [10]

```
1. Choose $\overline{\mathbf{x}}^0 \in \mathbb{R}^p$, $q \in \mathbb{N}_+$ and stepsize $\gamma > 0$.
2. For $k = 0, 1 \ldots$ perform:
2. $\mathbf{x}^0 = \overline{\mathbf{x}}^k$, $\mathbf{r}_0 = \frac{1}{n}\sum_{j=1}^n f_j(\overline{\mathbf{x}}^0)$
2a. $\mathbf{x}^1 = \mathbf{x}^0 - \gamma\mathbf{r}_0$
2b.  For $l = 1 \ldots, q-1$, perform:
        $\begin{cases} \text{pick } i_l \in \{1, \ldots, n\} \text{ uniformly at random,} \\ \mathbf{r}_l = \nabla f_{i_l}(\mathbf{x}^l) - \nabla f_{i_l}(\mathbf{x}^{l-1}) + \mathbf{r}_{l-1}, \\ \mathbf{x}^{l+1} = \mathbf{x}^l - \gamma\mathbf{r}_l. \end{cases}$
3 Update $\overline{\mathbf{x}}^{k+1} = \mathbf{x}^l$ where $l$ is chosen uniformly at random from $\{0, \ldots, q\}$ .
```

Recipe: *In a cycle of q inner iterations:*

- ○ Compute stochastic step direction by recursively adding and subtracting component gradients.

$$\mathbf{r}_l = \nabla f_{i_l}(\mathbf{x}^l) - \nabla f_{i_l}(\mathbf{x}^{l-1}) + \mathbf{r}_{l-1}.$$

- ○ Perform q SG-iterations with $\mathbf{r}_l$.
- ○ Update next iteration by picking uniformly at random from q previous iterations.

*Convergence of SARAH

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\left\{f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x})\right\}.}$$

Theorem (Convergence of SARAH [10])

Suppose that f is μ-strongly convex and that the stepsize γ and number of inner iterations q satisfies

$$\rho_q = \frac{1}{\mu\gamma(1+q)} + \frac{L_{\mathit{max}}\gamma}{2 - L_{\mathit{max}}\gamma} < 1.$$

Then

$$\mathbb{E}[\|\nabla f(\overline{\mathbf{x}}^k)\|^2] \leq \rho_q^k \|\nabla f(\overline{\mathbf{x}}^0)\|^2.$$

References I

[1] Dennis Amelunxen, Martin Lotz, Michael B. McCoy, and Joel A. Tropp.
Living on the edge: Phase transitions in convex programs with random data.
2014.
arXiv:1303.6672v2 [cs.IT].

[2] Peter L. Barlett and Shahar Mendelson.
Rademacher and Gaussian complexities: Risk bounds and structural results.
J. Mach. Learn. Res., 3, 2002.

[3] Léon Bottou and Oliver Bousquet.
The tradeoffs of large scale learning.
In *Advances in Neural Information Processing Systems*, 2007.

[4] John J Bruer, Joel A Tropp, Volkan Cevher, and Stephen Becker.
Time–data tradeoffs by aggressive smoothing.
In *Advances in Neural Information Processing Systems*, pages 1664–1672, 2014.

[5] John J Bruer, Joel A Tropp, Volkan Cevher, and Stephen R Becker.
Designing statistical estimators that balance sample size, risk, and computational cost.
IEEE Journal of Selected Topics in Signal Processing, 9(4):612–624, 2015.

References II

[6] Venkat Chandrasekaran and Michael I. Jordan.
Computational and statistical tradeoffs via convex relaxation.
Proc. Natl. Acad. Sci., 110(13):E1181–E1190, 2013.

[7] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky.
The convex geometry of linear inverse problems.
Found. Comput. Math., 12:805–849, 2012.

[8] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien.
Saga: A fast incremental gradient method with support for non-strongly convex composite objectives.
In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1646–1654. Curran Associates, Inc., 2014.

[9] Rie Johnson and Tong Zhang.
Accelerating stochastic gradient descent using predictive variance reduction.
In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

[10] Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takac.
Sarah: A novel method for machine learning problems using stochastic recursive gradient, 2017.

References III

[11] Samet Oymak, Christos Thrampoulidis, and Babak Hassibi.
Simple bounds for noisy linear inverse problems with exact side information.
2013.
arXiv:1312.0641v2 [cs.IT].

[12] Shai Shalev-Shwartz and Nathan Srebro.
Svm optimization: inverse dependence on training set size.
In *Proceedings of the 25th international conference on Machine learning*, pages 928–935, 2008.

[13] Lin Xiao and Tong Zhang.
A proximal stochastic gradient method with progressive variance reduction.
SIAM Journal on Optimization, 24, 03 2014.