

Iterative Randomized Robust Linear Regression

Yannis Kopsinis¹ Symeon Chouvardas² Sergios Theodoridis¹

¹University of Athens, Greece

²Mathematical and Algorithmic Sciences Lab,
Huawei, France

Wednesday, Apr 22, 2015

- 1 Randomized Linear Regression
- 2 Robust Randomized Linear Regression
- 3 Iterative Randomized Robust Regression
- 4 Performance Results
- 5 Randomized Low Rank matrix approximation

Randomized Methods

Major Principle that governs randomized methods

Instead of working with the original large-scale data matrices, operate on **compressed** versions of them. The compression is realized via computationally efficient **dimensionality reduction**, which is performed in a **randomized** rather than in a deterministic way.

Some Facts!

- It is a very appealing idea!
 - Data are highly compressible
 - Low speed memory units are the major bottleneck
- It is applicable to ubiquitous data analysis and ML tasks, even to basic matrix operations
 - Matrix Multiplication
 - Linear Regression
 - Low-rank Matrix approximation (Singular Value Decomposition)

Randomized Linear Regression

Linear LS Regression

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \boldsymbol{\eta}$$

$N \times 1 \quad N \times l \quad l \times 1 \quad N \times 1$

$N \gg l$, and N , at least, very large

$$\hat{\mathbf{x}}_{LS} = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$$

$$\hat{\mathbf{x}}_{LS} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

Computational complexity: $\mathcal{O}(Nl^2)$ via QR decomposition

Randomized Linear Regression

Linear LS Regression

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \boldsymbol{\eta}$$

$N \times 1 \quad N \times l \quad l \times 1 \quad N \times 1$

$N \gg l$, and N , at least, very large

$$\hat{\mathbf{x}}_{LS} = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$$

$$\hat{\mathbf{x}}_{LS} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

Computational complexity: $\mathcal{O}(Nl^2)$ via QR decomposition

Randomized Linear LS Regression

$$\underline{\mathbf{b}} = \mathbf{R} \mathbf{b}, \quad \underline{\mathbf{A}} = \mathbf{R} \mathbf{A}$$

$d \times 1 \quad d \times N \quad N \times 1 \quad d \times l \quad d \times N \quad N \times l$

where $d \ll N$

$$\hat{\mathbf{x}}_R = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\underline{\mathbf{b}} - \underline{\mathbf{A}}\mathbf{x}\|_2^2,$$

Computational complexity: $\mathcal{O}(dl^2) + \mathcal{T}(\mathbf{R}) + \mathcal{T}(\mathbf{R}\mathbf{A})$

Randomized Least Squares

Some theoretical results [Drineas 2011]

If $d = \mathcal{O}\left(l(\ln l)(\ln N) + \frac{l \ln N}{\epsilon}\right)$, then with probability 0.8

$$\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}_R\|_2 \leq (1 + \epsilon)\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}_{LS}\|_2$$

$$\|\mathbf{x}_{LS} - \mathbf{x}_R\|_2 \leq \sqrt{\epsilon} \left(\kappa(\mathbf{A}) \sqrt{\gamma^{-2} - 1} \|\mathbf{x}_{LS}\|_2 \right)$$

Worse Case Results!

Randomized Least Squares

Some theoretical results [Drineas 2011]

If $d = \mathcal{O}\left(l(\ln l)(\ln N) + \frac{l \ln N}{\epsilon}\right)$, then with probability 0.8

$$\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}_R\|_2 \leq (1 + \epsilon)\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}_{LS}\|_2$$

$$\|\mathbf{x}_{LS} - \mathbf{x}_R\|_2 \leq \sqrt{\epsilon} \left(\kappa(\mathbf{A}) \sqrt{\gamma^{-2} - 1} \|\mathbf{x}_{LS}\|_2 \right)$$

Worse Case Results!

Johnson–Lindenstrauss (JL) seminal work (1984)

For **any** set, S , of l points, $\mathbf{u}_1, \mathbf{u}_2, \dots$ in \mathbb{R}^N **there exist** a linear mapping $\mathbf{R} : \mathbb{R}^N \rightarrow \mathbb{R}^d$, with $d = \mathcal{O}(\epsilon^{-2} \log l)$, such that all the **pairwise** distances are approximately preserved:

$$\forall i, j \quad (1 - \epsilon)\|\mathbf{u}_i - \mathbf{u}_j\|_2^2 \leq \|\mathbf{R}\mathbf{u}_i - \mathbf{R}\mathbf{u}_j\|_2^2 \leq (1 + \epsilon)\|\mathbf{u}_i - \mathbf{u}_j\|_2^2$$

Randomized Least Squares

Some theoretical results [Drineas 2011]

If $d = \mathcal{O}\left(l(\ln l)(\ln N) + \frac{l \ln N}{\epsilon}\right)$, then with probability 0.8

$$\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}_R\|_2 \leq (1 + \epsilon)\|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}_{LS}\|_2$$

$$\|\mathbf{x}_{LS} - \mathbf{x}_R\|_2 \leq \sqrt{\epsilon} \left(\kappa(\mathbf{A}) \sqrt{\gamma^{-2} - 1} \|\mathbf{x}_{LS}\|_2 \right)$$

Worse Case Results!

JL Transforms (\mathbf{R} Matrix)

- Johnson and Lindenstrauss (1984): Choose \mathbf{R} uniformly at random from the space of projection matrices.
- Frankl and Maehara (1988): Random orthogonal matrix
- Indyk and Motwani (1998), DasGupta and Gupta (1999): entries chosen uniformly at random from $\mathcal{N}(0, \frac{1}{N})$

$\mathcal{T}(\mathbf{R}\mathbf{A}) \rightarrow \mathcal{O}(dNl)$ operations

Accelerating Johnson–Lindenstrauss (JL) Transforms

Achlioptas (2003)

$$a_{i,j} = \begin{cases} +1, & \text{with probability } \frac{1}{2}, \\ -1, & \text{with probability } \frac{1}{2}. \end{cases}$$

$$a_{i,j} = \sqrt{\frac{3}{d}} \begin{cases} +1, & \text{with probability } \frac{1}{6}, \\ 0, & \text{with probability } \frac{2}{3}, \\ -1, & \text{with probability } \frac{1}{6}. \end{cases}$$

then if $d \geq \frac{4+2\beta}{\epsilon^2/2-\epsilon^3/3} \log(l)$, each pairwise distance is preserved with probability at least $1 - l^{-\beta}$.

Accelerating Johnson–Lindenstrauss (JL) Transforms

Fast JL Transforms (e.g. Sarlos 2006, Drineas et al 2011)

$$\mathbf{R} = \mathbf{P}\mathbf{H}\mathbf{D}$$

- $\mathbf{D} \in \mathbb{R}^{N \times N}$ diagonal matrix with ± 1
- $\mathbf{H} \in \mathbb{R}^{N \times N}$ Hadamard matrix (normalized)
- $\mathbf{P} \in \mathbb{R}^{d \times N}$ a sparse matrix (or simply a Sampling Matrix)

Accelerating Johnson–Lindenstrauss (JL) Transforms

Fast JL Transforms (e.g. Sarlos 2006, Drineas et al 2011)

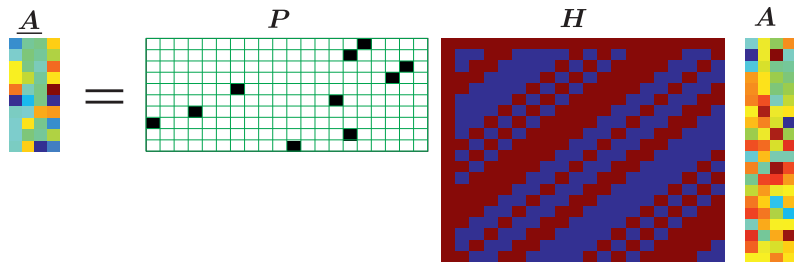
$$\mathbf{R} = \mathbf{PHD}$$

- $\mathbf{D} \in \mathbb{R}^{N \times N}$ diagonal matrix with ± 1
- $\mathbf{H} \in \mathbb{R}^{N \times N}$ Hadamard matrix (normalized)
- $\mathbf{P} \in \mathbb{R}^{d \times N}$ a sparse matrix (or simply a Sampling Matrix)

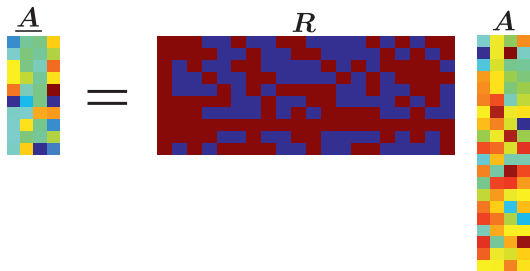
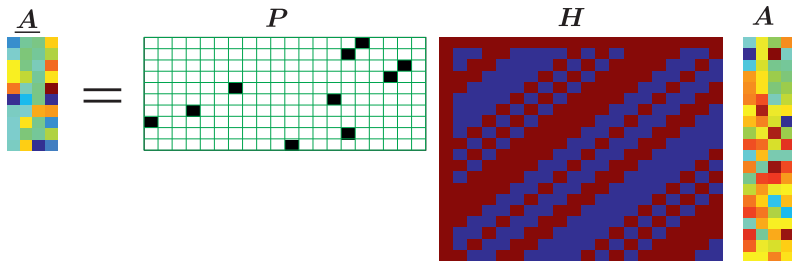
Computational Complexity / Facts

- It is called Randomized Hadamard Transform
- Multiplication with \mathbf{D} is just selective sign changes
- $\mathbf{H}\mathbf{a} \rightarrow \mathcal{O}(N \log k)$, where $k \leq N$, is the number Hadamard components needed
- Overall, $\mathbf{T}(\mathbf{R}\mathbf{A})$ takes $\mathcal{O}(lN \log d)$

Randomized projections

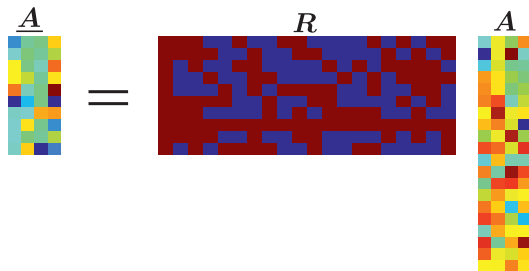


Randomized projections



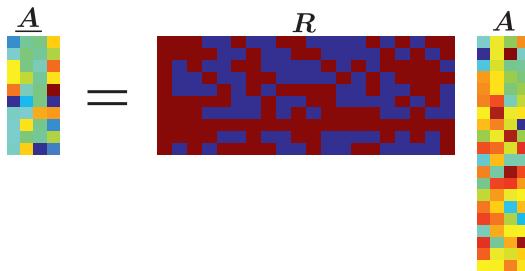
Randomized projections vs Randomized sampling

Randomized Projections

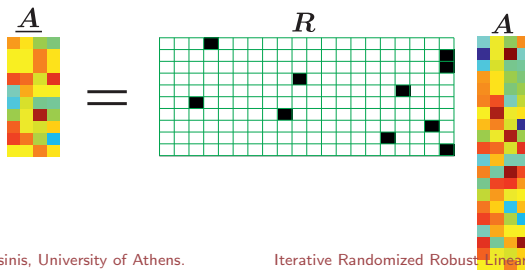


Randomized projections vs Randomized sampling

Randomized Projections



Randomized Sampling



Statistical Leverage

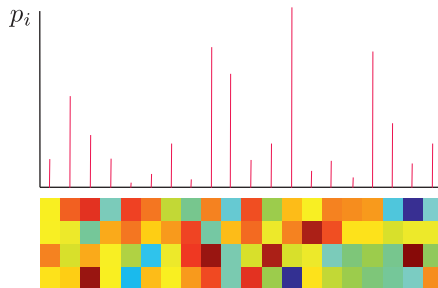
Statistical Leverage Scores

- Find any orthogonal matrix, \mathbf{U} spanning the column space of \mathbf{A} .

- $$l_i = \|\mathbf{U}_{i,\cdot}\|_2^2$$

- $l_i, i = 1 \dots N$ are referred to as **statistical leverage scores**

- Construct an importance sampling distribution $\{p_i\}_{i=1}^N$, with $p_i = \frac{l_i}{l}$.



Computation of the Statistical Leverage

“Naive way”

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

then \mathbf{U} is orthogonal spanning the column space of \mathbf{A} .

Alas, complexity $\mathcal{O}(N^2)$

Fast approximations (Drineas 2012)

- Exploit the fact that

$$l_i = \|(\mathbf{U}\mathbf{U}^T)_{i,\cdot}\|_2^2 = \|(\mathbf{A}\mathbf{A}^\dagger)_{i,\cdot}\|_2^2$$

- Construct two JL transform matrices, $\mathbf{\Pi}_1 \in \mathbb{R}^{r_1 \times N}$, $\mathbf{\Pi}_2 \in \mathbb{R}^{r_1 \times r_2}$
- Estimate leverage scores as

$$\hat{l}_i = \|(\mathbf{A}(\mathbf{\Pi}_1\mathbf{A})^\dagger\mathbf{\Pi}_2)_{i,\cdot}\|_2^2$$

- it is proved that $|l_i - \hat{l}_i| \leq \epsilon l_i, \forall i$

Robust Linear Regression

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \mathbf{n} + \mathbf{o}$$

$$\hat{\mathbf{x}}_{LAD} = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_1$$

- Least Absolute Deviations do not admit a closed form solution
- Linear programming using, e.g. interior-point methods $\mathcal{O}(\text{poly}(N))$
- Use approximate, iterative solutions, e.g. ADMM [Boyd 2011].

Robust Linear Regression

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \mathbf{n} + \mathbf{o}$$

$$\hat{\mathbf{x}}_{LAD} = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_1$$

- Least Absolute Deviations do not admit a closed form solution
- Linear programming using, e.g. interior-point methods $\mathcal{O}(\text{poly}(N))$
- Use approximate, iterative solutions, e.g. ADMM [Boyd 2011].

Randomized Sampling is better suited

$$\mathbf{R}\mathbf{b} = \mathbf{R}\mathbf{A}\mathbf{x}_* + \mathbf{R}\mathbf{n} + \mathbf{R}\mathbf{o}$$

- $\mathbf{R}\mathbf{o}$ is still sparse
- LAD can be applied
- Harder to compute the leverage scores
- Involves JL transforms with Cauchy distributed values (Fast Cauchy Transform [Clarkson et al 2014])

Reasoning behind our approach

Proposed Approach

- Apply a fast JL transform, $\underline{\mathbf{b}} = \mathbf{R}\mathbf{b}$, $\underline{\mathbf{A}} = \mathbf{R}\mathbf{A}$.
- **Progressively** clean the data from outliers **in the reduced dimensional space**
- Obtain final solution with ordinary LS.

In an ideal world...(I)

- Let $\Lambda \subset \{1, \dots, N\}$ be the index set indicating the corrupted data. **Assume Λ is known.**
- **Question:** Can we Clean the compressed data directly in the low dim domain?

$$\begin{aligned}\mathbf{R}\mathbf{b}_{\Lambda^c} &= \underline{\mathbf{b}} - \mathbf{R}_{\cdot, \Lambda} \mathbf{b}_{\Lambda} \\ \mathbf{R}\mathbf{A}_{\Lambda^c, \cdot} &= \underline{\mathbf{A}} - \mathbf{R}_{\cdot, \Lambda} \mathbf{A}_{\Lambda, \cdot}\end{aligned}$$

Reasoning behind our approach

In an ideal world...(II)

- Let randomized Hadamard transform be applied to the full data set

$$\underline{\mathbf{b}} = \underline{\mathbf{A}}\mathbf{x}_* + \underline{\mathbf{n}} + \mathbf{R}\mathbf{o}$$

- Assume that \mathbf{x}_* is known. Then

$$\boxed{\mathbf{z} = \mathbf{R}\mathbf{o} + \underline{\mathbf{n}}} \quad (1)$$

where \mathbf{z} is computed as $\underline{\mathbf{b}} - \underline{\mathbf{A}}\mathbf{x}_*$.

- Question:** Is it possible to estimate the support of \mathbf{o} , in the reduced dimensional space, based on (1)?
- Indeed, this is a typical compressed sensing scenario.

$$\hat{\mathbf{o}} = \min_{\mathbf{o}} (\|\mathbf{z} - \mathbf{R}\mathbf{o}\|_2) \text{ s.t. } \|\mathbf{o}\|_0 \leq K$$

- We only need to estimate the support (or a subset of it)

Reasoning behind our approach

Back in reality...

- Let randomized Hadamard transform is applied to the full data set

$$\underline{\mathbf{b}} = \underline{\mathbf{A}}\mathbf{x}_* + \underline{\mathbf{n}} + \mathbf{R}\mathbf{o}$$

- \mathbf{x}_* is not known but $\mathbf{R}\mathbf{o}$ is likely to be Normally distributed. so

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\underline{\mathbf{b}} - \underline{\mathbf{A}}\mathbf{x}\|_2$$

where $\hat{\mathbf{x}} = \mathbf{x}_* + \mathbf{x}_e$. Then, $\mathbf{z} = \underline{\mathbf{b}} - \underline{\mathbf{A}}\hat{\mathbf{x}}$

$$\mathbf{z} = \mathbf{R}\mathbf{o} + \underline{\mathbf{n}}'$$

- Request: Estimate **any part** of the support of \mathbf{o} .
- No Harm** to omit some healthy data as well!
- Suggestion: Just use the CoSaMP proxy, $\boldsymbol{\psi} = \mathbf{R}^T \mathbf{z}$, $\Lambda = \text{Supp}(|\boldsymbol{\psi}|, K)$

Iterative Randomized Robust LS: Concept

- Compress data: $\underline{\mathbf{b}} = \mathbf{R}\mathbf{b}$, $\underline{\mathbf{A}} = \mathbf{R}\mathbf{A}$
- **Start Iterations**
- Get a tentative estimate $\hat{\mathbf{x}}$ via $\arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\underline{\mathbf{b}} - \underline{\mathbf{A}}\mathbf{x}\|_2$
- Compute $\boldsymbol{\psi} = \mathbf{R}^T(\underline{\mathbf{b}} - \underline{\mathbf{A}}\hat{\mathbf{x}})$
- Define Λ as the set of indices of the K larger (in magnitude) components of $\boldsymbol{\psi}$.
Key remark: We are happy if Λ contains **some, not necessarily K** , outlier indices
- Exclude / Clear the data indexed in Λ from the compressed data set
$$\underline{\mathbf{b}} = \underline{\mathbf{b}} - \mathbf{R}_{\cdot, \Lambda} \mathbf{b}_{\Lambda}$$
$$\underline{\mathbf{A}} = \underline{\mathbf{A}} - \mathbf{R}_{\cdot, \Lambda} \mathbf{A}_{\Lambda, \cdot}$$
- **Return to hopefully get an improved $\hat{\mathbf{x}}$ or stop**

Proposed

- Once: $\mathcal{O}((l+1)N \log d)$
- Per iteration: $\mathcal{O}(dl^2) + d(l+1) + \mathcal{O}(Nd) + \mathcal{O}(N) + (dKl)$

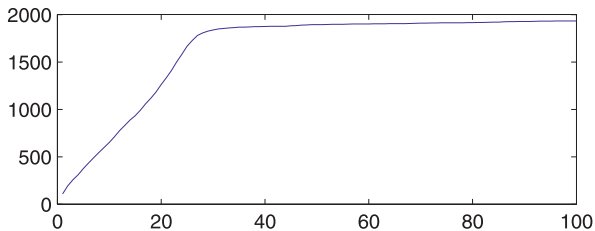
Random Sampling

- For the leverage Scores: $\mathcal{O}((l+1)N \log r_1 + lNr_2 + r_1l^2 + r_2l^2)$
- $r_1 = d$ and $r_2 = \mathcal{O}(\log l)$
- For LAD: $\text{poly}(d)$

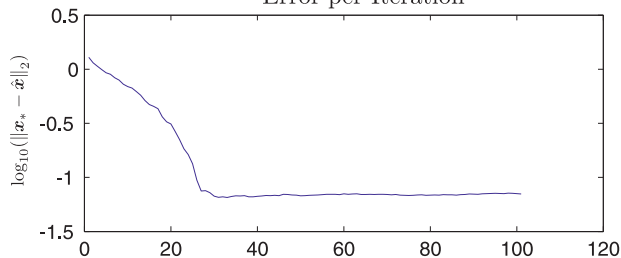
Some Results

$$N = 2^{17}, l = 100, S = 2000$$

Accumulated number detected outliers

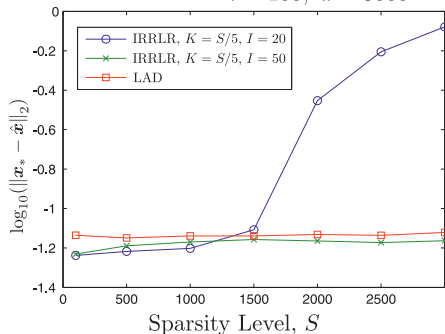


Error per Iteration

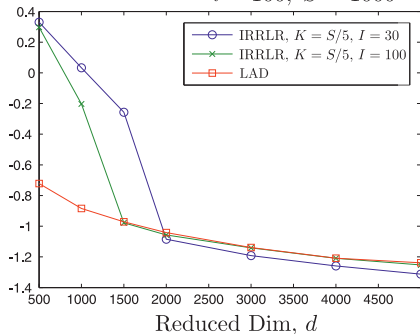


Some Results

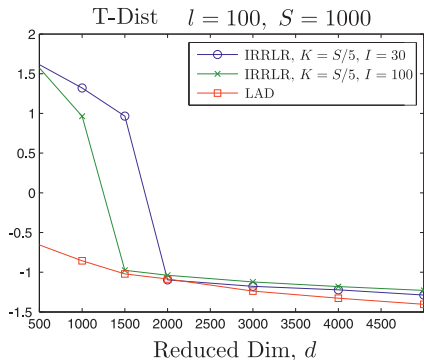
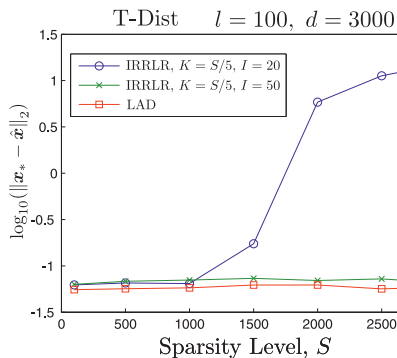
Normal Dist $l = 100, d = 3000$



Normal Dist $l = 100, S = 1000$



Some Results



- A New randomized-projections based Robust Linear regression technique was proposed
- Avoids computationally heavy LAD regression
- Offers competitive performance when the number of outliers is smaller than d
- Potential for improvements/extensions.

Thank you!

Hat Matrix

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \boldsymbol{\eta}$$

$$\hat{\mathbf{x}}_{LS} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

$$\hat{\mathbf{b}} = \mathbf{A} \hat{\mathbf{x}}_{LS} = \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \Rightarrow \hat{\mathbf{b}} = \mathbf{H} \mathbf{b}$$

Statistical Leverage Scores

- \mathbf{H}_{ij} measures the influence exerted on the prediction $\hat{\mathbf{b}}_i$ by observation \mathbf{b}_j
- $\ell_i = \mathbf{H}_{ii}$ measures the importance of \mathbf{b}_i in determining the best LS fit.
- $\ell_i, i = 1 \dots N$ are referred to as **statistical leverage scores**
- $\mathbf{H} = \mathbf{P}_A = \mathbf{U}\mathbf{U}^T$, for any orthogonal matrix spanning the column space of \mathbf{A} .

$$\ell_i = \|\mathbf{U}_{i,\cdot}\|_2^2$$

- Very large \mathbf{H}_{ii} are indicators for outliers in \mathbf{A} .

Sampling the column space is the key...

- Task: Let $\mathbf{A} \in \mathbb{R}^{n \times m}$, $\min_{\mathbf{X}: \text{rank}(\mathbf{X}=k)} \|\mathbf{A} - \mathbf{X}\|_F$
- **Randomized Projection based Range finder**
 - Generate matrix $\mathbf{R} \in \mathbb{R}^{m \times d}$
 - and compress: $\mathbf{Y} = \mathbf{A}\mathbf{R}$
 - some housekeeping: Replace \mathbf{Y} with \mathbf{Q} whose columns form an orthonormal basis for the range of \mathbf{Y}
- **SVD estimation in 3 steps**
- $\mathbf{B} = \mathbf{Q}^T \mathbf{A}$
- Compute low dimensional SVD: $\mathbf{B} = \tilde{\mathbf{U}} \tilde{\Sigma} \mathbf{V}^T$
- $\mathbf{U} = \mathbf{Q} \tilde{\mathbf{U}}$

Randomized Least Squares

Randomized Linear LS Regression

$$\underset{N \times 1}{\mathbf{b}} = \underset{N \times l}{\mathbf{A}} \underset{l \times 1}{\mathbf{x}_*} + \underset{N \times 1}{\boldsymbol{\eta}}$$

$$\underset{d \times 1}{\underline{\mathbf{b}}} = \underset{d \times N}{\mathbf{R}} \underset{N \times 1}{\mathbf{b}}, \quad \underset{d \times l}{\underline{\mathbf{A}}} = \underset{d \times N}{\mathbf{R}} \underset{N \times l}{\mathbf{A}}, \quad \text{where } d \ll N$$

$$\hat{\mathbf{x}}_R = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\underline{\mathbf{b}} - \underline{\mathbf{A}}\mathbf{x}\|_2^2,$$

Computational complexity: $\mathcal{O}(dl^2) + \mathcal{C}(\mathbf{R}) + \mathcal{T}(\mathbf{R}\mathbf{A})$

Randomized Least Squares

Randomized Linear LS Regression

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \boldsymbol{\eta}$$

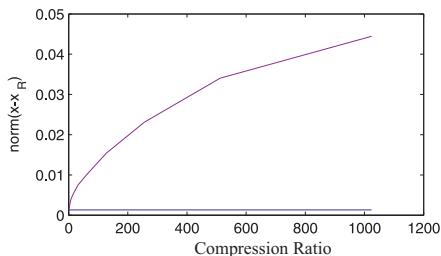
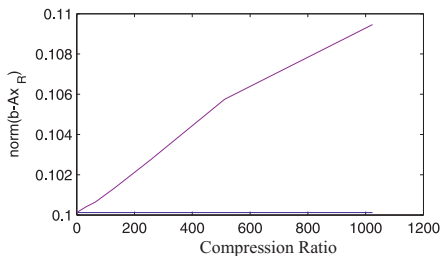
$N \times 1 \quad N \times l \quad l \times 1 \quad N \times 1$

$$\underline{\mathbf{b}} = \mathbf{R} \mathbf{b}, \quad \underline{\mathbf{A}} = \mathbf{R} \mathbf{A}, \quad \text{where } d \ll N$$

$d \times 1 \quad d \times N \quad N \times 1, \quad d \times l \quad d \times N \quad N \times l$

$$\hat{\mathbf{x}}_R = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\underline{\mathbf{b}} - \underline{\mathbf{A}}\mathbf{x}\|_2^2,$$

Computational complexity: $\mathcal{O}(dl^2) + \mathcal{C}(\mathbf{R}) + \mathcal{T}(\mathbf{R}\mathbf{A})$



Randomized Least Squares

Randomized Linear LS Regression

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \boldsymbol{\eta}$$

$N \times 1 \quad N \times l \quad l \times 1 \quad N \times 1$

$$\underline{\mathbf{b}} = \mathbf{R} \mathbf{b}, \quad \underline{\mathbf{A}} = \mathbf{R} \mathbf{A}, \quad \text{where } d \ll N$$

$d \times 1 \quad d \times N \quad N \times 1, \quad d \times l \quad d \times N \quad N \times l$

$$\hat{\mathbf{x}}_R = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\underline{\mathbf{b}} - \underline{\mathbf{A}}\mathbf{x}\|_2^2,$$

Computational complexity: $\mathcal{O}(dl^2) + \mathcal{C}(\mathbf{R}) + \mathcal{T}(\mathbf{R}\mathbf{A})$

Some theoretical results [Drineas 2011]

If $d = \mathcal{O}\left(l(\ln l)(\ln N) + \frac{l \ln N}{\epsilon}\right)$, then with probability 0.8

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_R\|_2 \leq (1 + \epsilon)\|\mathbf{b} - \mathbf{A}\mathbf{x}_{LS}\|_2$$

$$\|\mathbf{x}_{LS} - \mathbf{x}_R\|_2 \leq \sqrt{\epsilon} \left(\kappa(\mathbf{A}) \sqrt{\gamma^{-2} - 1} \|\mathbf{x}_{LS}\|_2 \right)$$

if $N \leq e^l$, and $\|\mathbf{U}_A \mathbf{U}_A^T \mathbf{b}\|_2 \geq \gamma \|\mathbf{b}\|_2, \gamma \in (0, 1]$

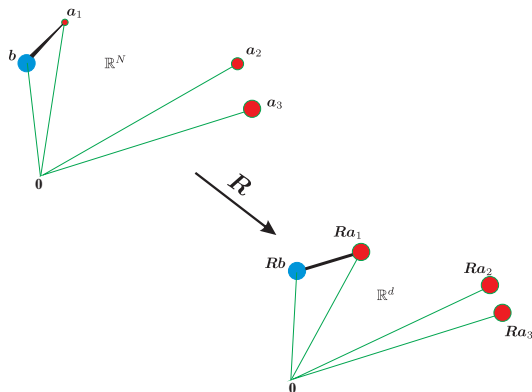
Johnson–Lindenstrauss (JL) seminal work

JL geometry in the Linear Regression case

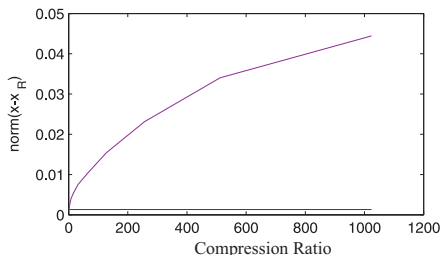
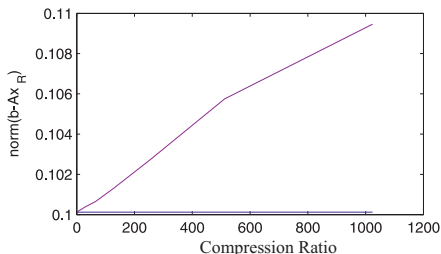
$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \boldsymbol{\eta}$$

$N \times 1$ $N \times l$ $l \times 1$ $N \times 1$

$$\hat{\mathbf{x}}_R = \arg \min_{\mathbf{x} \in \mathbb{R}^l} \|\mathbf{R}\mathbf{b} - \mathbf{R}\mathbf{A}\mathbf{x}\|_2^2,$$



Fast LS approximation Example



Randomized Hadamard Transform

Recall: $d = \mathcal{O}\left(l(\ln l)(\ln N) + \frac{l \ln N}{\epsilon}\right)$

- Example 1: $N = 10^6$, $l = 200$, $\epsilon = 0.1$.

$$\frac{Nl^2}{dl^2 + lN \log(d)} = 10, \quad \frac{N}{d} = 23$$

- Example 2: $N = 10^8$, $l = 1000$, $\epsilon = 0.1$.

$$\frac{Nl^2}{dl^2 + lN \log(d)} = 63, \quad \frac{N}{d} = 321$$

Hat Matrix

$$\mathbf{b} = \mathbf{A} \mathbf{x}_* + \boldsymbol{\eta}$$

$$\hat{\mathbf{x}}_{LS} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

$$\hat{\mathbf{b}} = \mathbf{A} \hat{\mathbf{x}}_{LS} = \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \Rightarrow \hat{\mathbf{b}} = \mathbf{H} \mathbf{b}$$

Statistical Leverage Scores

- \mathbf{H}_{ij} measures the influence exerted on the prediction $\hat{\mathbf{b}}_i$ by observation \mathbf{b}_j
- $\ell_i = \mathbf{H}_{ii}$ measures the importance of \mathbf{b}_i in determining the best LS fit.
- $\ell_i, i = 1 \dots N$ are referred to as **statistical leverage scores**
- $\mathbf{H} = \mathbf{P}_A = \mathbf{U}\mathbf{U}^T$, for any orthogonal matrix spanning the column space of \mathbf{A} .

$$\ell_i = \|\mathbf{U}_{i,\cdot}\|_2^2$$

- Very large \mathbf{H}_{ii} are indicators for outliers in \mathbf{A} .