Algorithms for Big Data (Fall 25)

Lecture 12

Applications of JL & Subspace Embedding

Ali Vakilian (vakilian@vt.edu)

Virginia Tech

Algorithms for Big Data
Fall 2025

Applications of Subspace Embedding

Regression

Applications of Subspace Embedding

Faster algorithms for approximate

- matrix multiplication
- regression
- SVD

Basic idea. Want to perform operations on matrix A with n data columns (in a large dimension $\mathbb{R}^h$) with small actual rank d.

Our goal is to reduce to a matrix of size roughly $\mathbb{R}^{d \times d}$ by spending time proportional to the number of non-zero entries in A.

Regression: Linear Model Fitting

A classic problem in **data analysis**

- n data points in $a_1, \cdots, a_n \in \mathbb{R}^d$
- Each data point a_i is associated with a value $b_i \in \mathbb{R}$

What model should one use to explain the data?

Simplest model? Linear fitting:

- $b_i = w_0 + \sum_{1 \le j \le d} w_j \cdot a_{i,j}$ for a vector $w \coloneqq (w_0, \cdots, w_d)$
- However, usually data is noisy and won't be able to satisfy for all data points
- Without loss of generality, we can restrict to $w_0 = 0$ by lifting to $d + 1$ dimensions

Regression

Goal: want to choose $w_1, \cdots, w_d$ to estimate $b_i \sim \sum_{1 \le j \le d} w_j \cdot a_{i,j}$

Let A be matrix with one row per data point a_i. We write $x_1, \ldots, x_d$ as variables for finding $w_1, \ldots, w_d$.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \ldots & a_{1,d} \\ a_{2,1} & a_{2,2} & a_{2,3} & \ldots & a_{2,d} \\ & & \vdots & & \\ a_{n,1} & a_{n,2} & a_{n,3} & \ldots & a_{n,d} \end{pmatrix}$$

Ideally: Find $x \in \mathbb{R}^d$ such that $Ax = b$

Best fit: Find $x \in \mathbb{R}^d$ to minimize $Ax - b$ under some norm

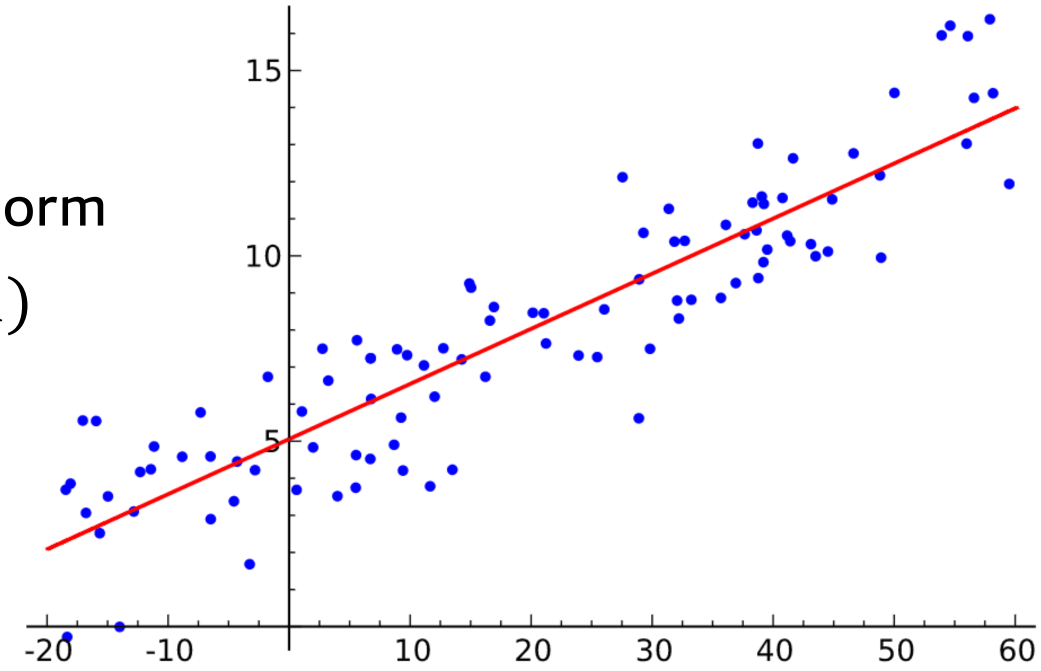
- $\|Ax - b\|_1, \|Ax - b\|_2, \|Ax - b\|_\infty$

Least Squares Error Regression

Given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^d$, find x to minimize $||Ax - b||_2$

Interesting when $n \gg d$; there is no solution to $Ax = b$ and want to find the best fit

- Ax is a linear combination of columns in A
- $z \in \text{colspace}(A)$ that is closest to b in ℓ_2-norm
- So, z is the projection of b onto $\text{colspace}(A)$



How to find it?

Least Squares Regression

Given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^d$, find x to minimize $\|Ax - b\|_2$

- Closest vector to b is the projection of b onto $\mathrm{colspace}(A)$
  - Find orthonormal basis $z_1, \ldots, z_r$ for the columns of A
  - Compute projection c of b to $\mathrm{colspace}(A)$ which is $c = \sum_{1 \le j \le r} \langle b, z_j \rangle \, z_j$
- Back to our question, what is x?
  - $Ax = c$. We need to solve the linear system.
  - By solving normal equation:
    - If columns of A are linearly independent, $A^\top A$ is full rank and $x^* = (A^\top A)^{-1} A^\top b$
    - Otherwise, there're multiple solutions and min-norm is $x^* = (A)^+ b$ (Moore-Penrose Pseudoinverse)

SVD & Moore-Penrose Pseudoinverse

- **Singular value decomposition (SVD).** For every $A \in \mathbb{R}^{m \times d}$, there exist matrices $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{d \times d}$ with orthonormal columns such that

$$A = U \, \Sigma \, V^\top ,$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_r, 0, \cdots, 0), \sigma_1 \geq \cdots, \geq \sigma_r > 0.$

Moore-Penrose Pseudoinverse of A is defined as $A^+ = V\Sigma^{-1}U^\top$ where $\Sigma = \mathrm{diag}(1/\sigma_1, \cdots, 1/\sigma_r, 0, \cdots, 0)$

- $AA^+A = A; (AA^+)^\top = AA^+$
- When A has linearly independent columns, then $A^+ = (A^\top A)^{-1}A^\top$
- AA^+: projection to the column span of A.
- $P \; = \; AA^+$; then $Pv = A(A^+v)$ which is a linear combination of columns in A
- Let $y \in colspan(A)$; $y = Ax$; then $Py = AA^+(Ax) = Ax = A$

Least Squares Regression

Given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^d$, find x to minimize $\|Ax - b\|_2$

- Closest vector to b is the projection of b onto $\mathrm{colspace}(A)$
  - Find orthonormal basis $z_1, \dots, z_r$ for the columns of A
  - Compute projection c of b to $\mathrm{colspace}(A)$ which is $c = \sum_{1 \le j \le r} \langle b, z_j \rangle \, z_j$
- Back to our question, what is x?
  - $Ax = c$. We need to solve the linear system.
  - By solving normal equation:
    - If columns of A are linearly independent, $A^\top A$ is full rank and $x^* = (A^\top A)^{-1} A^\top b$
    - Otherwise, there're multiple solutions and min-norm is $x^* = (A)^+ b$ (Moore-Penrose Pseudoinverse)
  - Naively requires $O(nd^2)$ time to compute

Can we speed up the process with some potential approximation?

LSE Regression via Subspace Embedding

Let E denote the subspace spanned by columns of A and b. It has dimension at most $d+1$.

Use Subspace Embedding Π on E with $k = O(d/\varepsilon^2)$ rows to reduce $\{A^{(1)}, A^{(2)}, \cdots, A^{(d)}, b\}$ to $\{A'^{(1)}, A'^{(2)}, \cdots, A'^{(d)}, b'\}$ which are in $\mathbb{R}^k$.

Solve $\min_{x' \in \mathbb{R}^d} \|A'x' - b'\|_2$

Lemma. With probability $1-\delta$,

$$(1-\varepsilon)\min_{x\in\mathbb{R}^d}\|Ax-b\|_2 \le \min_{x'\in\mathbb{R}^d}\|A'x'-b'\|_2 \le (1+\varepsilon)\min_{x\in\mathbb{R}^d}\|Ax-b\|_2$$

LSE Regression via Subspace Embedding

Lemma. With probability $1-\delta$,

$$(1-\varepsilon)\min_{x\in\mathbb{R}^d}\|Ax-b\|_2 \le \min_{x'\in\mathbb{R}^d}\|A'x'-b'\|_2 \le (1+\varepsilon)\min_{x\in\mathbb{R}^d}\|Ax-b\|_2$$

With probability $(1-\delta)$, via subspace embedding guarantee, for all $z \in E$,

$$(1-\varepsilon)\|z\|_2 \le \|\Pi z\|_2 \le (1+\varepsilon)\|z\|_2$$

- Let x^*, y^* be respectively the optimal solution to $\min_{x\in\mathbb{R}^d}\|Ax-b\|_2$ and $\min_{x'\in\mathbb{R}^d}\|A'x'-b'\|_2$
- Let $z = Ax^* - b$. Since $z \in E$, $\|Sz\|_2 \le (1+\varepsilon)\|z\|_2$.
- Since x^* is a feasible solution to $\min_{x'\in\mathbb{R}^d}\|A'x'-b'\|_2$,

$$\|A'y^*-b'\|_2 \le \|A'x^*-b'\|_2 \le (1+\varepsilon)\|Ax^*-b\|_2$$

- Since for any $y \in \mathbb{R}^d$, $\|A'y-b'\|_2 = \|\Pi Ay - \Pi b\|_2 \le (1+\varepsilon)\|Ay-b\|_2$

$$\|Ay^*-b\|_2 \le (1+\varepsilon)\|A'y^*-b'\|_2 \le (1+\varepsilon)\|A'x^*-b'\|_2 \le (1+3\varepsilon)\|Ax^*-b\|_2$$

Running Time

- Reduce the problem for d vectors in $\mathbb{R}^n$ to d vectors in $\mathbb{R}^k$ with $k = O(d/\varepsilon^2)$.
- Computing ΠA and Πb can be done in $nnz(A)$ via sparse/fast JL
- The reduced problem can be solved in time $O(d^3/\varepsilon^2)$
- Useful when $n \gg d/\varepsilon^2$

Approximate Matrix Multiplication

Matrix Multiplication

- A fundamental subroutine in countless computational tasks.
- Given two matrices $A \in \mathbb{R}^{n \times d}, B \in \mathbb{R}^{n \times p}$, we need to compute $A^\top B$
- The standard naïve approach requires $O(ndp)$.
- For the square matrices , significantly faster algorithms exist, running in $O(n^\omega)$, where ω is the exponent of Matrix Multiplication
  - $\omega \leq \log_2 7$ (Strassen)
  - $\omega \leq 2.376$ (Coppersmith, Winograd)
  - $\omega \leq 2.371339$ (Alman, Duan, Vassilevska Williams, Y. Xu, Z. Xu, Zhou)

Matrix Multiplication

Year	Best known (ω) bound	Authors
1969	2.8074	Strassen (Wikipedia)
1978	2.796	Pan (Wikipedia)
1979	2.780	Bini, Capovani, Romani (Wikipedia)
1981	2.522	Schönhage (Wikipedia)
1981	2.517	Romani (Wikipedia)
1981	2.496	Coppersmith, Winograd (Wikipedia)
1986	2.479	Strassen (laser method) (Wikipedia)
1990	2.3755	Coppersmith, Winograd (Wikipedia)
2010	2.3737	Stothers (Wikipedia)
2012	2.3729	Vassilevska Williams (Wikipedia)
2014	2.3728639	Le Gall (Wikipedia)
2020	2.3728596	Alman, Vassilevska Williams (SODA'21) (Wikipedia)
2022	2.371866	Duan, Wu, Zhou (FOCS'23) (Wikipedia)
2024	2.371552	Vassilevska Williams, Y. Xu, Z. Xu, Zhou (SODA'24) (arXiv)
2024	**2.371339** (current best)	Alman, Duan, Vassilevska Williams, Y. Xu, Z. Xu, Zhou (SODA'25) (arXiv)

Approximate Matrix Multiplication

- **Exact vs. Practical:** While theoretically fast, exact matrix multiplication can be complex.
- **Approx. is Often Enough:** In many modern applications (e.g., ML and data analysis), we don't need the perfect answer. A high-quality approx. is sufficient.
- **The Trade-Off:** This leads to **Approximate Matrix Multiplication**. We trade a small, controlled amount of precision for significant gains in speed.
- **The Formal Goal:** We want to quickly compute a matrix $C \in \mathbb{R}^{d \times p}$ that is close to the true answer, with a high probability of success.

$$\left\|A^\top B - C\right\|_F \leq \varepsilon, \text{ with probability } 1 - \delta$$

JL-Based Approach for Fast Matrix Mult.

Theorem. $\mathcal{D}$ is a DJL distribution of matrices with $O(\frac{1}{\varepsilon^2\delta})$ rows, then for A, B:

$$\Pr_{\Pi\sim\mathcal{D}}\left[\left\|A^\top B - (\Pi A)^\top(\Pi B)\right\|_F \geq \varepsilon\|A\|_F\|B\|_F\right] \leq \delta$$

Error Matrix:

- Let $M = A^\top B - (\Pi A)^\top(\Pi B)$;
- Our goal is to show that $\|M\|_F$ is small w.h.p.

Analyze a Single Entry of the Error Matrix:

- $M_{i,j} = \langle a_i, b_j\rangle - \langle \Pi a_i, \Pi b_j\rangle$
- Contribution of the entry to $\|M\|_F^2$ is $M_{i,j}^2 = \left(\langle \Pi a_i, \Pi b_j\rangle - \langle a_i, b_j\rangle\right)^2$

JL-Based Approach for Fast Matrix Mult.

Theorem. $\mathcal{D}$ is a DJL distribution of matrices with $O(\frac{1}{\varepsilon^2\delta})$ rows, then for A, B:

$$\Pr_{\Pi\sim\mathcal{D}}\left[\left\|A^\top B-(\Pi A)^\top(\Pi B)\right\|_F \geq \varepsilon\|A\|_F\|B\|_F\right] \leq \delta$$

Error Matrix: $M = A^\top B - (\Pi A)^\top(\Pi B)$;

Analyze a Single Entry of the Error Matrix: $M_{i,j} = \langle a_i, b_j\rangle - \langle \Pi a_i, \Pi b_j\rangle$

Use of Distributional JL Property: for unit norm vectors x, y, and Π with k rows,

$$\mathbb{E}[(\langle \Pi x, \Pi y\rangle - \langle x, y\rangle)^2] \leq 1/k$$

Bound the Expectation of the Error: bound the expected error, then apply Markov

Bound the Expected Error

$$\mathbb{E}\left[\|M\|_F^2\right] = \mathbb{E}\left[\sum_{i,j} M_{i,j}^2\right] = \sum_{i,j} \mathbb{E}\left[\left(\langle \Pi a_i, \Pi b_j \rangle - \langle a_i, b_j \rangle\right)^2\right]$$

- By the property of DJL: $\mathbb{E}\left[\left(\langle \Pi a_i, \Pi b_j \rangle - \langle a_i, b_j \rangle\right)^2\right] \le \frac{1}{k} \cdot \|a_i\|_2^2 \|b_j\|_2^2$
- So, summing over all i, j:

$\mathbb{E}\left[\|M\|_F^2\right] = \frac{1}{k} \cdot \|A\|_F^2 \cdot \|B\|_F^2$. Hence, applying Markov

$$\Pr\left[\|M\|_F^2 \ge \varepsilon^2 \cdot \|A\|_F^2 \cdot \|B\|_F^2\right] \le \frac{\mathbb{E}\left[\|M\|_F^2\right]}{\varepsilon^2 \cdot \|A\|_F^2 \cdot \|B\|_F^2} = \frac{\frac{1}{k} \cdot \|A\|_F^2 \cdot \|B\|_F^2}{\varepsilon^2 \cdot \|A\|_F^2 \cdot \|B\|_F^2} = \frac{1}{k\varepsilon^2}$$

- By choosing $k = O(1/(\varepsilon^2 \delta))$; the proof follows.

Runtime

The proposed Approximate Matrix Multiplication runs in

- Computing ΠA and ΠB in time $O(knd + knp)$
- Then, multiplying ΠA and ΠB in time $O(kdp)$
- Overall, runs in $O\big(k(nd + np + dp)\big) \ll O(ndp)$ as $k = O(1/(\varepsilon^2\delta))$

JL Map Preserves Dot Product

- Projection matrix $\Pi \in \mathbb{R}^{k\times d}$ have entries $\Pi_{r,c}$ that are independent r.v.s. with:
  - $\mathbb{E}[\Pi_{r,c}] = 0$
  - $\mathbb{E}[\Pi_{r,c}^2] = 1/k$ (this normalization simplifies the proof).
- $Z = \langle \Pi a, \Pi b\rangle = \sum_{r=1\ldots k} \langle \Pi_r, a\rangle\langle \Pi_r, b\rangle$

$$\begin{aligned}\mathbb{E}[Z] &= \textstyle\sum_{r=1\ldots k} \mathbb{E}[\langle \Pi_r, a\rangle\langle \Pi_r, b\rangle] \\ &= k \cdot \mathbb{E}[\langle \Pi_1, a\rangle\langle \Pi_1, b\rangle] = k \cdot \mathbb{E}\left[\left(\textstyle\sum_s a_s \pi_{1,s}\right)\left(\textstyle\sum_s b_s \pi_{1,s}\right)\right] \\ &= k \cdot \textstyle\sum_s a_s b_s \mathbb{E}\left[\pi_{1,s}^2\right] = \langle a, b\rangle\end{aligned}$$

- Similarly, we can bound the variance as $\mathrm{Var}(\langle \Pi a, \Pi b\rangle) \leq \frac{1}{k}\|a\|_2^2\|b\|_2^2$

So,

$$\mathbb{E}[(\langle \Pi a, \Pi b\rangle - \langle a, b\rangle)^2] = \mathbb{E}[(\langle \Pi a, \Pi b\rangle - \mathbb{E}[\langle \Pi a, \Pi b\rangle])^2] = \mathrm{Var}(\langle \Pi a, \Pi b\rangle) \leq \frac{1}{k}\|a\|_2^2\|b\|_2^2$$