New topic: Low-Rank Approximation via Randomized Algorithms

- We've learned a lot of methods to handle matrices:
  - When A has no structure: LU / QR / SVD
    (general but expensive)
  - When A is sparse (or Ax easy to evaluate): Krylov
    (sparse A usually arises in PDE problems)

- One of the most significant shifts in numerical analysis / applied math in recent years is the need to handle massive volumn of data.

  Challenge: 1) massive high-dim data sets / matrices

  2) The structure is less explicit in many cases

  3) Presence of noise and corruption in matrix entries

- How do we deal with high-dim data?

  Observation: high-dim data can often be approximated with low-rank matrices

$$
\begin{array}{cc} & A \\ m & \begin{bmatrix} \quad\quad & \quad\quad \\ & \\ & \end{bmatrix} \\ & n \end{array}
\approx
\begin{array}{cc} & B \\ m & \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix} \\ & k \end{array}
\cdot
\begin{array}{cc} C & \\ \begin{bmatrix} \quad\quad & \quad\quad \end{bmatrix} & k \\ n & \end{array}
$$

← cheaper to store and operate

$k < \min\{m, n\}$

Finding such B and C is not a new math problem.

We can formalize it as follows.

Goal: Given $A \in \mathbb{R}^{m \times n}$, $k < n$ (assume $m \geq n$)

Find $\min_{\operatorname{rank}(\hat{A}) \leq k} \| A - \hat{A} \|$ (*)

Here we take $\|\cdot\|$ to be 2-norm or Frobenius norm

Solution to (*) is given by the truncated SVD of A

Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ be singular value of A

$\mathbb{R}^{m\times m} \ni U = [u_1, u_2, \cdots, u_m]$ be left-singular vectors of A

$\mathbb{R}^{n\times n} \ni V = [v_1, v_2, \cdots, v_n]$ be right-singular vectors of A

$$A = U \Sigma V^T, \qquad \Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_n) \in \mathbb{R}^{m\times n}$$

$$= \sum_{i=1}^{n} \sigma_i u_i v_i^T \quad \longleftarrow \mathrm{rank}(A) = \#\{\text{nonzero } \sigma_i\text{'s}\}$$

Now we take the truncated SVD of A

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

$$= \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} - v_1^T - \\ \vdots \\ - v_k^T - \end{bmatrix} \in \mathbb{R}^{m\times n}$$

$$= \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix} \underbrace{\begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}}_{=:\Sigma_k} \begin{bmatrix} - v_1^T - \\ \vdots \\ - v_n - \end{bmatrix}$$

clearly $\mathrm{rank}(A_k) \leq k$,

$$\|A - A_k\|_2 = \|U(\Sigma - \Sigma_k)V^T\|_2 = \|\Sigma - \Sigma_k\|_2 = \sigma_{k+1}$$

$$\|A - A_k\|_F = \|\Sigma - \Sigma_k\|_F = \left(\sum_{j=k+1}^{n} \sigma_j^2\right)^{1/2}$$

<u>Thm</u> (Eckart-Young)

$$\min_{\mathrm{rank}(\hat{A}) \leq k} \|A - \hat{A}\|_2 = \sigma_{k+1}$$

$$\min_{\mathrm{rank}(\hat{A}) \leq k} \|A - \hat{A}\|_F = \left(\sum_{j=k+1}^{n} \sigma_j^2\right)^{1/2}$$

Pf: We prove the 2-norm case only.

It suffices to show that $\|A-\hat{A}\|_2 \geq \sigma_{k+1}$

for any $\mathrm{rank}(\hat{A}) \leq k$

It suffices to show that $\exists x \in \mathbb{R}^n$, s.t. $\frac{\|(A-\hat{A})x\|_2}{\|x\|_2} \geq \sigma_{k+1}$

I want to find $x \in \mathbb{R}^n$ such that $\hat{A}x = 0$

and $x \in \mathrm{span}\{v_1, \cdots, v_{k+1}\}$ $\left(x = \sum_{i=1}^{k+1} \alpha_i v_i, \ \sum_{i=1}^{k+1} \alpha_i^2 = 1 \right)$

$$\Rightarrow \frac{\|(A-\hat{A})x\|_2}{\|x\|_2} = \frac{\|Ax\|_2}{\|x\|_2} = \left\| \sum_{i=1}^{k+1} \sigma_i u_i \alpha_i \right\|_2 \geq \sigma_{k+1}$$

Such x always exists:

Since $\mathrm{rank}(\hat{A}) \leq k$, we know $\dim \mathrm{Null}(\hat{A}) \geq n-k$

but $\dim \mathrm{span}\{v_1, \cdots, v_{k+1}\} = k+1$

$$\Rightarrow \mathrm{Null}(\hat{A}) \cap \mathrm{span}\{v_1, \cdots, v_{k+1}\} \neq \phi$$

"The best rank k approximation of A is given by k-truncated SVD of A"

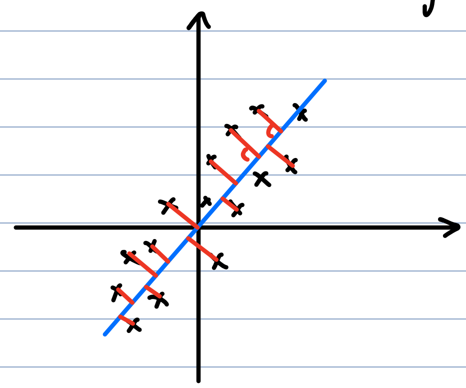
- An equivalent formulation of low-rank approximation is the Principal Component Analysis (PCA).

Let data points be m-dim vectors, stored in n columns of A. PCA aims to find k vectors whose span best contains the data points in A. We can assume these vectors are orthonormal basis and form the following problem:

$$\min_{\substack{Q \in \mathbb{R}^{m \times k} \\ Q^T Q = I_m}} \| A - \underbrace{QQ^T}_{=: P_Q} A \| \qquad (**)$$

P_Q projection onto span of Q

that is, $A \approx QQ^T A$, Q is an approximate basis for the range of A



PCA is essentially low-rank approximation:

- On one hand. $\displaystyle\min_{\text{rank}(\hat{A}) \le k} \|A - \hat{A}\| \le \min_{\substack{Q \in \mathbb{R}^{m \times k} \\ Q^T Q = I_m}} \|A - \underbrace{QQ^T A}_{\text{rank}(QQ^T A) \le k}\|$

- On the other hand, we take $Q = [u_1 \cdots u_k] \in \mathbb{R}^{m \times k}$

then $\underbrace{QQ^T}_{=:C} A = [u_1 \cdots u_k] \begin{bmatrix} - u_1^T - \\ \vdots \\ - u_k^T - \end{bmatrix} [u_1 \cdots u_m] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} - v_1^T - \\ \vdots \\ - v_n^T - \end{bmatrix}$

$= [u_1 \cdots u_k] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} - v_1^T - \\ \vdots \\ - v_k^T - \end{bmatrix} \longleftarrow$ truncated SVD of A

- How to efficiently compute low-rank approximation of A?
  - Apply SVD to A then truncate $\longleftarrow$ expensive! $O(mn^2)$
  - Workaround: two stage algorithm from PCA

    <u>Stage A</u>: compute orthonormal basis Q whose span approximates $\text{span}(A)$

$$\Rightarrow \quad A \approx QQ^TA \quad \text{and} \quad Q^TQ = I$$

<u>Stage B</u>: Compute $C = Q^TA \in \mathbb{R}^{k \times n}$ ← $O(kmn)$

then compute SVD of C (using whatever method):

$$B = \tilde{U}\Sigma V^T \quad \leftarrow O(k^2n)$$

then $A \approx (Q\tilde{U})\Sigma V^T$ ← $O(k^2m)$

- How to find Q? (Randomized "sketch")

Idea: Approximate $\text{span}\{u_1, \cdots, u_k\}$. the top k singular vectors of A, with a single power iteration

<u>Intuition 1</u>: Suppose A has exactly rank k, so the best rank k approximation of A is A itself. Suppose $\sigma_k > 0$

Compute $$Y = A\Omega, \quad \Omega \in \mathbb{R}^{n \times k}$$

$$[Y_1 \cdots Y_k] = [u_1 \cdots u_k]\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_k\end{bmatrix}\begin{bmatrix}- v_1^T - \\ \vdots \\ - v_k^T -\end{bmatrix}\underbrace{[\omega_1 \cdots \omega_k]}_{\omega_i \in \mathbb{R}^n \text{ random vectors}}$$

Y has linearly independent columns as long as $\text{rank}(V^T\Omega) = k$

For i.i.d normal entries of Ω, this happens almost surely.

$V^T\Omega$ still has iid normal entries

To get orthonormal basis, we compute $Y = QR$

Clearly $\|A - QQ^TA\|_2 = 0 = \sigma_{k+1}$ a.s.

<u>Intuition 2</u>: Now let $A = \underbrace{\sum_{i=1}^{k} \sigma_i u_i v_i^T}_{=: \hat{A}_k} + \underbrace{\sum_{j=k+1}^{n} \sigma_j u_j v_j^T}_{=: E}$

$(\|E\|_2 = \sigma_{k+1} \text{ small})$

then $\tilde{Y}_i = A\omega_i = \hat{A}_k\omega_i + E\omega_i$

$$\tilde{Y} = \underbrace{Y}_{\text{basis for } \operatorname{span}(\hat{A}_k)} + \underbrace{E\Omega}_{\text{small perturbation}}$$

The span of $\tilde{Y}$ is a good approximation to Y with high probability as long as we oversample, i.e. take $\Omega \in \mathbb{R}^{n\times(k+p)}$ instead (p is oversampling parameter)

<u>Algorithm</u>: (Randomized SVD)

<u>Stage A</u>: 1) Generate iid Gaussian random matrix

$$\Omega \in \mathbb{R}^{n\times(k+p)}$$

2) Compute $Y = A\Omega \in \mathbb{R}^{m\times(k+p)}$

3) Compute QR fact. $Y = QR$, $Q \in \mathbb{R}^{m\times(k+p)}$

<u>Stage B</u>: 1) Compute $B = Q^T A \in \mathbb{R}^{(k+p)\times n}$

2) Compute SVD : $B = \tilde{U}\Sigma V^T$

$\tilde{U} \in \mathbb{R}^{(k+p)\times(k+p)}$,

$\Sigma = \operatorname{diag}(\sigma_1, \cdots, \sigma_{k+p}) \in \mathbb{R}^{(k+p)^2}$

$V \in \mathbb{R}^{n\times(k+p)}$

3) Compute $U = Q\hat{U} \in \mathbb{R}^{m\times(k+p)}$

Output : $A \approx \sum_{i=1}^{k} \sigma_i u_i v_i^T$