DATA STRUCTURE AND ALGORITHM FOR MASSIVE DATASET

MINSEOK SONG

Three ways to deal with massive dataset

(1) Dimensional reduction: the purpose is to minimize the loss of information.
(2) Compressed representation: present data in a compact form, but not necessarily predicated on the retainment of information. i.e., it may prefer higher compression rates.
(3) Interpolation: only use discrete information of the distribution f. This is useful since we do not have a full function f available. Remember we used finite element method in numerical PDE, and the right space of function to discuss numerical stability etc was Sobolev space.

- All in all, it focuses on achieving lower computational/statistical complexity.
- To clarify, computational complexity deals with the resources(time and space), while statistical complexity with the intricacy of models(in the sense of how simpler model represents reduced data).

Theorem 1. *(Johnson-Lindenstrauss Lemma) Let Q be a finite set of vectors in $\mathbb{R}^d$. Let $\delta \in (0,1)$ and n be large enough integer such that*

$$\epsilon = \sqrt{\frac{6\log(2|Q|/\delta)}{n}} \leqslant 3 \tag{1}$$

With probability of at least $1-\delta$ over a choice of a random matrix $W \in \mathbb{R}^{n,d}$ such that each element of W is distributed normally with zero mean and variance of $1/n$ we have

$$\sup_{x\in\mathbb{Q}} |\frac{\|Wx\|^2}{\|x\|^2} - 1| < \epsilon \tag{2}$$

- We might think that W needs to be closer to identity matrix, but this lemma is all about preserving the distance.
- Do note that each element of W is generated by $N(0, 1/N)$, that k (the count of our vectors) is used here.
- The proof leans on the following lemma, which uses the concentration property of χ^2.

Lemma 2. *Fix some $x \in \mathbb{R}^d$. Let $W \in \mathbb{R}^{n,d}$ be a random matrix such that each $W_{i,j}$ is an independent normal random variable. Then, for every $\epsilon \in (0,3)$ we have*

$$\mathbb{P}[|\frac{\|(1/\sqrt{n})Wx\|}{\|x\|} - 1| > \epsilon] \leqslant 2e^{-\epsilon^2 n/6} \tag{3}$$

- Note that $W : \mathbb{R}^d \to \mathbb{R}^n$, and the result does not depend on d. This suggests that we can conduct dimensionality reduction in very high-dimensional spaces without much cost(!).
- This shows the existence of $T = \frac{1}{\sqrt{k}} \cdot R$ with $R \in \mathbb{R}^{k\times d}$, each element generated by $N(0,1)$, when $k \geqslant \Omega(\frac{\log k|Q|/\delta}{\epsilon^2})$, where k depends on δ as well.

Proof of Lemma 2. We can assume, WLOG, that $\|x\|^2 = 1$. Do note that $\|Wx\|^2$ has a χ^2_n distribution by construction, so we may use concentration of χ^2 inequality to get the result. □

Proof. In order to deal with $|Q|$, use the union bound. We can find appropriate ϵ afterward. □

- This says that the random projections do not distort Euclidean distances too much.

Efficient PCA We aim at solving the problem

$$\arg \min_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \|x_i - UWx_i\|_2^2$$

- It is then shown that the optimal solution is caculated by computing the eigenvectors of $A = \sum_{i=1}^{m} x_i^T x_i = XX^T$. This is the right eigenvectors of SVD. Do note that x_i is each column of X.
- This means the complexity is given by $O(d^3 + md^2)$
  (1) $O(d^3)$ for computing the eigenvectors and eigenvalues of $A = XX^T$.
  (2) $O(d^2 m)$ for computing the covariance matrix A.
- Instead of using XX^T, we can use the eigenvector of $B = X^T X$, that is, $A(X^T u) = \lambda(X^T u)$ where u is an eigenvector of B.
- This comes from the fact that $XX^T Xu = \lambda Xu$.
- Do note that B only requires calculating inner products $\langle x_i, x_j \rangle$.
- This reduces our complexity to $O(m^3 + dm^2)$, which is useful when d is very large.

Perturbation theory

- Instead of considering $Tx = X^*$, let's shift our focus on $X^* = X + \epsilon$.
- X^* is r-dimensional, and the QR decomposition gives $X^* = U^* z^*$, with X^* being orthonormal.
- Weyl's inequality relates to the singular values of perturbed matrix X^*.

Theorem 3 (Weyl's theorem)**.** *Let $X^* = X + \epsilon$. Then*

$$\|\tilde{\Lambda} - \Lambda\|_2 \leqslant \|\epsilon\|_2.$$

where Λ and $\tilde{\Lambda}$ are singular value matrices.

- This says that eigenvalue is stable under perturbation.

Theorem 4 (Wedin's theorem)**.** *Let $X = X^* + \epsilon$. Then*

$$\|U_{(r,X)} U_{(r,X)}^T - U_{(r,X^*)} U_{(X^*)}^T\|_F \leqslant \frac{\|\epsilon\|_F}{\sigma_r(X) - \sigma_{r+1}(X)}$$

- This theorem answers the question: if we slightly perturb our matrix, how much does the "important" subspaces (as captured by the dominant singular vectors) change? This change is inversely proportional to the gap at r'th singular value.
- In light of QR decomposition, we have $X^* = U^* z^*$ for some U^* with dimension $d \times r$ and z^* with dimension $r \times N$. Let $X = X^* + \Delta$. By Wedin's inequality, we have

$$\|U_r U_r^T |x^{(i)} - x^{(j)}|\| \leqslant (1 + O(\frac{\|\Delta\|_2}{\sigma_r(x^*)}))\|x^{(i)*} - x^{(j)*}\|_2 + O(\|\Delta\|_2)$$

- This gives the bound of the perturbation of $x^{(i)} - x^{(j)}$ in terms of Δ and r'th singular value.
- SVD has computational cost $O(dN^2)$ and JL has computational cost $O(dkN) = O(\frac{N \log N}{\epsilon^2})$
- We would still prefer the method in JL lemma.
- If the data matrix has rank r-dimensional, we can only require $O(\frac{\log r}{\epsilon^2})$ (?).
- Going forward, in summary, we use the perturbation theory + SVD to quantify the approximately dominant subspace of the matrix, and upgrade(?, wrong) JL lemma in reality using Weyl's/Wedin's lemma .

Proof of Theorem 3. First, assume that the matrix is Hermitian. We have

$$\lambda_n(A) = \max_{\|x\|=1} x^T Ax$$

for symmetric matrix A. It follows that $\min_{dim(A)n=i+1} \max_{x\in A,\|x\|=1} x^T Ax$. Using this, we can prove that

$$\lambda_i(A) + \lambda_j(B) \geqslant \lambda_{i+j-1}(A+B)$$

where each λ's are ordered. Similarly,

$$\lambda_i(A) + \lambda_j(B) \leqslant \lambda_{i+j-n}(A+B)$$

The first inequality shows that

$$|\lambda_i(A+B) - \lambda_i(A)| \leqslant \|B\|_2$$

which is equivalent to

$$|\lambda_i(X+\epsilon) - \lambda_i(X)| \leqslant \|\epsilon\|_2$$

in the setup of our theorem. We can generalize this by taking

$$\begin{pmatrix} 0 & M \\ M^* & 0 \end{pmatrix}$$

which gives

$$|\sigma_k(X+\epsilon) - \sigma_k(X)| \leqslant \sigma_1(\epsilon)$$

□

Fact 1. *If A is a bounded self-adjoint operator on a Hilbert space $\mathbb{H}$, then*

$$\|A\| = \sup_{\|x\|=1} |\langle x, Ax\rangle|$$

Proof. It suffices to show $\leqslant$. By definition, we have

$$\|A\| = \sup_{\|x\|=1} \|Ax\| = \sup\{|\langle y, Ax\rangle| \,|\, \|x\| = 1, \|y\| = 1\}$$

Now using polarization formula (choose appropriate y to delete imaginary part), we get,

$$|\langle y, Ax\rangle|^2 \leqslant \frac{1}{4}\alpha^2(\|x\|^2 + \|y\|^2)^2$$

□

- The polarization identity relates the inner product to the norms of linear combination of vectors. This is given by

$$\langle x, y\rangle = \frac{1}{4}(\|x+y\|^2 - \|x-y\|^2 + i\|x+iy\|^2 - i\|x-iy\|^2) = \sum_{k=0}^{3} \|x + i^k y\|^2.$$

Randomized SVD

- We are interested in finding an appropriate L_1 for $Y \approx L_1 L_1^T Y$.
- Setup:
  - m: original dimension
  - n: number of data
  - k: target dimension
  - r: random number.
  - $A : m \times n, L_1 : m \times k, L_2 : n \times k, V_r : n \times r, G : n \times k, G_1 : (n-r) \times k, G_2 : r \times k$
- We may proceed like this.
  (1) Multiply random matrix G on the right side of A ($O(mnk)$).
  (2) Perform QR decomposition, so that $YG = L_1 R$ ($O(k^2 m)$).
  (3) Compute $L_2 = L_1^T A$ and use $L_1 L_2 A$ ($O(mnk)$).
- Note that we have $L_1 L_1^T AG = L_1 L_1^T L_1 R = L_1 R = AG$.
- In what sense is $L_1 L_1^T A \approx A$?

(1) We can use Gordon's inequality (in random matrix theory) to show that

$$E(\|A - L_1 L_1^T A\|) \leqslant [1 + O(\frac{\sqrt{n} + \sqrt{k}}{\sqrt{k} - \sqrt{r}})]\|\Sigma_{>r}\|$$

(2) We can do "better" for Frobenius norm:

$$E(\|A - L_1 L_1^T A\|_F^2) \leqslant (1 + O(\frac{\sqrt{r}}{\sqrt{k} - \sqrt{r}})^2)\|\Sigma_{>r}\|_F^2$$

- w
- Since n is large, this can be costly; some variants to do this faster.
  (1) Structured random matrix; apply DFR decomposition on G, and we can calculate AG in better time complxity (this involves FFT)
  (2) Interpolation; use CUR decomposition.
  (3) Verify if the matrix has certain structures (like Toeplitz).
  (4) Process by blocks.
  (5) Adaptive methods; start with a small rank and increase it adaptively.
  (6) Perform SVD on a smaller matrix.
  (7) Power iterations algorithm.

Compressed Sensing

- This method juxtaposes PCA method.

  Definition 1. *A matrix $W \in \mathbb{R}^{n,d}$ is $(\epsilon, s) - RIP$ (with $\epsilon < 1$) if for all $x \neq 0$ s.t. $\|x\|_0 \leqslant s$ we have*

$$|\frac{\|Wx\|_2^2}{\|x\|_2^2} - 1| \leqslant \epsilon$$

  Theorem 5. *If $W \in \mathbb{R}^{n,d}$ is an $(\epsilon, 2s) \sim RIP$ matrix, and if x has less than s many nonzero elements, then x is the sparsest vector that gets mapped to Wx by the matrix W.*

  Proof. Assume $\tilde{x} \neq x$. Observe that $\|x - \tilde{x}\| \leqslant 2s$ and $W(x - \tilde{x}) = 0$. On the other hand, $|0 - 1| \leqslant \epsilon$, contradicting the definition of RIP matrix. □

- This implies that, for specific compression matrices and sparse data, the original data can be accurately recovered.

  Theorem 6. *Let $\epsilon < \frac{1}{1+\sqrt{2}}$ and let W be a $(\epsilon, 2s)$-RIP matrix. Let x be an arbitrary and let x_s be the vector which equals x on the x largest elements of x and equals 0 elsewhere. Let $x^* \in \arg\min_{v:Wv=y} \|v\|_1$. Then*

$$\|x^* - x\|_2 \leqslant 2\frac{1+\rho}{1-\rho} s^{-1/2} \|x - x_s\|_1$$

  where $\rho = \sqrt{2}\epsilon/(1-\epsilon)$.

  Remark 1. In particular, we have

$$x = \arg\min_{v:Wv=y} \|v\|_0 = \arg\min_{v:Wv=y} \|v\|_1$$

  for $\|x\|_0 \leqslant s$.

- We used L^1 since we are looking for solutions that are "almost sparse" rather than strictly sparse. Further, L^1 is convex so we can compute efficiently.

  Theorem 7. *Let $\epsilon, \delta \in (0,1)$. Let U be orthonormal matrix of size $d \times d$. Further, let $W \in \mathbb{R}^{n,d}$ be generated by $N(0, 1/n)$ where $n \geqslant 100\frac{s\log(40d/(\delta\epsilon))}{\epsilon^2}$ and $s \in [d]$. Then the matrix WU is (ϵ, s)-RIP.*

Remark 2. This is useful when the sparsity is hidden, i.e. $y = U\alpha$ where y is sparse.

- Connection of PCA with compressed sensing: While PCA identifies the dominant subspace in which most of the data's energy or variance lies, compressed sensing exploits the fact that signals often have sparse representations in some domain. These two concepts are related in the sense that they both exploit inherent structures in data (low-rank structure and sparsity, respectively) for efficient processing or recovery.
- As another note, we can also exploit the rank of the data matrix.