DATA STRUCTURE AND ALGORITHM FOR MASSIVE DATASET

MINSEOK SONG

Contents

1 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 n (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.

2 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) $O(d^2m)$ 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.

3 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
$$\|U_r U_r^T |x^{(i)} - x^{(j)}||| \leqslant (1 + O(\frac{\|\Delta\|_2}{\sigma_r(x^*)}))\|x^{(i)*} - x^{(j)*}\|_2 + O(\|\Delta\|_2)$$
- Even 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

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

4 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 $AG = L_1 R$ ($O(k^2 m)$).
  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":

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

- Analogous statement exists for spectral norm explained here.
- Let us take $A_k = L_1 L_1^T \in \mathbb{R}^{m\times m}$. Let us take $X^* = U^* z^*$ and $X = X^* + \Delta$. By the above inequality, we have

$$\begin{aligned}\|U_r^* U_r^{*T} - U_r U_r^T\|_F &\leqslant \frac{\Delta_F + C(k,r)\|\Sigma_{>r}(X)\|_F}{\sigma_r(X^*) - \sigma_{r+1}(X_k)} \\ &\leqslant \frac{\Delta_F + C(k,r)\|\Sigma_{>r}(X)\|_F}{\sigma_r(X^*) - (\sigma_{r+1}(X^*) + \|\Delta\|_F + C(k,r)\|\Sigma_{>r}(X)\|_F)} \\ &\leqslant \frac{\Delta_F + C(k,r)\|\Sigma_{>r}(X)\|_F}{\sigma_r(X^*) - (\sigma_{r+1}(X^*) + \|\Delta\|_F + C(k,r)\|\Sigma_{>r}(X)\|_F)}\end{aligned}$$

  where U_r comes from X, U^* comes from X^*.
- The second inequality comes from Weyl, and the third inequality from the inequality we acquired above.
- 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.

5 Kernel trick

- Imagine
$$\min_w f(\langle w, \psi(x_1)\rangle, \dots, \langle w, \psi(x_m)\rangle) + R(\|w\|)$$
where $f : \mathbb{R}^m \to \mathbb{R}$ is an arbitrary function, $R : \mathbb{R}_+ \to \mathbb{R}$ is a monotonically nondecreasing function, and ψ is a mapping from $\mathcal{X}$ to a Hilbert space.

Theorem 5. *(Representer Theorem) $w = \sum_{i=1}^m \alpha_i \psi(x_i)$ for some $\alpha \in \mathbb{R}^m$ gives an optimal solution.*

Proof. Use the property of Hilbert space, namely the orthogonalization based on the subspace. When use the monotonicity of $\|w\|$. □

- Note that ψ is an intereseted embedding function to the higher dimensional space.

Proof. Use the fact that Hilbert space has a basis (due to Gram-Schmidt), and express $w^* = \sum_{i=1}^m \alpha_i \psi(x_i) + u$ where w^* is an optimal solution. We have, by construction,
$$\|\sum_{i=1}^m \alpha_i \psi(x_i)\| \leqslant \|w^*\|$$
On the other hand, we have
$$f(\langle w, \psi(x_1)\rangle, \dots, \langle w, \psi(x_m)\rangle) = f(\langle w^*, \psi(x_1)\rangle, \dots, \langle w^*, \psi(x_m)\rangle)$$
Combining these, it follows that w gives an optimal solution. □

- Now substituting this formula, we have
$$\min_{\alpha \in \mathbb{R}^m} f(\sum_{j=1}^m \alpha_j K(x_j, x_1), \dots, \sum_{j=1}^m \alpha_j K(x_j, x_m)) + R(\sqrt{\sum_{i,j=1}^m \alpha_i \alpha_j K(x_j, x_i)})$$
where $K(x, x') = \langle \psi(x), \psi(x')\rangle$
- Note that we now only need to calculate the value of K.

Definition 1. *$G_{i,j} = K(x_i, x_j)$ is called Gram matrix.*

Example 1. Consider k degree polynomial kernel defined to be $K(x, x') = (1 + \langle x, x'\rangle)^k$. Then we have
$$K(x, x') = \sum_{J \in \{0,1,\dots n\}^k} \prod_{i=1}^k x_{J_i} \prod_{i=1}^k x'_{J_i} = \langle \psi(x), \psi(x')\rangle$$
by putting
$$\psi(x) = [\prod x_{J_{1,i}}, \prod x_{J_{2,i}}, \dots, \prod x_{J_{(n+1)^k,i}}]$$

Example 2. Let $K(x, x') = \exp(-\frac{\|x-x'\|^2}{2\sigma})$. By setting $\psi(x) = \frac{1}{\sqrt{n!}} \exp(-\frac{x^2}{2}) x^n, n = 0, 1, \dots,$ we have
$$\langle \psi(x), \psi(x')\rangle = \exp(-\frac{\|x - x'\|^2}{2})$$

- In gerneral, symmetric K is Kernel if and only if the associated gram matrix is positive semidefinite.
- For a given Kernel, we usually perform eigenvalue decomposition, which leaves us with infinite dimensional space, which is not suitable computationally.

- As a precursor, in order to decopose Kernel, we use the philosophy of interpolative decomposition (which doesn't work quite well for sparse matrix) and the detail of the kernel approximation is explained here.
- Polynomial approximation to kernel can be found here.

Kernel approximation

- We want to approximate a kernel $K : \Omega \times \Omega \to \mathbb{R}$. Let $[b_1, b_2, \dots, b_m]$ be polynomial basis, with each $b_i \in \mathbb{R}^{\infty \times m}$.
- To do that: say $\|K(x,y) - \sum\limits_{i,j=1}^{m} b_i(x)\tilde{\alpha}_{ij} b_j(y)\|_{L^2(\Omega\times\Omega)} \leqslant \epsilon_A^m$, an approximation error. Try to find $\tilde{\alpha}$ such that
$$K \approx B\tilde{\alpha}B^T$$
- Suppose B satisfies $\langle b_i, b_j \rangle = \delta_{ij}$.
- We can choose $\hat{\alpha}$ by truncating B to finite dimension. This procedure involves integration error. That is,
$$\tilde{\alpha} = \langle b_i, (Kb_j)\rangle \approx \sum_{l=0}^{m} b_i(q_l)(Kb_j)(q_l)w_l + \epsilon_I = \hat{\alpha}_{ij} + \epsilon_I$$
- It follows that
$$\|K - B\hat{\alpha}B^T\|_{L^2} \leqslant \epsilon_A + \epsilon_I$$
Let $K = K_m + E$. We have
$$\begin{aligned}\hat{\alpha} &= \sum_{l=0}^{m} b_i(q_l)(Kb_j)(q_l)w_l \\ &= \sum_{l=0}^{m} b_i(q_l)(K_m b_j)(q_l)w_l + b_i(q_l)(Eb_j)(q_l)w_l\end{aligned}$$
Note that
$$\sum_{ij=1}^{m} b_i(q_l)(Eb_j)(q_l)w_l$$
is bounded by approximation error scaled by m since $E = K - K_m$.
- All in all, what we do is to 1) find a basis B (approximation error) and 2) pick a weight corresponding to nodes (integration error).
- In the context of linear algebra, what we did is to
  1. First approximate by picking B: $B\tilde{\alpha}B^T = K$
  2. approximate the corresponding α using quadrature and weight: $\bar{B}\hat{\alpha}\bar{B}^T = \bar{K}$ where $\bar{B} = [\sum_l b_{i,j}(q_l)]$ and $\bar{K} = [K(q_i, q_j)]$
- We can approximate Gaussian kernel by polynomials and get an error $m \sim O(1/\epsilon)$. This seems to involve the smoothness of the kernel though. In fact, we can impose less regularity on Kernel, which motivates CUR. Notice also the form $\bar{B}\hat{\alpha}\bar{B}^T$, which reminds us of CUR decomposition.

6 Connection to CUR

- Let's deviate a bit. In CUR decomposition, what happens if C and R are degenerate? The following existential theorem gives the answer for the worst case.

Theorem 6 (ref)**.** *Let $A = A_r + F$ where $rank(A_r) \leqslant r$ and $\|F\|_2 \leqslant \epsilon = \sigma_{r+1}(A)$. Then there exists r column rows index I, J, and coefficient matrix G with dimension $r \times r$ such that*

$$\|A - CGR\|_2 \leqslant \epsilon(1 + \sqrt{\|\hat{U}^{-1}\|_2} + \sqrt{\|\hat{V}^{-1}\|_2})$$

where $A_r = U\Sigma V^T$, $\hat{U} = U(I, \cdot)$, and $\hat{V} = V^T(J, \cdot)$.

Proof.

$$\begin{aligned} A_r - CGR &= (UU^T + U_\perp(U_\perp)^T)(A_r - CGR)(VV^T + V_\perp(V_\perp)^T) \\ &= \end{aligned}$$

□

- The following proposition gives a further refinement of this bound.

  Proposition 7. *Let I be chosen such that $|det(U(I))|$ is maximized for $U \in \mathbb{R}^{m\times r}$. Then*

  $$\frac{1}{\sigma_{\min}(U(I, \cdot))} \leqslant \sqrt{r(m-r)+1}$$

  Proof. □

- Going back to what we needed for performing CUR decomposition, we require,
  1. find top singular vectors U, V (in an attempt to get a better error bound).
  2. find I, J via maximum volume on U, V (in an attempt to minimize the inverse of singular value).
- Even with the cost, CUR has interpretability advantage since we're using the column verbatim.
- With RSVD, the first procedure still takes $O(n^2)$; Second procedure is also combinatorially hard.
- We can bypass the second one by using greedy algorithm (”good” locally).
- We want to avoid computing U and V. Can we do that?

  Theorem 8. *Consider 2 by 2 block matrix $\{A_{ij}\}$. Suppose that A_{11} has maximal value over all r by r principal submatrix of A. Then*

  $$\|A_{22} - A_{21}A_{11}^{-1}A_{12}\|_\infty \leqslant (r+1)\sigma_{r+1}(A)$$

  Proof. (sketch) Let us prove in the case where A_{21}, A_{22} and A_{12} are one by one. We have $A^{-1} = \frac{1}{det(A)}adj(A)$. Define $(r+1) \times (r+1)$ matrix $A_2 = \begin{pmatrix} A_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$. Since A_{11} has maximum determinant, maximum value is achieved in $(r+1, r+1)$ index. It follows that

  $$\sigma_{r+1}(A)^{-1} = \|A^{-1}\|_2 \leqslant (r+1)|A^{-1}_{r+1,r+1}| = (r+1)|a_{22} - a_{21}A_{11}^{-1}a_{12}|^{-1}$$

  Rearranging the term, we get the result. □

- Finding the maximum submatrix is NP hard problem, so let us address this by usign greedy algorithm.

7 Greedy Adaptive Cross Approximation with complete pivot

1. **Input:** Initialization of matrix A.
2. **Output:** Sets I and J.
3. **Initialize:** $I = J = \varnothing$, $R_0 = A$.
4. **Iterate:**
   - (a) **Selection step:** Compute

     $$(i_k, j_k) = \arg\max_{i\notin I, j\notin J} |R_{k-1}(i,j)|$$

(b) **Update step:** Update the sets I and J:

$$I = I \cup \{i_k\}, \quad J = J \cup \{j_k\}$$

(c) **Pivot step:** Set the pivot value p_k:

$$p_k = R_{k-1}(i_k, j_k)$$

(d) **Matrix update:** Update the matrix R_k:

$$R_k = R_{k-1} - \frac{1}{p_k} R_{k-1}(\cdot, j_k) R_{k-1}(i_k, \cdot)$$

(5) **Repeat:** until some stopping criterion is met.

- Heuristically, the last step is like subtracting rank-1 approximation.

8 Intuition and Analysis of GACA

(i) intuition

- This corresponds to LU factorization with pivoting.
- Related to CUR decomposition, let us assume that $A = A(\cdot, J)A(I,J)^{+}A(I,\cdot) + \begin{pmatrix} 0 & 0 \\ 0 & A^{(k)} \end{pmatrix}$.
  Let $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ and $A_{11} = L_{11}U_{11}$. By simply rewriting the above formula, it follows that
  $$A = \begin{pmatrix} L_{11} & 0 \\ L_{21} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & A^{(k)} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & I \end{pmatrix}$$
  where $A^{(k)} = A_{22} - A_{21}A_{11}^{-1}A_{12}, L_{21} = A_{21}U_{11}^{-1}$ and $U_{12} = L_{11}^{-1}A_{12}$
- Therefore, if we stop at step k, we can pull out LU factorization of it.
- The crucial part is that we do not need to repeat LU factorization for A_{11} for each iteration.
- We can use the above equality to show that finding (i_k, j_k) in the greedy procedure is same as picking arg $\max\limits_{i \notin I, j \notin J} \det(A(I \cup i, J \cup j))$; just note that
  $$det(A(\tilde{I}, \tilde{J})) = det(L_{11})det(U_{11})|A^{(k)}(i,j)|$$
- This does not guarantee the global maximization of the volume of the submatrix, but we're picking the best possible index in each iteration(adding one more column and row).

(ii) analysis

Definition 2. $\rho_r = \sup_{rank(A) > r} \frac{\|A^{(r)}\|_\infty}{\|A\|_\infty}$ *is called growth factor.*

Theorem 9. *Let $A \in \mathbb{R}^{n \times n}$ be at least rank r. Then*

$$\|A - A(\cdot, J)A(I,J)^{-1}A(I,\cdot)\|_\infty \leqslant 4^r \rho_r \sigma_{r+1}(A)$$

- Usually, $\sigma_{r+1}(A)$ decreases exponentially.

  Proof. Let $I = \{1, \ldots, r\}$ and $J = \{1, \ldots, r\}$. Let $A_{11} = A[\tilde{I}, \tilde{J}]$. As before, we have
  $$A = \begin{pmatrix} \tilde{L}_{11} & 0 \\ \tilde{L}_{21} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & A^{(r+1)} \end{pmatrix} \begin{pmatrix} \tilde{U}_{11} & \tilde{U}_{12} \\ 0 & I \end{pmatrix}$$
  The key step involves using lemma related to the above equality, and use induction to get the diagonal elements of L_{11} and U_{11} (and their characteristics, namely that the diagonal elements of L_{11} is maximum in its column). We then use the lemma by Higham which gives the crucial bound, $\|A_{11}^{-1}\| \leqslant 4^r \min\{|p_1|, \ldots, |p_{r+1}|\}^{-1}$. The rest is to use the definition of growth factor. □
- Due to maximization step, we generally get $O(rn^2)$ complexity.

- With additional assumption for positive semi definite matrix, we have $O(rn)$.

9 Compressed Sensing

- Motivation: Say $y = Ax$, an underdetermined problem. Our goal is to compressed into y and reconstruct x. Two main question arises:
  (1) What matrices A should we use?
  (2) How can we reconstruct?
- The problem at hand is that we do not know the structure of x at hand. For example we only know that it is sparse, but doesn't quite know exactly the exact index.
- So to answer your question, one reason we might use the Fourier transform in compressed sensing is to transform the signal into a domain where its sparsity is more apparent, making it easier to find a sparse representation of the signal.

10 Compressed Sensing (2)

- This method juxtaposes PCA method.

Definition 3. *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 10. *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 unique 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 11. *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 for the s 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 3. 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 12. *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 4. 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.

11 Compressed Sensing (3)

- We need a number of points exponential in d in the approximation scheme of Kernel.
- We are only solving for $[\alpha_{ij}]_{i,j}$, which are m^2 coefficients, maybe we may require less points in some situation.
- Let $A_{ij} = b_j(q^{(i)})$ and $y_i = f(q^{(i)})$, and we want to solve $y_i = A\alpha$.
- in general, N, p have exponential scaling with respect to the dimension.
- The situation we're looking for is when α has "low complexity."
- Let α^* is s-sparse with $supp(\alpha^*) = S$. To determine $\alpha^* \in \mathbb{R}^d$, we need for A to satisfy RNP (restricted null space property), that is,

$$null(A) \cap \{\Delta \in \mathbb{R}^p | \|\Delta_S\|_1 \geqslant \|\Delta_{S^C}\|_1\} = \{0\}$$

Theorem 13. *Let α^* be s-sparse and we have $A\alpha = y$. Solving*

$$\min \|\alpha\|_1 \text{ such that } A\alpha = y$$

has a unique solution α iff A satisfies RNP.

Proof. For backward direction, it suffices to show that $\{\alpha | \|\alpha\|_1 \leqslant \|\alpha^*\|_1, A\alpha = y\} = \{\alpha^*\}$. The idea is to set $\Delta = \tilde{\alpha} - \alpha^*$ and show that $\Delta = 0$ using RNP. For forward direction, let $x \in null(A)$ and notice that $Ax_S = -Ax_{S^C}$, and so by uniqueness $\|x_{S^C}\|_1 > \|x_S\|_1$. It follows that $x = 0$. This means that A satisfies RNP. □

- In practice, people usually engineer such A to find α.
- Sufficient conditions for RNP
  1. Pairwise incoherence of A

     $$\|A_S^T A_S - I\|_\infty < \frac{1}{2s}, \forall |S| \leqslant s$$

     Proof. Let $\alpha = \alpha_S + \alpha_{S^C}$ where $\alpha \in null(A)$. So $A(\alpha_S + \alpha_{S^C}) = 0$, that is, $A\alpha_S = -A\alpha_{S^C}$. By algebraic manipulation, we get

     $$\|\alpha_S\|_1 \leqslant \frac{s\delta}{1 - s\delta}\|\alpha_{S^C}\|_1$$

     Since $\delta < \frac{1}{2s}$, we have $\|\alpha_S\|_1 < \|\alpha_{S^C}\|_1$. We showed that if $\alpha \in null(A)$, then $\|\alpha_S\|_1 < \|\alpha_{S^C}\|_1$. So A satisfies RNP. □

     - Asking for something more than S might be too much, so we restrict it to S.
     - This $A^T A$ situation in general comes up in regression problem; in $A^T Ax = A^T y$ we want $A^T A$ to be close to identity in order to recover x.
     - By JL lemma, we have $\|\frac{G^T G}{N} - I\|_\infty \leqslant \sqrt{\frac{\log p}{N}} = \epsilon$.
     - This is because in JL lemma, we have $N - O(\frac{\log p}{\epsilon^2})$. This achieves $N \sim s^2$, but in general, we want $N \sim s$.
  2. Restricted Isometry property: let $\delta_s = \max_{|S|=s} \|A_S^T A_S - I\|_2$. We require $\delta_{2S} < \frac{1}{3}$ to have a unique recovery.

Lemma 14. *Let supp(u)=S, supp(v)=T and supp(u)* $\cap$ *supp(v)* $= \phi$*. Then we have* $|\langle Au, Av\rangle| \leqslant \delta_{s+t}\|u\|_2\|v\|_2$

Proof. (lemma) Use the fact that $|\langle A_S u_S, A_S v_S\rangle| = |\langle (A_S^T A_S - I)u_S, v_S\rangle \leqslant \|A_S^T A_S - I\|_2\|u_S\|_2\|v_S\|_2$. □

Proof. (theorem) Let α satisfies $A\alpha = 0$. We want to show $\|\alpha_S\|_1 < \|\alpha_{S^C}\|_1$, or equivalently, $\|\alpha_S\|_1 < \frac{1}{2}\|\alpha\|_1$. To this end, we may reorder α so that the first s entires are the biggest s elements, and the next s entires are the next biggest s entries, and so on. The key step is as follows. First note that

$$\|\alpha_{S_0}\|^2 \leqslant \frac{1}{1-\delta_{2S}}\|A\alpha_{S_0}\|^2$$

We use the lemma to arrive $\leqslant \dfrac{\delta_{2s}}{1-\delta_{2s}}\|\alpha_{S_0}\| \sum\limits_{2k\geqslant 1}\|\alpha_{S_k}\|_2$. Further, the construction of α yields $\leqslant \dfrac{\delta_{2S}}{1-\delta_{2S}}\|\alpha_{S_0}\|_2 \sum\limits_{k\geqslant 1}\dfrac{\|\alpha_{S_{k-1}}\|}{\sqrt{s}}$. □

- The Iterative Hard Thresholding algorithm is given by the following steps:

$$a_{t+1} = z_t - A^T(Az_t - y), \tag{4}$$

$$z_{t+1} = P(a_{t+1}), \tag{5}$$

  where $P(a_{t+1}) = \min\limits_{z \text{ is s-sparse}} \|z - a_{t+1}\|$.
- This works when we know that the solution is s-sparse. Refer to here, page 76.