COMS E6998-9: Algorithms for Massive Data (Spring'19) Feb 12, 2019

Lecture 7: Numerical Linear Algebra: Least Square Regression, Fast Dimension Reduction

Instructor: *Alex Andoni* Scribes: *Yue Luo*

1 Review

- **Distributional Johnson-Lindenstrauss (DJL):**

  For any $\epsilon \in (0, \frac{1}{2})$, and given $\delta > 0$, $k = O(\frac{\lg \frac{1}{\delta}}{\epsilon^2})$, there $\exists$ a random linear mapping $\varphi : \mathbb{R}^n \to \mathbb{R}^k$, *s.t.*

  $$\forall x \in \mathbb{R}^n,\ Pr_{\varphi}(||\varphi(x)||_2 \in (1 \pm \epsilon)||x||_2) \geq 1 - \delta$$

  For any case, the mapping function $\varphi(x) = Sx$, and S is randomized.

- **Corollary(JL):**

  $$\forall x_1, ..., x_N \in \mathbb{R}^n,\ Pr(\forall x_i, x_j, ||\varphi(x_i) - \varphi(x_j)||_2 \in (1 \pm \epsilon)||x_i - x_j||_2) \geq 1 - \frac{1}{N}, when\ k = O(\frac{\lg N}{\epsilon^2}).$$

  The idea behind this is that, all points x_i are in $\mathbb{R}^n$ which is a high dimension. If we only care about the computation related to distances between the points, we can reduce their dimension by such mapping, which still preserves the distances with high probability.

Today we are going to see its applications in Numerical Linear Algebra.

2 Least Square Regression (LSR)

Definition 1. *Given A: $n \times d$ matrix, $b \in \mathbb{R}^n$, LSR is to solve*

$$x^* = \arg\min_{x \in \mathbb{R}^d} ||Ax - b||_2$$

Direct Solution: $\nabla_x ||Ax - b||_2^2 = 0 \Leftrightarrow A^T(Ax^* - b) = 0 \Leftrightarrow x^* = (A^T A)^{-1} A^T b$

Time: Compute the optimal solution is at least the time for computing $(A^T A)^{-1}$. Assume invertible, the time is O(nd^2) by direct computation, and can be further reduced to O(nd^{w-1}) by faster matrix multiplication where w=2.36. How can we go faster?

Observation 2. *Reduce time by allowing approximation: Find x' s.t.*

$$||Ax' - b||_2 \leq (1 + \epsilon)||Ax^* - b||_2.$$

Main Approach:

Suppose we are computing $\min_{x \in \mathbb{R}^d} ||Ax - b||_2$ approximately. We go faster by reducing the dimension and computing $\min_{x \in \mathbb{R}^d} ||\varphi(Ax - b)||_2$, where $\varphi(y) = Sy$ and $\varphi : \mathbb{R}^n \to \mathbb{R}^k$. Applying such transformation, it is to approximately compute

$$\min_{x \in \mathbb{R}^d} ||S(Ax - b)||_2 = \min_{x \in \mathbb{R}^d} ||SAx - Sb||_2 = \min_{x \in \mathbb{R}^d} ||A^{'}x - b^{'}||_2$$

where $A^{'}$ is a $k \times d$ matrix and $b^{'} \in \mathbb{R}^k$.

Then we got a LSR problem where $A^{'}$ has dimension $k \times d$.

Time: (Dim Reduction Time) + $O(kd^2)$. We will prove that the first term is not too large.

Proof. For original x^*,

$$||S(Ax^* - b)||_2 \leq (1 + \epsilon)||Ax^* - b||_2$$

So x^* is still a possible solution.

For $x \neq x^*$,

$$||S(Ax - b)||_2 \geq (1 - \epsilon)||Ax - b||_2 \geq (1 - \epsilon)||Ax^* - b||_2$$

Each of these statements happen for $\forall x$ with $Prob \geq 99\%$. ($\delta = 0.01$). But this is not enough since the probability is for any fixed x and it is about the random S. It means that we may still encounter some x $s.t.$ $||S(Ax - b)||_2$ is extremely small.

What we still need is

$$Pr(\forall x \in \mathbb{R}^d, ||S(Ax - b)||_2 \in (1 \pm \epsilon)||Ax - b||_2) \geq 1 - \delta$$

□

Theorem 3. ***Obilivious Subspace Embedding (OSE)***

*If $\varphi(y) = Sy$ satisfies **DJL**, then for $\forall$ d-dim subspace $U \subset \mathbb{R}^n$,*

$$Pr_{\varphi}(\forall y \in U, ||\varphi(y)||_2 \in (1 \pm \epsilon)||y||_2) \geq 1 - \delta, \ for \ k = O(\frac{d}{\epsilon^2} \cdot \lg \frac{1}{\delta})$$

Proof. Omitted here. You will need to prove it in the homework. □

Fact 4. *$Pr(\forall x \in \mathbb{R}^d, ||S(Ax - b)||_2 \in (1 \pm \epsilon)||Ax - b||_2) \geq 1 - \delta$ as we mentioned above follows from OSE for $\delta = 0.01$.*

U = span{columns of A, b} = $\{Ax - b; x \in \mathbb{R}^d\}$, is a (d+1)-dim subspace.

Then by OSE, we have

$$k = O(\frac{d + 1}{\epsilon^2} \lg \frac{1}{0.01}) = O(\frac{d}{\epsilon^2})$$

Analysis of Runtime Again:

$$\begin{aligned}
&(Dim\ Reduction\ Time) + O(kd^2)\\
&= Time(SA) + O(kd^2)\\
&= O(knd) + O(kd^2)\\
&= O(\frac{nd^2}{\epsilon^2}) + O(\frac{d^3}{\epsilon^2})
\end{aligned}$$

But this approximation method is even slower than LSR of original setting which takes O(nd^2). We hope to further reduce the time of dimension reduction to a time proportional to A, say O(nd). If A is sparse, we'd better get to O($nnz(A)$), where nnz stands for number of non-zero elements.

3 Fast Dimension Reduction

Theorem 5. ***The Fast Johnson-Lindenstrauss Transformation (FJLT) [Ailon-Chazelle '06]***

$$Given\ \varphi(x) = Sx, \varphi : \mathbb{R}^n \to \mathbb{R}^s, s = O(\frac{lg\frac{1}{\delta}}{\epsilon^2} \cdot \lg \frac{n}{\delta})$$

We have

$$for\ x \in R^n,\ Pr_\varphi(||Sx||_2 \in (1 \pm \epsilon)||x||_2) \geq 1 - \delta$$

The time for this is $O(n \lg n) + O(s)$.

Corollary 6. *For SA, we need to do this dimension reduction on every column of A, which takes time*

$$d(O(n \cdot \lg n) + O(s)) = O(nd \cdot \lg n) + O(sd)$$

Proof. In DJL, we are computing the multiplication of a dense Gaussian matrix S with x $\in \mathbb{R}^n$. The time will be proportional to the size of the matrix S. The idea here is to let Sx be the subsample of coordinates of x. Let's say we subsample s coordinates from x $\in \mathbb{R}^n$.

Pick $j_1, j_2, ..., j_s \in [n]$. Then $(Sx)_i$ is defined as $x_{j_i}, i = 1, 2, ..., s.$.

Now we look at $||Sx||_2^2$.

$$E[||Sx||_2^2] = E[\sum_{i=1}^{s} (Sx)_i^2] = sE[(Sx)_1^2] = s \cdot \frac{\sum_{i=1}^n x_i^2}{n} = \frac{s}{n}||x||_2^2$$

$$Var[||Sx||_2^2] = s \cdot Var[(Sx)_1^2] \leq s \cdot E[(Sx)_1^4] = s \cdot \frac{\sum_{i=1}^n x_i^4}{n} = \frac{s}{n}||x||_4^4$$

In order for $||Sx||_2^2$ to be a good estimator of $||x||_2^2$, we want

$$\sqrt{Var[||Sx||_2^2]} \ll \epsilon E[||Sx||_2^2]$$
$$\Rightarrow \sqrt{\frac{s}{n}}||x||_4^2 \ll \epsilon \frac{s}{n}||x||_2^2$$
$$\Rightarrow s \gg \frac{n}{\epsilon^2} \cdot \frac{||x||_4^4}{||x||_2^4}$$

Suppose $x = (1, 0, 0, ..., 0)$, we need to have $s \gg \frac{n}{\epsilon^2}$. This is not dimension reduction. If the mass is concentrated in some place of vector x, it is very unlikely those elements will be picked by subsampling.

Suppose $x = (1, 1, 1, ..., 1)$, we need to have $s \gg \frac{n}{\epsilon^2} \cdot \frac{n}{n^2} = \frac{1}{\epsilon^2}$. And this can be a good sampling.

More precisely, notice that $\|x\|_4^4 = \sum_i x_i^4 \leq (\max_j x_j)^2 \sum_i x_i^2 = \|x\|_\infty^2 \|x\|_2^2$. Thus, if $s \gg n/\varepsilon^2 \cdot \|x\|_\infty^2/\|x\|_2^2$, then $s \gg n/\varepsilon^2 \cdot \|x\|_4^4/\|x\|_2^4$. So we reach the conclusion that if x is spread around, the subsampling works.

Lemma 7. *Let P be a $s \times n$ matrix, s.t. $(Px)_i = X_{j_i}, j_i \in [n]$ randomly. Then*

$$\forall y \in \mathbb{R}^n,\ Pr(\frac{n}{s}||Py||_2^2 \in (1 \pm \epsilon)||y||_2^2) \geq 1 - \delta,\ for\ s = O(\frac{\lg \frac{1}{\delta}}{\epsilon^2} \cdot \frac{n \cdot ||y||_\infty^2}{||y||_2^2})$$

Here, $\frac{n \cdot ||y||_\infty^2}{||y||_2^2}$ is the quantization of spreadness. If $y = (1, 0, 0, ..., 0)$, this term will be n, meaning that we need to sample by a factor of n more. And if $y = (1, 1, 1, ..., 1)$, this term is 1, which does not have this factor.

Proof. From Chernoff. □

We want to first spread out x to be y first, and the subsampling on y could be faster. The idea behind that is to do a structured random rotation.

Define y $= H \cdot D \cdot x$, and $\varphi(x) = P \cdot H \cdot D \cdot x$. Here,

- P: sparse projection with dim $s \times n$.
- D: diagonal matrix with ±1 on its diagonal.
- H: Hadamard Matrix (Fourier Transform Matrix) with dim $n \times n$.

Fact 8. *H: $H_0 = (1)$; $H_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$; $H_t = \frac{1}{\sqrt{2}}\begin{pmatrix} H_{t-1} & H_{t-1} \\ H_{t-1} & -H_{t-1} \end{pmatrix}$ of dim $2^t \times 2^t$.*

Properties:
- It is a rotation: $||Hx||_2 = ||x||_2$.
- Can compute Hx in O($n \cdot \lg n$) time.
- If x is not spread around, then Hx is. He_1 has l_∞ norm = O($\frac{1}{\sqrt{n}}$). (Uncertainty Principle)

But there exists some vector that after processed by the Hadamard Matrix, the result becomes sparse. Try to avoid such cases, we include the diagonal matrix D.

Lemma 9. *$y = H \cdot D \cdot x$, assume $||x||_2 = 1$, then*

$$||y||_\infty \leq O(\sqrt{\frac{\lg n}{n}})\ with\ Prob \geq 1 - \frac{1}{n}$$

□