CS 395T: Sublinear Algorithms Fall 2014

Lecture 22 — Nov 11, 2014

Prof. Eric Price *Scribe: Taewan Kim*

1 Overview

In the last few lectures we covered

1. Fourier Transform
2. Sparse Fourier Transform
3. Fourier RIP

In this lecture, a new topic 'Oblivious Subspace Embeddings' is covered, especially algorithms introduced by Clarkson and Woodruff [CW13] for regression and low rank approximation problems.

2 Application

Oblivious Subspace Embedding (OSE) is a tool for faster numerical linear algebra. There are two possible applications where OSE can be applied: regression and low rank approximation.

2.1 Regression

Problem Statement Given $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, Find $x \in \mathbb{R}^d$ minimizing $\|Ax - b\|_2$. ($n \gg d$)

A is given data composed of n rows of size d which indicates the d different features. And for those n items, vector b is composed of n outcomes corresponding to each $1 \times d$ feature vector. By finding solution $x \in$ minimizing $\|Ax - b\|_2$, we can find an approximate linear mapping between A and b via x: $Ax \approx b$.

This problem can be relaxed by allowing ϵ error:

$$\text{Find } x' \text{ } s.t. \text{ } \|Ax' - b\|_2 \leq (1+\epsilon)\min_x \|Ax - b\|_2$$

An algorithm to find optimal solution of the regression problem ($\epsilon = 0$) is using Moore-Penrose pseudoinverse [1].

Algorithm

1. $x = A^+ b$, (A^+ is pseudoinverse of A)

[1] Details of Moore-Penrose pseudoinverse can be found in Wikipedia or Chapter 4 of Laub, Alan J. *Matrix analysis for scientists and engineers.* Siam, 2005.

2. When $rank(A) = d$ and $d \ll n$, $A^+ = (A^T A)^{-1} A^T$

Time complexity of this algorithm to calculate $x = A^+ b$ is $\mathcal{O}(d^2 n + d^3) = \mathcal{O}(d^2 n)$ when $d \ll n$. To speed up, sparsity of A can be utilized when A is sparse. If $nnz(A)$ represents the number of nonzero elements in A, time complexity can be improved to $\mathcal{O}(d \cdot nnz(A) + d^3)$. The actual computation should be done as follows: compute $A^T x$ first, which gives $\mathcal{O}(nnz(A))$, then compute $(A^T A)^{-1}$ which gives $\mathcal{O}(d^3)$, and finally compute $(A^T A)^{-1}(A^T x)$.

However, by using OSE of [CW13] one can achieve:

- $\mathcal{O}(nnz(A)) + \tilde{\mathcal{O}}(d^3/\epsilon^2)$
- $\mathcal{O}(nnz(A)\log(1/\epsilon)) + \tilde{\mathcal{O}}(d^3 \log(1/\epsilon))$
  (Here, $\tilde{\mathcal{O}}(f) \triangleq f \cdot \log^{\mathcal{O}(1)}(f)$)

2.2 Low Rank Approximation

Problem Statement Given a matrix $A \in \mathbb{R}^{n \times n}$, find a matrix B with $rank(B) = k$ which minimizes $\|A - B\|_F^2$.

This low rank approximation problem with Frobenius norm can also be relaxed by allowing ϵ error:

$$\text{Find } B' \text{ } s.t. \text{ } \|A - B'\|_F^2 \leq (1+\epsilon) \min_{\substack{B \\ rank(B)=k}} \|A - B\|_F^2$$

When $\epsilon = 0$, Singular Value Decomposition (SVD) gives the best rank-k approximation of A by selecting top k singular values and corresponding singular vectors. SVD requires $\mathcal{O}(n^3)$ of computational time.

However by using Power method/subspace iteration:

- Each iteration takes $\mathcal{O}(n^2 k)$ time.
- For Frobenius norm approximation, bound is not known.
- By allowing spectral error, $\tilde{\mathcal{O}}(n^2 k/\epsilon^2)$ is possible per iteration.

Also, utilizing OSE can give better time bound:

- $\mathcal{O}(nnz(A)) + \tilde{\mathcal{O}}(nk^2/\epsilon^4 + k^3/\epsilon^5)$

For a dense matrix A, rank-k matrix approximation using random projection was introduced by [Sarlos06, CW09].

3 Oblivious Subspace Embedding

Definition 1. *Defined on parameters $(m, n, d, \epsilon, \delta)$. An Oblivious Subspace Embedding (OSE) is a distribution on matrices $S \in \mathbb{R}^{m \times n}$, s.t. $\forall$ d-dimensional subspace U of $\mathbb{R}^n$, with probability $1 - \delta$ over S, we have $\forall x \in U$ that $\|Sx\|_2 = (1 \pm \epsilon)\|x\|_2$*

3.1 Regression with OSE

Now, we can solve the problem in easier way with lower dimension using OSE. Rather than solving $x^* = \arg\min_x \|Ax - b\|$, solve:

$$
\begin{aligned}
x' &= \arg\min_x \|SAx - Sb\| \\
&= \arg\min_x \|S(Ax - b)\|
\end{aligned}
$$

where $(Ax - b) \in Col(A \circ b)$.
($Col(A \circ b)$ means a column space of A adjoined with the vector b)

Then from the definition of OSE,

$$
\frac{\|Ax' - b\|}{\|Ax^* - b\|} \leq \left(\frac{1+\epsilon}{1-\epsilon}\right) \frac{\|S(Ax' - b)\|}{\|S(Ax^* - b)\|} \leq \left(\frac{1+\epsilon}{1-\epsilon}\right) \lesssim 1 + 3\epsilon \tag{1}
$$

Computational time is determined by "$Embedding\ time + Solve(m, d)$", where $Solve(m, d)$ represents the time to solve new regression problem with size $m \times d$ of SA and $m \times 1$ vector Sb.

One example of OSE is Gaussian random matrix which can be defined as:

$$
S_{i,j} = \mathcal{N}(0, 1/m) \tag{2}
$$

With Gaussian OSE, $m = \mathcal{O}(d/\epsilon^2)$. Therefore, embedding requires $\mathcal{O}(mnd) = \mathcal{O}(d^2n/\epsilon^2)$ and $Solve(d, m)$ requires $\mathcal{O}(d^3/\epsilon^2)$ computational time. So, total time is $\mathcal{O}(d^2n/\epsilon^2 + d^3/\epsilon^2)$.

3.2 Fast Johnson-Lindenstrauss

Now, we introduce an important lemma, which is called Johnson-Lindenstrauss (JL) lemma.

Definition 2 (Johnson-Lindenstrauss Lemma)**.** *If $m = \mathcal{O}((1/\epsilon^2)\log(1/\delta))$, then*

$$
\forall x,\ \|Ax\|_2^2 = (1 \pm \epsilon)\|x\|_2^2 \qquad w.p.\ 1 - \delta
$$

Think as this way: given d-dim. subspace U, take ϵ-net: $C = (1 + 1/\epsilon)^d$ points. If $m = \mathcal{O}((1/\epsilon^2)\log(1/\delta))$, then all are preserved, i.e. C can be covered.

$$
\begin{aligned}
x = x_1 &+ \epsilon x_2 + \epsilon^2 x_3 + \cdots \text{ for } x_1, \cdots \in C \\
\Rightarrow \|Ax\|_2 &\geq \|Ax_1\| - \epsilon\|Ax_2\| - \epsilon^2\|Ax_3\| - \cdots \\
&\geq 1 - \epsilon - (1+\epsilon)(\epsilon + \epsilon^2 + \cdots) \\
&\geq 1 - 3\epsilon
\end{aligned}
$$

Faster version of Johnson-Lindenstrauss embedding technique was introduced by [KW11]:

If A has RIP of order k, then AD has $(\epsilon, 2^{-k})$ JL property, where

$$
D = \begin{bmatrix} \pm 1 & & & \\ & \pm 1 & & \\ & & \ddots & \\ & & & \pm 1 \end{bmatrix}
$$

.

Last class, it is shown that $F_{\Omega\in[n]}$ satisfies (k,ϵ) RIP if $|\Omega| \geq (1/\epsilon^2)k\log^4 n$. So, if $m = |\Omega|$ is greater than $(d/\epsilon^2)\log(1/\epsilon)\log^4 n$, then subspace embeddings with $m = (d/\epsilon^2)\log^5 n$, and computational time is $n\log n$. So, with Fast JL, embedding requires $\mathcal{O}(dn\log n)$ and $Solve(m,d)$ requires $\mathcal{O}((d^3/\epsilon^2)\log^5 n)$.

3.3 [CW13]

To improve the complexity, [CW13] used the sparsity of A. In each column of S, exactly one element has ± 1 value defined with hash functions:

$$\begin{aligned} &h : [n] \to [m] && \leftarrow \text{2-independent} \\ &\sigma : [n] \to \{\pm 1\} && \leftarrow \text{4-independent} \end{aligned}$$

therefore OSE matrix S is defined as,

$$S_{h(i),i} = \sigma_i$$

Let's prove that above S is OSE by showing:

$$a, b \in \mathbb{R}^n \Rightarrow \langle Sa, Sb\rangle \approx \langle a, b\rangle$$

Proof. Denote $\delta_{r,i} = \mathbb{I}_{h(i)=r}$ (indicator function).

$$\begin{aligned} \langle Sa, Sb\rangle &= \sum_{r=1}^{m}\left[\left(\sum_{i=1}^{n}\delta_{r,i}\sigma_{r,i}a_i\right)\left(\sum_{j=1}^{n}\delta_{r,j}\sigma_{r,j}b_i\right)\right] \\ &= \left[\sum_{i=1}^{n}a_ib_i\left(\sum_{r=1}^{m}\delta_{r,i}^2\sigma_{r,i}^2\right)\right] + \sum_{r=1}^{m}\sum_{i\neq j}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}a_ib_j \\ &= \langle a, b\rangle + \sum_{r=1}^{m}\sum_{i\neq j}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}a_ib_j \\ \Rightarrow \mathbb{E}[\langle Sa, Sb\rangle] &= \langle a, b\rangle \end{aligned}$$

Now let's consider the variance, $Var[\langle Sa, Sb\rangle]$. (This proof can be referred to [NN13])

$$\begin{aligned} (Var[\langle Sa, Sb\rangle])^2 &= \sum_{r=1}^{m}\sum_{i\neq j}\mathbb{E}\left[\sigma_{r,i}^2\delta_{r,j}^2(a_i^2b_j^2 + a_ib_ja_jb_i)\right] \\ &\left(\begin{array}{ll} \because \text{Consider } (r,i),(r,j),(r',i'),(r',j') & \\ r = r' \text{ or } \{i,j\} = \{i',j'\} & \to \mathbb{E}[\ \cdot\ ] \neq 0 \\ \qquad\quad \text{Otherwise} & \to \mathbb{E}[\ \cdot\ ] = 0 \text{ by independence.} \end{array}\right) \\ \Rightarrow Var[\langle Sa, Sb\rangle] &= \frac{1}{m}\sum_{i\neq j}\left(a_i^2b_j^2 + a_ib_ja_jb_i\right) \\ &\leq \frac{2}{m}\sum_{i\neq j}a_i^2b_j^2 \\ &\leq \frac{2}{m}\sum_{i,j}a_i^2b_j^2 = \frac{2}{m}\|a\|_2^2\|b\|_2^2 \end{aligned}$$

Let $U \in \mathbb{R}^{n\times d}$ have orthonormal columns. We want,

$$\begin{aligned}
&\|SUx\|_2 = (1 \pm \epsilon)\|x\|_2 \ \ \forall x \in \mathbb{R}^d \\
\Leftrightarrow\ & x^T U^T S^T SUx = (1 \pm \epsilon) x^T x \\
\Leftrightarrow\ & \|U^T S^T SU - I\|_2 \leq \epsilon \\
\Leftarrow\ & \|U^T S^T SU - I\|_F^2 \leq \epsilon^2
\end{aligned}$$

So it is sufficient to show for Frobenius norm case.

$$\begin{aligned}
(U^T S^T SU)_{i,j} &= \langle SU_i, SU_j \rangle \ \ (U_i :\ i^{th} \text{ column of } U) \\
I_{i,j} &= \langle U_i, U_j \rangle
\end{aligned}$$

Also,

$$\begin{aligned}
\forall i,j\ \mathbb{E}[(U^T S^T SU - I)_{i,j}^2] &\leq \frac{2}{m} \\
\Rightarrow \mathbb{E}[\|U^T S^T SU - I\|_F^2] &\leq \frac{2d^2}{m} \leq 2\epsilon^2 \\
\Rightarrow \|U^T S^T SU - I\|_2 &\leq \epsilon
\end{aligned}$$

which shows that $\|SUx\|_2 = (1 \pm \epsilon)\|x\|_2 \ \ \forall x \in \mathbb{R}^d$, i.e. S is OSE. $\square$

With this setting of S by [CW13], complexity can be achieved to $\mathcal{O}(nnz(A) + (d^3/\epsilon^2)\log^5(d/\epsilon))$, which is $\mathcal{O}(nnz(A)) + \tilde{\mathcal{O}}(d^3/\epsilon^2)$.

Following Table compares the computational time for introduced algorithms when applied to regression problem. ($\mathcal{O}$ notation is omitted.)

No OSE:	$d \cdot nnz(A) + d^3 \ / \ d^2 n + d^3$		
with OSE:	$Embedding$	$+$	$Solve(d, m)$
Gaussian:	$mnd = d^2 n/\epsilon$	$+$	d^3/ϵ^2
Fast JL:	$dn \log n$	$+$	$(d^3/\epsilon^2)\log^5 n$
C-W:	$nnz(A)$	$+$	$Solve(d, d^2/\epsilon^2)$ $= d^4/\epsilon^2 \leftarrow$ bad! $\rightarrow (d^3/\epsilon^2)\log^5(d/\epsilon)$

Table 1: Comparing complexities for various algorithms for regression

References

[CW09] Clarkson, Kenneth L., and David P. Woodruff. Numerical linear algebra in the streaming model. *Proceedings of the forty-first annual ACM symposium on Theory of computing*, ACM, 2009.

[CW13] Clarkson, Kenneth L., and David P. Woodruff. Low rank approximation and regression in input sparsity time. *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, ACM, 2013.

[KW11] Krahmer, Felix, and Rachel Ward. New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis* 43.3 (2011): 1269-1281.

[NN13] Nelson, Jelani, and Huy L. Nguyn. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on.* IEEE, 2013.

[Sarlos06] Sarlos, Tamas. Improved approximation algorithms for large matrices via random projections. *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, IEEE, 2006.