Sketching Algorithm Constants

Taisuke Yasuda

April 22, 2018

1 Introduction

We use this file to record our derivation for tracking the constants used for some of the sketching algorithms that we implement.

2 Count Sketch

We refer to `http://www.cs.cmu.edu/afs/cs/user/dwoodruf/www/teaching/15859-fall17/weekTwo.pdf`, which gives that

$$k \geq \frac{6d^2}{\delta\varepsilon^2}$$

suffices for a $(1+\varepsilon)$ subspace embedding.

3 Gaussian sketch

We refer to theorem 2.1 in [DG03], which gives that

$$k \geq 4\left(\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3}\right)^{-1} \log n$$

suffices for a $(1+\varepsilon)$ subspace embedding.

4 Leverage score sampling

For a given matrix $A \in \mathbb{R}^{n\times d}$ and $i \in [n]$, define the *ith leverage score* to be

$$\ell_i := \sum_{j=1}^{d} U_{i,j}^2 = \|e_i U\|_2^2$$

where $A = U\Sigma V^\top$ is the singular value decomposition of A. Now consider a distribution $(q_1, \ldots, q_n)$ over the rows of A, where $\sum_{i=1}^n q_i = 1$ and the q_i satisfy

$$q_i \geq \frac{\beta\ell_i}{d}$$

where $\beta < 1$ is a fixed parameter. Then, we define the following leverage score sampling sketching matrix

$$S_{\text{leverage}} := D\Omega^\top$$

with $D \in \mathbb{R}^{k\times k}$ and $\Omega \in \mathbb{R}^{n\times k}$ as follows. For each column $j \in [k]$ of Ω and D, sample a row index i from the row distribution $(q_1, \ldots, q_n)$ and set $\Omega_{i,j} = 1$ and $D_{i,i} = (q_i k)^{-1/2}$. Here, Ω serves as a sampling matrix and D serves as a rescaling matrix. If $k = \Theta\left(\frac{d\log d}{\beta\varepsilon^2}\right)$, then S_{leverage} is a $(1+\varepsilon)$ subspace embedding.

4.1 Fast computation of leverage scores

4.1.1 First attempt

Let S be a $(1+\varepsilon)$ subspace embedding and let $SA = QR^{-1}$ be the QR decomposition of SA so that Q has orthonormal columns and R^{-1} is an upper triangular matrix. Now, we claim that

$$\ell_i' := |e_i AR|_2^2$$

is a $(1 \pm 6\varepsilon)$ approximation to the leverage scores of A. Since AR has the same column span as A, we may write $AR = UT^{-1}$. Then since S is a subspace embedding, we have that

$$\begin{aligned}(1-\varepsilon)\left\|ARx\right\|_2 &\leq \left\|SARx\right\|_2 = \left\|Qx\right\|_2 = \left\|x\right\|_2 \\ (1+\varepsilon)\left\|ARx\right\|_2 &\geq \left\|SARx\right\|_2 = \left\|Qx\right\|_2 = \left\|x\right\|_2\end{aligned}$$

Now note that for $\varepsilon \leq 1/2$,

$$\begin{aligned}\frac{1}{1-\varepsilon} &= 1 + \varepsilon + \varepsilon^2 + \cdots \leq 1 + 2\varepsilon \\ \frac{1}{1+\varepsilon} &= 1 - \varepsilon + \varepsilon^2 - \cdots \geq 1 - 2\varepsilon\end{aligned}$$

so

$$(1 \pm 2\varepsilon)\left\|Tx\right\|_2 = \left\|ARTx\right\|_2 = \left\|Ux\right\|_2 = \left\|x\right\|_2$$

and thus

$$\ell_i = \left\|e_i U\right\|_2^2 = \left\|e_i ART\right\|_2^2 = (1 \pm 2\varepsilon)^2 \left\|e_i AR\right\|_2^2 = (1 \pm 6\varepsilon)\ell_i'$$

by bounding $\varepsilon^2 \leq \varepsilon/2$ for $\varepsilon \leq 1/2$.

4.1.2 Further speedup

Note that computing $\ell_i' = \left\|e_i AR\right\|_2^2$ takes too long, since $A \in \mathbb{R}^{n\times d}$ and $R \in \mathbb{R}^{d\times d}$. Now recall that in order to get a subspace embedding out of leverage score sampling, we only used q_i with

$$q_i \geq \frac{\beta \ell_i}{d}.$$

Thus, we just need the result for $\beta = 1 - O(\varepsilon)$ a constant. Now note that by the above section, we can find a $(1 \pm 1/2)$ subspace embedding via a Gaussian sketch with

$$4\left(\frac{(1/2)^2}{2} - \frac{(1/2)^3}{3}\right)^{-1} \log n = 48 \log n$$

columns with probability at least $1 - 1/n^2$. Then, setting

$$\ell_i' := \left\|e_i ARG\right\|_2^2$$

allows for efficient computation while giving an approximation factor of

$$(1 \pm 6\varepsilon)(1 \pm 1/2) = 1 \pm (1/2 + 9\varepsilon).$$

Then, we want to set $\beta = 1 - (1/2 + 9\varepsilon)$ so we could set $\varepsilon = 1/36$ for $\beta = 1/4$ for example.

References

[DG03] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.