CS 15-859: Algorithms for Big Data Fall 2017

Lecture 4-2 — 09/28/2017

Prof. David Woodruff *Scribe: Zhengxu Chen*

1 Leverage Score Sampling

1.1 Definitions

Definition. For any $n \times d$ matrix U, define function $\ell(i) = \sum_{j=1}^{d} U_{i,j}^2$.

Definition. Let $(q_1, ..., q_n)$ be a distribution satisfying $q_i \geq \frac{\beta \ell(i)}{d}$, where β is a parameter less than 1.

Definition. Define sampling matrix $S_L = D \cdot \Omega^T$, where D is $k \times k$ and Ω is $n \times k$. Ω is a sampling matrix, and D is a rescaling matrix. For each column j of Ω, D, independently, and with replacement, select a row index in $[n]$ with probability of q_i, and then set $\Omega_{i,j} = 1$ and $D_{i,i} = \frac{1}{\sqrt{q_i k}}$.

Definition. Let $i(j)$ denote the index of the row of an orthonormal matrix U sampled in the j-th trial.

Definition. Let $X_j = I_d - \frac{U_{i(j)}^T U_{i(j)}}{q_{i(j)}}$, where $U_{i(j)}$ is the j-th sampled row of U.

1.2 Properties

In the last part we have proved that $E[X_j] = 0$, $|X_j|_2 \leq 1 + \frac{d}{\beta}$, and $|E[X^T X]|_2 \leq \frac{d}{\beta} - 1$.

1.3 Subspace Embedding

Fact 1. *(Matrix Chernoff Bound)* Let $X_1, ..., x_k$ be independent copies of a symmetric random matrix $X \in R^{d \times d}$ with $E[X] = 0, |X| \leq \gamma$, and $|E[X^T X]]_2| \leq \delta^2$. Let $W = \frac{1}{k} \sum_{j \in [k]} X_j$, then for any $\epsilon > 0$,

$$Pr[|W|_2 > \epsilon] \leq 2d \cdot e^{-k\epsilon^2/(\delta^2 + \frac{\gamma\epsilon}{3})}$$

where $|W|_2 = sup \frac{|Wx|_2}{|x|_2}$, when W is symmetric, $|W|_2 = sup_{|x|_2=1} x^T W x$.

Based on this bound, we want to prove Leverage Score Sampling can actually give us a sketching matrix:

Claim 1. For Leverage Score Sampling matrix S_L, $Pr[|I_d - U^T S^T S U| > \epsilon] \leq 2d \cdot e^{-k\epsilon^2 \Theta(\frac{\beta}{d})}$.

Proof. According to *Matrix Chernoff Bound*, we plug in $\gamma = 1 + \frac{d}{\beta}$, and $\delta^2 = \frac{d}{\beta} - 1$ in the bound. Therefore

$$\begin{aligned}\frac{1}{k}\sum X_{j\in[k]} &= \frac{1}{k}\sum_{j\in[k]} I_d - \frac{U_{i(j)}^T U_{i(j)}}{q_{i(j)}} \\ &= I_d - U^T S^T S U\end{aligned}$$

Hence we can plug in the sum of X_j, so $Pr[|I_d - U^T S^T S U|_2 > \epsilon] \leq 2d \cdot e^{-k\epsilon^2 \Theta(\frac{\beta}{d})}$. Set $k = \Theta(\frac{d \log d}{\beta \epsilon^2})$, then we are done. ■

Hence we according Matrxi Chernoff Bound, we can obtain a subspace embedding matrix by leverage scoring.

1.4 Fast Computation of Leverage Scores

We can always use an naive approach to compute leverage scores using SVD decomposition. Let S be a subspace embedding matrix for a $n \times d$ matrix A. It follows that we can decomposite $SA = QR^{-1}$ such that Q has orthonormal columns, with a fairly low cost.

Instead of getting actual $\ell(i)$, we want to approximate it. More specifically, set $\ell_i' = |e_i AR|_2^2$, where e_i is the i-th base. Note that $SAR = Q$, therefore it is a rotational matrix which does not change the norm of vectors, so

$$|SARx|_2 = |x|_2$$

Since S is a subpace embedding matrix, with a high probability,

$$|SARx|_2 \leq (1 \pm \epsilon)|ARx|_2$$

AR has the same column span of A, and $AR = UT^{-1}$, it follow that

$$(1 \pm O(\epsilon))|x|_2 = |ARx|_2 = |UT^{-1}x|_2 = |T^{-1}x|_2$$

Hence we can prove that

$$\ell(i) = |e_i ART|_2^2 = (1 \pm O(\epsilon))|e_i AR|_2^2 = (1 \pm O(\epsilon))\ell_i'$$

Note that it is sufficient to set ϵ to be a constant here, but there is a problem when we want to compute AR, which is expensive when A is big. To solve this, let G be a $d \times O(\log n)$ matrix of i.i.d. normal random variables. Note that $\forall$ vector z, $Pr[|zG|_2^2 = (1 \pm \frac{1}{2})|z|^2] \geq 1 - \frac{1}{n^2}$.

After we reduce dimension with G, we instead set $\ell_i' = |w_i ARG|_2^2$, and we can now compute ARG within $nnz(A) \log n + d^2 \log n$.

For a regression problem, the total time complexity with precision of $1 \pm \epsilon$ should be $nnz(A) \log n + poly(d \log n/\epsilon)$.

2 Distributed Low Rank Approximation

Currently we can compute low rank approximation for huge matrices with low time cost, however we might also want algorithms to scale in a distributed environment.

Suppose we have a huge matrix A, which is distributed among s servers, for $t = 1, ..., s$. Further, imagine the server t represents the $t - th$ shop, which has a customer-product matrix for itself, and we denote server t's matrix to be A^t.

The total matrix we want is $A = \sigma_{i=1}^{s} A^i$, and this model is called *arbitrary partition model*. This can actually be more general than *row-parition model*, where servers only store part of the rows respectively.

2.1 Communication Model in Arbitrary Partition Model

Suppose there is already *Server 1*, *Server 2*, ..., *Server s* in current setting. Then there is a central server called *Coordinator*. Each server should only talk to this *Coordinator* via 2-way channels.

Assume the capicity of *Coordinator* is large enough, then we can always simulate a point-t-point communication up to factor of 2, because we can just use the *Coordinator* as a middleman for arbitrary communication pair.

2.2 Communication Cost

We can further formulate the computation process for this distributed low rank approximation scenario:

Input: A $n \times d$ matrix A stored on s servers, and:

1. Server t has one $n \times d$ matrix A^t.
2. $A = \sigma_{i=1}^{s} A^i$.
3. Assume all the entries in A^t are $O(\log(nd))$-bit integers.

Output: Each server should output a k-dimensional space W, and:

1. $C = \sigma_{i=1}^{s} A^i P_W$, where AP_W denotes the projection of A onto W.
2. $|A - C|_F \leq (1 + \epsilon)|A - A|_F$.

The output can further be applied to k-means clustering process.

Resources: Minimize total communication and computation cost. We also want constant rounds of communication and input sparsity time.

2.3 Protocols

There are several protocols designed to solve *Distribtured Low Rank Approximation* problem, which is a natural derivation of single machine version of low rank approximation problems.

The first protocol for the row-partition model is proposed in [3]. It requires $O(sdk/\epsilon)$ real numbers of communication between servers. Note the time complexity does not depend on n here. This protocol do not analyze the bit comlexity in communication, which can be large during the process.

The second protocol proposed in [4] extend the model to arbitrary partition model, with the preservation of $O(skd/\epsilon)$ cost.

The third protocol proposed in [2] gives $O(skd) + poly(sk/\epsilon)$ words of communication in arbitrray partition model with input sparsity time. Note that this matches Ω words of communication lower bound. The intuition for this lower bound is that, there exists a underlying cost to have all s servers agree on a piece of $O(kd)$ information for each rank-k approximation.

There are several variants proposed in [1] about kernel low rank approximation, [5] about low approximation of an implicit matrix, and [2] about sparsity.

2.3.1 Coreset Construction

Let us take a look at the construction of *Coreset* proposed in [3]. Let an $n \times d$ matrix $A = U\Sigma V^T$ $U\Sigma V^T$ is an SVD decomposition form. Let $m = k+k/\epsilon$, where k represents the rank-k approximation we want to have. Let Σ_m be the matrix which only preserves the first m diagonal elements in the matrix Σ (and 0 for diagonal otherwise).

Claim 2. For all projection matrices $Y = I - X$ onto $(d-k)$-dimensional subspaces,

$$|\Sigma_m V^T Y|_F^2 = (1 \pm \epsilon)|AY|_F^2 + c \tag{1}$$

where $c = |A - A_m|_F^2$ does not depend on Y.

We can think of S as the corresponding version of U_m^T so that $SA = U_m^T U\Sigma V^T = \Sigma_m V^T$ is a sketch.

Proof.

$$\begin{aligned} |AY|_F^2 &= |U\Sigma_m V^T Y|_F^2 + |U(\Sigma - \Sigma_m)V^T Y|_F^2 \\ &\leq |\Sigma_m V^T Y|_F^2 + |A - A_m|_F^2 \\ &= |\Sigma V^T Y|_F^2| + c \end{aligned}$$

Also,

$$
\begin{aligned}
|\Sigma_m V^T Y|_F^2 + |A - A_m|_F^2 - |AY|_F^2 \\
&= |\Sigma_m V^T (I - X)|_F^2 + |A - A_m|_F^2 - |A(I - X)|_F^2 \\
&= |\Sigma_m V^T|_F^2 - |\Sigma_m V^T X|_F^2 + |A - A_m|_F^2 - |A|_F^2 + |AX|_F^2 \\
&= |AX|_F^2 - |\Sigma_m V^T X|_F^2 \\
&= |(\Sigma - \Sigma_m) V^T X|_F^2 \\
&\leq |(\Sigma - \Sigma_m) V^T|_2^2 \cdot |X|_F^2 \\
&\leq \sigma_{m+1}^2 k \\
&\leq \epsilon \sigma_{m+1}^2 (m - k) \\
&\leq \epsilon \sum_{i \in \{k+1, \ldots, m+1\}} \sigma_i^2 \\
&\leq \epsilon |A - A_k|_F^2 \\
&\leq \epsilon |A - X|_F^2 \\
&\leq \epsilon |AY|_F^2
\end{aligned}
$$

Therefore we prove that $|\Sigma_m V^T Y|_F^2 = (1 \pm \epsilon)|AY|_F^2 + c$. ■

References

[1] Maria-Florina Balcan et al. “Communication efficient distributed kernel principal component analysis”. In: *arXiv preprint arXiv:1503.06858* (2015).

[2] Christos Boutsidis, David P Woodruff, and Peilin Zhong. “Optimal principal component analysis in distributed and streaming models”. In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM. 2016, pp. 236–249.

[3] Dan Feldman, Melanie Schmidt, and Christian Sohler. “Turning Big Data into Tiny Data: Constant-size Coresets for K-means, PCA and Projective Clustering”. In: *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’13. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2013, pp. 1434–1453. ISBN: 978-1-611972-51-1. URL: `http://dl.acm.org/citation.cfm?id=2627817.2627920`.

[4] Ravi Kannan, Santosh Vempala, and David Woodruff. “Principal Component Analysis and Higher Correlations for Distributed Data”. In: *Proceedings of The 27th Conference on Learning Theory*. Ed. by Maria Florina Balcan, Vitaly Feldman, and Csaba SzepesvÃ¡ri. Vol. 35. Proceedings of Machine Learning Research. Barcelona, Spain: PMLR, 13–15 Jun 2014, pp. 1040–1057. URL: `http://proceedings.mlr.press/v35/kannan14.html`.

[5] David P Woodruff and Peilin Zhong. “Distributed low rank approximation of implicit functions of a matrix”. In: *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE. 2016, pp. 847–858.