Uniform Sampling for Matrix Approximation

CHEN Li

Institute of Operations Research and Analytics

April 30th, 2020

Problem

- For a tall-and-skinny matrix $A \in \mathbb{R}^{n \times d}$ ($n >> d$), find another matrix $B \in \mathbb{R}^{m \times d}$ such that $\|Ax\|_2 \approx \|Bx\|_2$ for any $x \in \mathbb{R}^d$
- We want $m << n$

Problem

- For a tall-and-skinny matrix $A \in \mathbb{R}^{n \times d}$ ($n >> d$), find another matrix $B \in \mathbb{R}^{m \times d}$ such that $\|Ax\|_2 \approx \|Bx\|_2$ for any $x \in \mathbb{R}^d$
- We want $m << n$
- Formally, find a spectral approximation of A,

Definition (Spectral approximation)

For a matrix A, a λ-spectral approximation of A where $\lambda > 1$ is another matrix $B \in \mathbb{R}^{m \times d}$ such that

$$\frac{1}{\lambda}\|Ax\|_2^2 \leq \|Bx\|_2^2 \leq \|Ax\|_2^2 \tag{1}$$

for any $x \in \mathbb{R}^d$. Equivalently,

$$\frac{1}{\lambda}A^TA \preceq B^TB \preceq A^TA$$

where $A^TA \succeq B^TB$ means $A^TA - B^TB$ is positive semidefinite.

An Example: Least-squares Approximation

- Given data points $(x_i, y_i), i = 1, ..., n$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, we want to find β by

$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (x_i^T \beta - y_i)^2$$

An Example: Least-squares Approximation

- Given data points $(x_i, y_i), i = 1, ..., n$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, we want to find β by

$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (x_i^T \beta - y_i)^2$$

- Note that

$$\sum_{i=1}^{n} (x_i^T \beta - y_i)^2 = \|X\beta - y\|_2^2 = \left\| (X, y) \begin{pmatrix} \beta \\ -1 \end{pmatrix} \right\|_2^2$$

where $X^T = (x_1, x_2, ..., x_n)$ and $y^T = (y_1, y_2, ..., y_n)$.

An Example: Least-squares Approximation

- Given data points $(x_i, y_i), i = 1, ..., n$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, we want to find β by

$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (x_i^T \beta - y_i)^2$$

- Note that

$$\sum_{i=1}^{n} (x_i^T \beta - y_i)^2 = \|X\beta - y\|_2^2 = \left\| (X, y) \begin{pmatrix} \beta \\ -1 \end{pmatrix} \right\|_2^2$$

  where $X^T = (x_1, x_2, ..., x_n)$ and $y^T = (y_1, y_2, ..., y_n)$.

- Find a spectral approximation of (X, y) and denote it as $(X', y') \in \mathbb{R}^{m \times (d+1)}$, we can solve a much smaller least-squares

$$\min_{\beta \in \mathbb{R}^d} \|X'\beta - y'\|_2^2$$

  and we know $\|X'\beta - y'\|_2^2 \approx \|X\beta - y\|_2^2$

Randomized Algorithm for Spectral Approximation

How to find a spectral approximation?

- Random Projection: recombine rows or columns from a large matrix to form a much smaller problem that approximates the original.
  - All of them are based on Johnson-Lindenstrauss Lemma
- Random Sampling: approximate large matrices by judiciously selecting (and reweighting) few rows or columns.
  - Simpler and faster than random projection
  - Challenge: How to efficiently compute the correct measure of "importance" for rows or columns such that more important rows or columns are selected with higher probability?

We focus on row sampling in the following.

A Quick Intuition of Row Sampling

- In the least-squres example, row sampling means selecting each data point (x_i, y_i) with some probability $p_i \in [0, 1]$ and stack the selected rows.

$$(X, y) = \begin{pmatrix} x_1^T & y_1 \\ x_2^T & y_2 \\ \vdots & \vdots \\ x_n^T & y_n \end{pmatrix} \longrightarrow (X', y') = \begin{pmatrix} x_{i_1}^T & y_{i_1} \\ x_{i_2}^T & y_{i_2} \\ \vdots & \vdots \\ x_{i_m}^T & y_{i_m} \end{pmatrix}$$

A Quick Intuition of Row Sampling

- In the least-squres example, row sampling means selecting each data point (x_i, y_i) with some probability $p_i \in [0, 1]$ and stack the selected rows.

$$(X, y) = \begin{pmatrix} x_1^T & y_1 \\ x_2^T & y_2 \\ \vdots & \vdots \\ x_n^T & y_n \end{pmatrix} \longrightarrow (X', y') = \begin{pmatrix} x_{i_1}^T & y_{i_1} \\ x_{i_2}^T & y_{i_2} \\ \vdots & \vdots \\ x_{i_m}^T & y_{i_m} \end{pmatrix}$$

- Approximate matrix multiplication:
  - Let $A = (a_1, ..., a_n)^T$, then $A^T A = \sum_{k=1}^{n} a_k a_k^T$.
  - Sample each row i according to probability p_i and scale it by $1/\sqrt{p_i}$. Let S_i be the indicator of the i-th row is selected and $B = (a_{i_1}, ..., a_{i_m})^T$ be the sampled matrix.
  - Then

$$\mathbb{E}\left[B^T B\right] = \mathbb{E}\left[\sum_{k=1}^{n} S_k \frac{a_k a_k^T}{p_k}\right] = \sum_{k=1}^{n} p_k \frac{a_k a_k^T}{p_k} = A^T A$$

Leverage Score Sampling

- How to determine the sampling distribution p?
- It is known sampling proportional to the leverage scores of each row is the best, i.e., $p_i \propto \tau_i(A)$.

Definition (Leverage scores of a matrix)

For a matrix $A \in \mathbb{R}^{n \times d}$, the leverage score of the i-th row of A is

$$\tau_i(A) := a_i^T (A^T A)^+ a_i \tag{2}$$

for each $i = 1, ..., n$ where $(A^T A)^+$ is the Moore-Penrose pseudoinverse of $A^T A$. We denote $\tau(A)$ as the n-dimensional vector of $\tau_i(A)$. The maximum of $\tau_i(A)$, i.e., $\|\tau(A)\|_\infty$ is called coherence of A.

- Roughly, leverage score is the measure of importance of each row with respect to the matrix.
- Leverage score sampling $O(d \log d/\epsilon^2)$ rows is enough to give a $(1+\epsilon)$-spectral approximation with high probability.

Challenge of Leverage Score Sampling

- However, computing leverage scores is of the same complexity of solving the least-squares problem due to inverting $A^T A$.
- To utilize it requires some approximation of leverage scores

Challenge of Leverage Score Sampling

- However, computing leverage scores is of the same complexity of solving the least-squares problem due to inverting $A^T A$.
- To utilize it requires some approximation of leverage scores
- Another simple sampling scheme is uniform sampling, i.e., each row is equally likely to be sampled. But it only works well for matrix with low coherence.

Challenge of Leverage Score Sampling

- However, computing leverage scores is of the same complexity of solving the least-squares problem due to inverting $A^T A$.
- To utilize it requires some approximation of leverage scores
- Another simple sampling scheme is uniform sampling, i.e., each row is equally likely to be sampled. But it only works well for matrix with low coherence.
- The idea of this talk is to approximate leverage scores by uniform row sampling, then we do leverage score sampling according to the approximation.

Preliminaries

- $\tau_i(A) \in [0, 1]$ for all i.
- $\sum_{i=1}^{n} \tau_i(A) = \mathrm{trace}(A(A^T A)^+ A) = \mathrm{rank}(A) \leq d$.
- We can also define the generalized leverage score of A respect to matrix $B \in \mathbb{R}^{m \times d}$as follows:

$$\tau_i^B(A) := \begin{cases} a_i^T (B^T B)^+ a_i & \text{if } a_i \perp Ker(B) \text{ or } a_i \in Range(B^T) \\ +\infty & \text{otherwise} \end{cases} \tag{3}$$

- If B is a λ-spectral approximation of A, i.e., $\frac{1}{\lambda} A^T A \preceq B^T B \preceq A^T A$, then

$$\tau_i(A) \leq \tau_i^B(A) \leq \lambda \tau_i(A)$$

Algorithm

Algorithm 1: Repeated Halving

input : A matrix $A \in \mathbb{R}^{n \times d}$

output: A spectral approximation B consisting of $O(d \log d)$ rescaled rows of A

Uniformly sample $n/2$ rows of A to form A_1;

if A_1 *has more than* $O(d \log d)$ *rows* **then**

  Recursively compute a spectral approximation of A_1, denoted as B_1

else

  $B_1 = A_1$

end

Compute generalized leverage score $\tau^{B_1}(A)$;

Do leverage score row sampling on A with estimates $\tau^{B_1}(A)$ to form B;

return B

Sampling Idea Illustration

$$\begin{array}{ccccccccc} A & \longrightarrow & A_1 & \longrightarrow & \cdots & \longrightarrow & A_{i-1} & \longrightarrow & A_i \\ \downarrow & \nwarrow & \downarrow & \nwarrow & \downarrow & \nwarrow & \downarrow & \nwarrow & \| \\ B & & B_1 & & \cdots & & B_{i-1} & & B_i \end{array} \tag{4}$$

- $\longrightarrow$ indicates uniform sampling.
- $\downarrow$ indicates leverage score sampling.
- $\nwarrow$ indicates leverage score approximation

For example, A_1 is uniformly sampled from A, B_1 is a spectral approximation of A_1, then we use B_1 to approximate the leverage score of A, and sample from A based on the leverage score approximation to get B as a spectral approximation of A.

Why it works?

Proof Outline:

- Given leverage score overestimates $u_i \geq \tau_i(A)$ for all i,, we can sample $O(\log d \sum_i u_i)$ rows to get a spectral approximation with high probability
- Uniform sampling m rows can reduce the sum of leverage score overestimates to $\frac{nd}{m}$.

Hence if we uniformly sample more rows to construct overestimates, we will sample fewer rows in leverage score sampling, which means a smaller spectral approximation.

- If we uniformly sample $O(n)$ rows, then the spectral approximation has $O(d \log d)$ rows.
- If we uniformly sample $O(d \log d)$ rows, then the spectral approximation has $O(n)$ rows.

The trade-off is addressed by the iterative sampling scheme.

Spectral Approximation via Leverage Score Sampling

Theorem

- *Given an error parameter* $\epsilon \in (0, 1)$,
- *let* u *be a vector of leverage score overestimates, i.e.,* $\tau_i(A) \leq u_i$ *for all* i.
- *Let* α *be a sampling rate parameter, let* c *be a fixed positive constant. For each row, we define a sampling probability* $p_i = \min\{1, \alpha c u_i \log d\}$.
- *Furthermore, we define a function* `Sample`(u, α), *which returns a random diagonal matrix* S *with independently chosen entries* $S_{ii} = 1/\sqrt{p_i}$ *with probability* p_i *and 0 otherwise.*
- *Setting* $\alpha = \epsilon^{-2}$, S *has at most* $\sum_i \min\{1, u_i \alpha c \log d\} \leq \|u\|_1 \alpha c \log d$ *non-zero entries and* $\frac{1}{\sqrt{1+\epsilon}} SA$ *is* $\frac{1+\epsilon}{1-\epsilon}$*-spectral approximation for* A *with probability at least* $1 - d^{-c/3}$.

Proof Sketch

The proof is based on a variant of matrix Chernoff bound, a very powerful tool of matrix concentration.

Lemma

Let $Y_1, ..., Y_k$ be independent random positive semidefinite matrices of $d \times d$ and $Y = \sum_{i=1}^{k} Y_i$, $Z = \mathbb{E}[Y]$. If $Y_i \preceq RZ$ for some $R > 0$, then

$$\begin{aligned} \mathbb{P}[Y \preceq (1-\epsilon)Z] &\leq d e^{-\frac{\epsilon^2}{2R}} \\ \mathbb{P}[Y \succeq (1+\epsilon)Z] &\leq d e^{-\frac{\epsilon^2}{3R}} \end{aligned} \tag{5}$$

- Let $Y_i = \frac{a_i a_i^T}{p_i}$
- Prove $a_i a_i^T \preceq \tau_i(A) A^T A$
- Check $Y_i \preceq \frac{1}{c\epsilon^{-2} \log d} A^T A$ for $p_i < 1$
- Split Y_i into $c\epsilon^{-2} \log d$ pieces with each piece equal to $\frac{a_i a_i^T}{c\epsilon^{-2} \log d} \preceq \frac{1}{c\epsilon^{-2} \log d} A^T A$.
- The sparsity of S is obtained by standard Chernoff bound.

Bound the sum of leverage score approximation via uniform sampling

Theorem (Leverage Score Estimation via Uniform Sampling)

Given any $A \in \mathbb{R}^{n \times d}$. Let $\mathcal{S}$ denote the set of row index of uniformly random sample of m rows from A and let $S \in \mathbb{R}^{n \times n}$ be its diagonal indicator matrix (i.e. $S_{ii} = 1$ if $i \in \mathcal{S}$ and 0 otherwise). Define

$$\tilde{\tau}_i := \begin{cases} \tau_i^{SA}(A) & \text{if } i \in \mathcal{S} \\ \frac{1}{1 + \frac{1}{\tau_i^{SA}(A)}} & \text{otherwise} \end{cases} \tag{6}$$

Then $\tilde{\tau}_i \geq \tau_i^{SA}(A)$ for all i and

$$\mathbb{E}\left[\sum_{i=1}^{n} \tilde{\tau}_i\right] \leq \frac{nd}{m} \tag{7}$$

Proof sketch

- The proof of overestimates is straightforward.
- Bound of expected sum of overestimates
  - Let $S^{(i)}$ be the diagonal indicator matrix for $\mathcal{S} \cup \{i\}$, then

$$\tilde{\tau}_i = \tau_i^{S^{(i)}A}(A).$$

    This can be checked directly using Sherman-Morrison formula. Then we split

$$\sum_{i=1}^{n} \tilde{\tau}_i = \sum_{i \in \mathcal{S}} \tilde{\tau}_i + \sum_{i \notin \mathcal{S}} \tilde{\tau}_i.$$

  - The first part is $\sum_{i \in \mathcal{S}} \tau_i(SA) = \operatorname{rank}(SA) \leq d$.
  - For the second part, construct two different but equivalent random processes: one is first randomly select $\mathcal{S}$, then randomly choose $i \notin \mathcal{S}$ and return value $\tilde{\tau}_i$. Another is randomly select a set $\mathcal{S}'$ of $m+1$ rows, then randomly select $i \in \mathcal{S}'$ and return its leverage score. The expectation of the first value is $\frac{1}{n-m}\mathbb{E}\left[\sum_{i \notin \mathcal{S}} \tilde{\tau}_i\right]$ and the second is bounded by $\frac{1}{m+1}d$. Put them together, we have $\mathbb{E}\left[\sum_{i \notin \mathcal{S}} \tilde{\tau}_i\right] \leq \frac{n-m}{m}d$.

An Interesing Fact

Theorem (Leverage Score Bounding Row Reweighting)

For any $A \in \mathbb{R}^{n \times d}$ and any vector $u \in \mathbb{R}^d$ with $u_i > 0$ for all i, there exists a diagonal matrix $W \in \mathbb{R}^{n \times n}$ with $0 \preceq W \preceq I$ such that:

$$\tau_i(WA) \leq u_i, \quad \forall i = 1, ..., n$$

and

$$\sum_{i:W_{ii} \neq 1} u_i \leq d$$

- Basically, for any matrix A, we can reweight a few rows of A to get a low coherence matrix.
- Let $u = \alpha \mathbf{1}$ for some coherence parameter $\alpha > 0$, there exists a diagonal matrix $W \in \mathbb{R}^{n \times n}$ with all entries in $[0, 1]$ and just d/α entries NOT equal to 1, such that $\tau_i(WA) \leq \alpha$ for all i.

Implications

- The fact also shows the power of uniform sampling: Since uniform sampling works well for low coherence matrix, and a large fraction of any matrix is of low coherence, so uniform sampling can give good leverage score approximation of that portion.

Implications

- The fact also shows the power of uniform sampling: Since uniform sampling works well for low coherence matrix, and a large fraction of any matrix is of low coherence, so uniform sampling can give good leverage score approximation of that portion.
- The fact also allows us to refine leverage score overestimates:

Theorem (Leverage Score Approximation via Undersampling)

Let u be a vector of leverage score overestimates, i.e., $\tau_i(A) \leq u_i$ for all i. For some undersampling parameter $\alpha \in (0, 1]$, let $S' = \sqrt{\alpha \cdot \frac{3}{4}}$ $\mathtt{Sample}(u, 9\alpha)$. Let $u'_i = \min\{u_i, \tau_i^{S'A}(A)\}$. Then with high probability, u'_i is a leverage score overestimate, i.e., $\tau_i(A) \leq u'_i$ for all i and

$$\sum_{i=1}^{n} u'_i \leq \frac{3d}{\alpha}$$

and S' has $O(\alpha \|u\|_1 \log d)$ nonzeros.

Proof sketch

- The overestimates part is easy from Lemma 4 with $\epsilon = 1/3$.
- Bound the leverage score overestimates:
  - There exists some reweighting matrix W such that $\tau_i(WA) \leq \alpha u_i$ and $\sum_{i:W_{ii}\neq 1} u_i \leq \frac{d}{\alpha}$.
  - Split

$$\sum_{i=1}^{n} u_i' = \sum_{i:W_{ii}\neq 1} u_i' + \sum_{i:W_{ii}=1} u_i'$$

  - Bound the first part by $\sum_{i:W_{ii}\neq 1} u_i \leq d/\alpha$.
  - For the second part, we bound it by

$$\sum_{i:W_{ii}=1} \tau_i^{S'A}(A) = \sum_{i:W_{ii}=1} \tau_i^{S'A}(WA) \leq \sum_{i:W_{ii}=1} \tau_i^{S'WA}(WA)$$

    and

$$\sum_{i:W_{ii}=1} \tau_i^{S'WA}(WA) \leq \frac{2}{\alpha} \sum_{i:W_{ii}=1} \tau_i(WA) \leq \frac{2}{\alpha} \cdot d$$

    and get the total sum bounded by $3d/\alpha$.

Another Algorithm

Let $\alpha = \frac{6d}{\|u\|_1}$ is enough to cut $\|u\|_1$ by a half, which leads to $O(d \log d)$ sampled rows.

Algorithm 2: Refinement Sampling

---

input : A matrix $A \in \mathbb{R}^{n \times d}$

output: A spectral approximation B consisting of $O(d \log d)$ rescaled rows of A

Initialize a leverage score overestimate vector $u = \mathbf{1}$;

while $\|u\|_1 \geq O(d)$ **do**

  Sample $O(d \log d)$ rows from A using u to obtain spectral approximation B;

  Compute generalized leverage score $\tau_i^B(A)$ for all i;

  $u_i \longleftarrow \min\{u_i, \tau_i^B(A)\}$ for all i;

end

Sample $O(d \log d)$ rows from A using u to obtain spectral approximation B;

return B

Runtime Analysis

- assumptions:
  - $n = O(poly(d))$, i.e., n is polynomial in d
  - $d \times d$ linear system can be solved in $O(d^w)$ time.
- Approximate generalized leverage score:

Lemma

Given B has $O(d \log d)$ rescaled rows of A, for any $\theta > 0$, it is possible to compute generalized leverage score estimate of $\tau^B(A)$, denoted as u, in $O(d^w \log d + nnz(A)\theta^{-1})$ time such that $u_i \geq \tau_i^B(A)$ and $u_i \leq d^\theta \tau_i^B(A$ with high probability.
By setting $\theta = O(\frac{1}{\log d})$, we can obtain a constant factor approximation of generalized leverage score in $O(d^w \log d + nnz(A) \log d)$ time.

- Runtime of Algorithm: $O(d^w \log d \log \frac{n}{d} + nnz(A) \log d \log \frac{n}{d})$. Roughly, there are $O(\log \frac{n}{d})$ iterations and the runtime of each iteration is bounded by $O(d^w \log d + nnz(A) \log d)$.

Thank You! Questions?