8 Sparsification Algorithms

All low rank matrix approximation algorithms including the fundamental ones such as Power method or Orthogonal iterations, involve lots of matrix-matrix or matrix-vector multiplications. These basic operations require time proportional to number of non-zero entries in matrices, as one need to read the entire matrix into memory. Sparsifying a matrix, i.e. decreasing number of non-zeros, and quantizing it, i.e. rounding up entries to a constant, accelerate such computations as well as saving space in representation.

First sparsification algorithm was by Achlioptas and McSherry[1], where they sampled and quantized entries of a given matrix $A \in \mathbb{R}^{n\times d}$ to lowered number of non-zeros and length of their representation. They observed acts of sampling and quantization can be viewed as adding a random noise matrix $E \in \mathbb{R}^{n\times d}$ to A, whose entries are independent random variables with zero mean and bounded variance. Since with high probability a random matrix has a weak spectral structure, it does not alter the the main spectrum of input matrix. Below we first state a theorem on norm of random matrices, then describe their algorithms.

8.1 Spectral Structure of Random Matrices

Theorem below[2] shows a well constructed random matrix has a weak spectral structure.

Theorem 8.1.1. *[2] Let $E \in \mathbb{R}^{n\times d}$ be a random matrix such that entries $E_{i,j} = r_{ij}$ are independent bounded random variables $r_{ij} \in [-k,k]$, with $\textbf{E}[r_{ij}] = 0$ and $\textbf{Var}(r_{ij}) \le \sigma^2$. For all $\alpha \ge 1, \varepsilon > 0$, and $n+d \ge 20$, if $k \le (\frac{4\varepsilon}{4+3\varepsilon})^3 \frac{\sigma\sqrt{n+d}}{\log^3(n+d)}$ then*

$$\textbf{Pr}\left[\|E\|_2 \ge (2+\varepsilon+\alpha)\sigma\sqrt{n+d}\right] < (n+d)^{-\alpha^2}$$

8.2 Additive Error Sparsification Algorithms

Using theorem 8.1.1, Achlioptas and McSherry[1] showed a carefully constructed random matrix $\hat{A} \in \mathbb{R}^{n\times d}$ can approximate spectral norm of A_k. Theorem 8.2.1 states their result.

Theorem 8.2.1. *Let $A \in \mathbb{R}^{n\times d}$ be an arbitrary matrix with $b = \max_{i,j}|A_{i,j}|$ being the maximum entry in absolute value. Let $\hat{A} \in \mathbb{R}^{n\times d}$ be a random matrix where entries $\hat{A}_{i,j}$ are independent random variables with $\textbf{E}[\hat{A}_{i,j}] = A_{i,j}$, $\textbf{Var}(\hat{A}_{i,j}) = (\sigma b)^2$ and $\|A_{i,j} - \hat{A}_{i,j}\|_2 \le \frac{\sigma b\sqrt{n+d}}{2\log^3(n+d)}$. Then for any $\alpha \ge 1$,*

$$\|A - \hat{A}_k\|_2 \le \|A - A_k\|_2 + (8+2\alpha)\sigma b\sqrt{n+d}$$

holds with probability atleast $1-(n+d)^{-\alpha^2}$.

Proof.

$$\begin{aligned}
\|A - \hat{A}_k\|_2 &\le \|A - \hat{A}\|_2 + \|\hat{A} - \hat{A}_k\|_2 && \text{triangle inequality}\\
&\le \|A - \hat{A}\|_2 + \|\hat{A} - A_k\|_2 && \text{For any rank } k \text{ matrix } D\text{: } \|\hat{A} - \hat{A}_k\|_2 \le \|\hat{A} - D\|_2\\
&\le \|A - \hat{A}\|_2 + \|\hat{A} - A\|_2 + \|A - A_k\|_2 && \text{triangle inequality}\\
&\le 2\|A - \hat{A}\|_2 + \|A - A_k\|_2
\end{aligned}$$

Setting $E = A - \hat{A}$ one can verify that E satisfies all conditions of theorem 8.1.1, as it has zero expectation $\textbf{E}[E_{i,j}] = A_{i,j} - \textbf{E}[\hat{A}_{i,j}] = 0$, bounded variance $\textbf{Var}(E_{i,j}) \le (\sigma b)^2$, and bouned entries $E_{i,j} \in [-\frac{\sigma b\sqrt{n+d}}{2\log^3(n+d)}, \frac{\sigma b\sqrt{n+d}}{2\log^3(n+d)}]$. Therefore taking $\varepsilon = 2$, the bound $\|A - \hat{A}\|_2 \le (4+\alpha)\sigma b\sqrt{n+d}$ holds with probability atleast $1-(n+d)^{-\alpha^2}$, and therefore $\|A - \hat{A}_k\|_2 \le (8+2\alpha)\sigma b\sqrt{n+d} + \|A - A_k\|_2$. $\square$

As theorem 8.2.1 holds for any random matrix $\hat{A}$ with above conditions, authors of [1] proposed two concrete constructions. The first construction is based on sampling; matrix $\hat{A}$ samples some entries of A and omits others, they show the stronger spectrum of input matrix is, the larger fraction of entries they can afford to lose. Theorem8.2.2 states their sampling result.

Theorem 8.2.2. *Let $A \in \mathbb{R}^{n\times d}$ be the input matrix and $b = \max_{i,j} |A_{i,j}|$ be the maximum entry in absolute value. Define matrix $\hat{A} \in \mathbb{R}^{n\times d}$ as*

$$\hat{A}_{i,j} = \begin{cases} 0 & \textit{w.p. } 1 - \frac{1}{s} \\ sA_{i,j} & \textit{w.p. } \frac{1}{s} \end{cases}$$

where $1 \leq s \leq \frac{n+d}{4\log^6(n+d)}$. Then with probability atleast $1 - 1/(n+d)$ the following error bound holds

$$\|A - \hat{A}\|_2 \leq \|A - A_k\|_2 + 10b\sqrt{s(n+d)}$$

Proof. It is easy to verify that matrix $\hat{A}$ satisfies all conditions of theorem 8.2.1:

- $\mathbf{E}[\hat{A}_{i,j}] = 0(1 - 1/s) + sA_{i,j}(1/s) = A_{i,j}$
- $\mathbf{Var}(\hat{A}_{i,j}) = \mathbf{E}[\hat{A}^2_{i,j}] - \mathbf{E}[\hat{A}_{i,j}]^2 = 1/s(sA_{i,j})^2 - A^2_{i,j} = (s-1)A^2_{i,j} \leq (\sqrt{s}b)^2$ therefore $\sigma = \sqrt{s} \leq \frac{\sqrt{n+d}}{2\log^3(n+d)}$
- $\forall i \in [1,n], j \in [1,d]: \quad |A_{i,j} - \hat{A}_{i,j}| \in \{A_{i,j}, sA_{i,j}\}$, and in both cases it is upper bounded by $|A_{i,j} - \hat{A}_{i,j}| \leq sA_{i,j} \leq \frac{(n+d)b}{4\log^6(n+d)}$

Fitting conditions of theorem 8.2.1, and using $\alpha = 1$, we obtain $\|A - \hat{A}_k\|_2 \leq 10b\sqrt{s(n+d)} + \|A - A_k\|_2$. □

In their second construction, they randomly quantize entries of A, and shorten the representation, this allows them to store each entry in one bit. Theorem 8.2.3 explains their result.

Theorem 8.2.3. *Let $A \in \mathbb{R}^{n\times d}$ be the input matrix and $b = \max_{i,j} |A_{i,j}|$ be the maximum entry in absolute value. Define matrix $\hat{A} \in \mathbb{R}^{n\times d}$ as*

$$\hat{A}_{i,j} = \begin{cases} +b & \textit{w.p. } \frac{1}{2} + \frac{A_{i,j}}{2b} \\ -b & \textit{w.p. } \frac{1}{2} - \frac{A_{i,j}}{2b} \end{cases}$$

Then $\|A - \hat{A}\|_2 \leq \|A - A_k\|_2 + 10b\sqrt{(n+d)}$ with probability atleast $1 - 1/(n+d)$.

Proof. Again it's easy to see that matrix $\hat{A}$ satisfies all conditions of theorem 8.2.1:

- $\mathbf{E}[\hat{A}_{i,j}] = b(\frac{1}{2} + \frac{A_{i,j}}{2b}) - b(\frac{1}{2} - \frac{A_{i,j}}{2b}) = A_{i,j}$
- $\mathbf{Var}(\hat{A}_{i,j}) = \mathbf{E}[\hat{A}^2_{i,j}] - \mathbf{E}[\hat{A}_{i,j}]^2 = b^2(\frac{1}{2} + \frac{A_{i,j}}{2b}) + b^2(\frac{1}{2} - \frac{A_{i,j}}{2b}) - A^2_{i,j} \leq b^2$ therefore $\sigma = 1$
- $\forall i \in [1,n], j \in [1,d]: \quad |A_{i,j} - \hat{A}_{i,j}| = |A_{i,j} \pm b| \leq 2b$

Fitting conditions of theorem 8.2.1, and using $\alpha = 1$ completes the proof

$$\|A - \hat{A}_k\|_2 \leq 10b\sqrt{(n+d)} + \|A - A_k\|_2$$

□

8.3 Relative Error Sparsification Algorithm

Latest result in using sparsification for low-rank approximation [3] takes advantange of a popular technique in matrix completion line of work, called as *alternating minimization*. We first give a brief review of this technique, then elaborate the main algorithm.

Often a target matrix A can be represented in a bi-linear form as $A = UV$ (matrices U, V are not necessarily orthonormal). Having this parametrization, the task of approximating A reduces to finding U and V that minimize an error metric, for example $\|A - UV\|_F$. The *alternating minimization* technique starts with some initial guess for U and V (say $U^{(0)}, V^{(0)}$), iteratively keep one of U, V fixed and optimize over the other, that is $V^{(i+1)} = \arg\min_V \|A - U^{(i)}V\|_F$, then switch and repeat until it converges.

In order to use this technique in matrix approximation, algorithm[3] samples some entries of matrix A, partition them into multiple subsets and iterates over those subsets to refine the approximation it obtained from first subset. The full method is described in algorithm 8.3.1 and 8.3.2.

Algorithm 8.3.1 Leverage Element Low Rank Approximation (LELA)

1: **Input:** $A \in \mathbb{R}^{d\times n}$, rank r, number of samples m, number of iterations T
2: **Output:** $P_\Omega(A), \Omega, r, \hat{q}, T$
3: $\Omega \subset [n] \times [d] \leftarrow$ indices of m independently sampled entries with probability $\hat{q}_{i,j} = \min\{1, q_{i,j}\}$ with $q_{i,j} = m.(\frac{\|A_{i,:}\|^2 + \|A_{:,j}\|^2}{2(n+d)\|A\|_F^2} + \frac{|A_{i,j}|}{2\|A\|_1})$
4: obtain $P_\Omega(A) \subset A$ as the matrix of sampled entries, using another pass over A
5: $\hat{A}_r = WAltMin(P_\Omega(A), \Omega, r, \hat{q}, T)$

Algorithm 8.3.2 Weighted Alternative Minimization

1: **Input:** $P_\Omega(A), \Omega, r, \hat{q}, T$
2: **Output:** $\hat{A}_r \in \mathbb{R}^{n\times d}$
3: For all $i, j \in [n] \times [d]$ set $w_{i,j} = 1/\hat{q}_{i,j}$ if $\hat{q}_{i,j} > 0$, otherwise $w_{i,j} = 0$
4: Divide Ω into $2T + 1$ equal uniformly random subsets $\Omega = \{\Omega_0, \cdots, \Omega_{2T}\}$
5: $R_{\Omega_0}(A) \leftarrow w. * P_{\Omega_0}(A)$
6: Set $U^{(0)}\Sigma^{(0)}(V^{(0)})^T = \mathsf{svd}(R_{\Omega_0}(A), r)$
7: **for** $t = 0$to $T - 1$ **do**
8: $\quad \hat{V}^{(t+1)} = \arg\min_{V\in\mathbb{R}^{d\times r}} \|R^{1/2}_{\Omega_{2t+1}}(A - \hat{U}^{(t)}V^T)\|_F^2$
9: $\quad \hat{U}^{(t+1)} = \arg\min_{U\in\mathbb{R}^{n\times r}} \|R^{1/2}_{\Omega_{2t+2}}(A - U(\hat{V}^{(t+1)})^T\|_F^2$
10: **return** $\hat{A}_r = \hat{U}^{(T)}(\hat{V}^{(T)})^T$

In the sampling phase, whose aim is to sparsify the matrix, each entry $A_{i,j}$ is sampled with a defined probability $q_{i,j}$ and weighted as $A_{i,j}/q_{i,j}$, so that sampled matrix $\hat{A} \in \mathbb{R}^{n\times d}$ has same frobenious norm as A in expectation. As decomposing a matrix takes time inversly proportional to the sparsity of the matrix authors spread non-zero entries of $\hat{A}$ equally and randomly amongst some fixed numbers of matrices $\hat{A}^{(j)} \in \mathbb{R}^{n\times d}$, therefore $\sum_{j=1} \hat{A}^{(j)} = \hat{A}$. Now that each $\hat{A}^{(j)}$ is a sparse random sample of $\hat{A}$, they take svd decomposition of $\hat{A}^{(1)}$ explicitly, i.e $[U, S, V] = \hat{A}^{(1)}$. Considering $\hat{A}^{(1)}$ in bi-linear form $\hat{A}^{(1)} = U(SV^T)$, they iterate over further matrices $\{A^{(j)}\}$ and minimize the fronbenious error of approximation.

They show that 8.3.1 needs $T = O(\log(\frac{\|A\|_2}{\varepsilon\|A-A_r\|_F}))$ iterations, runs in time $O(\mathsf{nnz}(A) + \frac{nr^5}{\varepsilon^2}\kappa^2 \log n)$ where $\kappa = \sigma_1/\sigma_r$ is the condition number of A, and achieves the relative error bound

$$\|A - \hat{A}_r\|_2 \leq \|A - A_r\|_2 + 2\varepsilon\|A - A_r\|_F$$

Bibliography

[1] Dimitris Achlioptas and Frank McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 611–618. ACM, 2001.

[2] Noga Alon, Michael Krivelevich, and Van H Vu. On the concentration of eigenvalues of random symmetric matrices. *Israel Journal of Mathematics*, 131(1):259–267, 2002.

[3] Srinadh Bhojanapalli, Prateek Jain, and Sujay Sanghavi. Tighter low-rank approximation via sampling the leveraged element. *arXiv preprint arXiv:1410.3886*, 2014.