稀疏矩阵的随机快速奇异值分解

李家郡

2019.5.8

背景与研究意义

常用的矩阵分解

- 奇异值分解 (特征值分解)
- QR 分解
- LU 分解

奇异值分解作用

- 主成分分析等价问题
- 社交网络分析
- 推荐系统
- 自然语言处理

Matrix Sketching

定义

给定 $A \in R^{m \times n}$ $S \in R^{n \times s}$ 为 sketching 矩阵 (random projection or column selection)

$C = AS \in R^{m \times s}$ 即为 A 的 sketch

这里的 C 可以比 A 小很多，但保留着很多 A 的性质

基本的稀疏矩阵随机奇异值分解算法

算法 1 基础 SVD

输入: $A \in R^{m\times n}$，秩参数 k，PI 参数 p，随机高斯矩阵Ω

输出:$U \in R^{m\times k}, S \in R^{k\times k}, V \in R^{n\times k}$

1: $Q = \mathrm{orth}(A\Omega)$

2: for i in 1: p

3: $\quad G = \mathrm{orth}(A^T Q)$

4: $\quad Q = \mathrm{orth}(AG)$

5: end for

6: $B = Q^T A$

7: $[U, S, V] = \mathrm{svd}(B)$

8: $U = QU$

9: $U = U(:, 1:k), S = S(1:k, 1:k), V = V(:, 1:k)$

本文算法改进方向

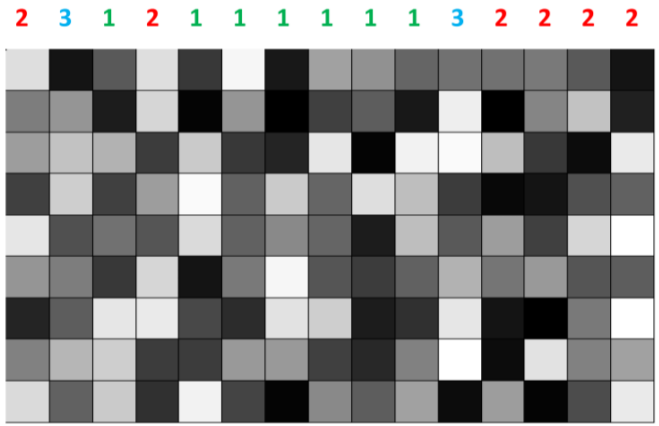
- LU 分解替代 QR 分解
- Count Sketch 替代 Gaussian Sketch

Count Sketch

等价的两种理解方式

- map-reduce fashion
- streaming fashion

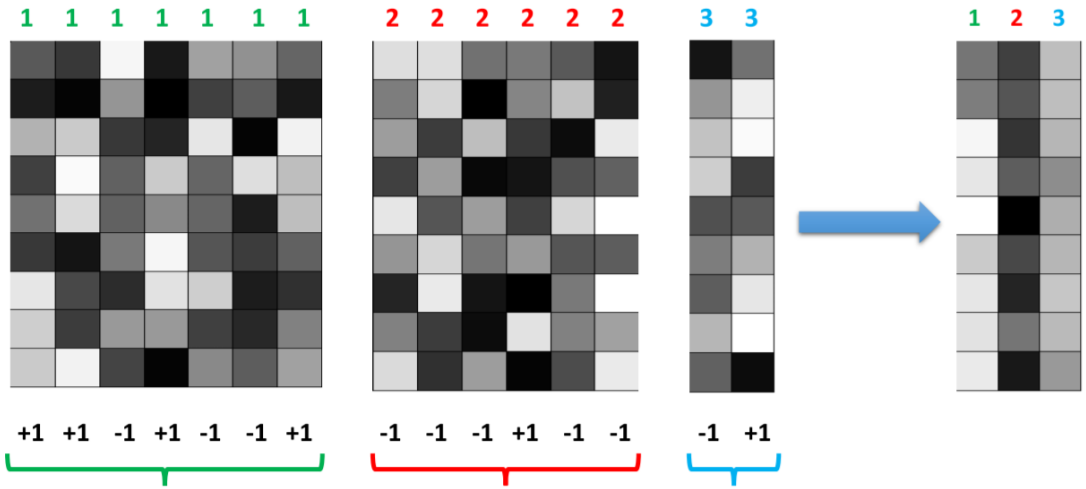
Count Sketch - map-reduce fashion



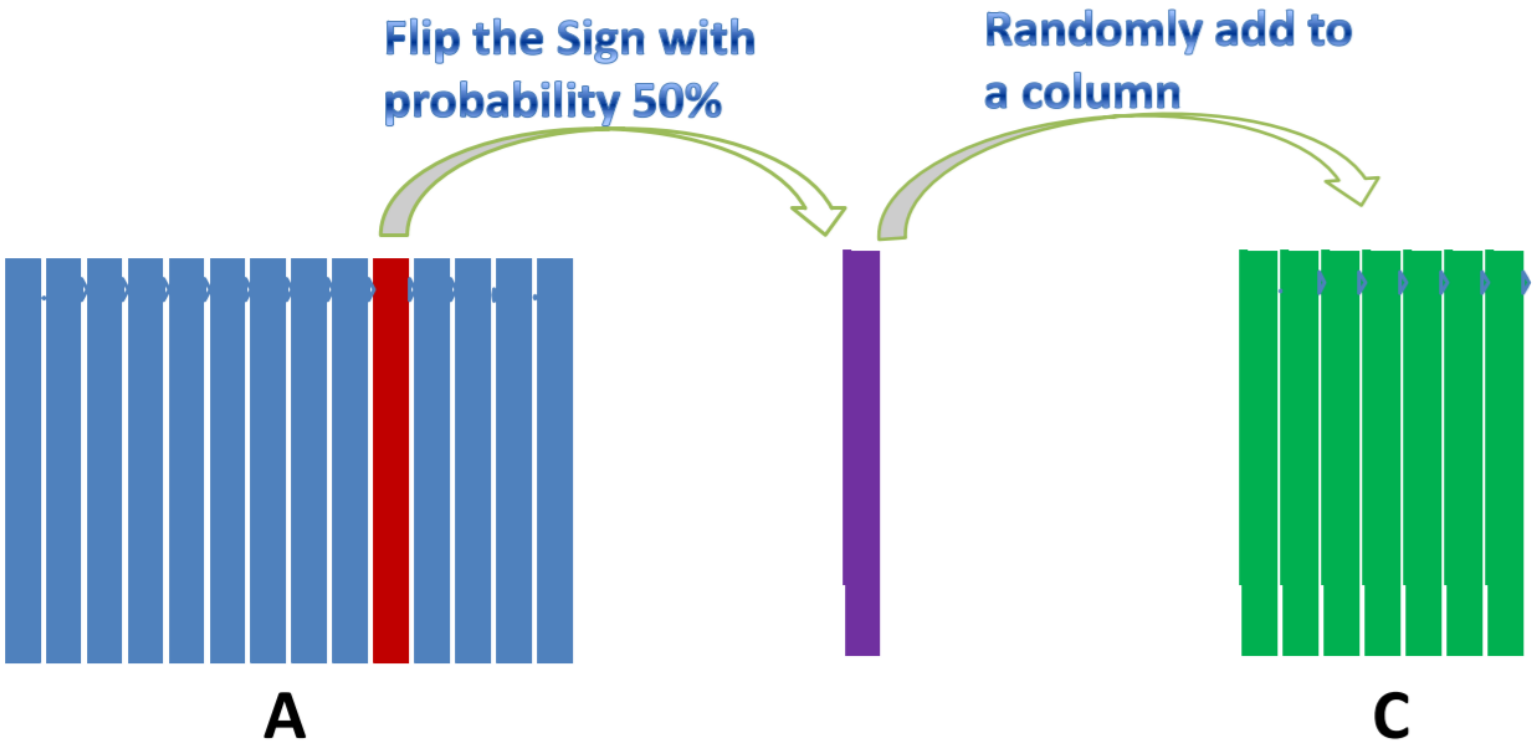
Example:

- Matrix size 9×15
- Sketch size $s = 3$

(a) Hash each column with a value uniformly sampled from $[s] = \{1, 2, 3\}$.



Count Sketch - streaming fashion



Count Sketch

$$S^T = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & -1 & 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

改进的算法

算法 3 CS-frSVD

输入：$A \in R^{m\times n}$，秩参数 k，PI 参数 q，Count Sketch 矩阵Ω

输出:$U \in R^{m\times k}, S \in R^{k\times k}, V \in R^{n\times k}$

1: if q%2 == 0::

2: $Q = \Omega(n, k+s)$

3: $Q = AQ$

4: if $q > 2$ $[Q, \sim] = lu(Q)$

5: else $[Q, \sim, \sim] = eigSVD(Q)$

6: else:

7: $Q = \Omega(m, k+s)$

8: for i in 1: $\left\lfloor \frac{q-1}{2} \right\rfloor$

9: if i == $\left\lfloor \frac{q-1}{2} \right\rfloor$ $[Q, \sim, \sim] = eigSVD(A(A^T Q))$

10: else $[Q, \sim] = lu(A(A^T Q))$

11: $[\hat{U}, \hat{S}, \hat{V}] = eigSVD(A^T Q)$

12: $index = k+s:-1:s+1$

13: $U = Q\hat{V}(:, index), V = \hat{U}(:, index), S = \hat{S}(index)$

数据说明

数据名称	维度	非零个数
SNAP(Leskovec and Krevl,2014)	82,168×82,168	948,464
Cit-Patents	3,774,768×3,774,768	116,514,648
Soc-LiveJournal	4,847,571×4,847,571	1,216,415,176
Pokec-relationships	1,632,803×1,632,803	625,101,587
Web-BerkStan	685,230×685,230	69,197,810
Web-Google	875,713×875,713	36,818,556

实验效果-时间

单线程下，实验迭代参数 q 为 3,LU 迭代次数为 1 时不同数据运行时间比较(s)

数据集	基础随机 SVD 基于 Eigen	frPCA 基于 MKL	CS-frSVD 基于 Armadillo
SNAP	225.41	3.41	2.38
Cit-Patents	14088.10	501.73	395.77
Soc-LiveJournal	50867.9	3321.01	3196.61
Soc-Pokec-relationships	22084.50	1752.79	2107.10
Web-BerkStan	3362.85	49.50	45.27
Web-Google	3262.71	74.20	45.90

实验效果-时间

多线程下, 实验迭代参数 q 为 3,LU 迭代次数为 1 时不同数据运行时间比较 (s)

数据集	Facebook-fbPCA	frPCA 基于 MKL	CS-frSVD 基于 MKL
SNAP	9.22	0.59	1.76
Cit-Patents	903.95	61.82	75.17
Soc-LiveJournal	6834.01	1726.22	1305.98
Soc-Pokec-relationships	2874.16	234.42	198.87
Web-BerkStan	149.74	6.00	13.26
Web-Google	180.63	7.56	9.76

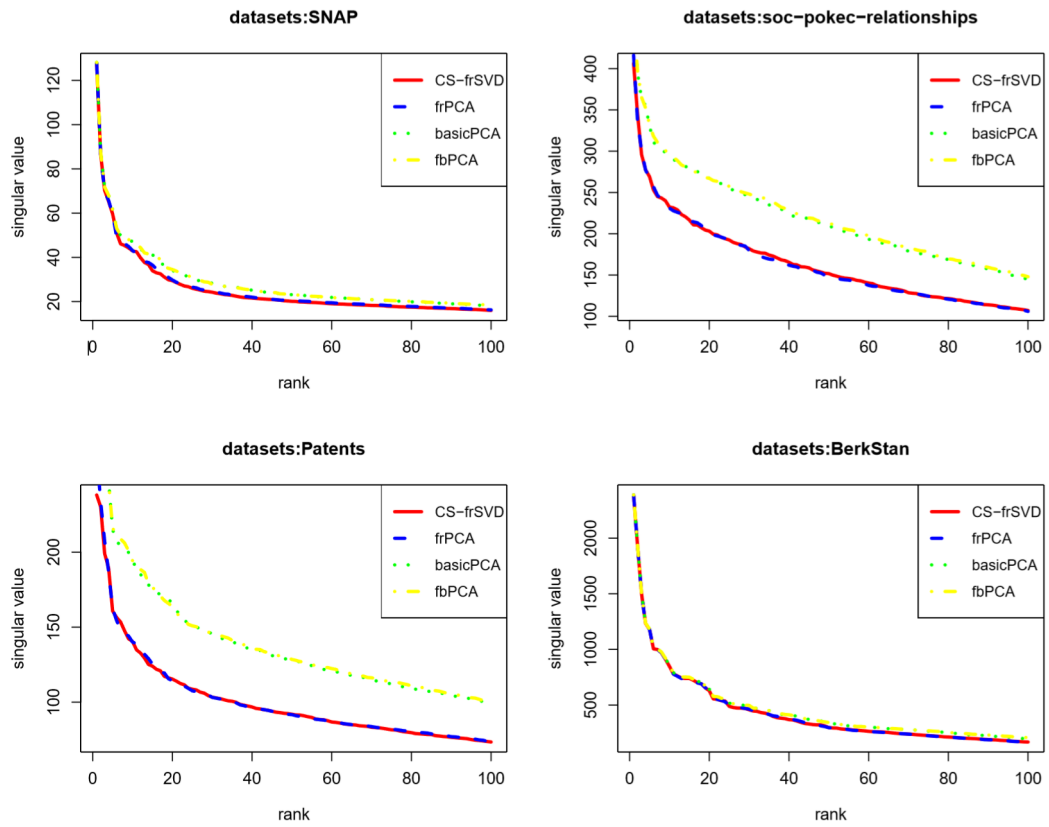
实验效果-精确度

迭代次数为 1 时，不同数据精确度比较 (前一百个奇异值平方和开根)

数据集	Facebook-fbPCA	基础随机 SVD 基于 Eigen	frPCA 基于 MKL	CS-frSVD 基于 Armadillo	CS-frSVD 基于 MKL	F-norm
SNAP	330.4	329.7	304.8	305.6	302.7	973.9
Cit-Patents	1459.9	1450.7	1063.1	1711.7	1055.9	18216.9
Soc-LiveJournal	3583.7	3563.4	2800.4	2807.5	2820.7	31157.7
Soc-Pokec-relationships	2299.1	2275.7	1721.7	1731.4	1731.5	20057.4
Web-BerkStan	5858.1	5846.4	5678.7	5676.7	5673.0	10302.2
Web-Google	2998.3	2982.5	2618.9	2624.4	2614.2	9631.0

实验效果-精确度

迭代分解次数为 1 时，不同数据集对应的前一百个奇异值分布



实验效果-精确度

迭代次数为 3 时，不同数据精确度比较 (前一百个奇异值平方和开根)

表格 5 迭代次数为 3 时，不同数据精确度比较(前一百个奇异值平方和开根)

数据集	Facebook-fbPCA	基础随机 SVD 基于 Eigen	frPCA 基于 MKL	CS-frSVD 基于 Armadillo	CS-frSVD 基于 MKL	F-norm
SNAP	346.7	351.9	350.2	350.2	350.1	973.9
Cit-Patents	1673.1	1729.9	1677.9	1711.7	1702.1	18216.9
Soc-LiveJournal	3952.9	4037.0	3943.6	4015.5	4011.2	31157.7
Soc-Pokec-relationships	2563.1	2621.2	2607.0	2606.2	2606.5	20057.4
Web-BerkStan	5931.4	5949.4	5943.2	5948.9	5945.3	10302.2
Web-Google	3160.4	3190.8	3168.2	3185.9	3183.3	9631.0

实验效果-精确度

迭代分解次数为 3 时，不同数据集对应的前一百个奇异值分布