Relative-Error CUR Matrix Decompositions

Arturo Fernandez

RandNLA Reading Group
University of California, Berkeley

Tuesday, April 7, 2015.

Motivation

"*study [low-rank] matrix approximations that are explicitly expressed in terms of a small numbers of columns and/or rows*"

"*Main algorithmic result are two randomized algorithms which take as input* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *and rank parameter* k"

Let $\mathbf{C} \in \mathbb{R}^{m \times c}$ be a subset of columns of $\mathbf{A}$.

1. $\mathbf{A}' = \mathbf{C}\mathbf{C}^{+}\mathbf{A}$
2. $\mathbf{A}' = \mathbf{CUR}$, where $\mathbf{R} \in \mathbb{R}^{r \times n}$ is a subset of rows of $\mathbf{A}$

For each, independently, with probability $1 - \delta$

$$\|\mathbf{A} - \mathbf{A}'\|_F \leq (1 + \varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$$

where $\mathbf{A}_k$ is the the thresholded SVD.

First Result/Theorem 1

$\mathbf{A}' = \mathbf{CX}$ is a column based matrix approximation to $\mathbf{A}$, or a $\mathbf{CX}$ *matrix decomposition*, for any $\mathbf{X} \in \mathbb{R}^{c \times n}$.

Among such a class

$$\mathbf{C}^{+}\mathbf{A} = \arg\min_{\mathbf{X}} \ \|\mathbf{A} - \mathbf{CX}\|_F$$

For a given $\mathbf{C}$, the optimal $\mathbf{CX} = \mathbf{CC}^{+}\mathbf{A} = \mathbf{P}_{\mathbf{C}}\mathbf{A}$.
i.e. $\mathbf{P}_{\mathbf{C}}$ is the projection matrix onto the colspace of $\mathbf{C}$

Theorem

Given $\mathbf{A}$ *and* $k << \min\{m, n\}$, *a randomized algorithm exists s.t. either exactly* $c = O(k^2 \varepsilon^{-2} \log(1/\delta))$ *columns of* $\mathbf{A}$ *are chosen to construct* $\mathbf{C}$, *or* $c = O(k \log k \, \varepsilon^{-2} \log(1/\delta))$ *in expectation, s.t. w.h.p.* $(1 - \delta)$

$$\|\mathbf{A} - \mathbf{CC}^{+}\mathbf{A}\|_F \leq (1 + \varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$$

Second Result/ Theorem 2

$\mathbf{A}' = \mathbf{CUR}$ is a column-row-based matrix approximation to $\mathbf{A}$, or a $\mathbf{CUR}$ *matrix decomposition*, for any $\mathbf{U} \in \mathbb{R}^{c \times r}$.

$\mathbf{U}$ will be a generalized inverse of the intersection between $\mathbf{C}$ and $\mathbf{R}$. For $\mathbf{C}$ ($\mathbf{R}$), let $\mathbf{S_C}$ ($\mathbf{S_R}$) denote its sampling operator and $\mathbf{D_C}$ ($\mathbf{D_R}$) a diagonal scaling matrix. Then

- $\mathbf{C} = \mathbf{A S_C D_C}$
- $\mathbf{R} = \mathbf{D_R S_R^T A}$
- $\mathbf{U} = (\mathbf{D_R S_R^T A S_C D_C})^+$

Theorem

Given $\mathbf{A}$ *and* $k << \min\{m, n\}$, *randomized algorithm exists s.t. either exactly* $c = O(k^2 \varepsilon^{-2} \log(1/\delta))$ *columns are chosen and then* $r = O(c^2 \varepsilon^{-2} \log(1/\delta))$ *rows are chosen to construct* $\mathbf{R}$, *OR* $c = O(k \log k \, \varepsilon^{-2} \log(1/\delta))$ *in expectation and then* $r = O(c \log c \, \varepsilon^{-2} \log(1/\delta))$ *in expectation, s.t. w.h.p.* $(1-\delta)$

$$\|\mathbf{A} - \mathbf{CUR}\|_F \leq (1+\varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$$

Main Idea: Subspace Sampling

Let $\mathbf{V}_{\mathbf{A},k} \in \mathbb{R}^{n \times k}$ be the top k singular vectors. For column subset selection, the *subspace sampling probabilities* $p_i, i \in [n]$ will satisfy

$$p_i \geq \beta \frac{\|[\mathbf{V}_{\mathbf{A},k}]_{(i)}\|_2^2}{k}, \quad i \in [n]$$

Exactly(c) algorithm: For $t = 1, \ldots, c$, 1. Pick $i_t \in [n]$ w.p p_i. 2. Set $S_{i_t,t} = 1$ 3. Set $D_{tt} = 1/\sqrt{cp_{i_t}}$

Expected(c) algorithm: Probabilites are now $\tilde{p}_i = \min\{1, cp_j\}$. Go through each element $j \in [n]$ and flip a coin with $\tilde{p}_i$ success probability. If picked, set $S_{j,t} = 1$ and $D_{tt} = 1/\sqrt{\tilde{p}_j}$

Relation to ℓ_2 regression

Given input $\mathbf{A}$ and target $\mathbf{B} \in \mathbb{R}^{m \times p}$, compute

$$Z = \min_{\mathbf{X}} \|\mathbf{B} - \mathbf{A}\mathbf{X}\|_F \quad \implies \mathbf{X}_{opt} = \mathbf{A}^+\mathbf{B}$$

Using sampling to get a subspace embedding, consider

$$\tilde{Z} = \min_{\mathbf{X}} \|\mathbf{D}\mathbf{S}^T\mathbf{B} - \mathbf{D}\mathbf{S}^T\mathbf{A}\mathbf{X}\|_F \quad \implies \tilde{\mathbf{X}}_{opt} = (\mathbf{D}\mathbf{S}^T\mathbf{A})^+\mathbf{D}\mathbf{S}^T\mathbf{B}$$

Theorem 3

Constant probability version of Result 1 (with remark to boost it up to $1 - \delta$).

Proof.

Let $\mathbf{P}_{\mathbf{A},k} = \mathbf{U}_{\mathbf{A},k}\mathbf{U}_{\mathbf{A},k}^T$ projection on to top k left singular vectors of $\mathbf{A}$

$$\begin{aligned}
\|\mathbf{A} - \mathbf{C}\mathbf{C}^+\mathbf{A}\|_F &= \|\mathbf{A} - (\mathbf{A}\mathbf{S}_\mathbf{C}\mathbf{D}_\mathbf{C})(\mathbf{A}\mathbf{S}_\mathbf{C}\mathbf{D}_\mathbf{C})^+\mathbf{A}\|_F \\
&\leq \|\mathbf{A} - (\mathbf{A}\mathbf{S}_\mathbf{C}\mathbf{D}_\mathbf{C})(\mathbf{P}_{\mathbf{A},k}\mathbf{A}\mathbf{S}_\mathbf{C}\mathbf{D}_\mathbf{C})^+\mathbf{P}_{\mathbf{A},k}\mathbf{A}\|_F \\
&= \|\mathbf{A} - (\mathbf{C})(\mathbf{P}_{\mathbf{A},k}\mathbf{C})^+\mathbf{P}_{\mathbf{A},k}\mathbf{A}\|_F \\
&= \|\mathbf{A} - (\mathbf{A}\mathbf{S}_\mathbf{C}\mathbf{D}_\mathbf{C})(\mathbf{A}_k\mathbf{S}_\mathbf{C}\mathbf{D}_\mathbf{C})^+\mathbf{A}_k\|_F \\
&\overset{\text{(Thm 5)}}{\leq} (1+\varepsilon)\|\mathbf{A} - \mathbf{A}\mathbf{A}_k^+\mathbf{A}_k\|_F \\
&= (1+\varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F
\end{aligned}$$

□

Sampling

Challenge: How sample s.t. the column-sampled version of the top k right singular vectors of $\mathbf{A}$ is full rank, i.e.

$$\text{rank}(\mathbf{V}_{\mathbf{A},k}^T \mathbf{S_C} \mathbf{D_C}) = \text{rank}(\mathbf{V}_{\mathbf{A},k}^T) = k$$

Answer: Use subspace sampling. Note that

$$\mathbf{A}^{(i)} = \mathbf{U}_k \mathbf{\Sigma}_k [\mathbf{V}_k^T]^{(i)} + \mathbf{U}_{\rho-k} \mathbf{\Sigma}_{\rho-k} [\mathbf{V}_{\rho-k}^T]^{(i)}$$

so $\|[\mathbf{V}_k^T]^{(i)}\|_2^2$ measures "how much" of $\mathbf{A}^{(i)}$ lies in the span of $\mathbf{U}_{\mathbf{A},k}$

CUR: Algorithm 2/Theorem 4

Picking rows? $q_i = \frac{1}{c}\|[\mathbf{U}_{\mathbf{C}}^T]^{(i)}\|_2^2$ (β-dependent accuracy fine)

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{C}$ columns subset of $\mathbf{A}$, $r \in \mathbb{Z}_{++}$ and ε
Output: $\mathbf{R} \in \mathbb{R}^{r \times n}$. $\mathbf{W} \in \mathbb{R}^{c \times r}$ (corresponding rows of $\mathbf{C}$, which gives $\mathbf{U} \in \mathbb{R}^{r \times c}$

1. Compute q_i
2. (Implicitly) construct $\mathbf{S_R}$ and $\mathbf{D_R}$ using Exactly(r) or Expected(r) algorithm
3. Construct $\mathbf{R} = \mathbf{D_R}\mathbf{S_R}^T\mathbf{A}$
4. Construct $\mathbf{W} = \mathbf{D_R}\mathbf{S_R}^T\mathbf{C}$
5. Let $\mathbf{U} = \mathbf{W}^+$

Full SVD of $\mathbf{C}$ is $O(c^2m)$ and $\mathbf{U}$ requires $O(c^2r)$ + lower order terms. So the dominating factor is $O(mn)$ in reading $\mathbf{A}$

ℓ_2-regression: Algorithm 3

Data : $A \in \mathbb{R}^{m \times n}$ that has rank no greater than k, $B \in \mathbb{R}^{m \times p}$, sampling probabilities $\{p_i\}_{i=1}^m$, and $r \leq m$.

Result : $\tilde{X}_{opt} \in \mathbb{R}^{n \times p}$, $\tilde{\mathcal{Z}} \in \mathbb{R}$.

- (Implicitly) construct a sampling matrix S and a diagonal rescaling matrix D with the EXACTLY(c) algorithm or with the EXPECTED(c) algorithm;
- Construct the matrix $DS^T A$ consisting of a small number of rescaled rows of A;
- Construct the matrix $DS^T B$ consisting of a small number of rescaled rows of B;
- $\tilde{X}_{opt} = \left(DS^T A\right)^+ DS^T B$;
- $\tilde{\mathcal{Z}} = \min_{X \in \mathbb{R}^{n \times p}} \left\|DS^T B - DS^T A \tilde{X}_{opt}\right\|_F$;

Theorem 5

Theorem

Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$ *has rank no greater than* k, $\mathbf{B} \in \mathbb{R}^{m \times p}$, $\varepsilon \in (0, 1]$, *and* $Z = \min_{\mathbf{X}} \|\mathbf{B} - \mathbf{A}\mathbf{X}\|_F$ *where* $\mathbf{X}_{opt} = \mathbf{A}^{+}\mathbf{B} = \mathbf{A}_k^{+}\mathbf{B}$.
Running Algorithm 3 with $p_i \geq \frac{\beta}{k}\|[\mathbf{U}_{\mathbf{A},k}]_{(i)}\|_2^2$ *for some* $\beta \in (0, 1]$ *giving output* $\tilde{\mathbf{X}}_{opt}$.
Then if $r = O(k^2/(\beta\varepsilon^2))$ *with Exactly(r) or* $r = O(k \log k/(\beta\varepsilon^2))$ *with Expected(r), we have with constant probability*

$$\|\mathbf{B} - \mathbf{A}\tilde{\mathbf{X}}_{opt}\|_F \leq (1 + \varepsilon)Z$$

Prior art

Sub-optimal and **randomized** algorithms.

	c	r	$\mathrm{rank}(\mathrm{U})$	$\|\mathrm{A}-\mathrm{CUR}\|_{\mathrm{F}}^2 \leq$	Time
1	k/ε^2	k/ε	k	$\|\mathrm{A}-\mathrm{A}_k\|_{\mathrm{F}}^2+\varepsilon\|\mathrm{A}\|_{\mathrm{F}}^2$	$nnz(\mathrm{A})$
2	k/ε^4	k/ε^2	k	$\|\mathrm{A}-\mathrm{A}_k\|_{\mathrm{F}}^2+\varepsilon\|\mathrm{A}\|_{\mathrm{F}}^2$	$nnz(\mathrm{A})$
3	$(k\log k)/\varepsilon^2$	$(k\log k)/\varepsilon^4$	$(k\log k)/\varepsilon^2$	$(1+\varepsilon)\|\mathrm{A}-\mathrm{A}_k\|_{\mathrm{F}}^2$	n^3
4	$(k\log k)/\varepsilon^2$	$(k\log k)/\varepsilon^2$	$(k\log k)/\varepsilon^2$	$(2+\varepsilon)\|\mathrm{A}-\mathrm{A}_k\|_{\mathrm{F}}^2$	n^3
5	k/ε	k/ε^2	k/ε	$(1+\varepsilon)\|\mathrm{A}-\mathrm{A}_k\|_{\mathrm{F}}^2$	n^2k/ε

References:

1. Drineas and Kannan. Symposium on Foundations of Computer Science, 2003.
2. Drineas, Kannan, and Mahoney. SIAM Journal on Computing, 2006.
3. Drineas, Mahoney, and Muthukrishnan. SIAM Journal on Matrix Analysis, 2008.
4. Drineas and Mahoney. Proceedings of the National Academy of Sciences, 2009.
5. Wang and Zhang. Journal of Machine Learning Research, 2013.

Lower bound

Theorem

Fix appropriate matrix $\mathsf{A} \in \mathbb{R}^{n \times n}$. *Consider a factorization* CUR,

$$\|\mathsf{A} - \mathsf{CUR}\|_{\mathrm{F}}^2 \leq (1+\varepsilon)\|\mathsf{A} - \mathsf{A}_k\|_{\mathrm{F}}^2.$$

Then, for any $k \geq 1$ *and for any* $\varepsilon < 1/3$*:*

$$c = \Omega(k/\varepsilon),$$

and

$$r = \Omega(k/\varepsilon),$$

and

$$\mathrm{rank}(\mathsf{U}) \geq k/2.$$

Extended lower bound in [Deshpande and Vempala, 2006], [Boutsidis et al, 2011], [Sinop and Guruswami, 2011]

Input-sparsity-time CUR

Theorem

There exists a randomized algorithm to construct a CUR with

$$c = O(k/\varepsilon)$$

and

$$r = O(k/\varepsilon)$$

and

$$\operatorname{rank}(U) = k$$

such that, with constant probability of success,

$$\|A - CUR\|_F^2 \leq (1+\varepsilon)\|A - A_k\|_F^2.$$

Running time: $O\left(nnz\left(A\right)\log n + (m+n)\cdot poly\left(\log n, k, 1/\varepsilon\right)\right).$

Adaptive Sampling

Adaptive Sampling method [Wang '13] works by

1. Approximating SVD (compute or random projection)
2. Dual Set Sparsification (DSS) Sampling
3. Adaptive Sampling (i.e. based on $\mathbf{E} = \mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}$)

Algorithm 2 Adaptive Sampling for CUR.

1: **Input:** a real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, target rank k, $\epsilon \in (0, 1]$, target column number $c = \frac{2k}{\epsilon}\big(1+o(1)\big)$, target row number $r = \frac{c}{\epsilon}(1+\epsilon)$;
2: Select $c = \frac{2k}{\epsilon}\big(1+o(1)\big)$ columns of $\mathbf{A}$ to construct $\mathbf{C} \in \mathbb{R}^{m \times c}$ using Algorithm 1;
3: Select $r_1 = c$ rows of $\mathbf{A}$ to construct $\mathbf{R}_1 \in \mathbb{R}^{r_1 \times n}$ using Algorithm 1;
4: Adaptively sample $r_2 = c/\epsilon$ rows from $\mathbf{A}$ according to the residual $\mathbf{A} - \mathbf{A}\mathbf{R}_1^\dagger\mathbf{R}_1$;
5: **return** $\mathbf{C}$, $\mathbf{R} = [\mathbf{R}_1^T, \mathbf{R}_2^T]^T$, and $\mathbf{U} = \mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger$.

Algorithm 1 here refers to the Near-Optimal Column Selection Algorithm of Boutsidis et al. (2011)

Towards More Efficient Nyström Approximation and CUR Matrix Decomposition [on Arxiv, March 29 2015]

References

Main Paper:
Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844881, September 2008.

Woodruff MMDS Slides: `http://researcher.watson.ibm.com/researcher/files/us-dpwoodru/mmds.pdf`

CUR with Adaptive Sampling Code:
`https://sites.google.com/site/zjuwss/`

CUR in **R**:
`http://cran.r-project.org/web/packages/rCUR/rCUR.pdf`