Approximate CUR Matrix Decomposition and Applications

Kathryn Linehan[1,2] Radu Balan[2]

[1]University of Virginia, Biocomplexity Institute, Social and Decision Analytics Division

[2]University of Maryland, Department of Mathematics, Applied Mathematics & Statistics and Scientific Computation Program

Joint Mathematics Meetings, January 7, 2023

Table of Contents

What is CUR?

A **low-rank matrix approximation**

$$\underset{m\times n}{\mathbf{X}} \approx \underset{m\times c}{\mathbf{C}} \; \underset{c\times r}{\mathbf{U}} \; \underset{r\times n}{\mathbf{R}},$$

where generally $c \ll n$ and $r \ll m$.

General idea:

1. Choose c columns of $\mathbf{X}$. Let $\mathbf{C}$ contain these columns.
2. Choose r rows of $\mathbf{X}$. Let $\mathbf{R}$ contain these rows.
3. Compute $\mathbf{U}$ so that $\mathbf{CUR}$ is a good approximation to $\mathbf{X}$.

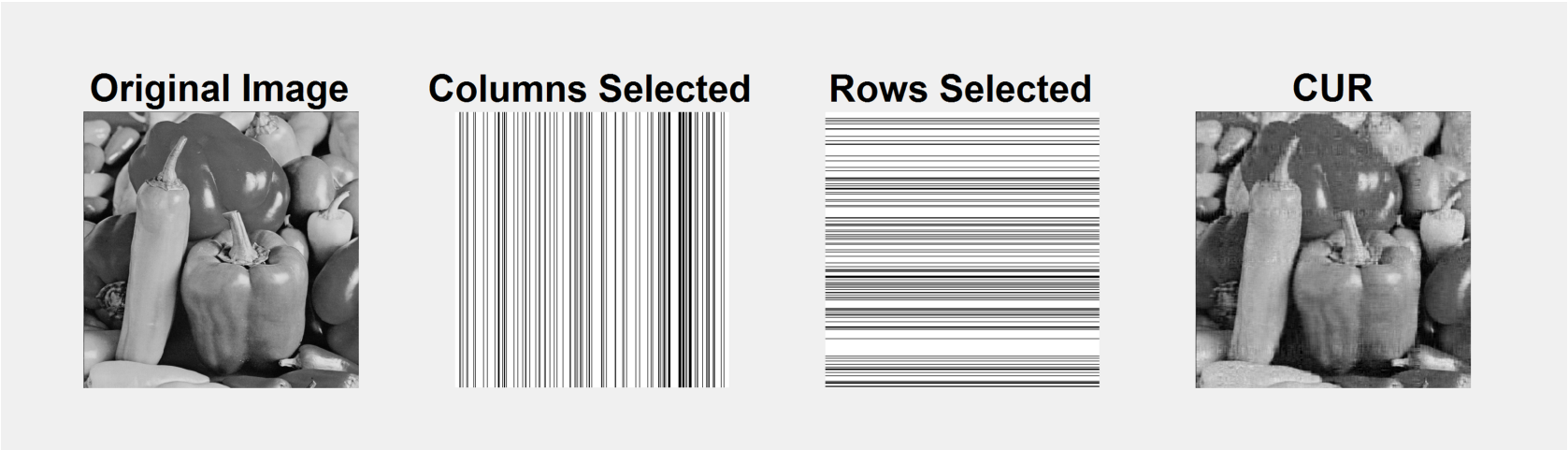
CUR algorithms can be randomized [2, 4, 5, 10] or deterministic [12].

CUR Example: Image Data

Original Image: 512×512

CUR Algorithm: Due to Mahoney and Drineas [10] (columns/rows randomly selected based on leverage score probability distribution)

- **Parameters**: $c = r = 100$
- **Relative Error of CUR Approximation**: 0.0093



Motivation for Using CUR

Low-rank matrix approximations are common application tools.

- Ex) Principal Component Analysis (PCA), signal denoising, least squares
- the truncated Singular Value Decomposition (SVD) can be used

CUR allows the scientist to interpret the results in terms of the original data.

- CUR preserves the structure of the original data in **C** and **R**.
- Ex) if the original data is sparse, **C** and **R** will be sparse

Motivation for Using CUR

CUR provides interpretation by selecting the most 'important' columns (rows) of the matrix.

Examples [10] [12]:

- influential terms in a set of documents
- classification using genetic data
- feature selection

Applications may not need the full CUR factorization. Sometimes an application only requires the matrix $\mathbf{C}$ (or $\mathbf{R}$).

CUR through Convex Optimization

There is some previous work in CUR algorithms using convex optimization [1, 9, 11].

Main Contributions:

- novel convex optimization formulation
- algorithm that solves for **C** and **R** separately and allows the user to select the number of columns in **C** and rows in **R** (common features in randomized CUR algorithms)
- an implementation utilizing the "surrogate functional" technique of [3] which we adapted for use with a new penalty function
- an algorithm and implementation strategy that can accommodate a variety of penalty functions, allowing the user a flexible framework for CUR through convex optimization

CUR through Convex Optimization - Compute **C**

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times m}} ||\mathbf{X}-\mathbf{X}\mathbf{W}\mathbf{X}||_F^2 + \lambda_C \sum_{i=1}^{n} ||\mathbf{W}(i,:)||_\infty$$

Intuition:

- The penalty function enforces sparsity in **W** such that some rows are zero and others are nonzero.
- The indices of the nonzero rows of **W** are the indices of the columns we should select from **X**.
- λ_C controls how many columns are selected from **X**.

CUR through Convex Optimization - Compute **C**

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times m}} ||\mathbf{X}-\mathbf{X}\mathbf{W}\mathbf{X}||_F^2 + \lambda_C \sum_{i=1}^{n} ||\mathbf{W}(i,:)||_\infty \tag{1}$$

Formally:

1. Find **W** from Equation 1.
2. Let J be the set of indices of nonzero rows of **W**.
3. If $|J|$ is the user-specified number of columns, go to step 4. Otherwise, use bisection to compute a new value of λ_C and repeat steps 1-3.
4. $\mathbf{C} = \mathbf{X}(:, J)$

CUR through Convex Optimization - Compute **R**

$$\min_{\mathbf{W}\in\mathbb{R}^{c\times m}} ||\mathbf{X} - \mathbf{CWX}||_F^2 + \lambda_R \sum_{j=1}^{m} ||\mathbf{W}(:,j)||_\infty \tag{2}$$

Formally:

1. Find **W** from Equation 2.
2. Let I be the set of indices of nonzero columns of **W**.
3. If $|I|$ is the user-specified number of rows, go to step 4. Otherwise, use bisection to compute a new value of λ_R and repeat steps 1-3.
4. $\mathbf{R} = \mathbf{X}(I,:)$

CUR through Convex Optimization - Compute **U**

Given **C** and **R**, solve the least squares problem

$$\min_{\mathbf{U}} ||\mathbf{X} - \mathbf{CUR}||_F.$$

The solution is given by $\mathbf{U} = \mathbf{C}^+\mathbf{X}\mathbf{R}^+$ [8].

Minimizations

There is potential for many minimizations to be solved in one run of the algorithm.

Small to Medium sized original data:

- can use a convex solver such as CVX in MATLAB [7].

Large sized original data:

- CVX and other packages either cannot accommodate the storage for the problem, or are too slow.
- we use the surrogate functional technique of [3] to make this algorithm computationally feasible.

Surrogate Functional to Compute C

We adapt the method of Daubechies, Defrise, and De Mol [3] to solve the minimizations in our algorithm.

Original Problem:

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times m}} ||\mathbf{X}-\mathbf{XWX}||_F^2 + \lambda_C \sum_{i=1}^{n} ||\mathbf{W}(i,:)||_\infty$$

Problem with Surrogate Functional: (for a given $\mathbf{Z}$)

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times m}} \underbrace{||\mathbf{X}-\mathbf{XWX}||_F^2 + \lambda_C \sum_{i=1}^{n} ||\mathbf{W}(i,:)||_\infty + \mu||\mathbf{W}-\mathbf{Z}||_F^2 - ||\mathbf{XWX}-\mathbf{XZX}||_F^2}_{\widehat{J}(\mathbf{W},\mathbf{Z})}$$

Surrogate Functional to Compute **C**

- Using the surrogate functional decouples the minimization problem so that we can solve for each row of **W** separately.
- Instead of solving one large problem, we solve n smaller proximal operator problems:

$$\min_{\mathbf{y}\in\mathbb{R}^m} \left[||\mathbf{y}-\mathbf{s}||_2^2 + \alpha||\mathbf{y}||_\infty\right].$$

How do we get the solution to our original problem?
Iterate: $\mathbf{W}_0 = 0$, $\mathbf{W}_k = \text{argmin}_{\mathbf{W}\in\mathbb{R}^{n\times m}} \left[\widehat{J}(\mathbf{W},\mathbf{W}_{k-1})\right]$

The sequence $\{\mathbf{W}_k\}$ converges to the solution of our original problem!

Convex Optimization CUR Framework

We can use this algorithm and implementation strategy with other penalty functions as long as the penalty function can be decoupled by columns and rows.

Examples:

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times m}} ||\mathbf{X}-\mathbf{XWX}||_F^2 + \lambda_C \sum_{i=1}^{n} ||\mathbf{W}(i,:)||_1$$

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times m}} ||\mathbf{X}-\mathbf{XWX}||_F^2 + \lambda_C ||\mathbf{W}||_F^2$$

Application - Natural Language Processing (NLP)

Data: TechTC document-term matrix, $\mathbf{X}$, created from 139 documents, each about Florida or Evansville, Indiana [10] [12].

- Florida: 71 documents, Evansville: 68 documents
- Preprocessing: terms of length four or fewer filtered out of the data, rows normalized to length one.
- Matrix size: $139 \times 15{,}170$

Results: We ran our CUR algorithm on $\mathbf{X}$, with $c = 20$, $r = 10$.

C: columns of $\mathbf{X}$ corresponding to the terms florida, click, miami, email, south, contact, first, please, information, service, their, business, events, about, other, links, services, spacer, indiana, evansville

Application - Genetics, Tumor Detection [12]

Data: NIH gene expression dataset

- 22,283 gene probes, 107 patients
- measurements of patient responses to each gene probe (mean centered)
- we know in advance which patients have a tumor (58) and which do not (49)

Goal: identify the most "important" probes to classify if a patient has a tumor or not (this is the subset of probes chosen for **R**)

Results: our CUR performs better than the deterministic CUR of [12], and the same as a deterministic variant of the CUR of [10].

Interpretation of **U**

We can write **X** as a sum of weighted column-row outer products.

$$\mathbf{X} \approx \mathbf{CUR} = \sum_{i=1}^{c} \sum_{j=1}^{r} \mathbf{U}_{ij} \mathbf{C}(:, i) \mathbf{R}(j, :)^T$$

- **U** contains the weighting factors.
- Interpretation: $\mathbf{U}_{ij}$ is how "important" the ith column - jth row pair is to reconstruction of the original matrix **X** [6].

Application - Joint Sensor Selection & Channel Assignment [6]

Data: Cognitive Radio Network

- 900 sensor locations, 32 frequency channels
- measurements of received power levels

Goal: Identify the best locations for a small number of sensors and determine which frequency channels to assign to each sensor (could vary by sensor)

Results: using the CUR algorithm of [6]

- **R**: contained the sensor locations (20 to 80)
- **U**: used to select 8 frequency channels for each sensor

Conclusions

- We presented a novel CUR algorithm that utilizes convex optimization.
- We showed how the surrogate functional technique of [3] can be used in the implementation of our algorithm; this allows for the use of big data.
- Our formulation and implementation strategy provide a flexible framework for use with a variety of penalty functions.
- CUR can be used in applications which require interpretation in terms of the original data.

References

[1] Jacob Bien, Ya Xu, and Michael W. Mahoney. "CUR from a Sparse Optimization Viewpoint". In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*. NIPS'10. Vancouver, British Columbia, Canada: Curran Associates Inc., 2010, pp. 217–225.

[2] Christos Boutsidis and David P. Woodruff. "Optimal CUR Matrix Decompositions". In: *SIAM Journal on Computing* 46.2 (2017), pp. 543–589. DOI: https://doi.org/10.1137/140977898.

[3] Ingrid Daubechies, Michel Defrise, and Christine De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Communications on Pure and Applied Mathematics* 57.11 (2004), pp. 1413–1457.

[4] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. "Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition". In: *SIAM Journal on Computing* 36.1 (2006), pp. 184–206. DOI: 10.1137/S0097539704442702.

[5] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. "Relative-Error *CUR* Matrix Decompositions". In: *SIAM Journal on Matrix Analysis and Applications* 30.2 (2008), pp. 844–881. DOI: 10.1137/07070471X.

[6] Ashkan Esmaeili et al. "Two-Way Spectrum Pursuit for CUR Decomposition and its Application in Joint Column/Row Subset Selection". In: *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. 2021, pp. 1–6. DOI: 10.1109/MLSP52302.2021.9596233.

[7] Michael Grant and Stephen Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. http://cvxr.com/cvx. Mar. 2014.

[8] Keaton Hamm and Longxiu Huang. "Perspectives on CUR decompositions". In: *Applied and Computational Harmonic Analysis* 48.3 (2020), pp. 1088–1099. DOI: https://doi.org/10.1016/j.acha.2019.08.006.

[9] Yasutoshi Ida et al. "Fast Deterministic CUR Matrix Decomposition with Accuracy Assurance". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 4594–4603. URL: https://proceedings.mlr.press/v119/ida20a.html.

[10] Michael W. Mahoney and Petros Drineas. "CUR matrix decompositions for improved data analysis". In: *Proceedings of the National Academy of Sciences* 106.3 (2009), pp. 697–702. DOI: 10.1073/pnas.0803205106.

[11] Julien Mairal et al. "Convex and Network Flow Optimization for Structured Sparsity". In: *J. Mach. Learn. Res.* 12.null (Nov. 2011), pp. 2681–2720. ISSN: 1532-4435.

[12] D. C. Sorenson and Mark Embree. "A DEIM induced CUR factorization". In: *SIAM Journal on Scientific Computing* 38.3 (2016), A1454–A1482. DOI: 10.1137/140978430.