Data Retrieval with Tensor Completion using t-CUR

Suvendu Kar
Indian Institute of Science
Bengaluru, KA, India
suvendukar@iisc.ac.in

March 29, 2024

Abstract

Matrix completion is a useful technique for data analysis; tensor completion is its higher level analog. In this work, we formulate the three-way tensor's missing data recovery problem as a tensor completion problem. We provide a new approach to tensor completion using tensor-CUR breakdown in order to calculate the missing data from small sample sizes. Computational experiments show that compared to other current methods, the suggested strategy performs better in certain cases.

Index Terms: Data recovery, tensor completion, tensor-CUR decomposition

1 Introduction

In this work, we address the issue of missing data recovery of a 3-way tensor and define it as a tensor completeness problem, drawing inspiration from several effective applications of the tensor pattern.CP[8−10] based decomposition requires to know (CP rank:The CP rank is defined by the minimum number of rank-one terms in CP decomposition) rank beforehand (calculating rank is NP complete[11]).Moreover, the best rank-K approximation to a tensor may not always exist in the CP approach. The tensor-SVD[12] algorithm is computationally expensive, since it requires computing the SVD decomposition of the approximate matrix at each iteration of the optimization.

Our suggested method(T product bases CUR decomposition i.e. t-CUR), which computes the low rank approximation of a given tensor using the tensor's actual rows and columns, extends matrix-CUR decomposition to tensor. Because it simply has to resolve a typical regression problem, it is computationally efficient. Furthermore, our suggested method may effectively compute the low rank approximation of a given tensor utilizing the tensor's real rows and columns without requiring knowledge of the rank beforehand.

2 Application of Tensor Completion

- Image Analysis[1,2]
- Recommender System[3,4]
- wireless spectrum map construction[5]
- seismology signal processing[6]
- computer vision[7]

3 Matrix Based CUR

Let us first tell briefly about the matrix based CUR decomposition[13,14], before jumping into Tensor T-product based CUR.

- RESULT1

  Let the matrix A $\in F^{n1\times n2}$ have rank(A)= r .Let I $\subset$ {1:n1} and J $\subset$ {1:n2} satisfying |I|$\geq r, |J| \geq r$. Let matrices C=A(:,J),R=(I,:),U=A(I,J),if rank(U)=rank(A),then A=CU†R.[U†, being pseudoinverse of U]

- THEOREM[15](Mahone and Drineas):

  CUR in O(n1×n2) time achieves $||A - CUR||_F \leq ||A - A_k||_F + \epsilon||A||_F$ with probability at least (1-δ), by picking O(klog(1/δ)/ϵ^2) columns , and O($\frac{k^2 log^3(1/\delta)}{\epsilon^6}$)

- RESULT[16]:

  Select c=O(klogk/ϵ^2) number of columns of Ausing column select algorithm[16], and select r=O(klogk/ϵ^2) rows of A using row select algorithm[16].Set U=W†, W being intersection of r rows and c columns.Then , we have $||A - CUR||_F \leq ||A - A_k||_F * (2 + \epsilon)$, with probability 98%.

We used a special sampling technique(based on finding row and columns with maximum relative Frobenius norm) for sampling row and columns.Here is the the pseudocode for row sampling(same method applied for column sampling)

Algorithm 1 ROW_SAMPLING_ALGO(Input: matrix A, number of sampled row c.Output:selected row indices and submatrix of A with selected rows.)

```
procedure ROW_SAMPLING_ALGO(A, c)
    m ← size(A, 1)
    n ← size(A, 2)
    if c > m then
        Print "Number of rows you want to consider exceeding total number of rows"
    end if
    p ← zeros(m, 1)
    indices ← zeros(c, 1)
    t ← norm(A, fro)^2
    for x from 1 to m do
        p(x) ← norm(A(x, :), fro)^2/t
    end for
    A_row_sub ← zeros(c, n)
    i ← 1
    while i ≤ c do
        [max_value, max_index] ← max(p)
        indices(i) ← max_index
        A_row_sub(i, :) ← A(max_index, :)
        p(max_index) ← min(p) − i
        i ← i + 1
    end while
    indices ← sort(indices)
    A_row_sub ← A(indices, :)
    return A_row_sub, indices
end procedure
```

4 Useful Definitions

- Definition (T-Product [12]): The T-product $\mathscr{A} * \mathscr{B}$ of tensor $\mathscr{A} \in R^{I_1 \times I_2 \times I_3}$ and $\mathscr{B} \in R^{I_2 \times I_4 \times I_3}$ is an $I_1 \times I_4 \times I_3$ tensor whose $(i,j)^{\text{th}}$ tube $\mathscr{C}(i,j,:)$ is given by

$$\mathscr{C}(i,j,:) = \sum_{k=1}^{I_2} \mathscr{A}(i,k,:) * \mathscr{B}(k,j,:),$$

where $*$ denotes the circular convolution between two tubes of the same size. We anchor the MatVec command to the frontal faces of the tensor. MatVec($\mathscr{A}$) takes an $I_1 \times I_2 \times I_3$ tensor and returns a block $I_1 I_2 \times I_3$ matrix, whereas the fold command undoes this operation

$$\text{MatVec}(\mathscr{A}) = \begin{bmatrix} \mathscr{A}^{(1)} \\ \mathscr{A}^{(2)} \\ \vdots \\ A^{(I_3)} \end{bmatrix}, \quad \text{fold (MatVec}(\mathscr{A})) = \mathscr{A}.$$

Then the T-product $\mathscr{A} * \mathscr{B}$ is the $I_1 \times I_4 \times I_3$ tensor

$$\mathscr{A} * \mathscr{B} = \text{fold}(\text{circ}(\mathscr{A}) \cdot \text{Mat Vec}(\mathscr{B}))$$

Zhang and Aeron [12] showed that for any tensor $\mathscr{A} \in R^{I_1 \times I_2 \times 3}$ and $\mathscr{B} \in R^{I_2 \times l \times 3}$, then $\mathscr{A} * \mathscr{B} = \mathscr{C} \Leftrightarrow \overline{\mathscr{A}\mathscr{B}} = \overline{\mathscr{C}}$. We are able to transform the t-product into its equivalent form in Fourier domain. On the other hand, we can also transform an operator in Fourier domain back to the original domain as needed.

- Definition (Identity Tensor [12]): The identity tensor $\mathscr{I} \in R^{n_1 \times n_1 \times n_3}$ is defined to be a tensor whose first frontal slice $\mathscr{I}^{(1)}$ is the $n_1 \times n_1$ identity matrix and all other frontal slices $\mathscr{I}^{(i)}; i = 2, \ldots, n_3$ are zero.
- Definition (Orthogonal Tensor [12]): A tensor $\mathscr{A} \in R^{n_1 \times n_1 \times n_3}$ is orthogonal if it satisfies

$$\mathscr{A}^T * \mathscr{A} = \mathscr{A} * \mathscr{A}^T = \mathscr{I}$$

- Definition (Inverse of Tensor [12]): The inverse of a tensor $\mathscr{A} \in R^{I_1 \times I_2 \times I_3}$ is written as $\mathscr{A}^{-1}$ satisfying

$$\mathscr{A}^{-1} * \mathscr{A} = \mathscr{A} * \mathscr{A}^{-1} = \mathscr{I}$$

5 Used Tensor Decompositions

5.1 Tensor t-CUR algorithm

Suppose we sample a 3-way tensor $\mathscr{A} \in R^{I_1 \times I_2 \times I_3}$ at the set of indices in a set Ω. Let P_Ω denotes the sampling operator

$$P_\Omega : R^{I_1 \times I_2 \times I_3} \to R^{I_1 \times I_2 \times I_3},$$

which is defined by

$$P_\Omega(\mathscr{A})_{ijk} = \begin{cases} \mathscr{A}_{ijk} & (i,j,k) \in \Omega \\ 0 & otherwise \end{cases}.$$

- Algorithm : Tensor-CUR Approximation From Partially Observed Entries (T-CUR) Require: Observation data $P_\Omega(\mathscr{A}) \in R^{I_1 \times I_2 \times I_3}$, vector of sample columns/rows number $d/l \in R^{1 \times I_3}$

1. $\hat{\mathscr{A}} = FFT\left(P_\Omega(\mathscr{A}), [], 3\right)$;
2. for $i \leftarrow 1, \ldots, I_3$ do
3. $\left[\hat{\mathscr{C}}^{(i)}, \hat{\mathscr{U}}^{(i)}, \hat{\mathscr{R}}^{(i)}\right] = \text{M-CUR}\left(P_\Omega\left(\hat{\mathscr{A}}^{(i)}\right), d_i, l_i\right)$

4.end for

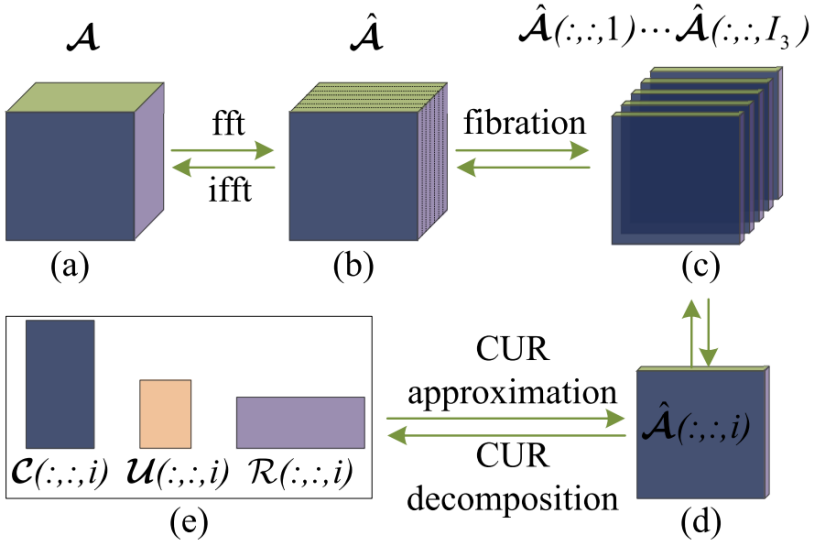
5.$\mathscr{C} = \text{IFFT}(\hat{\mathscr{C}}, [], 3); \mathscr{U} = \text{IFFT}(\hat{\mathscr{U}}, [], 3); \mathscr{R} = \text{IFFT}\ (\ \hat{\mathscr{R}}, [], 3)$

6.Return a tensor cur approximation of $\mathscr{A} : \mathscr{Y} = \mathscr{C} * \mathscr{U} * \mathscr{R} \in R^{I_1 \times I_2 \times I_3}$

5.2 t-SVD

```
function [C, U, R] = TENSOR_CUR(A, d1, d2)
    % Input: A - a tensor of size n1 x n2 x n3
    %        d1 - sample size for rows
    %        d2 - sample size for columns

    A_tilde = fft(A,[],3);

    % Sample row index I and column index J
    %I = randsample(size(A,1), d1);
    %J = randsample(size(A,2), d2);

    C = zeros([size(A,1), d2, size(A,3)]);
    R = zeros([d1, size(A,2), size(A,3)]);
    U = zeros([d2,d1,size(A_tilde ,3)]);

    for k= 1:size(A_tilde ,3)
        C(:,:,k) = A_tilde(:,1:d2,k);
        R(:,:,k) = A_tilde(1:d1,:,k);
        U(:,:,k) = MAT_PSEUDOINV(A_tilde(1:d1,1:d2,k));
    end

    C = ifft(C,[],3);
    R = ifft(R,[],3);
    U = ifft(U,[],3);
end
```

(a) Used Algorithm



(b) Pictorial Presentation

Figure 1: t-CUR

Algorithm 2 tsvd(Input: 3 dimensional Tensor A.Output: Three dimensional tensor U,orthonormal tensors with left singular vectors;S, diagonal tensor with singular values; V, orthogonal tensor with right singular values)

procedure tsvd(X, opt)
  if ¬exist(opt) then
    $opt \leftarrow' full'$
  end if
  $(n1, n2, n3) \leftarrow \text{size}(X)$
  $X \leftarrow \text{fft}(X, [], 3)$
  if *opt* is 'skinny' or *opt* is 'econ' then
    $min12 \leftarrow \min(n1, n2)$
    $U \leftarrow \text{zeros}(n1, min12, n3)$
    $S \leftarrow \text{zeros}(min12, min12, n3)$
    $V \leftarrow \text{zeros}(n2, min12, n3)$
    $halfn3 \leftarrow \lceil (n3+1)/2 \rceil$
    for $i \leftarrow 1$ to $halfn3$ do
      $(U(:,:,i), S(:,:,i), V(:,:,i)) \leftarrow \text{svd}(X(:,:,i),' econ')$
    end for
    for $i \leftarrow halfn3 + 1$ to $n3$ do
      $U(:,:,i) \leftarrow \text{conj}(U(:,:,n3+2-i))$
      $V(:,:,i) \leftarrow \text{conj}(V(:,:,n3+2-i))$
      $S(:,:,i) \leftarrow S(:,:,n3+2-i)$
    end for
    if *opt* is 'skinny' then
      $s1 \leftarrow \text{diag}(\sum(S,3)/n3^2)$
      $tol \leftarrow \max(n1, n2) \times \text{eps}(\max(s1))$
      $trank \leftarrow \sum(s1 > tol)$ ▷ tensor tubal rank
      $U \leftarrow U(:, 1:trank, :)$
      $V \leftarrow V(:, 1:trank, :)$
      $S \leftarrow S(1:trank, 1:trank, :)$
    end if
  else if *opt* is 'full' then
    $U \leftarrow \text{zeros}(n1, n1, n3)$
    $S \leftarrow \text{zeros}(n1, n2, n3)$
    $V \leftarrow \text{zeros}(n2, n2, n3)$
    $halfn3 \leftarrow \lceil (n3+1)/2 \rceil$
    for $i \leftarrow 1$ to $halfn3$ do
      $(U(:,:,i), S(:,:,i), V(:,:,i)) \leftarrow \text{svd}(X(:,:,i))$
    end for
    for $i \leftarrow halfn3 + 1$ to $n3$ do
      $U(:,:,i) \leftarrow \text{conj}(U(:,:,n3+2-i))$
      $V(:,:,i) \leftarrow \text{conj}(V(:,:,n3+2-i))$
      $S(:,:,i) \leftarrow S(:,:,n3+2-i)$
    end for
  end if
  $U \leftarrow \text{ifft}(U, [], 3)$
  $S \leftarrow \text{ifft}(S, [], 3)$
  $V \leftarrow \text{ifft}(V, [], 3)$
  return U, S, V
end procedure

5.3 t-Eigen Value Decomposition

Algorithm 3 T_evd_reduced(Input: Tensor A, and number of eigen values we want to consider.Output: Corresponding Eigen values, and Eigen vectors)

```
procedure T_evd_reduced(A, n)
    n1 ← size(A, 1)
    n2 ← size(A, 2)
    n3 ← size(A, 3)
    if ¬(n1 == n2) then
        Print "Eigen value decomposition is not possible"
    end if
    if n > min(n1, n2) then
        Print "Change the number of eigen values you want to consider as it is exceeding the actual number."
    end if
    A ← fft(A, [], 3)
    U ← zeros([n1, n1, n3])
    S ← zeros([n1, n1, n3])
    U_inv ← zeros([n1, n1, n3])
    for i ← 1 to n3 do
        [U(:, :, i), S(:, :, i)] ← eig(A(:, :, i))
        U_inv(:, :, i) ← MAT_PSEUDOINV(U(:, :, i))
    end for
    S_temp ← zeros(size(S))
    for i ← 1 to n3 do
        S_temp(1 : n, 1 : n, i) ← S(1 : n, 1 : n, i)
    end for
    U_ifft ← ifft(U, [], 3)
    S_ifft ← ifft(S_temp, [], 3)
    U_inv_ifft ← ifft(U_inv, [], 3)
    return U_ifft, S_ifft, U_inv_ifft
end procedure
```

6 CUR vs Other Methods

6.1 Problem Formulation

Definition (Tensor CUR Decomposition (T-CUR)): For a 3-way tensor $\mathscr{A} \in R^{I_1 \times I_2 \times I_3}$, the tensor-CUR decomposition is given by

$$\mathscr{A} = \mathscr{C} * \mathscr{U} * \mathscr{R},$$

where $\mathscr{C}, \mathscr{U}$ and $\mathscr{R}$ are tensors of size $I_1 \times I_1 \times I_3, I_1 \times I_2 \times I_3$ and $I_2 \times I_2 \times I_3$ respectively.

The missing data recovery problem(with low rank approximation $\mathscr{Y}$) can be formulated as a tensor completion problem with the goal of finding its missing entries through the following optimization

$$\min_{\mathscr{U} \in R^{I_1 \times I_2 \times I_3}} \frac{1}{2} \|P_\Omega(\mathscr{Y} - \mathscr{C} * \mathscr{U} * \mathscr{R})\|_F^2$$
$$\text{subject to } P_\Omega(\mathscr{Y}) = P_\Omega(\mathscr{A})$$

This section compares t-CUR with past approaches.

6.2 CP vs CUR

Suppose we sample a 3-way tensor $\mathscr{A} \in R^{I_1 \times I_2 \times I_3}$ at the set of indices in a set Ω. Let P_Ω denotes the sampling operator

$$P_\Omega : R^{I_1 \times I_2 \times I_3} \to R^{I_1 \times I_2 \times I_3},$$

which is defined by

$$P_\Omega(\mathscr{A})_{ijk} = \begin{cases} \mathscr{A}_{ijk} & (i,j,k) \in \Omega \\ 0 & otherwise \end{cases}.$$

Jain and Oh [17] tried to complete the tensor by solving the following optimization problem

$$minimize_{\hat{\mathscr{A}},rank(\hat{\mathscr{A}})=r}||P_\Omega(\mathscr{A}-\Sigma_{l=1}^{r}\sigma_l(a_l^{(1)}\circ a_l^{(2)}\circ a_l^{(3)}))||_F^2.$$

They showed that under certain standard assumptions, the proposed method can recover a three-mode $n\times n\times n$ dimensional rank-r tensor exactly from $O(n^{3/2}\cdot r^5 log^4 n)$ randomly sampled entries.But knowing r is NP complete.

6.3 Tucker Decomposition vs CUR

Liu et al. [7] proposed a tensor completion algorithm based on minimizing tensor n-rank in Tucker decomposition format. It uses the matrix nuclear norm instead of matrix rank, and try to solve the convex problem as follows

$$\min_{\mathscr{X}} \sum_{i=1}^{n} \alpha_i \left\|X_{(i)}\right\|_*$$
$$\text{subject to } P_\Omega(\mathscr{X}) = P_\Omega(\mathscr{A}),$$

where $X_{(i)}$ is the mode- n matricization of $\mathscr{X}$ and $\alpha_{(i)}$ are prespecified constants satisfying $\alpha_{(i)}\geq 0, \sum_{i=1}^{n}\alpha_{(i)}=1$.

Unlike the optimal dimensionality reduction in matrix PCA which can be obtained by truncating the SVD, there is no trivial multi-linear counterpart to dimensionality reduction for Tucker type. Alternatively, suitable choices of the truncation values of n-rank are not likely to be known a priori [18]. As a contrast, our proposed algorithm use the actual rows and columns of the tensor to efficiently compute the low rank approximation of a given tensor, without having to know the rank a prior.

6.4 SVD vs CUR

The tensor tubal rank is used in tensor-SVD, also referred to as tensor rank. Zhang et al. tried to complete the tensor by solving the following convex optimization problem

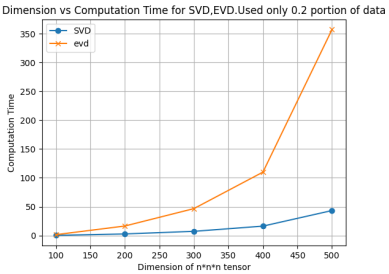
$$\min_{\mathscr{X}} \|\mathscr{X}\|_{TNN}$$
$$\text{subject to } P_\Omega(\mathscr{X}) = P_\Omega(\mathscr{A}),$$

where the tensor nuclear norm is taken as the convex relaxation of tensor tubal rank, $\|*\|_{TNN}$ is the tensor nuclear norm. Zhang et al. showed that one can perfectly recover a tensor of size $I_1\times I_2\times I_3$ with rank r under tensor-SVD as long as $O\left(rI_1I_3\log\left(\left(I_1+I_2\right)I_3\right)\right)$ samples are observed.

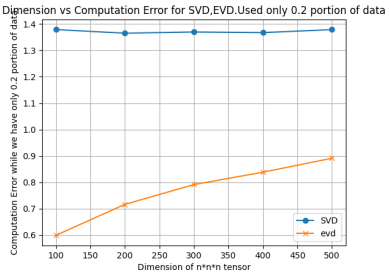
Our suggested method, which computes the low rank approximation of a given tensor using the tensor's actual rows and columns, extends matrix-CUR decomposition to tensor. Because it simply has to resolve a typical regression problem, it is computationally efficient. Furthermore, our suggested method may effectively compute the low rank approximation of a given tensor utilizing the tensor's real rows and columns without requiring knowledge of the rank beforehand.

6.5 SVD vs EVD

Let's consider $n\times n\times n$ Tensors with n=[100,200,300,400,500], and let's consider that we have only 0.2portion of data in hand.Now let's analyze their computation time and error in approximation.



(a) Their Taken Time(sec) of Computation



(b) SVD vs EVD in terms of error

Figure 2: SVD vs EVD

7 Obtained Results

7.1 Mathematical Guarantees

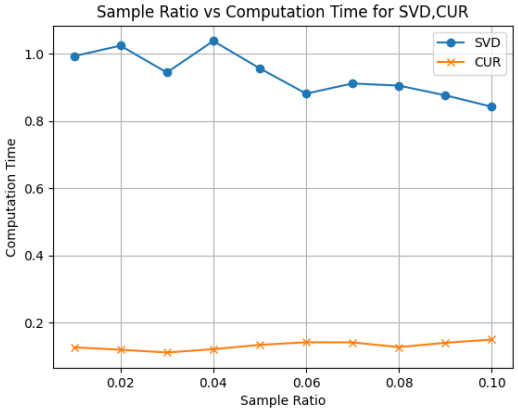
1)CUR decomposition of a real valued tensor always exists.(Because CUR decomposition of a matrix is guaranteed, and with the isomorphism operation fft, we are having a isomorphic relation between matrix space and tensor space)

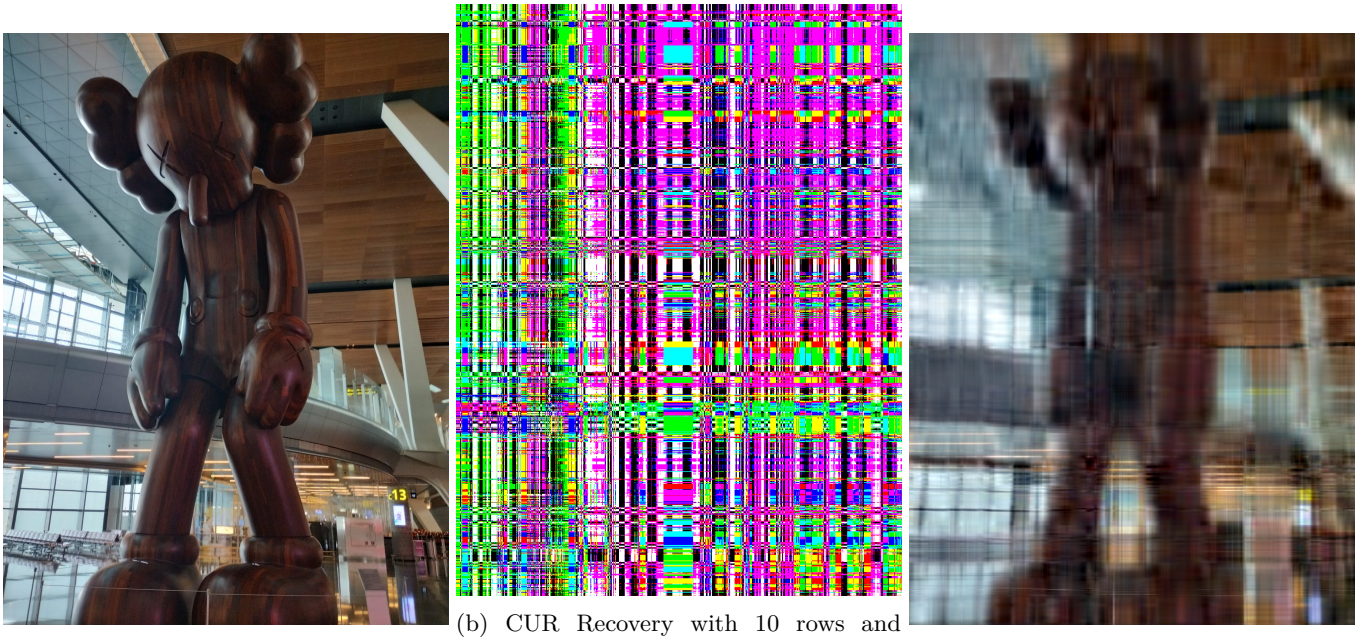
2)If $\mathscr{B}$ is the approximation of tensor $\mathscr{A}$, then the used error formulation for the approximation was: $\frac{||\mathscr{A}-\mathscr{B}||_F}{||\mathscr{A}||_F}$

3)Let the tensor $\mathscr{A} \in \mathrm{F}^{n1\times n2\times n3}$ have multi-rank($\mathscr{A}$)= R[Let $\hat{\mathscr{A}}$=fft($\mathscr{A}$), and U^i be its i the frontal slice.multi rank of tensor $\mathscr{A}$ is an array of length n3, where ith entry is rank of U^i.And tubal rank of A is maximum entry in the array of multi rank] , and tubalrank=r.Let I $\subset$ {1:n1} and J $\subset$ {1:n2} satisfying $|\mathrm{I}| \geq r, |J| \geq r$. Let tensors $\mathscr{C} = \mathscr{A}$(:,J,:),$\mathscr{R} = \mathscr{A}$(I,:,:),$\mathscr{U} = \mathscr{A}$(I,J,:),if multirank($\mathscr{U}$)=multirank($\mathscr{A}$),then $\mathscr{A} = \mathscr{C}\mathscr{U}^{\dagger}\mathscr{R}$[$\mathscr{U}^{\dagger}$, being pseudoinverse of $\mathscr{U}$]

7.2 Observation

» For a certain Tensor SVD took time of 2.074059 sec, where CUR took time of 1.144329 sec, which indeed tells that CUR is comparatively less computationally expensive.
» We created a 1000*1000*3 tensor with each frontal slice is diagonal as got the following:[svd took 10X more time for such small tensor size!!]

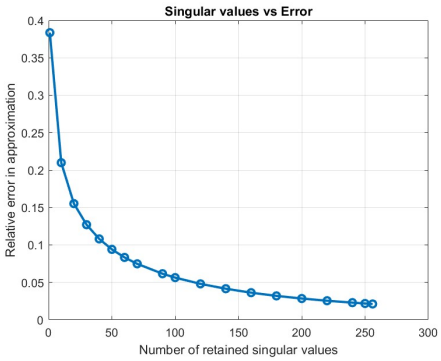




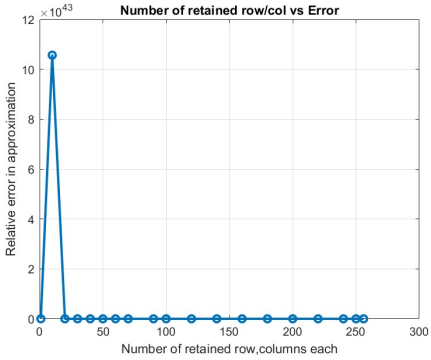
(a) Actual Image (b) CUR Recovery with 10 rows and columns (c) SVD Recovery with 10 singular values

Figure 3: Performance of SVD,CUR for a specific instance

(a) Actual Image



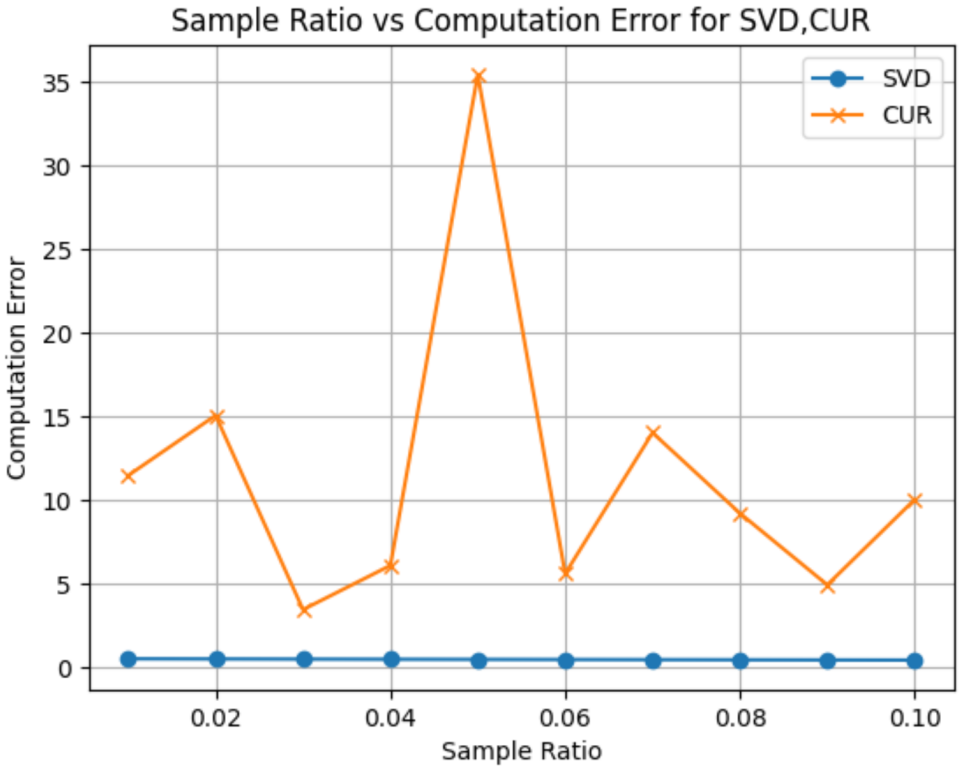
(b) Error in SVD based recovery



(c) Error in CUR based recovery.Clearly the tubalrank of the image is <20

Figure 4: SVD vs CUR computation error

For a fixed tensor A(saved for regeneration purpose) of size 1000X1000X3 with each frontal slice being of full rank , we got the error as:



8 Conclusion

The missing data recovery problem of a 3-way tensor was defined as a tensor completion problem in this study.Our suggestion was a new tensor completion approach that could efficiently retrieve the absent data from small samples.It's a novel approach of turning the matrix-CUR into a three-way tensor. This means that one may now express a 3-way tensor as a product of other 3-way tensors.Using our own created tensors as test subjects, we discovered that although CUR is time-efficient, it is not always

possible to find a smaller error (when rank of $\mathscr{W}$, which is the intersection of $\mathscr{C}, \mathscr{R}$ is less than tubalrank of tensor).
TOWARDS END-TERM I WILL EXPLORE DIFFERENT SAMPLING TECHNIQUES INCLUDES TERM SAMPLING /REJECTION SAMPLING FOR ROW AND COLUMN WHICH PROVIDES THEORETICAL GURANTEE OF MORE EFFICIENT APPROXIMATION. THIS SURELY WILL BE A NICE RESEARCH WORK.

9 References

1[13])J. A. Bengua, H. N. Phiem, H. D. Tuan, and M. N. Do, ‘‘Efficient tensor completion for color image and video recovery: Low-rank tensor train,’’ IEEE Trans. Image Process., vol. 26, no. 5, pp. 2466–2479, May 2017.
2[14])] N. Li and B. Li, ‘‘Tensor completion for on-board compression of hyperspectral images,’’ in Proc. IEEE Int. Conf. Image Process., Sep. 2010, pp. 517–520.
3[15]) A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, ‘‘Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering,’’ in Proc. ACM Conf. Recommender Syst. (RecSys), Barcelona, Spain, Sep. 2010, pp. 79–86.
4[16])W. W. Sun, J. Lu, H. Liu, and G. Cheng, ‘‘Provable sparse tensor decomposition,’’ J. Roy. Stat. Soc., vol. 79, no. 3, pp. 899–916, 2016.
5[17])M. Tang, G. Ding, Q. Wu, Z. Xue, and T. A. Tsiftsis, ‘‘A joint tensor completion and prediction scheme for multi-dimensional spectrum map construction,’’ IEEE Access, vol. 4, no. 99, pp. 8044–8052, 2016.
6[18])Y. Zhang, C. D. Silva, R. Kumar, and F. J. Herrmann, ‘‘Massive 3D seismic data compression and inversion with hierarchical tucker,’’ in Proc. SEG, 2017, pp. 1347–1352.
7[19])J. Liu, P. Musialski, P. Wonka, and J. Ye, ‘‘Tensor completion for estimating missing values in visual data,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 208–220, Jan. 2013.
8[20])B. Ran, H. Tan, Y. Wu, and P. J. Jin, ‘‘Tensor based missing traffic data completion with spatial–temporal correlation,’’ Phys. A, Stat. Mech. Appl., vol. 446, pp. 54–63, Mar. 2016.
9[21])E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, ‘‘Scalable tensor factorizations for incomplete data,’’ Chemometrics Intell. Lab. Syst., vol. 106, no. 1, pp. 41–56, 2011.
10[22]) H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, ‘‘A tensorbased method for missing traffic data completion,’’ Transp. Res. C, Emerg. Technol., vol. 28, pp. 15–27, Mar. 2013.
11[23])J. Håstad, ‘‘Tensor rank is NP-complete,’’ J. Algorithms, vol. 11, no. 4, pp. 644–654, 1990.
12[25])Z. Zhang and S. Aeron, ‘‘Exact tensor completion using t-SVD,’’ IEEE Trans. Signal Process., vol. 65, no. 6, pp. 1511–1526, Mar. 2017.
13) Cai, H., Hamm, K., Huang, L., Li, J., Wang, T. (2021). Rapid robust principal component analysis: CUR accelerated inexact low rank estimation. IEEE Signal Process. Lett. 28:116–120. DOI: 10.1109/LSP.2020.3044130.
14) Hamm, K., Huang, L. (2020). Perspectives on CUR decompositions. Appl. Comput. Harmon. Anal. 48(3):1088–1099. DOI: 10.1016/j.acha.2019.08.006.
15)CUR Decomposition
16)Advanced CUR Decomposition
17[30])P. Jain and S. Oh, ‘‘Provable tensor factorization with missing data,’’ in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 1431–1439.
18[24])N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover, ‘‘Facial recognition using tensor-tensor decompositions,’’ SIAM J. Imag. Sci., vol. 6, no. 3, pp. 437–463, 2013.