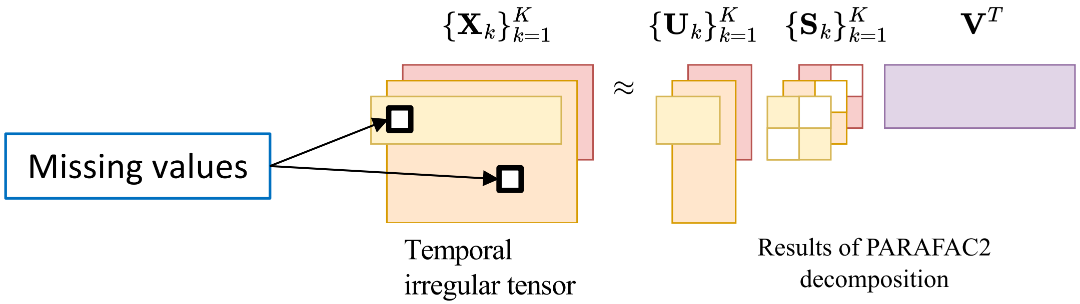
Accurate PARAFAC2 Decomposition for Temporal Irregular Tensors with Missing Values

BigData 2022

Jun-Gi Jang, Jeongyoung Lee,
Jiwon Park, and U Kang
Data Mining Lab
Dept. of CSE
Seoul National University

Overview

- **Q.** Given a temporal irregular tensor with missing values, how can we perform accurate decomposition for the tensor?
  - Temporal irregular tensor: a collection of matrices whose columns have the same size and rows corresponding to the time dimension have different sizes from each other
- **A.** **ATOM,** an accurate decomposition method that carefully handles missing values in a temporal irregular tensor

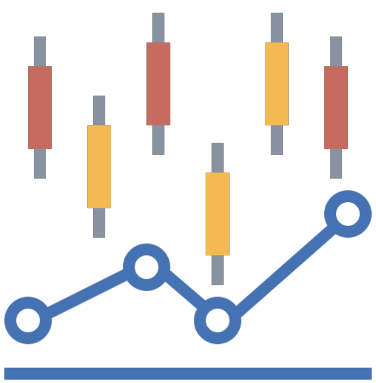


Outline

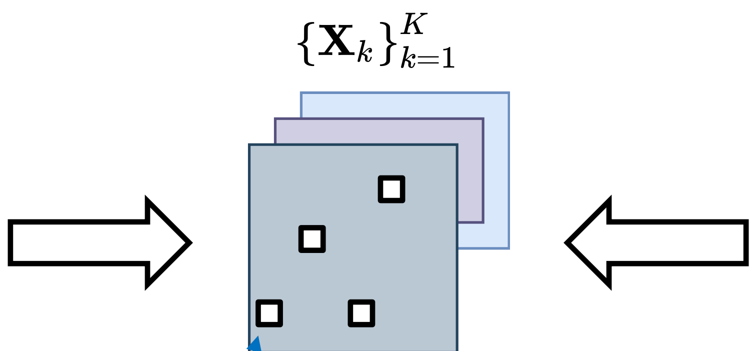
- **Introduction**
- Proposed Method
- Experiments
- Conclusion

Temporal Irregular Tensors

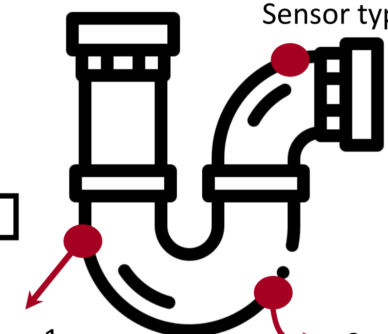
- There are many **temporal irregular tensors** in nature
  - A temporal irregular tensor is a collection of matrices whose columns have the same size and rows have different sizes from each other
    - Rows correspond to the time dimension



Stocks
Index: (time, feature, stock)
Value: measurement



A temporal irregular tensor



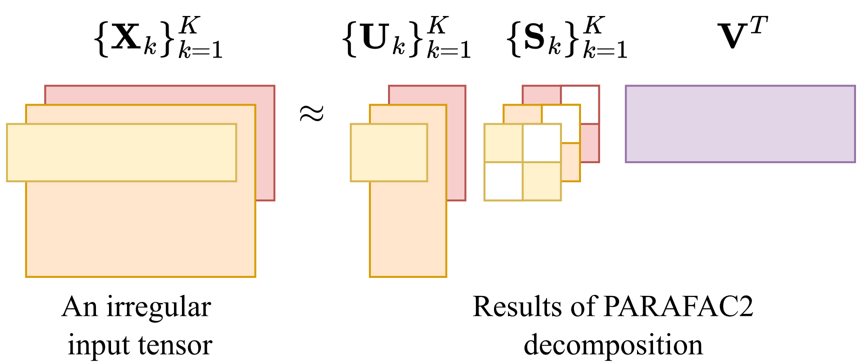
Sensor
Index: (time, location, sensor)
Value: measurement

Missing values can exist in a temporal irregular tensor

PARAFAC2 Decomposition

- **PARAFAC2 Decomposition**
  - A fundamental tool to analyze irregular tensors
- **Given** an irregular tensor $\{\mathbf{X}_k\}_{k=1}^{K}$, rank R
  - Slice matrix $\mathbf{X}_k \in \mathbb{R}^{I_k \times J}$
- **Obtain** obtain factor matrices $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}$, $\mathbf{S}_k \in \mathbb{R}^{R \times R}$, $\mathbf{V} \in \mathbb{R}^{J \times R}$ for k=1,…,K
- **Objective function**

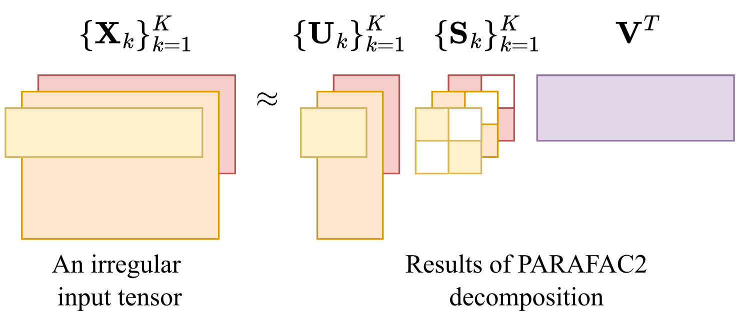
$$\min_{\{\mathbf{U}_k\},\{\mathbf{S}_k\},\mathbf{V}} \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{U}_k\mathbf{S}_k\mathbf{V}^T\|_F^2$$



- **Several applications for PARAFAC2 decomposition**
  - Dimensionality reduction, anomaly detection, trend analysis, and phenotype discovery

Alternating Least Square

- **Many methods use Alternating Least Square (ALS) to obtain factor matrices of PARAFAC2 decomposition**
  - ALS *Iteratively* updates a target factor matrix while fixing all other factor matrices
    - For example, update the matrix $\mathbf{V}$ while fixing all other factor matrices $\mathbf{U}_k$ and $\mathbf{S}_k$ for all k



Main Goal

- **Our goal is to obtain** accurate results of PARAFAC2 decomposition when a temporal irregular tensor with missing values is given
  - **Given**
    - A temporal irregular tensor $\{\mathbf{X}_k\}_{k=1}^{K}$,
      - Rows of slice matrices $\mathbf{X}_k \in \mathbb{R}^{I_k \times J}$ correspond to the time dimension
      - Missing values exist in $\mathbf{X}_k$
    - Target rank R
  - **Results**
    - $\mathbf{U}_k \in \mathbb{R}^{I_k \times R}, \mathbf{S}_k \in \mathbb{R}^{R \times R}, \mathbf{V} \in \mathbb{R}^{J \times R}$ for k=1,…,K

Limitation of Previous Works

- **Limitations of previous PARAFAC2 decomposition methods**
  - They handle missing values as zero
    - Loss function with a matrix form implies treating missing values as 0 $\Rightarrow$ generate inaccurate factor matrices
    - Need to fully exclude missing values in obtaining factor matrices
  - They fail to effectively capture temporal patterns inherent in many temporal irregular tensors

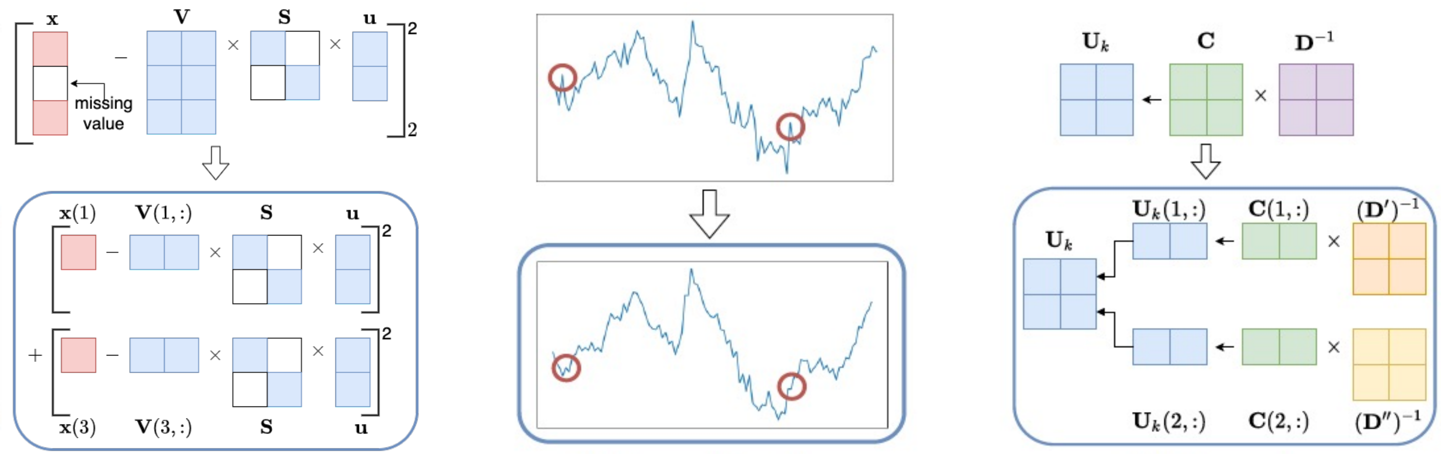
How can we address the above limitations to improve the accuracy of PARAFAC2 decomposition?

Outline

- Introduction
- **Proposed Method**
- Experiments
- Conclusion

Proposed Method

- We propose **ATOM** (**A**ccurate PARAFAC2 decomposition method for **T**emporal irregular tens**O**rs with **M**issing values)



(a) Loss function with a scalar form (b) Temporal Smoothness (c) Update row by row

  - **(Idea 1)** Reformulating the loss function of PARAFAC2 decomposition as a scalar form
  - **(Idea 2)** Exploiting Temporal regularization that makes nearby temporal factor vectors similar to each other
  - **(Idea 3)** Proposing Row-wise update procedure for factor matrices

Loss function with a Scalar Form

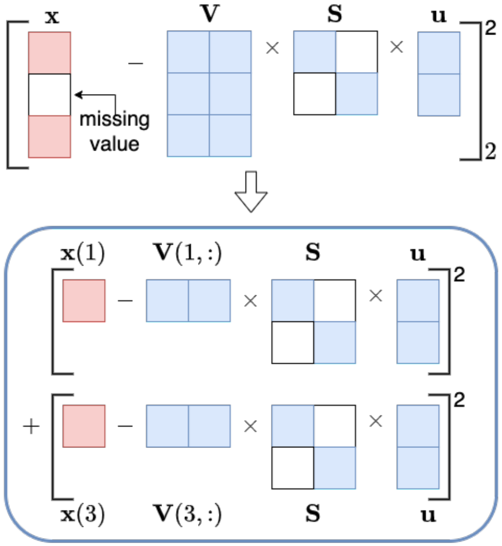
- **Reformulate the loss function of a matrix form into a scalar form**

$$\sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{U}_k \mathbf{S}_k \mathbf{V}^T\|_F^2$$

$\mathbf{U}_k \in \mathbb{R}^{I_k \times R}, \mathbf{S}_k \in \mathbb{R}^{R \times R}, \mathbf{V} \in \mathbb{R}^{J \times R}$: factor matrices

Ω_k: the observable entries of the kth slice matrix

$$\sum_{k=1}^{K} \sum_{(i,j)\in\Omega_k} \left(\mathbf{X}_k(i,j) - \mathbf{U}_k(i,:)\mathbf{S}_k\left(\mathbf{V}(j,:)\right)^T\right)^2$$

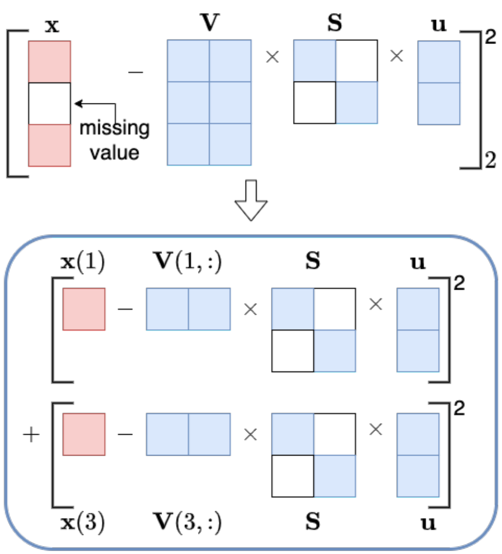


Example

- **This reformulation fully excludes missing values so that it lays the groundwork for obtaining accurate factor matrices**
  - Without reformulation, factor matrices are learned to set missing values to zero

Example

- Given a vector $\mathbf{x} \in \mathbb{R}^3$ whose second element has a missing value, find factor matrices (or vectors) by alternating optimization



Loss function

Update $\mathbf{u}$ derived by $\frac{\partial L}{\partial \mathbf{u}} = 0$ while fixing $\mathbf{V}$ and $\mathbf{S}$

$$\|\mathbf{x} - \mathbf{VSu}\|_F^2 \Rightarrow \mathbf{u} \leftarrow (\mathbf{SV}^T\mathbf{VS})^{-1}\mathbf{SV}^T\mathbf{x}$$

$$(\mathbf{x}(1) - \mathbf{V}(1,:)\mathbf{Su})^2 + (\mathbf{x}(3) - \mathbf{V}(3,:)\mathbf{Su})^2 \Rightarrow \mathbf{u} \leftarrow \left(\mathbf{S}\left(\mathbf{V}(1,:)^T\mathbf{V}(1,:) + \mathbf{V}(3,:)^T\mathbf{V}(3,:)\right)\mathbf{S}\right)^{-1} \times \mathbf{S}(\mathbf{V}(1,:)^T + \mathbf{V}(3,:)^T)\mathbf{x}$$

- The main difference is the inside of the inverse term (blue-colored text)
- Including $\mathbf{V}(2,:)^T\mathbf{V}(2,:)$ degrades the performance when updating factor matrices

Temporal Regularization

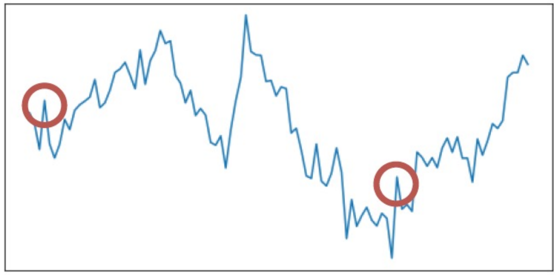
- Use **temporal smoothness regularization** that captures temporal patterns which change gradually over time
  - Make near temporal factor vectors similar to each other
- **Temporal smoothness regularization**

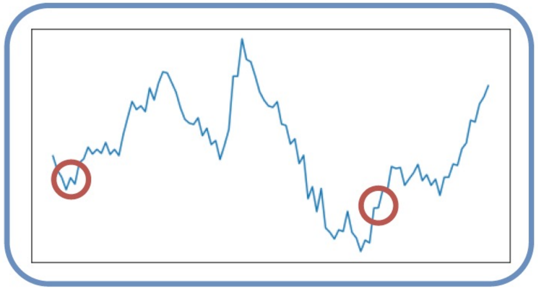
$$\lambda_s \sum_{i=1}^{I_k-1} \|\mathbf{U}_k(i,:) - \mathbf{U}_k(i+1,:)\|_2^2$$

$\mathbf{U}_k$: kth temporal factor matrix
I_k: the row size of $\mathbf{X}_k$ (or $\mathbf{U}_k$)
λ_s: hyperparameter for temporal smoothness

- Example for $\mathbf{U}_k(:,j)$ with temporal smoothness

Without temporal smoothness





With temporal smoothness

Additional Regularization

- Use effective regularizations that mix well with our reformulated loss function
  - **Uniqueness**

$$\lambda_u \|\mathbf{U}_k - \mathbf{Q}_k \mathbf{H}\|_F^2$$

  - **L2 Regularization**

$$\lambda_l \|\mathbf{V}\|_F^2 \text{ and } \lambda_l \|\mathbf{S}_k\|_F^2$$

$\mathbf{Q}_k \in \mathbb{R}^{I_k \times R}$: kth column orthogonal matrix
$\mathbf{H} \in \mathbb{R}^{R \times R}$: common factor matrix over slice matrices
λ_u and λ_l: hyperparameter for uniqueness and L2 regularization, respectively

- These regularizations make our method avoid overfitting and prevent the computation of the inverse of a singular matrix in our update procedure

Proposed Loss Function

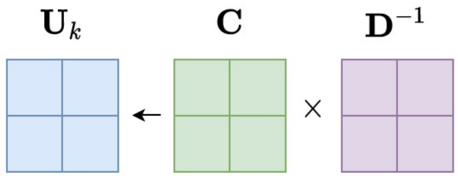
- Propose the following loss function
  - Fully excludes missing values and includes effective regularizations

$$\sum_{k=1}^{K}\left(\sum_{(i,j)\in\Omega_k}\left(\mathbf{X}_k(i,j)-\mathbf{U}_k(i,:)\mathbf{S}_k\left(\mathbf{V}(j,:)\right)^T\right)^2 + \lambda_s\sum_{i=1}^{I_k-1}\|\mathbf{U}_k(i,:)-\mathbf{U}_k(i+1,:)\|_2^2 + \lambda_u\|\mathbf{U}_k-\mathbf{Q}_k\mathbf{H}\|_F^2+\lambda_l\|\mathbf{S}_k\|_F^2\right) + \lambda_l\|\mathbf{V}\|_F^2$$

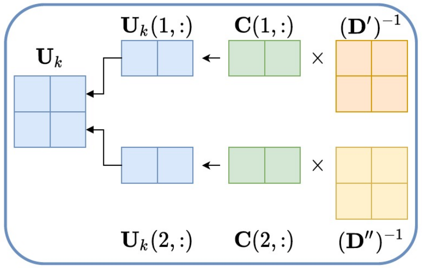
(green)	PARAFAC2 Decomposition Loss
(blue)	Temporal Regularization
(yellow)	Additional Regularization

Row-wise Update Procedure

- **Update factor matrices row by row based on alternating optimization**
  - We need to design a new update procedure appropriate for the reformulated loss function
  - Rows of a slice matrix have different sparsity patterns and all the rows of a factor matrix are independent
    - Sparsity patterns: missing values in a tensor are not aligned but spread out all over the tensor
- **Example**
  - When updating $\mathbf{U}_k(i,:)$ in the reformulated function, $\mathbf{U}_k(i,j)$ for all j is updated by using the same inverse term



Conventional update procedure in the loss function of the matrix form



Row wise update procedure in the reformulated loss function

Row-wise Update Procedure

- **Procedure for updating factor matrices row by row**
  - Fix all the factor matrices except for the target matrix $\mathbf{U}_k$
  - For all k, update each row $\mathbf{U}_k(i,:)$ of the target matrix by setting $\frac{\partial L}{\partial \mathbf{U}_k(i,:)}$ to zero

$$\mathbf{U}_k(i,:) \leftarrow \left(\mathbf{X}_k(i,:)\mathbf{V}\mathbf{S}_k + \lambda_u \mathbf{Q}_k(i,:)\mathbf{H} + \lambda_s\left(\mathbf{U}_k(i-1,:) + \mathbf{U}_k(i+1,:)\right)\right) \times \left((\lambda_u + 2\lambda_s)\mathbf{I} + \sum_{(i,j)\in\Omega_{k,i}} \mathbf{S}_k\mathbf{V}(j,:)^T\mathbf{V}(j,:)\mathbf{S}_k\right)^{-1}$$

  - Update other factor matrices $\mathbf{S}_k$ and $\mathbf{V}$ in the same manner
    - 1) Fix other factor matrices except for a target matrix
    - 2) For all rows,
      - 2-1) Compute the derivation with respect to the row of the target matrix
      - 2-2) Update the row of the target factor matrix
  - Update $\mathbf{Q}_k$ and $\mathbf{H}$

$\Omega_{k,i}$: the observable entries of the ith row of the kth slice matrix
$\mathbf{I}$: identity matrix

From PARAFAC2 decomposition loss

From temporal regularization

From uniqueness regularization

Outline

- Introduction
- Proposed Method
- **Experiments**
- Conclusion

Experimental Questions

- **Q1. Performance.** How accurately does **ATOM** predict missing values for real-world irregular tensors?
- **Q2. Ablation Study.** Do the ideas of temporal smoothness and handling of missing values contribute to the performance of **ATOM**?
- **Q3. Hyperparameter Sensitivity.** How much do the regularization hyperparameters affect the performance of **ATOM**?

Dataset

- **Dataset**

	Dataset	Max Dim. I_k	Dim. J	Dim. K	# of nnz	Summary
Dense	US Stoctk[1] [23]	7,883	88	4,742	1.3B	Stock
	Korea Stock[2] [14]	5,270	88	3,619	859M	Stock
	China Stock	2,431	88	219	46M	Stock
	Japan Stock	2,204	88	215	41M	Stock
	VicRoads[3] [29]	1,084	96	2,033	197M	Traffic
	PEMS-SF[4]	440	144	963	60M	Traffic
Sparse	ML-100k[5]	40	1,682	344	50K	Movie rating

- Each slice matrix of a temporal irregular tensor has different I_k
- J is the column size of slice matrices
- K is the number of slice matrices in a temporal irregular tensor

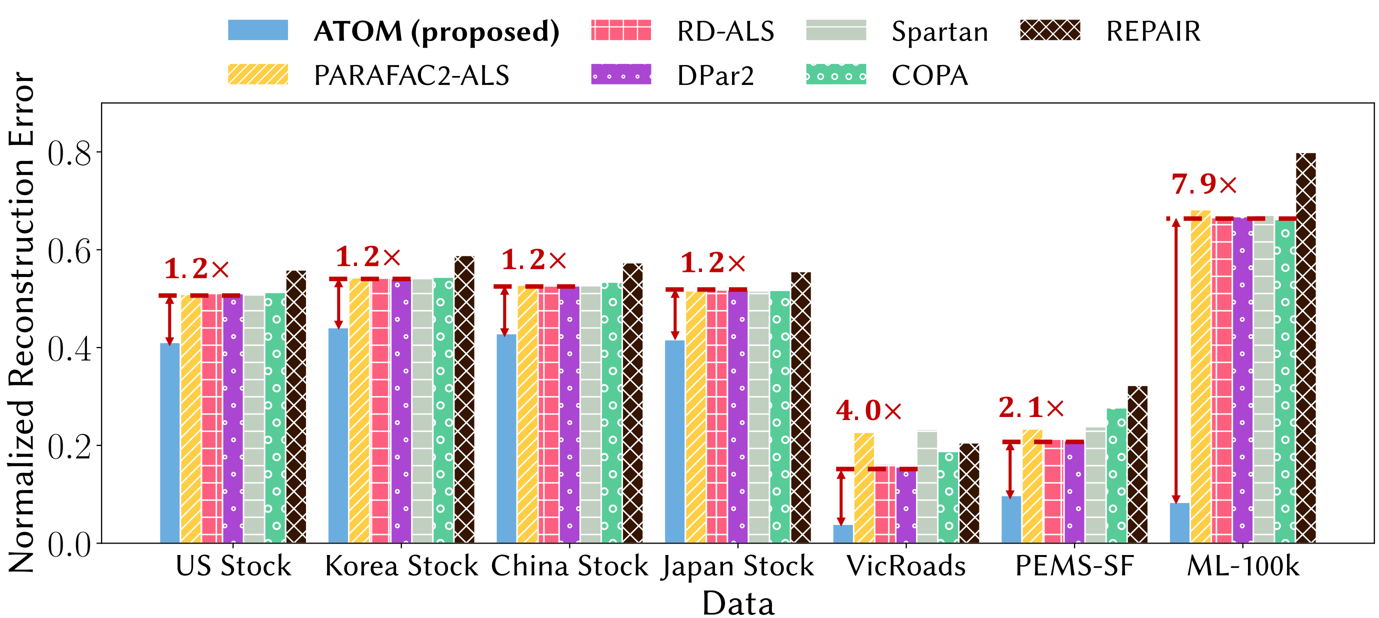
Experimental Setting

- **Competitors**
  - 6 existing PARAFAC2 decomposition methods for irregular tensors
    - **PARAFAC2-ALS, RD-ALS, DPar2, SPARTAN, COPA, and REPAIR**
- **Task**
  - To evaluate the performance of ATOM, we perform a missing value prediction task
    - Randomly split observed entries of a given data into training and test entries with the ratio: $(70\%, 30\%)$ in this presentation
    - learn factor matrices using training entries of irregular tensors, and predict values of test entries using the learned factor matrices
- **Metric**
  - **Normalized Reconstruction Error**

$$\left(\frac{\sum_{k=1}^{K}\sum_{(i,j)\in\Omega_{k,test}}\left(\mathbf{X}_k(i,j)-\hat{\mathbf{X}}_k(i,j)\right)^2}{\sum_{k=1}^{K}\sum_{(i,j)\in\Omega_{k,test}}\left(\mathbf{X}_k(i,j)\right)^2}\right)$$

Q1. Performance Trade-off

- **ATOM** **outperforms** the competitors, giving up to 7.9× **lower error** than competitors



Q2. Ablation Study

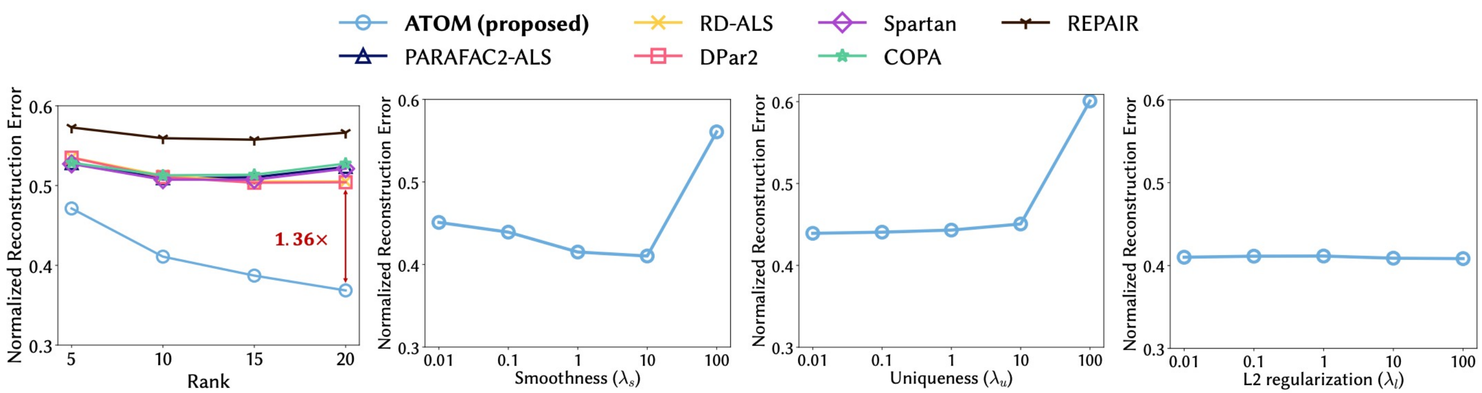
- **Ablation study**
  - -S and –M indicate the elimination of temporal smoothness regularization and no consideration of missing values, respectively

Missing Value Prediction - Missing Ratio: 30%							
Model	US Stock	Korea Stock	China Stock	Japan Stock	VicRoads	PEMS-SF	ML-100k
ATOM- M - S	0.5046 ± 0.0017	0.5341 ± 0.0012	0.5206 ± 0.0012	0.5141 ± 0.0006	0.2171 ± 0.0007	0.2228 ± 0.0006	0.6757 ± 0.0011
ATOM- M	0.4956 ± 0.0008	0.5252 ± 0.0014	0.5136 ± 0.0019	0.5066 ± 0.0009	0.2157 ± 0.0012	0.2213 ± 0.0005	0.6798 ± 0.0021
ATOM- S	0.4533 ± 0.0042	0.4822 ± 0.0040	0.4571 ± 0.0013	0.4510 ± 0.0010	0.0409 ± 0.0001	0.0986 ± 0.0001	0.1268 ± 0.0013
ATOM	$\mathbf{0.4094 \pm 0.0027}$	$\mathbf{0.4399 \pm 0.0017}$	$\mathbf{0.4269 \pm 0.0021}$	$\mathbf{0.4210 \pm 0.0008}$	$\mathbf{0.0391 \pm 0.0001}$	$\mathbf{0.0976 \pm 0.0001}$	$\mathbf{0.1005 \pm 0.0016}$

- Compared to **ATOM**-M, our update procedure with the loss function of the scalar form effectively reduces prediction errors
- Atom-S is not as good as **ATOM**, but it has better performance than **ATOM**-M-S and existing methods

Q3. Hyperparameter Sensitivity

- Evaluate the hyperparameter sensitivity by measuring prediction errors while varying hyperparameters
  - target rank, temporal smoothness, uniqueness, and L2 regularization



(a) Target rank for US Stock (b) Smoothness for US Stock (c) Uniqueness for US Stock (d) L2 regularization for US Stock

- In contrast to existing methods, **ATOM** decreases the errors as target rank increases
- For **ATOM,** we need to select smoothness and uniqueness regularizations hyperparameters λ_s and λ_u that do not degrade the prediction performance

Outline

- Introduction
- Proposed Method
- Experiments
- **Conclusion**

Conclusion

- We propose **ATOM** to obtain accurate factor matrices from a temporal irregular tensor with missing values
- Main ideas
  - Reformulate the loss function that fully excludes missing values
  - Add effective regularizations including temporal smoothness regularization that captures temporal patterns
  - Propose a row-wise update procedure appropriate for the reformulated function
- We experimentally show that **ATOM** achieves the lowest errors for missing value prediction

Thank you !

https://jungijang.github.io

Appendix
Row-wise Update Procedure

- **Procedure for updating factor matrices row by row**
  - Fix all the factor matrices except for the target matrix $\mathbf{W}(k,:)$ where $\mathbf{W}(k,:)$ corresponds to the diagonal elements of $\mathbf{S}_k$
  - Update each row $\mathbf{W}(k,:)$ of the target matrix by setting $\frac{\partial L}{\partial \mathbf{W}(k,:)}$ to zero

$$\mathbf{W}(k,:) \leftarrow \left(vec(\mathbf{X}_k)^T(\mathbf{V} \odot \mathbf{U}_k)\right) \times \left(\lambda_l \mathbf{I} + \sum_{(i,j)\in\Omega_k} \mathbf{V}(j,:)^T\mathbf{V}(j,:) * \mathbf{U}_k(i,:)^T\mathbf{U}_k(i,:)\right)^{-1}$$

$\mathbf{W} \in \mathbb{R}^{K\times R}$: factor matrix consisting of the diagonal elements of $\mathbf{S}_k$ for all k
Ω_k: the observable entries of the kth slice matrix
$\odot$: Khatri-rao product
$*$: element-wise product
$vec(\cdot)$: vectorization of a matrix

From PARAFAC2 decomposition loss

From L2 regularization

Appendix
Row-wise Update Procedure

- **Procedure for updating factor matrices row by row**
  - Fix all the factor matrices except for the target matrix $\mathbf{V}$
  - Update each row $\mathbf{V}(j,:)$ of the target matrix by setting $\frac{\partial L}{\partial \mathbf{V}(j,:)}$ to zero

$$\mathbf{V}(j,:) \leftarrow \left(\sum_{k=1}^{K} \mathbf{X}_k(:,j)^T \mathbf{U}_k \mathbf{S}_k\right) \times \left(\lambda_l \mathbf{I} + \sum_{(\boldsymbol{i},\boldsymbol{j}) \in \Omega_{k,j}} \mathbf{S}_k \mathbf{U}_k(i,:)^T \mathbf{U}_k(i,:) \mathbf{S}_k\right)^{-1}$$

$\Omega_{k,j}$: the observable entries of the jth column of the kth slice matrix

From PARAFAC2 decomposition loss

From L2 regularization

Appendix
Performance

- Randomly split observed entries of a given data into training and test entries with the ratios: $(90\%, 10\%)$, $(80\%, 20\%)$
- **ATOM** **outperforms** the competitors for all the test ratios

Missing Value Prediction - Test Ratio: 10%

Model	US Stock	Korea Stock	China Stock	Japan Stock	VicRoads	PEMS-SF	ML-100k
PARAFAC2-ALS	0.4148 ± 0.0018	0.4554 ± 0.0002	0.2620 ± 0.0016	0.2695 ± 0.0001	0.0958 ± 0.0011	0.1269 ± 0.0003	0.6124 ± 0.0037
RD-ALS [21]	0.4154 ± 0.0002	0.4508 ± 0.0001	0.2584 ± 0.0005	0.2682 ± 0.0001	0.0666 ± 0.0004	0.1204 ± 0.0010	0.6028 ± 0.0010
DPar2 [23]	0.4139 ± 0.0002	0.4487 ± 0.0004	0.2588 ± 0.0006	0.2654 ± 0.0003	0.0637 ± 0.0004	0.1174 ± 0.0004	0.6042 ± 0.0016
SPartan [16]	0.4117 ± 0.0005	0.4500 ± 0.0010	0.2600 ± 0.0021	0.2670 ± 0.0000	0.1049 ± 0.0020	0.1307 ± 0.0014	0.6080 ± 0.0022
COPA [17]	0.4178 ± 0.0018	0.4583 ± 0.0000	0.2738 ± 0.0015	0.2713 ± 0.0000	0.1123 ± 0.0003	0.2192 ± 0.0008	0.6023 ± 0.0019
REPAIR [18]	0.4547 ± 0.0052	0.4866 ± 0.0000	0.3136 ± 0.0044	0.3039 ± 0.0000	0.1225 ± 0.0007	0.2393 ± 0.0013	0.7562 ± 0.0010
ATOM	$\mathbf{0.3677 \pm 0.0013}$	$\mathbf{0.3994 \pm 0.0003}$	$\mathbf{0.2086 \pm 0.0023}$	$\mathbf{0.2196 \pm 0.0007}$	$\mathbf{0.0365 \pm 0.0001}$	$\mathbf{0.0854 \pm 0.0002}$	$\mathbf{0.0778 \pm 0.0014}$

Missing Value Prediction - Test Ratio: 20%

Model	US Stock	Korea Stock	China Stock	Japan Stock	VicRoads	PEMS-SF	ML-100k
PARAFAC2-ALS	0.4618 ± 0.0006	0.4894 ± 0.0011	0.3072 ± 0.0027	0.3154 ± 0.0008	0.1578 ± 0.0013	0.1719 ± 0.0006	0.6426 ± 0.0032
RD-ALS [21]	0.4630 ± 0.0005	0.4855 ± 0.0002	0.3044 ± 0.0003	0.3153 ± 0.0002	0.1041 ± 0.0006	0.1575 ± 0.0012	0.6312 ± 0.0013
DPar2 [23]	0.4610 ± 0.0002	0.4831 ± 0.0002	0.3027 ± 0.0005	0.3130 ± 0.0004	0.0999 ± 0.0003	0.1552 ± 0.0001	0.6305 ± 0.0011
SPartan [16]	0.4598 ± 0.0007	0.4867 ± 0.0009	0.3035 ± 0.0005	0.3142 ± 0.0005	0.1662 ± 0.0024	0.1757 ± 0.0018	0.6428 ± 0.0065
COPA [17]	0.4664 ± 0.0027	0.4922 ± 0.0007	0.3169 ± 0.0021	0.3164 ± 0.0000	0.1410 ± 0.0007	0.2383 ± 0.0008	0.6283 ± 0.0040
REPAIR [18]	0.5109 ± 0.0037	0.5322 ± 0.0025	0.3659 ± 0.0053	0.3535 ± 0.0000	0.1569 ± 0.0010	0.2732 ± 0.0015	0.7764 ± 0.0028
ATOM	$\mathbf{0.3942 \pm 0.0022}$	$\mathbf{0.4147 \pm 0.0028}$	$\mathbf{0.2200 \pm 0.0026}$	$\mathbf{0.2357 \pm 0.0014}$	$\mathbf{0.0374 \pm 0.0001}$	$\mathbf{0.0912 \pm 0.0001}$	$\mathbf{0.0798 \pm 0.0022}$