Video Imputation and Prediction Models in Context of Space Weather Monitoring

Hu Sun

Zhijun Hua, Jiaen Ren, Shasha Zou, Yuekai Sun, Yang Chen

1Department of Statistics, University of Michigan, Ann Arbor
2Department of Climate and Space Sciences and Engineering, University of Michigan, Ann Arbor

April 27, 2021

1Click here for the preprint version of our paper.
Overview

1 Background
- Total Electron Content (TEC) map
- Matrix Completion Problem
- Spherical Harmonics

2 Method
- Conceptual Framework
- Algorithm
- Theoretical Guarantees

3 Empirical Analysis
- Simulation Study
- Imputing TEC map

4 Conclusion & Future Plan
Ionosphere Total Electron Content (TEC) is defined as the total number of electrons in the path between satellite\(^2\) radio transmitter and ground-based receiver. (1 TEC unit (TECU) = \(10^{16}\) electrons/m\(^2\))
Ionosphere Total Electron Content (TEC) is defined as the total number of electrons in the path between satellite\(^2\) radio transmitter and ground-based receiver. (1 TEC unit (TECU) = \(10^{16}\) electrons/m\(^2\))

TEC affects the propagation of radio waves, leading up to 10s meters positioning error in the GNSS Positioning, Navigation and Timing (PNT) services. Better knowledge of TEC map will make PNT services more accurate.
Figure: TEC map from the Madrigal Database (A) without median filter, (B) with a $3^\circ \times 3^\circ$ median filter and (C) TEC map from the International GNSS Service (IGS).
The goal of the project is to reasonably “fill in“ the missing values within TEC maps. Pan et al. (2020) used DCGAN-based models for TEC map completion, relying on IGS TEC maps as either reference or training data. But overall the IGS data is of low-resolution, and we want to preserve the high-resolution nature of the TEC map.
To impute the TEC maps, we adopt classical statistics techniques called **matrix completion**.
To impute the TEC maps, we adopt classical statistics techniques called **matrix completion**.

Matrix completion is a commonly used method in designing recommender systems. With a user-item rating matrix, for example, matrix completion can infer the potential rating a user would give to an item he/she has never consumed.
Matrix Completion Problem

Rank-Restricted SVD (Mazumder et al., 2010)

\[
\min_{M_t} \quad H(M_t) := \frac{1}{2} \| P_{\Omega_t}(X_t - M_t) \|_F^2 + \lambda \| M_t \|_*
\]

where \( \| M_t \|_* \) is the nuclear norm, i.e. sum of all singular values, of \( M_t \).

Following the notation in (Candès and Tao, 2010), the projection \( P_{\Omega_t}(X_t) \) is an \( m \times n \) matrix keeping all observed entries of \( X_t \) and replacing all missing entries with 0.

- It is a well-known result that the solution is \( M_t = U_r S_\lambda(D_r) V_r^T \), where \( r = \min(m, n) \) and \( U_r, D_r, V_r \) are the components of rank-\( r \) SVD of \( X_t \). \( S_\lambda(D_r) = \text{diag} [(\sigma_1 - \lambda)_+, (\sigma_2 - \lambda)_+, \ldots, (\sigma_r - \lambda)_+] \) is the soft-thresholding operator.
Matrix Completion with Factorization

Maximum-margin Matrix Factorization (MMMF) (Srebro et al., 2005)

\[
\min_{A_t, B_t} F(A_t, B_t) := \frac{1}{2} \left\| \mathbf{P}_{\Omega_t} (X_t - A_t B_t^T) \right\|_F^2 + \frac{\lambda_1}{2} \left( \|A_t\|_F^2 + \|B_t\|_F^2 \right) \tag{2}
\]

with solution \( \hat{A}_t = U_r S_\lambda(D_r)^{\frac{1}{2}} \) and \( \hat{B}_t = V_r S_\lambda(D_r)^{\frac{1}{2}} \)
Matrix Completion with Factorization

Maximum-margin Matrix Factorization (MMMF) (Srebro et al., 2005)

\[
\min_{A_t, B_t} F(A_t, B_t) := \frac{1}{2} \|P_{\Omega_t}(X_t - A_t B_t^T)\|_F^2 + \frac{\lambda_1}{2} (\|A_t\|_F^2 + \|B_t\|_F^2) \quad (2)
\]

with solution \( \hat{A}_t = U_r S_\lambda (D_r)^{1/2} \) and \( \hat{B}_t = V_r S_\lambda (D_r)^{1/2} \)

Such a factorization setup has direct interpretations in the factor matrices \( A, B \). For example, the original map is of size \( m \times n \), where \( m, n \) corresponds to latitude and longitude. Then each row in \( A \) and \( B \) can be considered as the “latent feature“ of each latitude and longitude. The final imputation at any location is the inner product of the feature vectors of the corresponding latitude and longitude.
SoftImpute-Alternating Least Square (Hastie et al., 2015)

\[
\min_{A_t, B_t} F(A_t, B_t) := \frac{1}{2} \| \hat{X}_t - A_t B_t^T \|_F^2 + \frac{\lambda_1}{2} (\| A_t \|_F^2 + \| B_t \|_F^2) \tag{2}
\]

where \( \hat{X}_t \) is a "filled-in" \( m \times n \) matrix, with \( \hat{X}_t = P_{\Omega_t}(X_t) + P_{\Omega^\perp_t}(\tilde{A}_t \tilde{B}_t^T) \), and \( \tilde{A}_t, \tilde{B}_t \) are the two factor matrices in the previous iterative step.
Matrix Completion with Factorization

Figure: TEC maps: observed (left) and fitted by the SoftImpute approach (right).
Apart from the matrix completion method, one can also impute each TEC map $X_t$ with **Spherical Harmonics (SH)**.
Apart from the matrix completion method, one can also impute each TEC map $X_t$ with **Spherical Harmonics (SH)**.

Spherical Harmonics is approximating data on a surface with a linear combination of several basis functions. For TEC map, we can think of TEC value distributed on the globe, and we use Spherical Harmonics to approximate this surface of TEC values.
Figure: Example of Spherical Harmonics Fitting
Spherical Harmonics

Figure: Source: Nortje et al., 2015
Our final framework has the following features:

- Impute a consecutive sequence of TEC maps (i.e. TEC videos)
Conceptual Framework

Our final framework has the following features:

- Impute a consecutive sequence of TEC maps (i.e. TEC videos)
- Use a matrix factorization formulation as the imputed matrix
Conceptual Framework

Our final framework has the following features:

- Impute a consecutive sequence of TEC maps (i.e. TEC videos)
- Use a matrix factorization formulation as the imputed matrix
- Use spherical harmonics as a warm-start (we call it “auxiliary data“)
Conceptual Framework

Our final framework has the following features:

- Impute a consecutive sequence of TEC maps (i.e. TEC videos)
- Use a matrix factorization formulation as the imputed matrix
- Use spherical harmonics as a warm-start (we call it “auxiliary data“)
- Penalizes the matrix norm of the factor matrices (soft constraint on rank)
Conceptual Framework

Our final framework has the following features:

- Impute a consecutive sequence of TEC maps (i.e. TEC videos)
- Use a matrix factorization formulation as the imputed matrix
- Use spherical harmonics as a warm-start (we call it “auxiliary data“)
- Penalizes the matrix norm of the factor matrices (soft constraint on rank)
- Reinforce smoothness of the imputed results along the temporal dimension
Conceptual Framework

Our final framework has the following features:

- Impute a consecutive sequence of TEC maps (i.e. TEC videos)
- Use a matrix factorization formulation as the imputed matrix
- Use spherical harmonics as a warm-start (we call it “auxiliary data“)
- Penalizes the matrix norm of the factor matrices (soft constraint on rank)
- Reinforce smoothness of the imputed results along the temporal dimension
- Objective function has the form:

\[
\text{Imputation Loss} + \lambda_1 \times \text{Matrix Norm Penalty} \\
+ \lambda_2 \times \text{Temporal Smoothness Penalty} \\
+ \lambda_3 \times \text{Auxiliary Data Penalty}
\]
Our model has a name “Video Imputation with SoftImpute, Temporal smoothing and Auxiliary data“ (VISTA)
Conceptual Framework

Objective Function

\[
\min_{A_1:T, B_1:T} \left\{ F(A_1:T, B_1:T) \triangleq \frac{1}{2} \sum_{t=1}^{T} \| P_{\Omega_t} (X_t - A_t B_t^T) \|_F^2 
+ \frac{\lambda_1}{2} \sum_{t=1}^{T} (\| A_t \|_F^2 + \| B_t \|_F^2) 
+ \frac{\lambda_2}{2} \sum_{t=2}^{T} \| A_t B_t^T - A_{t-1} B_{t-1}^T \|_F^2 
+ \frac{\lambda_3}{2} \sum_{t=1}^{T} \| Y_t - A_t B_t^T \|_F^2 \right\}
\]

where \( Y_1, Y_2, \ldots, Y_T \) are \( m \times n \) auxiliary data with no missing values.
An alternative perspective to interpret the objective function is to think about it under a Bayesian setup:

\[ X_t \sim N(A_tB_t^T, \sigma^2) \]  
(Data generating model)

\[ A_t \sim N(0, \frac{1}{\lambda_1}\sigma^2) \]  
(Prior of A)

\[ B_t \sim N(0, \frac{1}{\lambda_1}\sigma^2) \]  
(Prior of B)

\[ A_tB_t^T \sim N(A_{t-1}B_{t-1}^T, \frac{1}{\lambda_2}\sigma^2) \]  
(Random walk assumption)

\[ A_tB_t^T \sim N(Y_t, \frac{1}{\lambda_3}\sigma^2) \]  
(Prior of \(AB^T\))

And the objective function is maximizing the posterior likelihood based on T frames of data.
There are in total $T$ frames to be imputed at the same time, and each frame has its own $A_t, B_t$ factors.
There are in total $T$ frames to be imputed at the same time, and each frame has its own $A_t, B_t$ factors.

Update the factors $A_1, A_2, \ldots, A_T, B_1, B_2, \ldots, B_T$ cyclically:
$A_1 \rightarrow A_2 \rightarrow \cdots \rightarrow A_T \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_T \rightarrow A_1 \rightarrow A_2 \rightarrow \ldots$.
Algorithm Outline

- There are in total $T$ frames to be imputed at the same time, and each frame has its own $A_t, B_t$ factors.
- Update the factors $A_1, A_2, \ldots, A_T, B_1, B_2, \ldots, B_T$ cyclically: $A_1 \rightarrow A_2 \rightarrow \cdots \rightarrow A_T \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_T \rightarrow A_1 \rightarrow A_2 \rightarrow \ldots$.
- Fix $2T - 1$ matrices and update one matrix at a time with majorization-minimization (MM) algorithm. The final form is simply doing a least square.
Update Matrix with Least Square

Suppose in the $k$-th round, we wish to update $A_t$. The current values for the other factors are: $A_1^{(k+1)}, A_2^{(k+1)}, \ldots, A_{t-1}^{(k+1)}, A_t^{(k)}, \ldots A_T^{(k)}$ and $B_1^{(k)}, B_2^{(k)}, \ldots, B_T^{(k)}$. Keeping every matrix other than $A_t$ fixed at their current values, the convex optimization problem is reduced to the following optimization problem:

$$
\min_{A_t} \left\{ Q(A_t|A_1^{(k+1)}, A_t^{(k)}, A_{t+1:T}, B_1^{(k)}) \right. \\
\hspace{1cm} \left. \triangleq \frac{1}{2} \| P_{\Omega_t} (X_t - A_t (B_t^{(k)})^T) \|_F^2 + \frac{\lambda_1}{2} \| A_t \|_F^2 + \frac{\lambda_3}{2} \| Y_t - A_t (B_t^{(k)})^T \|_F^2 \\
\hspace{1cm} + \frac{\lambda_2}{2} I_{\{t>1\}} \| A_t (B_t^{(k)})^T - A_{t-1}^{(k+1)} (B_{t-1}^{(k)})^T \|_F^2 \\
\hspace{1cm} + \frac{\lambda_2}{2} I_{\{t<T\}} \| A_{t+1} (B_{t+1}^{(k)})^T - A_t (B_t^{(k)})^T \|_F^2 \right\}
$$
The very first term is \( \| P_{\Omega_t} (X_t - A_t (B_t^{(k)})^T) \|_F^2 \), which can be upper bounded easily by:

\[
\| P_{\Omega_t} (X_t - A_t (B_t^{(k)})^T) \|_F^2 \leq \| P_{\Omega_t} (X_t) + P_{\Omega_t^\perp} (A_t^{(k)} (B_t^{(k)})^T) - A_t (B_t^{(k)})^T \|_F^2
\]
Update Matrix with Least Square

Substituting the first term with its upper bound, and denote the new objective function as $\tilde{Q}(A_t|A_{1:t-1}^{(k+1)}, A_{t+1:T}^{(k)}, B_{1:T}^{(k)})$. Then one can take the derivative of $\tilde{Q}$ w.r.t. $A_t$ and sets it to zero and get:

$$A_t^{(k+1)} = \left[(1 + \lambda_2(I_{\{t<T\}} + I_{\{t>1\}}) + \lambda_3)(B_t^{(k)})^T B_t^{(k)} + \lambda_1 I\right]^{-1} Z_t^{(k)} B_t^{(k)}$$

where

$$Z_t^{(k)} = P_{\Omega_t}(X_t) + P_{\Omega_t^\perp}(A_t^{(k)}(B_t^{(k)})^T$$

$$+ \lambda_2 \left(I_{\{t>1\}} A_{t-1}^{(k+1)}(B_{t-1}^{(k)})^T + I_{\{t<T\}} A_{t+1}^{(k)}(B_{t+1}^{(k)})^T \right)$$

$$+ \lambda_3 Y_t$$
Algorithm 1 softImpute-ALS with Temporal Smoothing and Auxiliary Data

**Input:** $m \times n$ Sparse data $X_1, X_2, \ldots, X_T$, $m \times n$ auxiliary data $Y_1, Y_2, \ldots, Y_T$, operating rank $r$. Maximum iteration $K$ and convergence threshold $\tau$.

**Output:** Imputation of sparse data $A_1 B_1^T, A_2 B_2^T, \ldots, A_T B_T^T$.

1. **Initialization:** For $1 \leq t \leq T$, $A_t^{(1)} = U_t D_t$, $B_t^{(1)} = V_t D_t$, where $U_t, V_t$ are $m \times r, n \times r$ randomly chosen matrix with orthogonal columns. $D_t$ is $I_{r \times r}$.

2. **Update A:**

   3. for $t = 1 : T$ do

      4. a. Let $X_t^{(k)} = P_{\Omega_t} (X_t) + P_{\Omega_t^+} (A_t^{(k)} (B_t^{(k)})^T)$, which is the “filled-in” version of $X_t$

      5. b. Let $Z_t^{(k)}$ be the weighted label in equation (11)

      6. c. $A_t^{(k+1)}$ is updated as equation (13)

   7. end for

8. **Update B:** For every $t$, repeat a,b,c steps above, with $X_t^{(k)}, Z_t^{(k)}$ being replace by $X_t^{(k+\frac{1}{2})}, Z_t^{(k+\frac{1}{2})}$. $B_t^{(k+1)}$ is calculated following equation (14).

9. Repeat updating $A_{1:T}$ and $B_{1:T}$ until convergence. The algorithm converges when $\max \{ \nabla F_1^{(k)}, \nabla F_2^{(k)}, \ldots, \nabla F_T^{(k)} \} < \tau$, with $\nabla F_t^{(k)}$ defined in (15).

10. For any $t$, denote the final output as $A_t^*, B_t^*$. Let $X_t^* = P_{\Omega_t} (X_t) + P_{\Omega_t^+} (A_t^* (B_t^*)^T)$.

11. Do SVD for $A_t^* (B_t^*)^T = U_t^* (D_t^*)^2 (V_t^*)^T$.

12. Define $M_t = X_t^* V_t^*$ and do SVD for $M_t = \tilde{U}_t \tilde{D}_t R_t^T$.

13. Do soft-thresholding on $\tilde{D}_t$: $\tilde{D}_{t, \lambda_1} = \text{diag} [(\sigma_1 - \lambda_1)_+, (\sigma_2 - \lambda_1)_+, \ldots, (\sigma_r - \lambda_1)_+]$

14. Output imputation for time $t$ as $\bar{U}_t \tilde{D}_{t, \lambda_1} (V_t^* R_t)^T$.
Convergence Guarantee

Across the iterations of our algorithm, we denote the iterative value of $A_{1:T}, B_{1:T}$ in the $k$-th round of algorithm as $A^{(k)}_{1:T}, B^{(k)}_{1:T}$. Then we can prove the following property of our algorithm:
Convergence Guarantee

Across the iterations of our algorithm, we denote the iterative value of $A_{1:T}, B_{1:T}$ in the $k$-th round of algorithm as $A^{(k)}_{1:T}, B^{(k)}_{1:T}$. Then we can prove the following property of our algorithm:

**Objective Function is Non-Increasing**

Define the descent of objective function value at iteration $k$ as

$$\Delta_k = F(A^{(k)}_{1:T}, B^{(k)}_{1:T}) - F(A^{(k+1)}_{1:T}, B^{(k+1)}_{1:T}).$$

Then the value of the objective function is non-increasing, i.e.,

$$F(A^{(k)}_{1:T}, B^{(k)}_{1:T}) \geq F(A^{(k+1)}_{1:T}, B^{(k)}_{1:T}) \geq F(A^{(k+1)}_{1:T}, B^{(k+1)}_{1:T}),$$

thus $\Delta_k \geq 0$, for all $k \geq 1$. 
Convergence Rate

Convergence Rate Lower Bound

Let the limit of the objective function $F(A_{1:T}^{(k)}, B_{1:T}^{(k)})$ be $f^\infty$, we have:

$$\min_{1 \leq k \leq K} \Delta_k \leq \frac{F(A_{1:T}^{(1)}, B_{1:T}^{(1)}) - f^\infty}{K}$$

where $K$ is the total number of iterations.

These results suggest that our algorithm is converging at a rate of $O(1/K)$. 
Empirical Analysis: Data Pipeline

**Input Video**
- Fit each frame with spherical harmonics, order = 11, v = 0.1
- Box-Cox Transformation with λ = 0.1
- Calculate the mean (μ) and standard deviation (σ) of all observed entries
- Standardize all observed entries with the μ, σ
- Standardize with the μ, σ of the input video
- Box-Cox Transformation with λ = 0.1
- Impute Algorithm
  - Multiply all entries with σ and add μ
  - Inverse Box-Cox transformation
  - Output Video

**Figure:** Data Pipeline
In our simulation study, we use the IGS dataset of TEC map, which is of low resolution but is fully observed without missing values. We fit our model on several days of IGS data in Sept. 2017. Each day contains data of size $181 \times 361 \times 96$, where every matrix is of size $181 \times 361$. To mimic some data missing patterns typically observed in Madrigal database (high-res TEC maps), we artificially “created” some missingness.
In our simulation study, we use the IGS dataset of TEC map, which is of low resolution but is fully observed without missing values. We fit our model on several days of IGS data in Sept. 2017. Each day contains data of size $181 \times 361 \times 96$, where every matrix is of size $181 \times 361$.

To mimic some data missing patterns typically observed in Madrigal database (high-res TEC maps), we artificially “created“ some missingness.
Simulation Study: Missingness Design

Figure: Create Missing Data
Simulation Study: Missingness Design

- Random missingness (sub-figure B): for each matrix, randomly drop 30%/50%/70% of the pixels.
Simulation Study: Missingness Design

- **Random missingness** (sub-figure B): for each matrix, randomly drop 30%/50%/70% of the pixels.

- **Temporal missingness** (sub-figure C): for the first matrix, randomly drop 30%/50%/70% of pixels, and let the missing mask move 6 columns horizontally (direction shown as the red arrow).
Simulation Study: Missingness Design

- **Random missingness (sub-figure B):** for each matrix, randomly drop 30%/50%/70% of the pixels.

- **Temporal missingness (sub-figure C):** for the first matrix, randomly drop 30%/50%/70% of pixels, and let the missing mask move 6 columns horizontally (direction shown as the red arrow).

- **Random patch missingness (sub-figure E):** for each frame, randomly pick a center on a fixed bounding box around high TEC value region (sub-figure D) and create a $27 \times 27$ or $45 \times 45$ or $63 \times 63$ patch as missing.
Simulation Study: Missingness Design

- Random missingness (sub-figure B): for each matrix, randomly drop 30%/50%/70% of the pixels.
- Temporal missingness (sub-figure C): for the first matrix, randomly drop 30%/50%/70% of pixels, and let the missing mask move 6 columns horizontally (direction shown as the red arrow).
- Random patch missingness (sub-figure E): for each frame, randomly pick a center on a fixed bounding box around high TEC value region (sub-figure D) and create a $27 \times 27$ or $45 \times 45$ or $63 \times 63$ patch as missing.
- Temporal patch missingness (sub-figure F): similar to patch missingness, but the center of the $27 \times 27/45 \times 45/63 \times 63$ patch moves along the bounding box at the speed of 6 columns(rows) per matrix (anti-clockwise as shown by the red arrow).
Simulation Study: Models & Metrics

We consider fitting the following VISTA models on each of the missing pattern:

1. **soft**: softImpute as in Hastie et al., 2015: $\lambda_1 = 0.9, \lambda_2 = 0, \lambda_3 = 0$.  
   (Benchmark model)

2. **TS**: softImpute + temporal smoothing: $\lambda_1 = 0.9, \lambda_2 = 0.05, \lambda_3 = 0$.

3. **SH**: softImpute + auxiliary data based on spherical harmonics: $\lambda_1 = 0.9, \lambda_2 = 0, \lambda_3 = 0.01$.

4. **TS+SH**: softImpute + temporal smoothing + auxiliary data based on spherical harmonics: $\lambda_1 = 0.9, \lambda_2 = 0.05, \lambda_3 = 0.01$. 
To evaluate the performance of the imputation, we compute **Relative Squared Error (RSE)**:

$$\text{RSE}(X_t, X^*_t, \Omega_t) = \frac{\|P_{\Omega_t^\perp}(X^*_t - X_t)\|_F}{\|P_{\Omega_t^\perp}(X_t)\|_F},$$

where $X_t$ is the fully-observed IGS data. $\Omega_t$ is the bitmap indicating the observed pixels. $P_{\Omega_t^\perp}(.)$ is a projection operator onto the missing pixels. $X^*_t$ is the imputation of $P_{\Omega_t}(X_t)$ and $\|.\|_F$ is the Frobenius norm.
Figure: Random missing and temporal missing results. Three variants of our method are considered: TS, SH, TS+SH. The scatter points show the average test set RSE margin over baseline softImpute method, positive means performance better than softImpute. Error bar gives the 95% confidence interval.
Figure: Random patch missing and temporal patch missing results. Three variants of our method are considered: TS, SH, TS+SH. The scatter points show the average test set RSE margin over baseline softImpute method, positive means performance better than softImpute. Error bar gives the 95% confidence interval.
Simulation Study: Imputation Example

Figure: Example of imputing IGS data with temporal patch missingness.
To impute the final Madrigal TEC map, we fit VISTA on each day of TEC map, which is of size $181 \times 361 \times 288$. Every matrix is of size $181 \times 361$. We showcase our results based on two days of data: Sept-08-2017 (storm day), Sept-03-2017 (non storm day).
To impute the final Madrigal TEC map, we fit VISTA on each day of TEC map, which is of size $181 \times 361 \times 288$. Every matrix is of size $181 \times 361$. We showcase our results based on two days of data: Sept-08-2017 (storm day), Sept-03-2017 (non storm day).

Tuning parameters ($\lambda_1, \lambda_2, \lambda_3$) are determined with grid-search.
To impute the final Madrigal TEC map, we fit VISTA on each day of TEC map, which is of size $181 \times 361 \times 288$. Every matrix is of size $181 \times 361$. We showcase our results based on two days of data: Sept-08-2017 (storm day), Sept-03-2017 (non storm day).

- Tuning parameters ($\lambda_1, \lambda_2, \lambda_3$) are determined with grid-search.

- Since Madrigal TEC map contains missing values, it not possible to directly validate the fitted model on the missing values. We instead randomly drop 20% of the observed pixels and use them as test set, and we fit our model only on the rest 80% of the observed pixels.
## Empirical Analysis

### Imputing Madrigal TEC map: Result

<table>
<thead>
<tr>
<th>Storm Day</th>
<th>Model</th>
<th>test RSE</th>
<th>test MSE</th>
<th># matrices better than softImpute</th>
<th># matrices worse than Full model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>softImpute ($\lambda_1 = 0.9$)</td>
<td>10.895%</td>
<td>2.675</td>
<td>/</td>
<td>285 (98.96%)</td>
</tr>
<tr>
<td></td>
<td>TS ($\lambda_1 = 0.9$, $\lambda_2 = 0.2$)</td>
<td>9.643%</td>
<td>2.106</td>
<td>284 (98.62%)</td>
<td>267 (92.71%)</td>
</tr>
<tr>
<td></td>
<td>SH ($\lambda_1 = 0.9$, $\lambda_3 = 0.021$)</td>
<td>9.936%</td>
<td>2.227</td>
<td>287 (99.65%)</td>
<td>274 (95.14%)</td>
</tr>
<tr>
<td></td>
<td>Full ($\lambda_1 = 0.9$, $\lambda_2 = 0.2$, $\lambda_3 = 0.021$)</td>
<td>9.357%</td>
<td>1.983</td>
<td>285 (98.96%)</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>Directly use Spherical Harmonics</td>
<td>17.354%</td>
<td>6.720</td>
<td>0 (0%)</td>
<td>288 (100%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Storm Day</th>
<th>Model</th>
<th>test RSE</th>
<th>test MSE</th>
<th># matrices better than softImpute</th>
<th># matrices worse than Full model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>softImpute ($\lambda_1 = 0.9$)</td>
<td>10.424%</td>
<td>1.324</td>
<td>/</td>
<td>283 (98.26%)</td>
</tr>
<tr>
<td></td>
<td>TS ($\lambda_1 = 0.9$, $\lambda_2 = 0.31$)</td>
<td>8.880%</td>
<td>0.958</td>
<td>281 (97.57%)</td>
<td>235 (81.60%)</td>
</tr>
<tr>
<td></td>
<td>SH ($\lambda_1 = 0.9$, $\lambda_3 = 0.03$)</td>
<td>9.231%</td>
<td>1.032</td>
<td>287 (99.65%)</td>
<td>278 (96.53%)</td>
</tr>
<tr>
<td></td>
<td>Full ($\lambda_1 = 0.9$, $\lambda_2 = 0.31$, $\lambda_3 = 0.03$)</td>
<td>8.592%</td>
<td>0.895</td>
<td>283 (98.26%)</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>Directly use Spherical Harmonics</td>
<td>15.732%</td>
<td>2.893</td>
<td>0 (0%)</td>
<td>288 (100%)</td>
</tr>
</tbody>
</table>

Table 1: Empirical study results from the madrigal database.

**Figure:** Imputation Result
Imputing Madrigal TEC map: Non-storm Day Example

Figure: 2017-09-03/00:02:30 UT Result
Imputing Madrigal TEC map: Storm Day Example

Figure: 2017-09-08/00:02:30 UT Result
Conclusion

- We propose a new imputation method (VISTA), combining matrix completion with soft rank constraint, temporal smoothing and spherical harmonics in a unified framework, to impute Total Electron Content (TEC) maps with over 50% data missing.
Conclusion

- We propose a new imputation method (VISTA), combining matrix completion with soft rank constraint, temporal smoothing and spherical harmonics in a unified framework, to impute Total Electron Content (TEC) maps with over 50% data missing.
- Matrix completion provides the basic low-rank structure of the imputation.
Conclusion

- We propose a new imputation method (VISTA), combining matrix completion with soft rank constraint, temporal smoothing and spherical harmonics in a unified framework, to impute Total Electron Content (TEC) maps with over 50% data missing.

- Matrix completion provides the basic low-rank structure of the imputation.

- Temporal smoothing borrows information from TEC maps at adjacent timestamp and smooth the low-rank structure.
Conclusion

- We propose a new imputation method (VISTA), combining matrix completion with soft rank constraint, temporal smoothing and spherical harmonics in a unified framework, to impute Total Electron Content (TEC) maps with over 50% data missing.
- Matrix completion provides the basic low-rank structure of the imputation.
- Temporal smoothing borrows information from TEC maps at adjacent timestamp and smooth the low-rank structure.
- Spherical harmonics provides a warm-start of imputation values at big patches of missingness.
Conclusion

- We propose a new imputation method (VISTA), combining matrix completion with soft rank constraint, temporal smoothing and spherical harmonics in a unified framework, to impute Total Electron Content (TEC) maps with over 50% data missing.
- Matrix completion provides the basic low-rank structure of the imputation.
- Temporal smoothing borrows information from TEC maps at adjacent timestamp and smooth the low-rank structure.
- Spherical harmonics provides a warm-start of imputation values at big patches of missingness.
- Empirical results suggest improvements on both global-scale and meso-scale reconstruction.
Future Plan

- We plan to release a data product containing the imputed TEC maps based on VISTA for the last solar cycle (2009-2020).
Future Plan

- We plan to release a data product containing the imputed TEC maps based on VISTA for the last solar cycle (2009-2020).
- We plan to use matrix/tensor-based factor model and other machine learning methods to do TEC map predictions using our VISTA data product as inputs. The ultimate goal is to provide a complete imputation-prediction pipeline for operational use.


