Matrix Completion Methods for the Total Electron Content Video Reconstruction[1]

Yang Chen

Department of Statistics, University of Michigan, Ann Arbor

Joint work with H. Sun, Z. Hua, Y. Sun (UM Statistics) and J. Ren, S. Zou (UM CLaSP)

October 8th, 2021

[1]Click here for the preprint version of our paper.

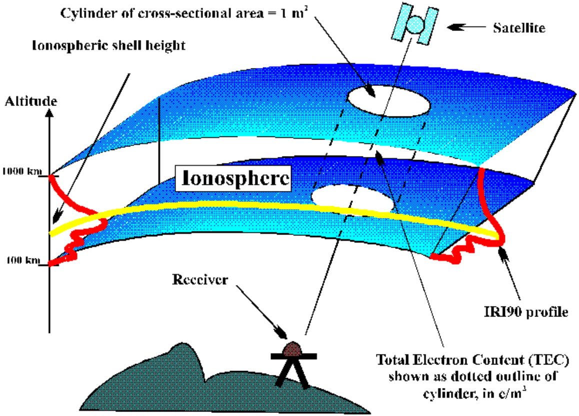
Overview

Outline

Outline

Total Electron Content (TEC) map

Ionosphere: layer in the upper atmosphere 70-1000 km above Earth.
Ionosphere TEC: total number of electrons in the path between satellite (The Global Navigation Satellite Systems (GNSS)) radio transmitter and ground-based receiver. (1 TEC unit = 10^{16} electrons/m²).



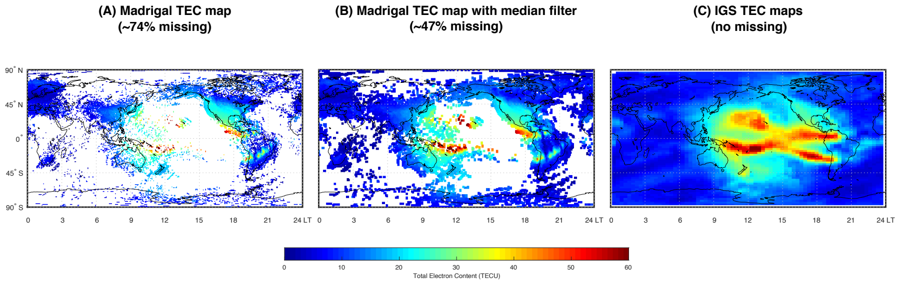
Total Electron Content (TEC) map

Real time monitoring of TEC is important.

- 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.
- TEC measurement has been used in earthquake monitoring, modelling and prediction: a significant reduction in TEC is observable for at least 3 days before major earthquakes.

Total Electron Content (TEC) map

- The Madrigal Database: global maps of vertical TEC measurements with a spatial resolution of $1^\circ \times 1^\circ$ latitude by longitude and a temporal resolution of 5 minutes.
- International GNSS Service (IGS) TEC maps: spherical harmonics fitted TEC maps with a spatial resolution of 2.5° latitude by 5° longitude and a highest temporal resolution of 15 minutes.



Total Electron Content (TEC) map

- Goal: "fill in" missing values in Madrigal data to create high spatial & temporal resolution *full* TEC maps.

Total Electron Content (TEC) map

- Goal: "fill in" missing values in Madrigal data to create high spatial & temporal resolution *full* TEC maps.
- Pattern of missingness in Madrigal TEC maps:

Total Electron Content (TEC) map

- Goal: "fill in" missing values in Madrigal data to create high spatial & temporal resolution *full* TEC maps.
- Pattern of missingness in Madrigal TEC maps:
  - Big patches of missingness in ocean area.
  - Scattered (not random) missingness in land area.
  - Temporally moving missingness patches.

Outline

Framed as a Matrix Completion Problem

- To impute the TEC maps, we adopt classical statistics techniques called **matrix completion**.

Framed as a Matrix Completion Problem

- To impute the TEC maps, we adopt classical statistics techniques called **matrix completion**.
- Matrix completion is a commonly used method in designing recommendation 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 with Factorization

SoftImpute-Alternating Least Square (Hastie et al., 2015)

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

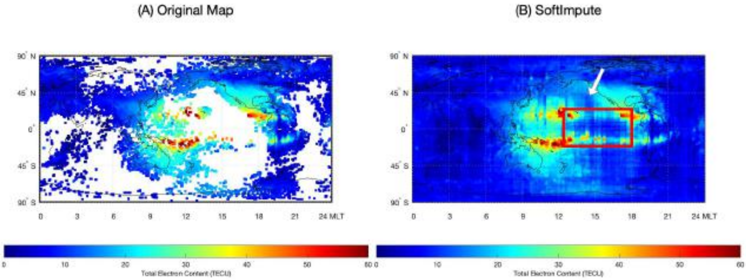
where $\widehat{X}_t$ is a "filled-in" $m \times n$ matrix, with $\widehat{X}_t = \mathrm{P}_{\Omega_t}(X_t) + \mathrm{P}_{\Omega_t^\perp}(\tilde{A}_t \tilde{B}_t^T)$, and $\tilde{A}_t, \tilde{B}_t$ are the two factor matrices in the previous iterative step.

Matrix Completion with Factorization

SoftImpute-Alternating Least Square (Hastie et al., 2015)

$$\min_{A_t, B_t} \quad F(A_t, B_t) := \frac{1}{2}\|\widehat{X}_t - A_t B_t^T\|_F^2 + \frac{\lambda_1}{2}(\|A_t\|_F^2 + \|B_t\|_F^2) \qquad (1)$$

where $\widehat{X}_t$ is a "filled-in" $m \times n$ matrix, with $\widehat{X}_t = \mathrm{P}_{\Omega_t}(X_t) + \mathrm{P}_{\Omega_t^\perp}(\tilde{A}_t \tilde{B}_t^T)$, and $\tilde{A}_t, \tilde{B}_t$ are the two factor matrices in the previous iterative step.



Spherical Harmonics

Approximating data on a surface (TEC values around the globe) with a linear combination of several basis functions.

Spherical Harmonics

Approximating data on a surface (TEC values around the globe) with a linear combination of several basis functions.

2.1. Spherical Harmonics. For a sufficiently smooth surface, represented by a function $f(\theta, \phi)$, an infinite series of spherical harmonic basis functions can be used to represent it in the following form [23]:

$$f(\theta, \phi) \simeq \sum_{l=0}^{l_{\max}} \sum_{m=-l}^{l} a_l^m Y_l^m(\theta, \phi), \qquad (1)$$

where θ and ϕ are the polar and azimuth angles in a spherical coordinate system. As $l_{\max} \to \infty$, this representation becomes an exact description of the surface $f(\theta, \phi)$. The

Source: Nortje et al., 2015

Spherical Harmonics

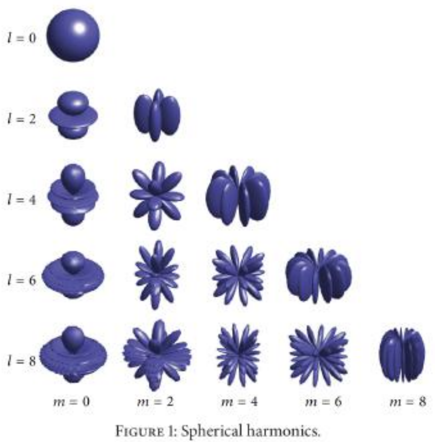


Figure: Source: Nortje et al., 2015

Spherical Harmonics

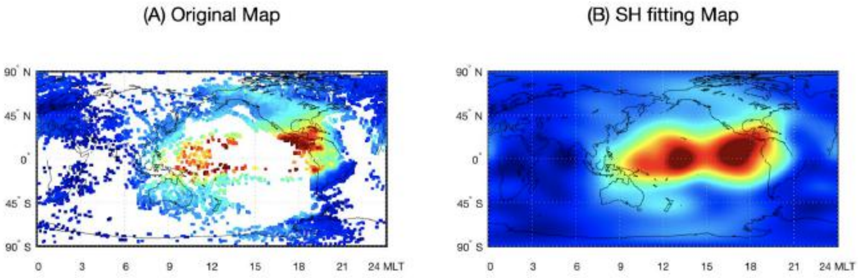
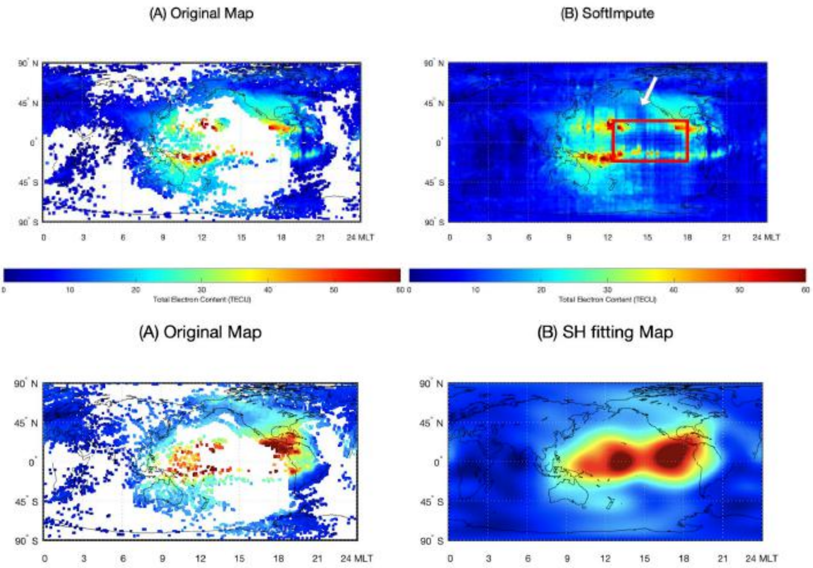


Figure: Example of Spherical Harmonics Fitting

Summary: SoftImpute versus Spherical Harmonics



Outline

Outline

Proposed Method: VISTA

Our final framework has the following features:

- Impute a time series of TEC maps (i.e. TEC videos)

Proposed Method: VISTA

Our final framework has the following features:

- Impute a time series of TEC maps (i.e. TEC videos)
- Use a matrix factorization formulation as the imputed matrix

Proposed Method: VISTA

Our final framework has the following features:

- Impute a time series 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")

Proposed Method: VISTA

Our final framework has the following features:

- Impute a time series 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

Proposed Method: VISTA

Our final framework has the following features:

- Impute a time series 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
- Reinforce smoothness of the imputed results temporally

Proposed Method: VISTA

Our final framework has the following features:

- Impute a time series 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
- Reinforce smoothness of the imputed results temporally

- Objective function has the form:

$$\begin{aligned}\text{Imputation Loss} &+ \lambda_1 \times \text{Matrix Norm Penalty}\\ &+ \lambda_2 \times \text{Temporal Smoothness Penalty}\\ &+ \lambda_3 \times \text{Auxiliary Data Penalty}\end{aligned}$$

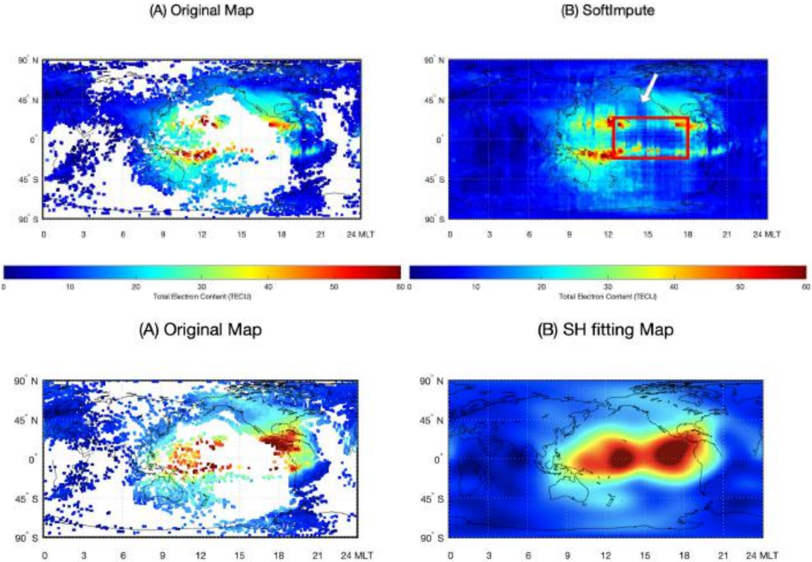
Proposed Method: VISTA

Our model has a name "Video Imputation with SoftImpute, Temporal smoothing and Auxiliary data" (**VISTA**).

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

where $Y_1, Y_2, \ldots, Y_T$ are $m \times n$ auxiliary data with no missing values.

Recall: SoftImpute versus Spherical Harmonics



Outline

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.

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: $A_1 \to A_2 \to \cdots \to A_T \to B_1 \to B_2 \to \cdots \to B_T \to A_1 \to A_2 \to \ldots$.

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: $A_1 \to A_2 \to \cdots \to A_T \to B_1 \to B_2 \to \cdots \to B_T \to A_1 \to A_2 \to \ldots$.
- Fix $2\mathrm{T} - 1$ matrices and update one matrix at a time with majorization-minimization (MM) algorithm.

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: $A_1 \to A_2 \to \cdots \to A_T \to B_1 \to B_2 \to \cdots \to B_T \to A_1 \to A_2 \to \ldots$.
- Fix $2\mathrm{T} - 1$ matrices and update one matrix at a time with majorization-minimization (MM) algorithm.
- The final step in MM 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: $A_1^{(k+1)}, A_2^{(k+1)}, \dots, A_{t-1}^{(k+1)}, A_t^{(k)}, \dots A_T^{(k)}$ and $B_1^{(k)}, B_2^{(k)}, \dots, B_T^{(k)}$. Keeping every matrix other than A_t fixed at their current values, the convex optimization problem is reduced to the following optimization problem:

$$
\begin{aligned}
&\min_{A_t} \Big\{ Q(A_t | A_{1:t-1}^{(k+1)}, A_{t+1:T}^{(k)}, B_{1:T}^{(k)}) \\
&\triangleq \frac{1}{2}\| P_{\Omega_t}(X_t - A_t(B_t^{(k)})^T)\|_F^2 + \frac{\lambda_1}{2}\|A_t\|_F^2 + \frac{\lambda_3}{2}\|Y_t - A_t(B_t^{(k)})^T\|_F^2 \\
&\quad + \frac{\lambda_2}{2}\mathrm{I}_{\{t>1\}}\|A_t(B_t^{(k)})^T - A_{t-1}^{(k+1)}(B_{t-1}^{(k)})^T\|_F^2 \\
&\quad + \frac{\lambda_2}{2}\mathrm{I}_{\{t<T\}}\|A_{t+1}^{(k)}(B_{t+1}^{(k)})^T - A_t(B_t^{(k)})^T\|_F^2 \Big\}
\end{aligned}
$$

Update Matrix with Least Square

$$\min_{A_t}\left\{\frac{1}{2}\|P_{\Omega_t}(X_t - A_t(B_t^{(k)})^T)\|_F^2 + \frac{\lambda_1}{2}\|A_t\|_F^2 + \frac{\lambda_3}{2}\|Y_t - A_t(B_t^{(k)})^T\|_F^2 + \frac{\lambda_2}{2}\mathrm{I}_{\{t>1\}}\|A_t(B_t^{(k)})^T - A_{t-1}^{(k+1)}(B_{t-1}^{(k)})^T\|_F^2 + \frac{\lambda_2}{2}\mathrm{I}_{\{t<T\}}\|A_{t+1}^{(k)}(B_{t+1}^{(k)})^T - A_t(B_t^{(k)})^T\|_F^2\right\}$$

The first term $\|P_{\Omega_t}(X_t - A_t(B_t^{(k)})^T)\|_F^2$ can be upper bounded by:

$$\|P_{\Omega_t}(X_t - A_t(B_t^{(k)})^T) + P_{\Omega_t^\perp}(A_t^{(k)}(B_t^{(k)})^T - A_t(B_t^{(k)})^T)\|_F^2$$
$$= \|P_{\Omega_t}(X_t) + P_{\Omega_t^\perp}(A_t^{(k)}(B_t^{(k)})^T) - A_t(B_t^{(k)})^T\|_F^2$$

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:

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

where

$$\begin{aligned} Z_t^{(k)} &= P_{\Omega_t}(X_t) + P_{\Omega_t^{\perp}}(A_t^{(k)}(B_t^{(k)})^T \\ &+ \lambda_2 \left(\mathsf{I}_{\{t>1\}} A_{t-1}^{(k+1)}(B_{t-1}^{(k)})^T + \mathsf{I}_{\{t<T\}} A_{t+1}^{(k)}(B_{t+1}^{(k)})^T\right) \\ &+ \lambda_3 Y_t \end{aligned}$$

Final Algorithm

Algorithm 1 softImpute-ALS with Temporal Smoothing and Auxiliary Data

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

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

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

Update A:

for $t = 1 : T$ **do**

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

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

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

end for

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)

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

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

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

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

Do soft-thresholding on $\tilde{D}_t$: $\tilde{D}_{t,\lambda_1} = \mathbf{diag}\left[(\sigma_1 - \lambda_1)_+, (\sigma_2 - \lambda_1)_+, \dots, (\sigma_r - \lambda_1)_+\right]$

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

Outline

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 $A_{1:T}^{(k)}, B_{1:T}^{(k)}$. 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 $A_{1:T}^{(k)}, B_{1:T}^{(k)}$. 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 $\Delta_k = F(A_{1:T}^{(k)}, B_{1:T}^{(k)}) - F(A_{1:T}^{(k+1)}, B_{1:T}^{(k+1)})$. Then the value of the objective function is non-increasing, i.e.,

$$F(A_{1:T}^{(k)}, B_{1:T}^{(k)}) \geq F(A_{1:T}^{(k+1)}, B_{1:T}^{(k)}) \geq F(A_{1:T}^{(k+1)}, B_{1:T}^{(k+1)}),$$

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^∞, we have:

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

where K is the total number of iterations.

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

Parameter Tuning

There are multiple hyper-parameters in our imputation model:

- (VISTA Model Fitting) $\lambda_1, \lambda_2, \lambda_3$ controlling the relative weights of the penalty on matrix norm, temporal smoothness and auxiliary data.
- (Auxiliary Data Generating) Number of basis functions and L1-penalty in Spherical Harmonics fitting.

Parameter Tuning

There are multiple hyper-parameters in our imputation model:

- (VISTA Model Fitting) $\lambda_1, \lambda_2, \lambda_3$ controlling the relative weights of the penalty on matrix norm, temporal smoothness and auxiliary data.
- (Auxiliary Data Generating) Number of basis functions and L1-penalty in Spherical Harmonics fitting.

We do not choose the VISTA model parameters based on any data-driven methods for the TEC data. In general, we adopt cross validation.

The model exhibits stable performances across many different choices of $(\lambda_1, \lambda_2, \lambda_3)$, with $\lambda_1 \in [0.5, 1.5], \lambda_2 \in [0.2, 0.3], \lambda_3 \in [0.01, 0.03]$ as recommended ranges for TEC data.

Parameter Tuning

There are multiple hyper-parameters in our imputation model:

- (VISTA Model Fitting) $\lambda_1, \lambda_2, \lambda_3$ controlling the relative weights of the penalty on matrix norm, temporal smoothness and auxiliary data.
- (Auxiliary Data Generating) Number of basis functions and L1-penalty in Spherical Harmonics fitting.

Parameter Tuning

There are multiple hyper-parameters in our imputation model:

- (VISTA Model Fitting) $\lambda_1, \lambda_2, \lambda_3$ controlling the relative weights of the penalty on matrix norm, temporal smoothness and auxiliary data.
- (Auxiliary Data Generating) Number of basis functions and L1-penalty in Spherical Harmonics fitting.

One can tune the auxiliary data parameters via cross-validation: masking out a portion of observed pixels and fit on the rest, then decide the best tuning parameters based on test set RMSE.

A single run of our model to impute one-day TEC map ($T = 288, m = 181, n = 361$) takes $\sim 10 - 20$ minutes on a single-core machine with 16 GB memory.

Outline

Empirical Analysis: Data Pipeline

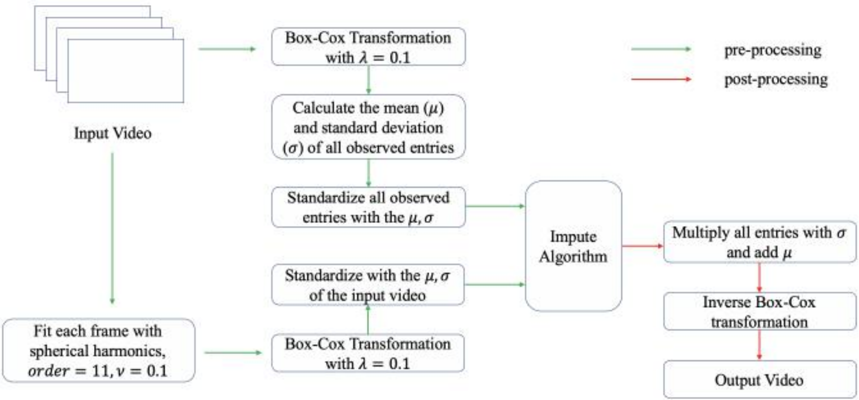


Figure: Data Pipeline

Outline

Simulation Study: Data

- Benchmark: IGS dataset of TEC map, which is of low resolution but is fully observed without missing values.

Simulation Study: Data

- Benchmark: 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.

Simulation Study: Data

- Benchmark: 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 181×361.

Simulation Study: Data

- Benchmark: 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 181×361.

- Missingness patterns chosen to mimic some data missing patterns typically observed in Madrigal database (high-res TEC maps).

Simulation Study: Missingness Design

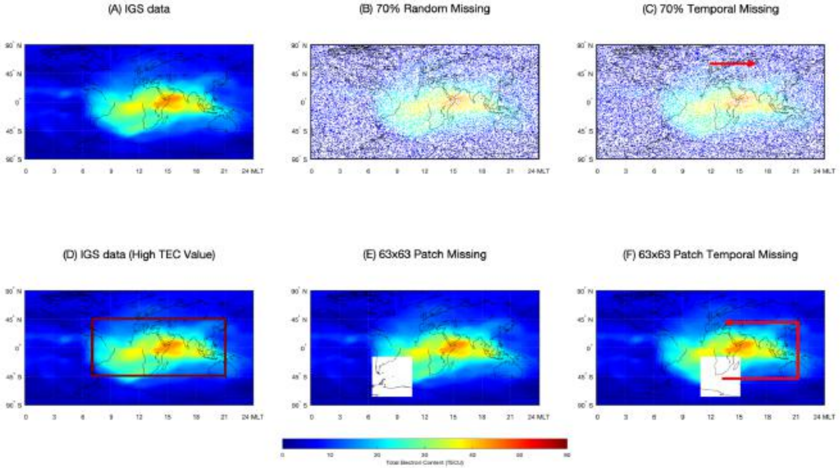


Figure: Create Missing Data

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 27×27 or 45×45 or 63×63 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 27×27 or 45×45 or 63×63 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 fit the following 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: $\lambda_1 = 0.9, \lambda_2 = 0.05, \lambda_3 = 0.01$.

Simulation Study: Models & Metrics

To evaluate the performance of the imputation, we compute **Relative Squared Error** (RSE):

$$\mathrm{RSE}(X_t, X_t^*, \Omega_t) = \frac{\|\mathrm{P}_{\Omega_t^\perp}(X_t^* - X_t)\|_F}{\|\mathrm{P}_{\Omega_t^\perp}(X_t)\|_F},$$

where X_t is the fully-observed IGS data. Ω_t is the bitmap indicating the observed pixels. $\mathrm{P}_{\Omega_t^\perp}(.)$ is a projection operator onto the missing pixels. X_t^* is the imputation of $\mathrm{P}_{\Omega_t}(X_t)$ and $\|.\|_F$ is the Frobenius norm.

Simulation Study: Result of Random Missingness

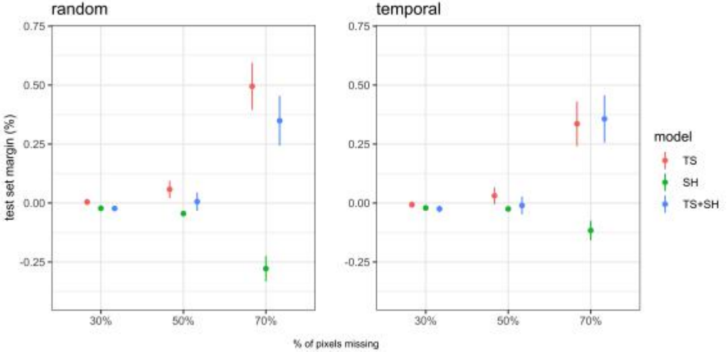


Figure: Random missing and temporal missing results. Three variants of our method are considered: TS, SH, TS+SH. The scatter points show the average test set RSE margin over baseline softImpute method, positive means performance better than softImpute. Error bar gives the 95% confidence interval.

Simulation Study: Result of Patch Missingness

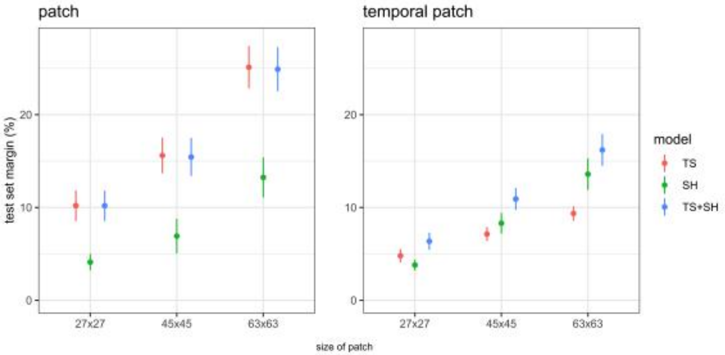


Figure: Random patch missing and temporal patch missing results. Three variants of our method are considered: TS, SH, TS+SH. The scatter points show the average test set RSE margin over baseline softImpute method, positive means performance better than softImpute. Error bar gives the 95% confidence interval.

Simulation Study: Imputation Example

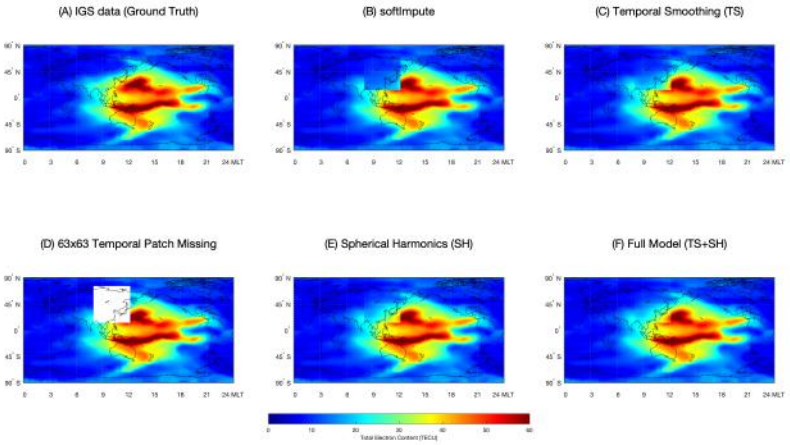


Figure: Example of imputing IGS data with temporal patch missingness.

Outline

Imputing Madrigal TEC map: Data

- We fit VISTA on each day of TEC map, which is of size $181 \times 361 \times 288$. Every matrix is of size 181×361.

Imputing Madrigal TEC map: Data

- We fit VISTA on each day of TEC map, which is of size $181 \times 361 \times 288$. Every matrix is of size 181×361.
- We showcase our results based on two days of data: Sept-08-2017 (storm day), Sept-03-2017 (non storm day).

Imputing Madrigal TEC map: Data

- We fit VISTA on each day of TEC map, which is of size $181 \times 361 \times 288$. Every matrix is of size 181×361.
- 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.

Imputing Madrigal TEC map: Data

- We fit VISTA on each day of TEC map, which is of size $181 \times 361 \times 288$. Every matrix is of size 181×361.
- 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.
- We 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.

Imputing Madrigal TEC map: Result

Storm Day				
Model	test RSE	test MSE	# matrices better than softImpute	# matrices worse than Full model
softImpute ($\lambda_1 = 0.9$)	10.895%	2.675	/	285 (98.96%)
TS ($\lambda_1 = 0.9, \lambda_2 = 0.2$)	9.643%	2.106	284 (98.62%)	267 (92.71%)
SH ($\lambda_1 = 0.9, \lambda_3 = 0.021$)	9.936%	2.227	287 (99.65%)	274 (95.14%)
Full ($\lambda_1 = 0.9, \lambda_2 = 0.2, \lambda_3 = 0.021$)	9.357%	1.983	285 (98.96%)	/
Directly use Spherical Harmonics	17.354%	6.720	0 (0%)	288 (100%)
Non-Storm Day				
Model	test RSE	test MSE	# matrices better than softImpute	# matrices worse than Full model
softImpute ($\lambda_1 = 0.9$)	10.424%	1.324	/	283 (98.26%)
TS ($\lambda_1 = 0.9, \lambda_2 = 0.31$)	8.880%	0.958	281 (97.57%)	235 (81.60%)
SH ($\lambda_1 = 0.9, \lambda_3 = 0.03$)	9.231%	1.032	287 (99.65%)	278 (96.53%)
Full ($\lambda_1 = 0.9, \lambda_2 = 0.31, \lambda_3 = 0.03$)	8.592%	0.895	283 (98.26%)	/
Directly use Spherical Harmonics	15.732%	2.893	0 (0%)	288 (100%)

Table 1: Empirical study results from the madrigal database.

Figure: Imputation Result

Imputing Madrigal TEC map: Non-storm Day Example

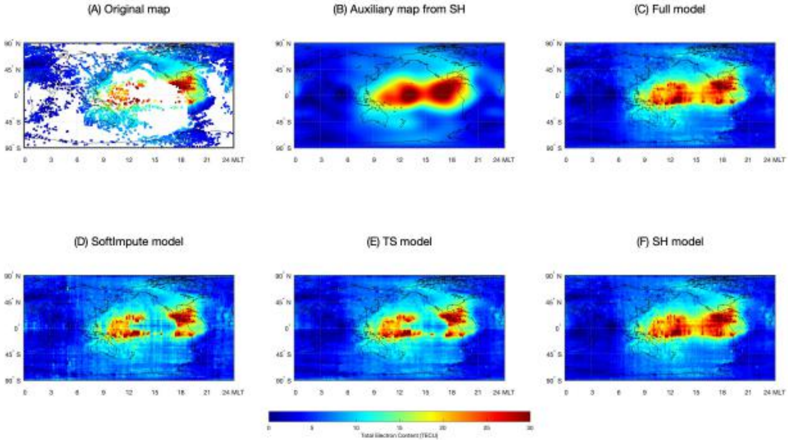


Figure: 2017-09-03/00:02:30 UT Result

Imputing Madrigal TEC map: Storm Day Example

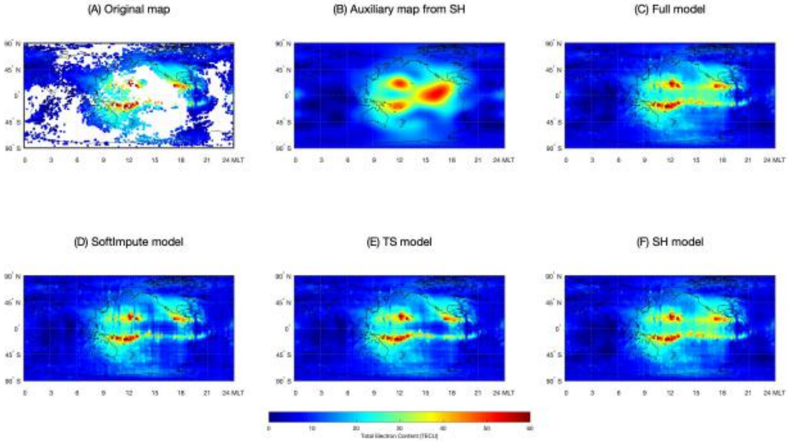


Figure: 2017-09-08/00:02:30 UT Result

Outline

Model Overview

We build a matrix-based auto-regressive model for forecasting TEC maps hours into the future with the following highlights:

- (**VISTA database**) We use the VISTA-imputed TEC map for training, validating and testing our model.

Model Overview

We build a matrix-based auto-regressive model for forecasting TEC maps hours into the future with the following highlights:

- (**VISTA database**) We use the VISTA-imputed TEC map for training, validating and testing our model.
- (**Multi-modality Data**) We utilize both the past VISTA-imputed TEC videos and the past time-series of relevant global plasma parameters (referred to as solar-wind parameters hereafter) to jointly predict the current/future TEC maps.

Model Overview

We build a matrix-based auto-regressive model for forecasting TEC maps hours into the future with the following highlights:

- (**VISTA database**) We use the VISTA-imputed TEC map for training, validating and testing our model.
- (**Multi-modality Data**) We utilize both the past VISTA-imputed TEC videos and the past time-series of relevant global plasma parameters (referred to as solar-wind parameters hereafter) to jointly predict the current/future TEC maps.
- (**Algorithm**) We use iterative least square to alternatively estimate the coefficients of the predictors of different modality.

The Model

For TEC maps $X_{t-P}, X_{t-P+1}, \ldots, X_{t-1}, X_t$ ($m \times n$ matrices) and solar-wind parameters $Z_{t-S}, Z_{t-S+1}, \ldots, Z_{t-1}, Z_t$ ($d \times 1$ vectors), we have an auto-regressive model:

$$X_t = \sum_{p=1}^{P} A_p X_{t-p} B_p + \sum_{k=1}^{K} [Z_{t-1}^T : Z_{t-2}^T : \cdots : Z_{t-S}^T]\beta_k \cdot Y_k + E_t$$

where $A_p, B_p, p = 1, 2, \ldots, P$ are the autoregressive coefficients and $Y_1, Y_2, \ldots, Y_K$ are some pre-specified $m \times n$ matrix basis functions and $\beta_1, \beta_2, \ldots, \beta_K$ are the coefficients for solar-wind parameters. E_t is the error matrix. We use spherical harmonics basis functions as our $Y_1, Y_2, \ldots,$ in the TEC forecasting task.

The Model: Interpretation

If one applies a vectorization operator $\mathrm{vec}(.)$ to both side of the model, one gets:

$$\mathrm{vec}(X_t) = \sum_{p=1}^{P} [B_p \otimes A_p] \mathrm{vec}(X_{t-p}) + \mathcal{Y}\mathcal{B}^T [Z_{t-1}^T : Z_{t-2}^T : \cdots : Z_{t-S}^T]^T + \mathrm{vec}(E_t)$$

where $\mathcal{Y} = [\mathrm{vec}(Y_1) : \mathrm{vec}(Y_2) : \cdots : \mathrm{vec}(Y_K)]$, $\mathcal{B} = [\beta_1 \beta_2 \ldots \beta_K]$.

The Model: Interpretation

The vectorized model looks very similar to a traditional vector auto-regressive model (**VAR**). Each term has its own interpretations in the context of TEC map prediction:

- $\sum_{p=1}^{P}[B_p \otimes A_p]\mathrm{vec}(X_{t-p})$: Auto-regressive (**AR**) part with structured coefficient matrix $B_p \otimes A_p$. The matrix A_p, B_p captures the latitude-latitude and longitude-longitude interaction in TEC forecasting.

The Model: Interpretation

The vectorized model looks very similar to a traditional vector auto-regressive model (**VAR**). Each term has its own interpretations in the context of TEC map prediction:

- $\sum_{p=1}^{P}[B_p \otimes A_p]\mathrm{vec}(X_{t-p})$: Auto-regressive (**AR**) part with structured coefficient matrix $B_p \otimes A_p$. The matrix A_p, B_p captures the latitude-latitude and longitude-longitude interaction in TEC forecasting.
- $\mathcal{Y}\mathcal{B}^T[Z_{t-1}^T : Z_{t-2}^T : \cdots : Z_{t-S}^T]^T$: Semi-parametric part. We have pre-specified a series of matrix basis function Y_k and solar-wind parameter only needs to predict the coefficients to linearly combine all basis functions.

The Model: Interpretation

The vectorized model looks very similar to a traditional vector auto-regressive model (**VAR**). Each term has its own interpretations in the context of TEC map prediction:

- $\sum_{p=1}^{P}[B_p \otimes A_p]\mathrm{vec}(X_{t-p})$: Auto-regressive (**AR**) part with structured coefficient matrix $B_p \otimes A_p$. The matrix A_p, B_p captures the latitude-latitude and longitude-longitude interaction in TEC forecasting.
- $\mathcal{Y}\mathcal{B}^T[Z_{t-1}^T : Z_{t-2}^T : \cdots : Z_{t-S}^T]^T$: Semi-parametric part. We have pre-specified a series of matrix basis function Y_k and solar-wind parameter only needs to predict the coefficients to linearly combine all basis functions.
- $\mathrm{vec}(E_t)$: The error process. We now simply assume i.i.d. errors but structured covariance matrices can be easily modeled.

Estimation Algorithm

To estimate the coefficients of the predictors, i.e. $A_1, \ldots, A_P, B_1, \ldots, B_P$ and $\beta_1, \beta_2, \ldots, \beta_K$ (or simply $\mathcal{B}$), we introduce the following optimization problem:

$$\min_{\substack{A_1,\ldots,A_P \\ B_1,\ldots,B_P \\ \mathcal{B}}} \sum_t \left\| X_t - \sum_{p=1}^{P} A_p X_{t-p} B_p - \mathcal{Y}\mathcal{B}^T [Z_{t-1}^T : Z_{t-2}^T : \cdots : Z_{t-S}^T]^T \right\|_F^2$$

We can attach penalties for $A_1, \ldots, A_P, B_1, \ldots, B_P, \mathcal{B}$ following the prediction loss above, if needed.

Estimation Algorithm

To estimate the coefficients, we optimize the all the coefficients, within each iteration of the algorithm, in the order of: $A_1 \to B_1 \to A_2 \to B_2 \to \cdots \to A_P \to B_P \to \mathcal{B}$. Whenever we update one of the coefficient in this chain, we fix all other coefficients.

Estimation Algorithm

To update the coefficients, we simply derive the first-order condition and solve the matrix equations accordingly:

- A_j:

$$\sum_t A_j X_{t-j} \hat{B}_j^T \hat{B}_j X_{t-j}^T =$$

$$\sum_t \left(X_t - \sum_{i \neq j} \hat{A}_i^T X_{t-i} \hat{B}_i^T - \sum_k \mathcal{Y} \hat{\mathcal{B}}^T \mathcal{Z}_t \right) \hat{B}_j X_{t-j}^T$$

where $\mathcal{Z}_t = [Z_{t-1}^T : Z_{t-2}^T : \cdots : Z_{t-S}^T]^T$ and $\hat{A}, \hat{B}, \hat{\mathcal{B}}$ are the current estimates of the coefficients.

Estimation Algorithm

To update the coefficients, we simply derive the first-order condition and solve the matrix equations accordingly:

- B_j:

$$\sum_t B_j X_{t-j}^T \hat{A}_j^T \hat{A}_j X_{t-j} =$$

$$\sum_t \left(X_t - \sum_{i \neq j} \hat{A}_i^T X_{t-i} \hat{B}_i^T - \sum_k \mathcal{Y} \hat{\mathcal{B}}^T \mathcal{Z}_t \right)^T \hat{A}_j X_{t-j}$$

Estimation Algorithm

To update the coefficients, we simply derive the first-order condition and solve the matrix equations accordingly:

- $\mathcal{B}$:

$$\left(\sum_t \mathcal{Z}_t \mathcal{Z}_t^T\right) \mathcal{B}(\mathcal{Y}_*^T \mathcal{Y}) =$$
$$\sum_t \left\{ \left[\mathrm{vec}\left(X_t - \sum_i \hat{A}_i^T X_{t-i} \hat{B}_i^T \right)^T \mathcal{Y} \right] \otimes \mathcal{Z}_t \right\}$$

where $\mathcal{Y}_* = [\mathrm{vec}(Y_1^T) : \mathrm{vec}(Y_2^T) : \cdots : \mathrm{vec}(Y_K^T)]$

Preliminary Empirical Results

We conduct a first-step empirical analysis of the TEC forecast problem using our model with our VISTA database in August, 2017:

- VISTA TEC map is available at 5-min cadence. Solar-wind parameter is available at 1-min cadence. We down-sample the data to 15-min cadence for both datasets.
- Take the first 20 days of August, 2017 as training set and the trailing 5 days as the testing set. ($\sim 2,000$ frames of TEC map for model training, ~ 500 frames for testing the prediction performance)
- Predictors are normalized to have standard Gaussian distribution before model training.

Preliminary Empirical Results

Table 1: Preliminary Results for the TEC Forecasting Model, pixel-wise forecast Mean-Squared Error

Model		15-min			1h			3h		
AR	SW	$K = 36$	$K = 81$	$K = 121$	$K = 36$	$K = 81$	$K = 121$	$K = 36$	$K = 81$	$K = 121$
P = 1	S = 0		1.430			4.566			13.844	
P = 2	S = 0		1.011			3.913			12.993	
P = 3	S = 0		1.017			3.889			12.948	
P = 1	S = 4	1.103	1.103	1.104	4.295	4.295	4.296	13.987	13.993	13.995
P = 1	S = 8	1.119	1.120	1.121	4.409	4.412	4.413	14.840	14.848	14.851
P = 1	S = 12	1.128	1.130	1.132	4.516	4.522	4.524	15.149	15.168	15.172

* P is the number of lagged term in the auto-regressive part. S is the number of lagged term in the solar-wind parameter part. K is the number of basis functions. MSE is calculated for all pixels in the test set.

Outline

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

Imputation

Release a data product containing the imputed TEC maps based on VISTA for the last solar cycle (2009-2020).

Future Plan

Imputation

Release a data product containing the imputed TEC maps based on VISTA for the last solar cycle (2009-2020).

Prediction

- Give uncertainty quantification of the model prediction.

Future Plan

Imputation

Release a data product containing the imputed TEC maps based on VISTA for the last solar cycle (2009-2020).

Prediction

- Give uncertainty quantification of the model prediction.
- Derive the joint asymptotic properties of the estimator for $A_1, A_2, \ldots, A_P, B_1, B_2, \ldots, B_P$ and $\mathcal{B}$.

Future Plan

Imputation

Release a data product containing the imputed TEC maps based on VISTA for the last solar cycle (2009-2020).

Prediction

- Give uncertainty quantification of the model prediction.
- Derive the joint asymptotic properties of the estimator for $A_1, A_2, \ldots, A_P, B_1, B_2, \ldots, B_P$ and $\mathcal{B}$.
- Provide a complete imputation-prediction pipeline for operational use.

Thank you!

Questions?

Email Yang Chen, ychenang@umich.edu

References I

Hastie, Trevor, Rahul Mazumder, Jason D Lee, and Reza Zadeh (2015). "Matrix completion and low-rank SVD via fast alternating least squares". In: *The Journal of Machine Learning Research* 16.1, pp. 3367–3402.

Nortje, Caitlin R, Wil OC Ward, Bartosz P Neuman, and Li Bai (2015). "Spherical harmonics for surface parametrisation and remeshing". In: *Mathematical Problems in Engineering* 2015.