



Completion of Marine Wireless Sensor Monitoring Data Based on Tensor Mode-n Rank and Tucker Operator

Peng Xu¹, Huayang Chen¹, Chunhua Yu^{1(✉)}, Tongtong Wang²,
and Yechao Bai¹

¹ Nanjing University, Nanjing 210023, China
yuch@nju.edu.cn

² Tianjin Artificial Intelligence Innovation Center (TAIIC),
Tianjin 300450, China

Abstract. The marine monitoring system is one of the frontier technologies actively developed by major countries in the world today. The current ocean monitoring system mainly relies on technologies such as positioning, control, and wireless sensors. The buoy equipped with a variety of wireless sensors continuously collects data on the ocean. However, due to natural environmental influences and malicious tampering by the enemy, the data directly obtained by the wireless sensor buoys may contain large errors. Therefore, we construct the obtained original data into a tensor model, and at the same time replace the data with larger errors into null data “0”. Based on the tensor mode-n rank, we use the alternating direction method of multipliers (ADMM) framework, combined with the tensor Tucker decomposition, and introduce the tensor Tucker singular value operator to it. The overall data can be completed from the existing original data with missing values, so as to optimize the original ocean monitoring data. The method is compared with the data completion based on tensor Tucker decomposition and the linear regression prediction based on principal component analysis. Numerical experiments are given to confirm the superiority of the proposed method.

Keywords: Marine monitoring · Wireless sensors · Tensor completion · Tucker operator

1 Introduction

Our country is rich in marine resources. Marine environment monitoring is also one of the necessary technical means to study marine resources, which is conducive to safeguarding our country’s marine rights and interests, early warning of marine natural environmental risks, and protection of the marine environment.

The monitoring system for the marine environment mainly relies on radar satellite positioning, system control, and wireless sensor [1] buoy technologies. Buoys equipped with multiple wireless sensors are placed on the sea to continuously collect data from the ocean, and then transfer the data to the control center for processing through the relevant equipment. In general, data measured by buoys equipped with wireless

sensors have multiple dimensions, including latitude and longitude, temperature, pressure, and salinity, etc. As buoys equipped with wireless sensors are in an open environment and highly independent, they are more susceptible to the impact of natural environment or malicious attacks, resulting in large errors in the monitoring data directly obtained by wireless sensors. In some target sea areas, there will also be insufficient monitoring buoys, resulting in missing data.

At present, there are several commonly used methods to complete missing data at home and abroad, including matrix completion [2], regression prediction based on BP (back propagation) neural network [3], linear regression prediction based on principal component analysis, and tensor completion based on Tucker decomposition, etc. When there are no restrictions on matrix completion, the solution is infinite and unsolvable. It can be seen from [4] that low rank is a prerequisite for good performance of matrix completion. In addition, matrix completion is applicable to low-dimensional data but cannot handle high-dimensional data. The regression prediction based on the BP neural network depends on the number of layers of the network, the number of neurons in each layer, and the initial value. During the training process, the calculations are complicated and the efficiency is low. Linear regression prediction based on principal component analysis is simple and easy to implement, but its accuracy is low, which may cause prediction failure. Tensor completion based on Tucker decomposition extracts the underlying structure between elements, and then extracts the principal components to complete the missing data, but it will lose some high-dimensional information, resulting in a large error.

In this paper, the existing buoy monitoring data is constructed into a tensor model. Based on tensor mode- n rank related knowledge, we combine the alternating direction method of multipliers (ADMM) framework with tensor Tucker decomposition and introduce the Tucker singular value operator for the missing data complement to obtain the finally optimized ocean monitoring data. During the experiment, the error between the data after the final completion and the original data was measured using MATLAB R2017a software to verify the effectiveness of our proposed method. For comparison, the linear regression prediction algorithm based on principal component analysis and the tensor completion algorithm based on Tucker decomposition are considered.

2 Linear Regression Based on Principal Component Analysis

2.1 Principal Component Analysis

The essence of multivariate principal component analysis is a statistical method for reducing dimensions. It converts multiple relevant variables of the research object into a set of linearly uncorrelated variables through orthogonal transformation called principal component. The original variable can be replaced by a linear combination of this group of principal components.

The principal component analysis method is to transform the covariance matrix of the original random variable into a diagonal matrix in algebra. In geometry, it can be regarded as transforming the original coordinate system into a new orthogonal coordinate system that points to the orthogonal direction with the greatest variability in the sample points, and then reduce the dimension of multidimensional variables.

Let $X = (X_1, X_2, \dots, X_P)^T$ be a P-dimensional random variable, and its principal component Z_i should meet the following conditions.

- (1) $Z_i = e_i^T X, e_i^T e_i = 1$, Where e_i is a P*1-dimensional eigenvector.
- (2) $Z_i(i = 1, 2, \dots, k, k \leq P)$ are not related.
- (3) The value of e_i makes the variance of Z_i as large as possible.

Corresponding to the eigenvector e_i , the eigenvalue is $\lambda_i, \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_P \geq 0$, thus the cumulative contribution rate of the i-th principal component is defined as

$$\gamma_i = \frac{\sum_{k=1}^i \lambda_k}{\sum_{k=1}^P \lambda_k} \tag{1}$$

If the cumulative contribution rate of the first i factors exceeds the set threshold, the first i factors are selected as the principal components. At this time, the number of factors is reduced from P to i , and they are not related to each other.

2.2 Linear Regression Prediction

The expression of the classical linear regression analysis prediction method [5] is

$$Y = f(B, Z) \tag{2}$$

where B is the parameter variable of the prediction model, Z is the principal component independent variable obtained from Sect. 2.1, and Y is the dependent variable that needs to be predicted.

In the prediction process, the fitting value at time t is set to \hat{Y}_t and the difference between the actual value Y_t and the fitting value \hat{Y}_t is

$$\begin{aligned} v_t &= Y_t - \hat{Y}_t = Y_t - f(B, Z_t), t = 1, 2, \dots, n \\ \min Q &= \sum_{t=1}^n v_t^2 \end{aligned} \tag{3}$$

For linear regression prediction, the expression is shown in (4).

$$Y = BZ = b_0 + b_1 z_1 + \dots + b_n z_n \tag{4}$$

where $B = [b_0, b_1, \dots, b_n], Z = [z_1, z_2, \dots, z_n]$. The specific algorithm is as follows and the final calculation error expression of the algorithm is

$$RSE = 10 * \log_{10} \left(\left\| \hat{Y} - X \right\| / \|X\| \right), dB \tag{5}$$

Algorithm 1 Linear regression prediction based on principal component analysis (Linear-PCR)

Input: Initial data X and principal component threshold ε

1. Standardize X .
2. Remove the rows that contain missing values in the initial data X to form new data X' .
3. Perform principal component analysis on X' , find feature values and feature vectors, apply equation (1) to make γ_i greater than threshold ε , and finally obtain principal component Z .
4. The regression analysis method is used to regress the dependent variable on the principal component Z , and the regression model shown in equation (4) is obtained.
5. Substituting the dependent variable of missing values into the model to obtain standardized prediction data Y .
6. Inverse standardization of Y .
7. Compute the RSE (Relative square error).

Output: Predictive value Y and the RSE

3 Tensor Decomposition Related Knowledge

Tensors are mathematical tools for studying high-dimensional data, and can be seen as a higher-order generalization of one-dimensional vectors and two-dimensional matrices. In the era of information technology explosion today, more and more attention is paid to it. It makes great contribution to several research fields such as high-dimensional signal processing, data mining and machine learning.

3.1 Tensor Representation

Tensor in this paper refers to a multi-dimensional array, in which one-dimensional vectors and two-dimensional matrices belong to special tensors. The different dimensions of a tensor are called the tensors' mode. The tensor model constructed in this paper is a three-dimensional form. When it is expressed as a vector, the mode-1 vector is a column vector, the mode-2 vector is a row vector, and the mode-3 vector is a depth vector. When it is expressed as a matrix, the slices corresponding to a certain dimension of the tensor are expanded and arranged into a long matrix. The matrix representation on the n-mode is called mode-n matrix expansion of the tensor [6].

3.2 Tensor Rank and Tucker Decomposition

There are multiple definitions of the rank of tensors, which lead to a variety of low rank tensor decomposition methods. The definition of the rank of the tensor is mainly divided into CP rank and mode-n rank, and the low rank tensor decomposition methods determined therefrom are CP decomposition and Tucker decomposition. This paper uses mode-n rank and Tucker decomposition, so only these two methods are described.

The mode-n rank of a tensor refers to the number of maximally linearly independent groups of its expansion matrix in the mode-n form, which is denoted as $rank_n(\alpha)$.

Tucker decomposition, also known as high-order singular value decomposition (HOSVD), is essentially a high-order principal component analysis [7]. For a known tensor $\alpha \in R^{I_1 \times I_2 \times \dots \times I_N}$, the mathematical expression of its tucker decomposition is

$$\begin{aligned} \alpha &= G \times_1 U^{(1)} \times_2 U^{(2)} \times_3 \dots \times_N U^{(N)} \\ \alpha &= \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \dots \sum_{j_N=1}^{J_N} g_{j_1 j_2 \dots j_N} (u_{j_1}^{(1)} \circ u_{j_2}^{(2)} \circ \dots \circ u_{j_N}^{(N)}) \end{aligned} \tag{6}$$

where $G \in R^{J_1 \times J_2 \times \dots \times J_N}$ is the core tensor, $U^{(n)} \in R^{I_n \times J_n}$ indicates that the tensor $\alpha \in R^{I_1 \times I_2 \times \dots \times I_N}$ corresponds to the n th order orthogonal transformation matrix (also called factor matrix), and the core tensor G is calculated as follows.

$$G = \alpha \times_1 U^{(1)T} \times_2 U^{(2)T} \times_3 \dots \times_N U^{(N)T} \tag{7}$$

In the above Eq. (6) and (7), \times_n represents the n -mode product operator, and its expression is

$$\begin{aligned} b_{i_1, i_2, \dots, i_{n-1}, j, i_{n+1}, \dots, i_N} &= \sum_{i_n=1}^{I_n} a_{i_1, i_2, \dots, i_n} c_{j, i_n} \\ \beta &= \alpha \times_n C, \alpha \in R^{I_1 \times I_2 \times \dots \times I_N}, C \in R^{J \times I_n} \end{aligned} \tag{8}$$

$A \circ B$ is the outer product, and its expression is

$$[A \circ B]_{m,n,p,q} = a_{m,n} b_{p,q}, A \in R^{M \times N}, B \in R^{P \times Q} \tag{9}$$

3.3 Tensor Completion

The essence of tensor completion is to solve the following convex optimization problem [8].

$$\begin{aligned} \min \sum_{i=1}^n \mu_i \text{rank}(X_i) &\rightarrow \min \sum_{i=1}^n \mu_i \|X_i\|_* \\ \text{subject to } P_\Omega(\chi) &= P_\Omega(\alpha) \end{aligned} \tag{10}$$

where $\|X_i\|$ represents Frobenius norm, which is defined as

$$\|X\| = \left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N}^* \right)^{1/2} = \langle X, X \rangle^{1/2}, X \in R^{I_1 \times I_2 \times \dots \times I_N} \tag{11}$$

χ is the intermediate tensor variable, Ω refers to the set of non-missing values in tensor α . X_i represents the mode- n expansion matrix of χ . μ_i is a constant set in advance, and needs to satisfy Eq. (12) and $\mu_i = 1/3$ in this paper.

$$\mu_i \geq 0, \sum_{i=1}^n \mu_i = 1 \tag{12}$$

Based on tensor tucker decomposition to complete the data, the principle is to use an iterative method to decompose the original tensor α with the missing data, and then reorganize it into a new tensor $\hat{\alpha}$ to achieve the purpose of tensor completion. The specific algorithm is as follows.

Algorithm 2 Tensor completion based on Tucker decomposition (Tucker completion)

Input: The original tensor α , rank k_1, k_2, \dots, k_N and the maximum number of iterations Max .

1. Set 0, 1 tensor S and meet

$$S_{i,j,k} = \begin{cases} 1 & (i, j, k) \in \Omega \\ 0 & (i, j, k) \notin \Omega \end{cases} \tag{13}$$

2. Set initial χ_0 and satisfy

$$\chi_{i,j,k} = \begin{cases} a_{i,j,k} & (i, j, k) \in \Omega \\ 0 & (i, j, k) \notin \Omega \end{cases}, \chi_0 = S * \alpha \tag{14}$$

3. Repeat Max times

4. for $n=1:N$

$$X(n) = U^{(n)} \sum (n) V^{(n)T} \tag{15}$$

5. end

6. Update core tensor G :

$$G = \chi_0 \times_1 U^{(1)T} \times_2 U^{(2)T} \times_3 \dots \times_N U^{(N)T} \tag{16}$$

7. Update intermediate tensor χ_m :

$$\chi_m = G \times_1 U^{(1)} \times_2 \dots \times_N U^{(N)} \tag{17}$$

8. Update estimated tensor α :

$$\alpha = \chi_m * (1 - S) \tag{18}$$

9. Compute the RSE.

Output: Estimated tensor $\hat{\alpha}$ and the RSE

The above Algorithm 2 removes some minor components in the process of completing data, which reduces its performance when completing data with more detailed features. Therefore, this paper introduces a tensor completion method based on mode-n rank, and on this basis, introduces the tensor Tucker singular value operator. The

essence of performing tensor completion based on mode- n rank is to minimize the mode- n rank $rank_n(\alpha)$ of tensor α . The singular value of the tensor expansion matrix is used to update the intermediate variable χ for iterative estimation, and then the missing value of the tensor is completed. In the iterative process, the framework used is the alternating direction multiplier method (ADMM) [9], which combines with the relevant characteristics of tensor Tucker decomposition. The tensor tucker singular value operator $TD_\tau(\bullet)$ is introduced and is defined as follows.

$$\chi \approx G(1 : r_1, 1 : r_2, \dots, 1 : r_N) \times_1 U^{(1)}(:, 1 : r_1) \times_2 \cdots \times_N U^{(N)}(:, 1 : r_N) \quad (19)$$

where $r_i = \text{sum}(\text{diag}(\max(\sigma_i - \tau, 0)))$ and $TD_\tau(\bullet)$ meets the following theorem [10].

$$TD_\tau(y) = \arg \min_{\chi} \frac{1}{2} \|\chi - y\|_F^2 + \tau \|\chi\|_* \quad (20)$$

Similar to the matrix singular value operator $TD_\tau(\bullet)$ in [10], it can be seen that the tensor Tucker singular value operator is the optimal solution to the tensor completion convex optimization problem, and because it uses high-dimensional tensor data structure, it can improve the utilization of tensor high-dimensional structure.

In this algorithm, the equation of $TD_\tau(\bullet)$ can be rewritten as

$$\chi = \arg \min_{\chi} \frac{\mu}{2} \left\| \chi - \left(\alpha + \frac{1}{\mu} y \right) \right\|_F^2 + \tau \|\chi\|_* \quad (21)$$

From Eq. (20), the intermediate variable χ can be solved as

$$\chi = TD_{\frac{\mu}{2}}\left(\hat{\alpha} + \frac{y}{\mu}\right) \quad (22)$$

The values of estimated tensor $\hat{\alpha}$ and auxiliary variable y_m can be calculated from it. The specific algorithm is as follows.

Algorithm 3 Tensor completion based on mode-n rank and Tucker operator (Mode-n and Tucker completion)

Input: The original tensor α , the maximum number of iterations Max and parameter ρ

1. Set 0, 1 tensor S and meet equation (13).
2. Set initial χ_0 and satisfy equation (14).
3. Set auxiliary variable $y_1, y_2, y_3 \in R^{n_1 \times n_2 \times n_3}$ and the elements are all “0”.
4. for 1: Max
5. Update intermediate tensor χ :

$$\chi_m = TD_\tau(\chi_{m-1} + \frac{y_m}{\rho}), m = 1, 2, 3 \tag{23}$$

6. Update estimated tensor α :

$$\alpha = (1 - S) * \left\{ \frac{1}{3} * \sum_{m=1}^3 (\chi_m - \frac{y_m}{\rho}) \right\} + S * \chi_m \tag{24}$$

7. Update auxiliary variable y_m :

$$y_m = y_m - \rho * (\chi_m - \alpha), m = 1, 2, 3 \tag{25}$$

8. end
9. Compute the RSE.

Output: Estimated tensor α and RSE

Among them $\tau = \frac{1}{\rho}$.

4 Experimental Verification

This experiment selects the relevant data of the Argo Real-time Data Center of China (taken from some data in the number 0469). The dimensions of selected data are latitude and longitude, pressure, temperature and salinity. Each dimension has 12 data, and there are 60 in total. This data is constructed as a three-dimensional tensor $\alpha \in R^{6 \times 2 \times 5}$. Among them, the first 24 data belong to the latitude and longitude dimensions, and the last 36 data belong to the dimensions of pressure, temperature and salinity. Randomly select 1–20 of these 36 data (step size is 1) as null data “0”, use it to simulate missing data or tamper with data, that is, construct tensor $S * \alpha$ and then complete it with tensor. Perform 10 experiments respectively to calculate the final average error.

The initial data of the experiment as shown in Table 1 below.

Table 1. Initial data of longitude, latitude, pressure, temperature and salinity

Longitude	110.119	110.114	110.125	109.944	109.849	109.786
	109.637	109.403	109.128	108.813	108.534	108.334
Latitude	17.810	17.837	17.865	18.193	18.437	18.610
	18.707	18.798	18.846	18.814	18.750	18.641
Pressure (DBAR)	1025.200	1063.900	1064.100	1064.000	1064.900	1065.500
	1064.800	1065.000	1064.700	1064.500	1065.300	1064.800
Temperature (DEG C)	5.022	4.913	4.847	4.845	4.957	4.993
	5.012	5.031	4.941	4.938	4.901	4.921
Salinity (PSU)	34.638	34.636	34.638	34.638	34.638	34.634
	34.636	34.632	34.637	34.638	34.637	34.638

4.1 Convergence Experiment

During the experiment, randomly set the S tensor of 6 empty data (two in each of the three dimensions), and then complete the tensor of the missing data, and explore the convergence of Algorithm 3.

The missing position of the tensor data and the RSE of the data completion using Algorithm 3 are shown in Table 2, and the convergence of RSE is shown in Fig. 1.

Table 2. Random missing positions and completion data

	Missing position	Completion value
Pressure (DBAR)	3	992.300
	5	992.900
Temperature (DEG C)	9	4.972
	10	4.972
Salinity (PSU)	1	34.574
	5	34.608

In this experiment, there are 6 missing data (two in each of the three dimensions), and ADMM iteration is performed 1000 times. As can be seen from Fig. 1, in the iterative process, the RSE of data after tensor completion gradually converges, reaching stability after about 350 iterations, indicating that Algorithm 3 has convergences well for tensor data completion.

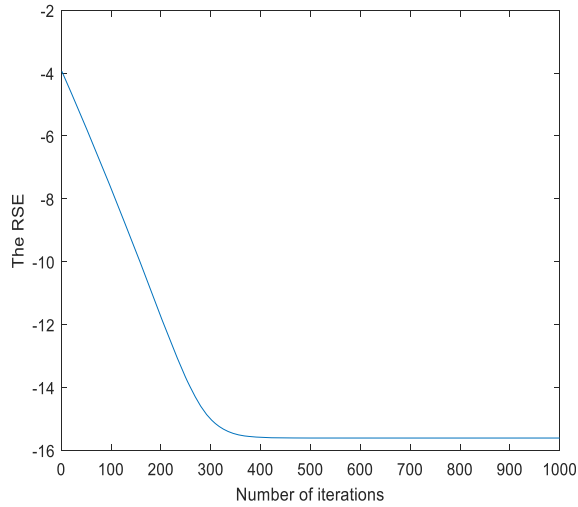


Fig. 1. The RSE during the tensor completion iteration

4.2 Comparative Experiment of Data Completion Effect

This experiment also applies three algorithms to complete the missing data tensor $S. * \alpha$. During the experiment, the missing values are randomly set. In order to make the experimental results more credible, the number of missing data ranges from 1 to 20. Each time a new missing value is added, the last missing value is guaranteed to be unchanged, thereby simulating the effect of initial data tensor completion under different missing degrees. The completion process of each proportion value is iterated 50 times, and finally find the average. The experimental results are shown in Fig. 2.

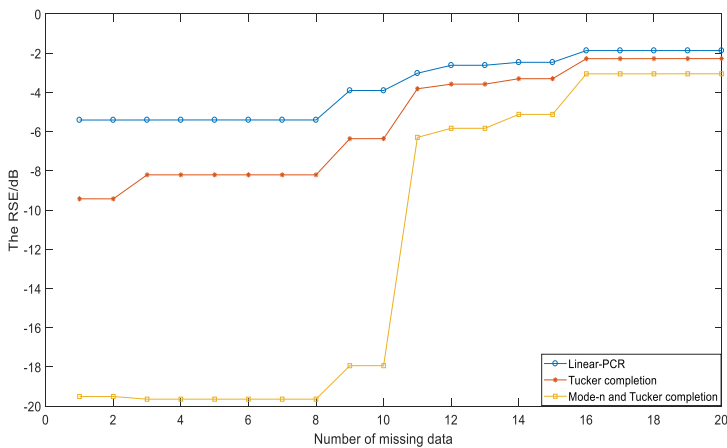


Fig. 2. Comparison of the completion effects of three tensor completion algorithms

It can be seen from Fig. 2 that as the value of “0” in the initial tensor data (that is, the missing value or tampered value in the ocean monitoring data) increases, the RSE after completing the data through the tensor also becomes larger and larger. Overall, compared with the other two algorithms, the tensor completion Algorithm 3 proposed in this paper has a relatively ideal effect on the completion of the monitoring data of marine wireless sensors. In addition, in the process of missing data from 1 to 10, data completion effect is better.

5 Conclusion

In this paper, the marine wireless sensor monitoring data is constructed into a three-dimensional tensor. Using the high-dimensional characteristics of the tensor, the tensor Tucker decomposition operator is introduced on the basis of the traditional tensor mode-n rank completion algorithm. After compared with data completion based on tensor Tucker decomposition and linear regression prediction based on principal component analysis, it is proved that this proposed algorithm can effectively improve the accuracy of tensor completion data, and its convergence is well, which is of great significance for the optimization of monitoring data of marine wireless sensors.

References

1. Wang, J., Zhou, W., Shen, Y.: Research on marine environment monitoring system based on wireless sensor network. *Comput. Eng. Des.* **2008**(13), 3334–3337 (2008)
2. Shi, J., Li, X.: Meteorological data estimation based on matrix completion. *Meteorol. Sci. Technol.* **047**(003), 420–425 (2019)
3. Shen, F., Guo, J.: Data fusion method based on BP neural network. *Autom. Instrum.* **2005** (05), 66–68+71 (2005)
4. Candes, E.J., Recht, B.: Exact matrix completion via convex optimization. *Found. Comput. Math.* **9**(6), 717–772 (2009)
5. Niu, D., Cao, S., Zhao, L., Zhang, W.: *Power Load Forecasting Technology and its Application*. China Electric Power Press, Beijing (1998)
6. Cichocki, A., Mandic, D., Lathauwer, L.D.: Tensor decompositions for signal processing application: from two-way to multiway component analysis. *IEEE Signal Process. Mag.* **32** (2), 145–163 (2015)
7. Zeng, K., He, L., Yang, X.: Support for higher-order tensor machines based on multilinear principal component analysis. *J. Nanjing Univ. Nat. Sci.* **2014**(02), 113–121 (2014)
8. Wang, L.: *Data Collection and Recovery Based on Sparse Representation Theory*. Hunan University, Changsha (2017)
9. Boyd, S., Parikh, N., Chu, E.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2010)
10. Zhou, J.: *Research on Data Recovery Method Based on Tensor*. Shaanxi Normal University, Xi'an (2014)