Recovering Incomplete Data using Tucker Model for Tensor with Low-n-rank

Thao Nguyen Thieu, Hyung-Jeong Yang, Tien Duong Vu
Department of Electronics and Computer Engineering,
Chonnam National University, 500-757 Gwangju, South Korea

Sun-Hee Kim
Department of Brain and Cognitive Engineering,
Korea University, Seoul, South Korea

ABSTRACT

Tensor with missing or incomplete values is a ubiquitous problem in various fields such as biomedical signal processing, image processing, and social network analysis. In this paper, we considered how to reconstruct a dataset with missing values by using tensor form which is called tensor completion process. We applied Tucker factorization to solve tensor completion which was built base on optimization problem. We formulated the optimization objective function using components of Tucker model after decomposing. The weighted least square matric contained only known values of the tensor with low rank in its modes. A first order optimization method, namely Nonlinear Conjugated Gradient, was applied to solve the optimization problem. We demonstrated the effectiveness of the proposed method in EEG signals with about 70% missing entries compared to other algorithms. The relative error was proposed to compare the difference between original tensor and the process output.

Key words: *Tucker Decomposition, Missing Value and Tensor Completion.*

1. INTRODUCTION

A tensor is studied as a definition of a data in multi-dimensional space. It may be regarded as the higher generation of vector and matrix. More formally, an N-way or Nth-order tensor is an element of the tensor product of N vector spaces [1]. Tensor analysis is an approach that has many applications in various fields such as physics, mechanics, and information sciences. Tensor Completion and Tensor Decomposition are two most important application models of tensor [2], [3]. Tensor Decomposition is used to divide a tensor into many smaller parts but still contain the vital information of the original tensor, so that the process will become faster and simpler. Tensor Completion is a method to recover missing data of a tensor. As mathematical definition, tensor completion methods can reconstruct a tensor from a smaller part of its entries.

In recent years, recovering a tensor with missing entries is an open problem, especially tensor with low rank in its mode. The main idea of low rank was first introduced to solve the problems related to the lost information in a matrix that has low rank [4]. Lost information or also called "missing values" usually occurs when collecting data as recording signals, taking a picture in a bad condition or when transforming data from the raw form into the more logical structure. Generally, for more understandable, low-rank matrix completion is a method that can recover a matrix from a subset of its entries. Low-rank matrix is a very useful method for signal processing, computer vision [2], biomedical and various other areas [5]-[8]. However, a real dataset is presented not only in second-order form as a matrix but also in higher-order form as a tensor. Therefore, recovery multidimensional missing data or low-rank tensor completion is being focused on studying more and more commonly.

The minimization problem of low-n-rank tensor recovery is a difficult non-convex problem, because *rank* is nonconvex [5]. Therefore, Nuclear Norm Approximation (NNA) is approached to relax low-n-rank tensor completion problem into the convex problem. NNA is applied by replacing the rank function of tensor with nuclear norm. Therefore, the tensor completion problem become convexity and we can solve it easier than former non-convex one. Based on this idea, most of previous approaches related to tensor completion used NNA for their studies. The idea of NNA is to compute the sum of singular values of that matrix. NNA promotes a low-rank solution, which is a key idea that can be applied to many applications such as recommender system, dimension reduction in multivariate regression, and multi-task learning [2]. Generally, tensor completion is developed based on optimization idea: minimizing or maximizing a function to

obtain the optimal solution. The optimal solution of the optimization problem is the reconstructed tensor.

In this paper, we applied Tensor Completion on missing values in EEG signals. In detail, we propose a method using Tucker factorization for recovering lost EEG data. First, we use Tucker factorization to build an objective function for the optimization problem. Then, we apply Nonlinear Conjugate Gradient algorithm [9] to find the optimal solution for the reconstructed EEG signals.

Our contributions are summarized as: (i) we build a method for Tucker factorization of missing data by using the weight least square function, (ii) the proposed method applied successfully not only on element-missing level but also channel-missing level. (iii) We proposed to use approximately true rank of tensor, by rank of matricization form along to each dimension, in order to relax unknown true rank constraint, which is an important constraint in non-convex methods such as Tucker decomposition. Our work is also the first attempt on element-missing level in EEG data and shows high efficiency with relative error less than 5%. On both levels of missing entries, the proposed method provides the best performance in the aspect of accuracy when comparing it to the other methods in literature.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works. Section 3 introduces the proposed method. Experimental results are shown in Section 4 to demonstrate the performance of the proposed algorithm and the final conclusions are presented in Section 5.

2. RELATED WORK

Tensor completion is an extension of matrix completion which is usually used to recover data in a matrix form. Matrix which has low rank becomes the subject for simple recovery [7], [8]. Low-rank matrix completion arises in many experiments because it can recover a matrix from several known entries. The development of processing and storage data techniques has led to multidimensional forms of data, so we need recovery techniques on higher level, rather than matrix, to deal with multidimensional data, preferably formed as tensor. Therefore, many literatures have been investigating tensor recovery, and applied tensor completion widely in areas such as mathematical modelling, data compression, image recovering [5], [6], [10]-[12]. Fig. 1 shows an example of a third-order tensor with missing entries.
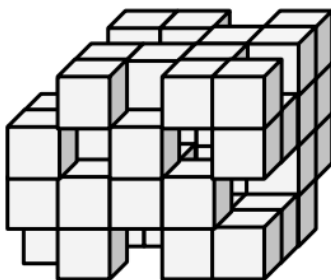

Fig. 1. A third-order tensor with missing entries

The optimization problem is non-convex since function *rank* is non-convex. Therefore, trace norm is used to approximate rank of matrices, aims to relax the object function to convex optimization problem. In essence, the trace norm of a tensor is a convex combination of the trace norms of all matrices unfolded along each mode.

Similar approaches were used in Gandy [5] and Tan [6], Liu et al. [11] showed that the above problem still hard to solve because the matrix nuclear norm terms are interdependent. Therefore, Liu et al. [11] proposed three algorithms to tackle this problem, namely SiLRTC, FaLRTC and HaLRTC.

The SiLRTC algorithm employs a relaxation technique to separate the dependent relationships and uses the block coordinate descent (BCD) method to achieve a globally optimal solution. The SiLRCT is simple and can be adjusted trade-off between running time and accuracy through turning of a parameter theta. The FaLRTC algorithm utilizes a smoothing scheme to transform the original non-smooth problem into a smooth problem. The HaLRTC algorithm applies the alternating direction method of multipliers (ADMM) [13]. These three methods are more accurate and robust than heuristic approaches such as Tucker, PARAFAC and SVD. The FaLRTC and HaLRTC are more significantly efficient than SiLRTC about running time, when they have same accuracy. FaLRTC is more efficient to a low accuracy solution and HaLRTC is preferred if a high accuracy solution is desired.

However, there are some problems with these approaches. Generally, NNA is the sum of singular values of a matrix. Therefore, when applying these approaches, every iteration of the SVD needs to be computed. This makes these algorithms slower and the complexity of problem will be increased. In addition, these methods were just proven effectively in graphic, video processing and reflected data, no guarantee that they might be good for EEG recovery.

Acar et al. [10] also built a model to recover a low rank tensor of EEG with missing data but using different approach to recover a low rank tensor with missing values to avoid using SVD. In [10], an objective function formed from the components is proposed after PARAFAC decomposing and its weight least matrix which just contains 0 and 1. The value of this matrix is 1 if it is known values. Otherwise, it is 0. The objective function used in their paper for third-order tensor is sum of squares error between initial tensor and reconstructed elements built from factor matrices after CANDECOMP/ PARAFAC decomposition.

In fact, Tucker Decomposition and PARAFAC are two major tensor decomposition methods [12]. As in [3], Tucker is considered as a more flexible model than PARAFAC. Therefore, based on Tucker's studies, Marko [12] developed another method for recovering a tensor. He mentioned that his proposed method can recover underlying low-n-rank tensor even when the true tensor ranks are unknown. His approach required an assumption that tensor has low-rank on every mode. When true rank is overestimated, Marko's method provided an accurate reconstructed tensor, and it demonstrated on some stimulation datasets, 3D images and protein dataset. However, one problem is overestimated rank setting not same on his experiments and if estimated rank is assumed equal to full rank, his proposed method became worse much than nuclear norm

based methods. This is completely realistic, because when we didn't know true rank, the most certain assumption is assigning full rank to estimated rank. In case of underestimated rank, this method resulted in approximately reconstructed tensor, but author's experiment just was done in several pictures made artificially low rank. Therefore, it didn't guarantee good performance in generally practical problem. Moreover, this paper concentrated on numerical demonstration only and there are no theoretical guarantees for proposed method, since it is based on non-convex optimization.

From impressive experiment results of [12], we proposed using Tucker decomposition as a non-convex method on EEG, similar to the above mentioned papers [10], [12]. However, the rank of tensor used in this paper will be computed by a detail equation to obtain the best one.

3. PROPOSED METHOD

3.1 Mathematical notations

In this paper, all of examples of tensor is denoted by tensor $\mathcal{X}$. The order of a tensor is the number of dimensions, known as ways or modes. *Fibers* are the higher order analogue of matrix rows and columns, defined by fixing every index but one. Third-order tensors have column, row, and tube fibers, denoted by $x_{:jk}, x_{i:k}$ and $x_{ij:}$, respectively. Fig. 2 shows an illustration of fibers in a third-order tensor.
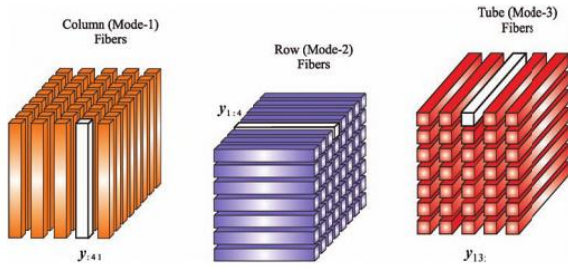

Fig. 2. Fibers of a third-order tensor

Slices are two-dimensional sections of a tensor, defined by fixing all but two indices. Third-order tensors $\mathcal{X}$ include three slices: horizontal, lateral and frontal slices denoted by $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$ and $\mathbf{X}_{::k}$, respectively. The detail description of the symbols used in this paper is also showed in Table 1.

Table 1. Some mathematical symbols and its definition used in this paper

Symbols	Definitions
a	Vectors
A	Matrices
$\mathcal{X}$	Higher-order tensors
a	Scalars
a_i	The ith entry of a vector **a**
a_{ij}	Element (i, j) of a matrix **A**
x_{ijk}	Element (i, j, k) of a tensor $\mathcal{X}$
$\mathbf{X}_{(n)}$	Mode-n matricization of a tensor $\mathcal{X}$

The *norm* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is the square root of the sum of the squares of all its elements, i.e.

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2} \quad (1)$$

Furthermore, based on the definition of the mode-n matricization, we can calculate the n-rank of tensor X. The n-rank of a N-order tensor is the tuple of the ranks of the mode-n matricizations as follow:

$$n\text{-rank}(\mathcal{X}) = (\text{rank } \mathbf{X}_{(1)}, \text{ rank } \mathbf{X}_{(2)}, \dots, \text{rank } \mathbf{X}_{(N)}) \quad (2)$$

A tensor can be multiplied together or with a matrix or with a vector. The n-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$, $1 \le n \le N$ is denoted by $\mathcal{X} \times_n \mathbf{U}$ which is of size $I_1 \times I_2 \times \dots \times J \times I_{k+1} \times \dots \times I_N$. Therefore, we have:

$$(\mathcal{X} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_N} u_{j i_n} \quad (3)$$

3.2 Tucker decomposition and Nonlinear Conjugate Gradient

Tucker decomposition is an application of tensor considering the higher order singular value decomposition. Tucker decomposes a tensor into the multiple of another tensor $\mathcal{G}$, core tensor, and some matrices $\mathbf{A}_n$ along to each mode of tensor. Therefore, Tucker decomposition can be indicated as following:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \dots \times_N \mathbf{A}_N = [\![\mathcal{G}; \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N]\!] \quad (4)$$

Let assume that the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ given in this paper is low-rank tensor. Ranks of a tensor can be chose as a lowest rank or chose randomly. However, in these cases, we can assume that we have some approximations of true ranks.

$\mathcal{W}$ is denoted as a tensor with missing observations and $\mathcal{W}$ has the same size as $\mathcal{X}$. Tensor $\mathcal{W}$ can be defined as:

$$w_{i_1 i_2 \dots i_N} = \begin{cases} 1, & \text{if } x_{i_1 i_2 \dots i_N} \in \Omega \\ 0, & \text{if } x_{i_1 i_2 \dots i_N} \in \Omega^C \end{cases} \quad (5)$$

where, Ω is a subset of $\{1, \dots, I_1\} \times \{1, \dots, I_2\} \times \dots \times \{1, \dots, I_N\}$, including positions of known entries of tensor and Ω^C is its complement.

Following Tucker model, tensor $\mathcal{X}$, which has rank of each mode be $\mathbf{R}_n$ $(n = 1, \dots, N)$, can be factorized as:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \dots \times_N \mathbf{A}_N + \text{error} \quad (6)$$

where $\mathcal{G}$ is $r_1 \times r_2 \times \dots \times r_N$ core tensor and $\mathbf{A}_n \in \mathbb{R}^{I_n \times R_n}$ are factor matrices along each mode of tensor.

Our goal is to find core tensor $\mathcal{G}$ and matrices $\mathbf{A}_n$ $(n = 1, \dots, N)$ by minimizing the objective function which will be defined as below. Most of the optimization problems are

computed based on an objective function and its gradient [13]. As the definition of Tucker decomposition, when decomposing a tensor in $\mathbb{R}^n$, we will obtain a core tensor and some factor matrices along to each mode. However, in reality, the result of the mode-n product of the core tensor and n factor matrices is not same as the original tensor, just an approximate reconstructed tensor. This difference between reconstructed tensor and initial tensor is formed as an "error tensor".

The N -way objective function for the optimization problem is defined by Marko [12] as (7).

$$
\begin{aligned}
&f_{\mathcal{W}}(\mathcal{G}, \mathbf{A_1}, \mathbf{A_2}, \dots, \mathbf{A_N}) \\
&= \|\mathcal{W} * (\mathcal{X} - \mathcal{G} \times_1 \mathbf{A_1} \times_2 \mathbf{A_2} \times_3 \dots \times_N \mathbf{A_N})\|_F^2 \\
&= \sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \dots \sum_{i_N}^{I_N} w_{i_1 i_2 \dots i_N}^2 \left\{ x_{i_1 i_2 \dots i_N}^2 \right. \\
&\quad - 2 x_{i_1 i_2 \dots i_N} \sum_{r_1}^{R_1} \sum_{r_2}^{R_2} \dots \sum_{r_N}^{R_N} g_{r_1 r_2 \dots r_N} \prod_{n=1}^{N} a_{i_n r_n}^{(n)} \\
&\quad \left. + \left(\sum_{r_1}^{R_1} \sum_{r_2}^{R_2} \dots \sum_{r_N}^{R_N} g_{r_1 r_2 \dots r_N} \prod_{n=1}^{N} a_{i_n r_n}^{(n)} \right)^2 \right\}
\end{aligned}
\tag{7}
$$

The gradient can be obtained by computing the partial derivatives of objective function f_W to each element of the core tensor and the factor matrices after decomposing. The gradient equation can be rewritten in the matrix notation case as follows:

$$
\begin{aligned}
\frac{\partial f_W}{\partial \mathbf{A_n}} = 2 &\left\{ [\mathcal{W} * (\mathcal{G} \times_1 \mathbf{A_1} \times_2 \mathbf{A_2} \times_3 \dots \times_N \mathbf{A_N} - \mathcal{X})]_{(n)} \cdot \right. \\
&\left. [(\mathcal{G} \times_1 \mathbf{A_1} \times_2 \dots \times_{n-1} \mathbf{A_{n-1}} \times_{n+1} \mathbf{A_{n+1}} \times_{n+2} \dots \times_N \mathbf{A_N})_{(n)}]^T \right\}
\end{aligned}
\tag{8}
$$

Gradient of the objective function on the core tensor is:

$$
\begin{aligned}
\frac{\partial f_W}{\partial \mathcal{G}} = 2 \{ &\mathcal{W} * (\mathcal{G} \times_1 \mathbf{A_1} \times_2 \mathbf{A_2} \times_3 \dots \times_N \mathbf{A_N} - \mathcal{X}) \\
&\times_1 \mathbf{A_1}^T \times_2 \dots \times_N \mathbf{A_N}^T \}
\end{aligned}
\tag{9}
$$

In this paper, we used Nonlinear Conjugate Gradient (NCG) because of its speed. For applying NCG method for our problem, we used Poblano Toolbox [14].

We proposed a method to recover a tensor with low rank in its mode by using Tucker model and a first-order optimization problem. The outputs of Tucker model will be combined with a least square matrix to form the objective function for the optimization problem. The output of the optimization problem is the recovery of the tensor with missing entries.

4. EXPERIMENTS

In this paper, we conducted experiments on two datasets. The first experiment is done with element-missing level on epilepsy dataset of four dogs and the other one is done with channel (or fiber)-missing level on a EEG dataset collected from a stimulation experiment on human.

4.1 Experiment on element-missing level data

In this experiment, we use an EEG dataset recorded from four dogs with naturally occurring epilepsy using an ambulatory monitoring system. The recording process was acquired on 16 channels. Data are recorded continuously at a sampling frequency of 400 Hz and referenced to the group average. This data set is open freely at the International Epilepsy Electrophysiology portal and was developed by the University of Pennsylvania and the Mayo Clinic [15].

The dataset will be set into a tensor of size 16×400×200 as channel × time-point × segments for each subject. Each segment is the signals recorded in one second. In this paper, 200 segments recorded in 200 first seconds are used. The low-n-rank of this tensor is 15×213×200 computed by equation (2).

The relative error will be estimated for determining the difference of the original data and recovered data by this formula:

$$
r_{err} = \frac{\|\widehat{\mathcal{X}} - \mathcal{X}\|_F}{\|\mathcal{X}\|_F}
\tag{10}
$$

Where $\widehat{\mathcal{X}}$ denotes the output of the proposed method, and $\mathcal{X}$ denotes the original tensor with low rank in each mode. The relative error is always nonnegative and the best possible value is 0. The core and factor matrices are initialized by Tucker algorithm in N-way Toolbox [16].

The results of proposed method are shown in Table 2. This table displays the results of the relative error of two tensors: the original tensor and the output of the proposed method. This error is computed from equation (10). The rank of core tensor is 15×213×200 as the low feasible rank of this tensor. The obtained results of four dogs are approximately less than 4% error. In other words, the similarity of the original tensor and its reconstruction using Tucker model is about 96%.

Table 2. The relative error of proposed method in different fractions of missing data

Missing data	Dog 1	Dog 2	Dog 3	Dog 4
30%	1.2%	1.69%	1.59%	1.85%
50%	1.76%	3.3%	1.59%	3.34%
70%	1.49%	4%	2.88%	3.5%

As an example of the original signals of the first dog, its missing versions and the recovered signals are shown in Fig. 3: fig. (a) shows the first EEG segment with 70% of missing values of Dog 1 and fig. 3.(b) is the recovery of (a). These figures show that the proposed method can successfully recover a signal with up to 70% missing values.

In this experiment, we also compare the effectiveness of the proposed method with the others that are from Ji Liu's paper [11]. Simple low-rank tensor completion (SiLRTC) is used as an optimization method that can be applied to a simple convex structure which can be solved by block coordinate descent. A faster version of SiLRTC is also proposed as

another comparison method, called FaLRTC. This method uses the smooth model of the convex problem to solve.

The comparison of the proposed method and two Ji Liu's methods is showed in Fig. 4 with average related errors of four dogs.
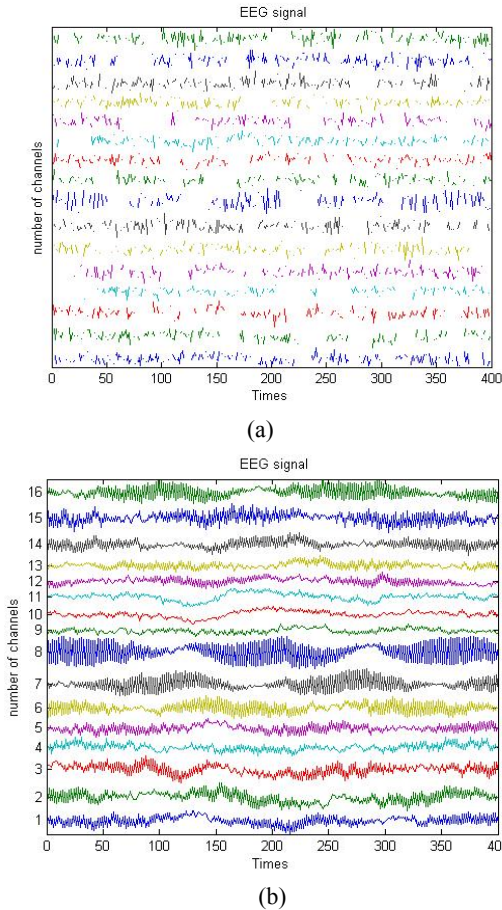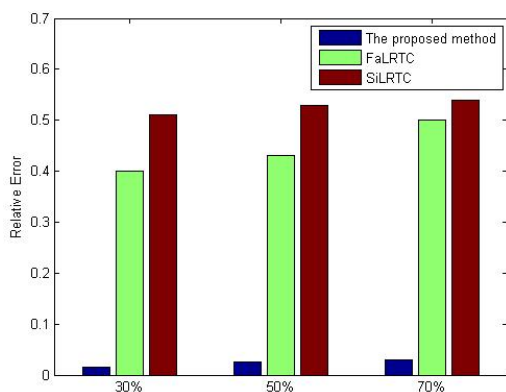

(a)

(b)

Fig. 3. The EEG segment of Dog 1

Fig. 4. Comparison of the proposed method and the others

This result shows the big difference output between these methods. These scores are computed by the relative errors for the whole tensor. The relative errors computed after applying SiLRTC and FaLRTC are close to 50%. It also shows that these two methods not good when applying for EEG data. We think

that these high errors occurred since the natural characteristics of continuous elements in EEG data absolutely different to 3-D graphic data. In EEG, the correlation of a point along to each mode (dimension) is majority, differ to the correlation of neighbor pixels (multi-dimension) as in 3-D space of graphic data.

We also studied the effect of this method in the case of some channels are disconnected. This is one of the reasons why the EEG data is lost information when collecting. To study tensor completion for this case, we remove one or more of channels of each segment. Fig. 4 shows the example results of Dog 3 for this case. Four channels: 3, 8, 10 and 16 are randomly removed. The detail scores are represented in Table 3. The proposed method still works well even though being removed a half of channels, means about 50% entries is lost.

Table 3. Relative error after removing some channels

Number of Removed Channels	Dog 1	Dog 2	Dog 3	Dog 4
1	0.0069	0.001	0.0078	0.035
4	0.0055	0.0089	0.007	0.026
8	0.0044	0.0049	0.0054	0.017

4.2 Experiment on fiber-missing level data

For the second experiment, we used same EEG dataset as Acar [11] for comparison. This dataset contains 64 channels recorded from 14 subjects during stimulation of left and right hand. Therefore, each measurement is arranged as a *channel by frequency by time*, which is 3D form of size 64×61×72. Time and frequency will be vectorized of length 4392 (i.e. 61×72); in other words, each measurement is represented by a *channel by time-frequency matrix*. Therefore, whole dataset was arranged as a *channel by time-frequency by measurements tensor* of size 64×4392×28.

(a)
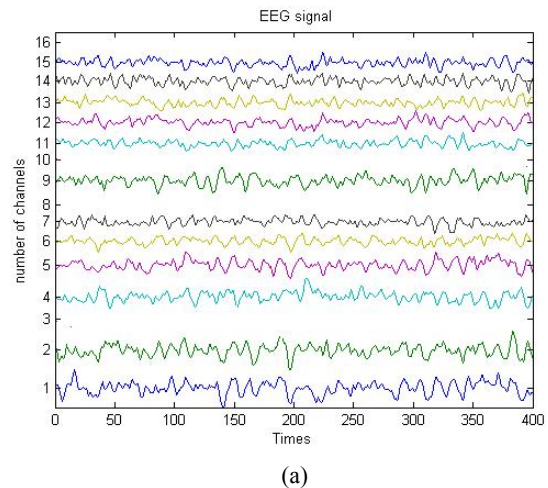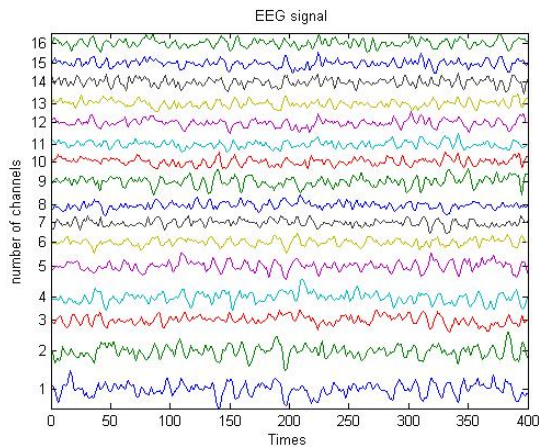
(b)

Fig. 5. Reconstruction of signals of Dog 3 after removing four channels: 3, 8, 10, 16. (a) original data before removing channels. (b) the output after applying the proposed method

The rank of this tensor computed by equation (2) is 64×71×28 [17]. Following Acar's paper [11], EEG analysis is easily lost information because one or some of channels are disconnected. This corresponds to the missing fibers in a tensor. To reflect such cases of missing data, Acar set randomly data for one or some of the current channels for each measurement to be missing, and then the PARAFAC model was applied to the above low rank to obtain the results of the proposed method. Same as Acar's work, we also randomly remove one channel or more of 64 channels (1, 10, 20 respectively) for each measurement of the whole dataset. Fig. 5 illustrate effectiveness of proposed method when applying for tensor missing four channels. It shows EEG data before and after recovering.

Relative errors compared to Acar's method are showed in Table 4. The error between two tensors: the original tensor and the reconstructed tensor is computed by formula (10). This experiment shows that the proposed method still give better results than Acar's method if removing 20 channels, i.e. about 30% entries of the data. This result also demonstrates remarkable improvement about performance of Tucker compared to PARAFAC.

Table 4. The comparison of the relative error between the proposed method and Acar's method

Number of missing channels	The proposed method	Acar's method
1	$1.19e - 4$	0.0009
10	$1.53e - 4$	0.0065
20	0.0137	0.1656

5. CONCLUSION

The aim of this paper is to focus on the problem of low-rank tensor completion used to reconstruct a tensor based on a subset of its entries. This approach is applied to recover missing values of the dataset in multidimensionality. By using Tucker model and its weight function for known entries, we can form an objective function for the optimization problem. The first-order optimization, Nonlinear Conjugate Gradient, is applied to solve the weighted least squares objective function because its speed and simplification. This paper also shows that an EEG signals in tensor form has sufficiently low-n-rank can be recovered exactly from a subset of its entries, about 70% missing entries, with less than 4% relative error. We also show the applicability of the proposed method on different datasets.

In the future work, we will research deeper about applications of non-convex methods in other areas besides EEG and graphic data. And we want to extend current work to include some constraints on factors in the model, such as orthogonality or non-negativity.

REFERENCES

[1] Tamara G. Kolda and Brett W. Bader, "Tensor decomposition and applications," SIAM review, vol. 51, no. 3, 2009, pp. 455-500.

[2] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye, "Tensor Completion for Estimating Missing Values in Visual data," IEEE Transactions on pattern analysis and machine intelligence, vol. 35, no. 1, 2013.

[3] Wei Chu and Zoubin Ghahramani, "Probabilistic Models for Incomplete Multi-dimensional Arrays," 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, Florida, USA, 2009.

[4] Emmanuel J. Candes and Benjamin Rech, "Exact matrix completion via convex optimization," Foundations of Computational Mathematics, vol. 9, no. 6, 2009, pp. 717-772.

[5] Silvia Gandy, Benjamin Rech, and Isao Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," Inverse Problems, vol. 27, no. 2, Jan. 2011.

[6] Huachun Tan, Bin Cheng, Wuhong Wang, Yu-Jin Zhang, and Bin Ran, "Tensor completion via a multi-linear low-n-rank factorization model," Neurocomputing, vol. 133, 2014, pp. 161-169.

[7] Silvia Gandy and Isao Yamada, "Convex optimization techniques for the efficient recovery of a sparsely corrupted low-rank matrix," Journal of Math-for-Industry, vol. 2, 2010, pp. 147-156.

[8] B. Vandereycken, "Low-rank matrix completion by Riemannian Optimization," SIAM Journal of Optimization, vol. 23, no. 2, 2013, pp. 1214-1236.

[9] Jonathan Barzilai and Jonathan M. Borwein, "Two-point step size gradient methods," IMA Journal of Numerical Analysis, vol. 8, no. 1, Jan. 1988, pp. 141-148.

[10] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations with missing data," Proceedings of the Tenth SIAM International Conference on Data Mining, SIAM, 2010, pp. 701-712.

[11] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye, "Tensor Completion for Estimating Missing Values in Visual data," IEEE ICCV, 2009.

[12] Marko Filipovic and Ante Julie, "Tucker factorization with missing data with application to low-n-rank tensor completion," Multidimensional Systems and Signal Processing, vol. 26, 2015, pp. 677-692.

[13] R. Tomioka, K. Hayashi, and H. Kashima, *Estimation of low-rank tensors via convex optimization,* Technical report, 2011.

[14] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar, *Poblano v1.0: A Matlab Toolbox for Gradient-Based Optimization*, Sandia National Laboratories, Albuquerque, NM and Livermore, CA.

[15] The International Epilepsy Electrophysiology portal: https://www.ieeg.org/

[16] C. A. Andersson and R. Bro, "The N-way toolbox for Matlab," Chemometrics and Intelligent Laboratory Systems, vol. 52, Aug. 2000, pp. 1-4.

[17] M. Mørup, L. K. Hansen, and S. M. Arnfred, "ERPWAVELAB a toolbox for multi-channel analysis of time-frequency transformed event related potentials," Journal of Neuroscience Methods, vol. 161, no. 2, 2007, pp. 361-368.

Thao Nguyen Thieu
She received the B.S. from Ho Chi Minh University of Science, Vietnam. She received her M.S. from in Chonnam National University, Korea. She researches in bio-signals as EEG and related areas.

Hyung-Jeong Yang
She received her B.S., M.S. and Ph.D. degrees from Chonbuk National University, Korea. She was a Post-doc researcher at Carnegie Mellon University, USA. She is currently a professor at Dept. of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea. Her main research interests include multimedia data mining, pattern recognition, artificial intelligence, e-Learning, and e-Design.

Tien Duong Vu
He received his B.S. from Ho Chi Minh University of Technology, Viet Nam. He is studying M.S. degree in Chonnam National University, Korea. His main research interests include Data Mining, Pattern Recognition, Machine Learning and Bioinformatics

Sun-Hee Kim
She received her M.S. from in Dongguk University, Korea. She received Ph.D. degree at Dept. Electronics and Computer Engineering, Chonnam National University, Korea. She is currently a research professor at Department of Brain and Cognitive Engineering, Korea University, Korea. Her research interests focus on datamining, sensor mining and stream mining, bioinformatics.