A Noise Inference Method Based on Fast Context-Aware Tensor Decomposition

Qingfu FAN\textsuperscript{(a)}, Nonmember, Lei ZHANG\textsuperscript{(b)}, Member, and Wen LI\textsuperscript{†}, Nonmember

SUMMARY Existing noise inference algorithms neglected the smooth characteristics of noise data, which results in executing slowly of noise inference. In order to address this problem, we present a noise inference algorithm based on fast context-aware tensor decomposition (F-CATD). F-CATD improves the noise inference algorithm based on context-aware tensor decomposition algorithm. It combines the smoothness constraint with context-aware tensor decomposition to speed up the process of decomposition. Experiments with New York City 311 noise data show that the proposed method accelerates the noise inference. Compared with the existing method, F-CATD reduces 4-5 times in terms of time consumption while keeping the effectiveness of the results.

key words: noise inference, context-aware tensor decomposition, fast context-aware tensor decomposition, smoothness constraint

1. Introduction

Although the development of urbanization modernizes people's lives, the city's noise pollution has become a problem derived from the urbanization development, affecting human behavior, well-being, and health\cite{1}. In recent years, the research of urban noise situation has become a hot topic. Among these researches, noise inference is a very important research direction. A number of countries, such as the United Kingdom and Germany, have started to monitor noise pollution using a noise map. Luca et al. \cite{2} used wireless sensor networks to monitor noise pollution in urban areas. Since the cost of collecting noise map data and deploying dedicated sensor networks is very expensive, NoiseTube \cite{3} and Ear-phone \cite{4} - two mobile terminal noise pollution monitoring software have been widely applied. Due to the fact that the urban noises are usually a mixture of multiple sound sources, diagnosing urban noise pollution solely by monitoring the noise decibel level data is not thorough. Analyzing the composition of the noises is a key to solve the noise pollution.

To address this problem, Zheng Yu et al. \cite{5} proposed a noise inference method based on context-aware tensor decomposition (CATD). CATD integrates 311 noise complaint data, social media data, POIs and road network data into a context-aware tensor decomposition model to infer the urban noise pollution situation (consisting of a noise pollution indicator and the composition of noises). The CATD algorithm first builds a tensor to model urban noise. Each entry of the tensor stores the number of noise complaints about a particular noise category in a particular region and a particular time slot, and then fills in the tensor's missing entries. After that the value of an entry is used to learn the composition of the noises in arbitrary locations and to diagnose the urban noise situation. However, this method has a drawback. It neglects the smooth characteristics of noise data, which results in executing slowly of noise inference. To improve the efficiency of tensor decomposition, Y. Wang et al. \cite{6} proposed fast and randomized tensor decomposition algorithms based on sketching. This algorithm is built on the idea of count sketches. Jeremy et al. \cite{7} decomposed a nonnegative tensor into decomposable terms in a compressed domain. They made use of the orthogonal structure of the transformation matrices to ease a fast compression-decompression scheme. Yokota et al. \cite{8} proposed Gaussian radial basis functions nonnegative tensor decomposition method (GRBF-NTD) with additional smoothness constraints. As the GRBF-NTD method used smoothing constraint which is compatible with the smooth data studied in this paper, we leverage the GRBF-NTD to speed up the computation of the CATD that Zheng Yu et al. have proposed and present a noise inference method based on fast context-aware tensor decomposition (F-CATD). The method combines the smoothness constraint with context-aware tensor decomposition by using a linear combination of smoothing functions when the tensor is decomposed into a low-dimensional matrix. It's aim is to do approximation processing and to accelerate the process of noise inference.

2. Noise Inference

The main idea of noise inference is to build a tensor to model urban noise based on the original urban noise data. It then uses tensor decomposition to fill in the tensor's missing entries so as to obtain the noise situation of the wider regions. Each value of the tensor represents the noise pollution indicator of a region.

2.1 Tensor Construction

311 is NYC's governmental non-emergency service number, allowing people to complain about the city's issues. Among these complaints, noise is the third largest category of complaints of the 311 data. Each noise complaint is associated
with a location, a time stamp, and a noise category. A tensor 
\( Y \in \mathbb{R}^{N \times M \times L} \) with three dimensions denoting \( N \) regions, \( M \) noise categories and \( L \) time slots is used to model the urban

noises. Due to the different noise types on weekdays and weekends, two tensors are built separately.

Region dimension: The first dimension denotes regions
\( r = \{r_1, r_2, \ldots, r_N\} \).

Category dimension: This dimension denotes the different
noise categories \( c = \{c_1, c_2, \ldots, c_M\} \).

Time span dimension: A day is divided into equal time
slots \( t = \{t_1, t_2, \ldots, t_k, \ldots, t_L\} \).

An entry: An entry \( Y(i, j, k) \) stores the total number of
311 complaints of category \( c_j \) in region \( r_i \) and time slot \( t_k \).
The value of each entry in the tensor \( Y \) is then normalized to
\([0, 1]\) for decomposition.

In order to fill the missing entries of tensor \( Y \), we decom-
pose \( Y \) into the multiplication of three matrices, \( \Re \in \mathbb{R}^{N \times R_1}, \Ce \in \mathbb{R}^{M \times R_2}, \Te \in \mathbb{R}^{L \times R_3} \) and a core tensor \( \G \in \mathbb{R}^{R_1 \times R_2 \times R_3} \), based on non-zero entries. \( R_1, R_2 \) and \( R_3 \) denote the number of latent factors (usually very small). Using a tucker decom-
position model [9], the objective function is defined as:

\[
F = \frac{1}{2} \| Y - G \times_1 R \times_2 C \times_3 T \|^2 \\
+ \frac{\lambda_1}{2} (\| G \|^2 + \| R \|^2 + \| C \|^2 + \| T \|^2)
\]  

(1)

where \( \times_k \) denotes the \( k \)-th way tensor-matrix product. By
 minimizing the objective function, we can get the optimized
\( R, C \) and \( T \). The missing values in \( Y \) can be recovered by
Eq. (2):

\[
Y_{re} = G \times_1 R \times_2 C \times_3 T
\]

(2)

Then we can get the distribution of city noise from \( Y_{re} \).
However, the 311 data is very sparse, resulting in a sparse
tensor. Filling in the missing entries of the tensor just based
on the non-zero entries is not accurate enough. To deal
this problem, Zheng Yu et al. proposed a method based on
context-aware tensor decomposition (CATD). The CATD
extracted additional features from POI/road network data,
user check-ins, and 311 data, which contains geographical
features, human mobility features and noise category fea-
tures denoted by matrices \( X, D, Z \). These additional infor-
mation sources are applied as contexts in the decomposition
process to improve the accuracy of noise inference. 
Matrix \( X \in \mathbb{R}^{N \times P} \) (\( P \) denotes the dimension of the geographical features), incorporates the similarity between two regions
in terms of their geographic features (regions with similar
geographic features have a similar noise situation). \( X \)
can be factorized into the multiplication of two matrices,
\( X = RU \), where \( R \in \mathbb{R}^{N \times R} \) and \( U \in \mathbb{R}^{R \times P} \) are low rank latent
factor matrices for regions and geographical features.
Matrix \( D \in \mathbb{R}^{L \times N} \) stores the correlation between different time
slots in terms of distribution of check-ins over different re-
gions (two time slots sharing a similar user check-in pat-
ttern have a similar noise situation), which can be factorized
into the multiplication of two matrices, \( D = TF \) (\( T \in \mathbb{R}^{L \times R} \)
is a low rank latent factor matrix for time slots). Matrix
\( Z \in \mathbb{R}^{M \times N} \) stores the correlation between different noise cate-
gories. Decomposing \( Y \) with feature matrices \( X, D \) and \( Z \)
collaboratively, the objective function is defined as Eq. (3):

\[
F = \frac{1}{2} \| Y - G \times_1 R \times_2 C \times_3 T \|^2 \\
+ \frac{\lambda_1}{2} (\| X - RU \|^2 + \| D - TR^T \|^2) \\
+ \frac{\lambda_2}{2} (\| C \|^2 + \| T \|^2 + \| U \|^2) \\
+ \frac{\lambda_3}{2} (\| G \|^2 + \| R \|^2 + \| C \|^2 + \| T \|^2)
\]

(3)

where \( \| \cdot \|^2 \) denotes the \( l_2 \) norm; the first part is to control the
decomposing \( Y \) error; the second part is to control the error
of factorization of \( X \); the fourth part is to control the error
of factorization of \( D \); the fifth part is a regularization penalty to
avoid over-fitting; \( \lambda_1, \lambda_2, \lambda_3 \) and \( \lambda_4 \) are parameters control-
ling the contribution of each part during the decomposition;
the third part \( tr(C^T L_Z C) \) is to ensure the similarity between different categories, where \( tr(\cdot) \) denotes the matrix trace, \( L_Z \) is the Laplacian matrix of the category
correlation graph. We recover the tensor \( Y \) by Eq. (2) and
fill the missing values to conduct noise inference. However,
this method neglected the characteristics that the noise data
are smooth in the adjacent areas and time slot, which results
in a slower rate of decomposition.

2.2 Fast Context-Aware Tensor Decomposition
In order to speed decomposition’s rate, we use a linear func-
tion to combine the smoothness constraint with context-
aware tensor decomposition. This combination makes an
approximation of processing when the tensor is decomposed
into a low-dimensional matrix, thereby reducing the compu-
tation time.

The smoothness means that the differences between
neighboring values are small in some domain. Considering
that the noise data in adjacent area and time slots are smooth, we extend Gaussian radial basis functions nonneg-
tive tensor decomposition (GRBF-NTD) [8] to the context-
aware tensor decomposition to improve the performance of
the noise inference algorithm. We use a linear combination
of smoothing function to obtain a low-dimensional matrix
\( R \). The objective function is defined as Eq. (4):

\[
F = \frac{1}{2} \| Y - G \times_1 \Phi W \times_2 C \times_3 T \|^2 \\
+ \frac{\lambda_1}{2} (\| X - RU \|^2 + \| D - TR^T \|^2) \\
+ \frac{\lambda_2}{2} (\| C \|^2 + \| T \|^2 + \| U \|^2) \\
+ \frac{\lambda_3}{2} (\| G \|^2 + \| R \|^2 + \| C \|^2 + \| T \|^2)
\]

(4)

where \( W \) is a non-negative matrix. The entry of matrix
\( \Phi \) uses GRBF with a standard deviation \( \sigma \) and can be ex-
pressed as Eq. (5):

\[
\Phi_{ij} = \frac{1}{\sigma} \exp \left( -\frac{(x_i - x_j)^2}{\sigma^2} \right)
\]
$$\varphi(i, n) \leftarrow \Gamma_\varphi(i, n) := \exp\left[-\frac{(i-n\Delta t)^2}{2\sigma^2}\right]$$  \hfill (5)

where $\sigma$ is a trade-off parameter that indicates the Gaussian function Bandwidth, $\Delta t$ represents the time interval. The pseudo code of F-CATD can be described as follows:

**Algorithm 1.** F-CATD algorithm

Input: NYC noise tensor $Y$, geographic feature matrix $X$, human mobility feature matrix $D$, noise category correlation matrix $Z$, the size of core tensor $G$ is $R_1, R_2, R_3$, smooth constraints matrix $\Phi$

Output: low rank matrices $R, C, T, W$, core tensor $G$

1. Initialize $G, R, C, T, W$ randomly;
2. $E \leftarrow Y - G \times_1 \Phi W \times_2 C \times_3 T$;
3. Set $i = 0$; /*The initial value of the number of iterations
4. Set $\eta$ as step length;
5. repeat
6. \[ \xi_1, \ldots, \xi_{R_1} ]^T \leftarrow G_1(C^T \otimes T)^T; /\{ G_1 \in \mathbb{R}^{R_1 \times R_2 \times R_3} \}
7. \text{for } r = 1, \ldots, R_1
8. Z_r \leftarrow E_1 + (\Phi w_r) \xi_r^T ; /\{ E_1 \in \mathbb{R}^{N \times ML} \}
9. \xi_r \leftarrow [Z_r^T(\Phi w_r)]^T;
10. \xi_r \leftarrow \xi_r/||\xi_r||;
11. w_r \leftarrow [w_r \otimes (\Phi T(Z_r \xi_r))] /\{ \Phi T(\Phi w_r) \};
12. E_1 \leftarrow Z_r - (\Phi w_r) \xi_r^T ;
13. end for
14. R \leftarrow \Phi W;
15. compute $F_i$; /* Use Eq. (4) to calculate
16. compute the gradients $\nabla_R F_i, \nabla_C F_i, \nabla_T F_i, \nabla_U F_i, \nabla_G F_i$;
17. $R_{i+1} = R_i - \eta \nabla_R F_i$; $C_{i+1} = C_i - \eta \nabla_C F_i$; $T_{i+1} = T_i - \eta \nabla_T F_i$; $U_{i+1} = U_i - \eta \nabla_U F_i$; $G_{i+1} = G - \eta \nabla_G F_i$;
18. compute $F_{i+1}$; /* Use Eq. (4) to calculate
19. $i++$;
20. until $(F_{i+1} - F_i)/F_i < \epsilon$ // The convergence threshold

In general, there is no closed form solution for $F$, so we use the gradient descent method to solve this problem, for example the gradient $\nabla_T F$ is computed as $\nabla_T F = \partial F/\partial T$. The purpose of introducing GRBF-NTD in context-aware tensor decomposition is to replace $R$ with $\Phi W$ approximately.

The key to this problem is the update of $W$. The lines 6-14 give the update procedure for $W$, where $[x]_+ := \max(x, 0)$, $e$ is usually very small (typically, $e = 10^{-16}$), @ and @ denotes element-wise multiplication (Hadamard product) and element-wise division, @ denotes Kronecker product. The lines 15-20 give the solution of the target matrix $R, C, T, G$ by the gradient descent. Finally, we can fill the missing values in tensor $Y$ by Eq. (2). We can use F-CATD to accelerate the process of noise inference.

3. Experiments and Analysis

In order to validate the performance of F-CATD proposed in this paper, the noise inference system NFS is developed, which is implemented using the Tensor Toolbox of Matlab R2009a. It is worth mentioning that all of the experiments were run on a commodity computer with Intel Core i5 CPU (2.3GHz) and 4GB RAM. The four real data sets [10] are taken as the experimental dataset.

3.1 Datasets

In this paper, we use four data sets: 311 noise dataset, Road Networks dataset, Check-ins dataset, POIs dataset. Table 1 summarizes the description of the datasets. Each 311 noise complaint data contains a timestamp, a location and a complaint category. Each user check-in (Foursquare and Gowalla) has a time stamp and a geospatial coordinate. A road network is comprised of a set of road segments. Each road segment has two terminal nodes, a series of intermediate points between the two terminals. POI is a venue in a physical world, which has a name, address, coordinates and category.

Road segments with a level are used to partition NYC, resulting in 891 regions. As weekdays and weekends have different noise situations, we build two tensors for them. Setting 1 hour as a time slot, the size of the two tensors is $891 \times 14 \times 24$. We feed the 311 data of 168 weekdays and 68 weekends into the two tensors. Finally, we use noise inference to fill the missing values in the two tensors.

3.2 Evaluation on the Method

We randomly remove part of non-zero entries from the tensor and fill in these entries using our method. Then we use the original values of these entries as a ground truth to measure the inferred values. Table 2 shows the comparison results between the noise inference algorithms based on context-aware tensor decomposition (CATD) and our noise inference algorithm based on F-CATD. The performance is measured by two metrics: Root Mean Square Error (RMSE) and time consumption. RMSE is computed as Eq. (6):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y}_i)^2}{n}}$$  \hfill (6)

where $\hat{y}_i$ is an inference, $y_i$ is the ground truth and $n$ is the number of instances. We split the whole data sets into training and validation sets with a different ratio. We held out the validation set and constructed the model using the training set with different values for the model parameters. The proposed method has some hyper-parameters in order to control sparseness, smoothness, or a level of regularization, we set

<table>
<thead>
<tr>
<th>Table 1</th>
<th>The information of datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sets</td>
<td>Period</td>
</tr>
<tr>
<td>311 noise data</td>
<td>5/23/2013-1/31/2014</td>
</tr>
<tr>
<td>Foursquare</td>
<td>4/24/2009-10/13/2013</td>
</tr>
<tr>
<td>Gowalla</td>
<td>5/5/2008-7/25/2011</td>
</tr>
<tr>
<td>POIs</td>
<td>2013</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>The comparison of two methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Weekdays</td>
</tr>
<tr>
<td>CATD</td>
<td>RMSE</td>
</tr>
<tr>
<td>F-CATD</td>
<td>0.0408</td>
</tr>
</tbody>
</table>

**IEICE TRANS. INF. & SYST., VOL.E100–D, NO.6 JUNE 2017**
\( \Delta t = 1, \sigma = 1, R_1 = R_2 = R_3 = 10 \) and \( \eta = 10 \). We employed the same stopping criterion for all the experiments. When the difference in the values of the objective function \( (F_{i+1} - F_i)/F_i \) updates is smaller than \( \epsilon = 10^{-2} \), the algorithm was stopped. Table 2 summarizes the experimental results of statistics, where we set the model parameters \( \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.0001 \) and remove 30\% non-zero entries from the tensor as the validation set (The remaining 70\% data as train data). We ran the parameter-sweeping experiments 15 times for each parameter value and averaged the RMSE. The size of the tensor \( Y \) is \( 891 \times 14 \times 24 \) (\( N=891, M=14, L=24 \)), the size of core tensor \( G \) is \( 10 \times 10 \times 10 \) (\( R_1 = R_2 = R_3 = 10 \)). The smaller values of RMSE denote the better results. Our approach (F-CATD) is very similar to CATD in filling errors (RMSE), but we consume less time and provide a significant improvement on the performance. Due to the fact that the noise situation is more complex at the weekend, we will use weekend tensor to do noise inference.

We varied the ratio of the training data from 60\% to 80\% in order to show the impact on performance in different parameters. As shown in Fig. 1, the RMSE of F-CATD and CATD have almost the same accuracy to infer the missing values when the tensor remain 60\%, 70\% and 75\% (removing other non-entries from the tensor as test data) of the original non-zero entries as training data. In addition, the RMSE of F-CATD is smaller than that of CATD when non-zero entries of the tensor remain 65\% and 80\%.

As shown in Fig. 2, we can find that there is the large difference between the two methods in the time consumption in noise inference. F-CATD spends around 50s, while CATD takes nearly 220s to 250s. In general, F-CATD reduces 4-5 times in terms of time consumption compared with CATD.

In summary, it can be concluded that the RMSE of F-CATD remain unchanged while time consumption reduces a lot compared with CATD. F-CATD shows the better performance.

4. Conclusion

In this paper, we propose a noise inference algorithm based on fast context-aware tensor decomposition to address the slower rate of noise inference problem. The method improves the noise inference method based on context-aware tensor decomposition. It combines the smoothness constraint into context-aware tensor decomposition when a tensor is decomposed into a low-dimensional matrix to do approximation processing so as to speed up the process of decomposition. Experiments with real datasets show that F-CATD accelerates the noise inference.

Acknowledgements

This work was supported by the Research and Practice Project on the Teaching Reform of Graduate Education in China University of Mining and Technology (YJSJG2016-32) and the Undergraduate Innovation and Entrepreneurship Aviation Funds Supported Undergraduate Innovation Project in China University of Mining and Technology (DC201641).

References