Comodule Estimation of Cognitive Sensor Networks Based on Partial Clustering for Partial Observed Data

Abdul bin Ismail

School of Engineering and Computing, Manipal International University, Malaysia
E-mail: engabdulbinismail@gmail.com

Abstract

The proposed study is on the partial clustering algorithms for cognitive sensor networks that deal with partially observed data. The proposed algorithms aim to estimate clusters in the presence of missing values and leverage data imputation techniques to fill in the gaps in the target and station device matrices. A modified loss function is introduced to shape the cluster centers, and robust Non-negative Matrix Factorization (NMF) algorithms are utilized to enhance the robustness of the clustering process. This research contributes to the field of cognitive sensor networks by providing insights into the challenges of partial clustering and presenting effective algorithms to address them. The proposed methods have the potential to enhance the performance of clustering tasks in various domains, including sensor networks, by accounting for missing data and producing accurate cluster reconstructions.

Keywords: Comodule Estimation, Partial Clustering, Sensor Network, Partial Observed Data

1. Introduction

Partial clustering of sensor networks is important for modern communications [1-3]. Partial clustering significantly reduces the overhead associated with complete clustering. In complete clustering, where all nodes participate in the clustering process, the exchange of clustering messages and the reconfiguration of clusters can consume significant network resources. With partial clustering, clustering is performed selectively, only when it is

DOI: https://doi.org/10.36548/jsws.2023.3.006
Received: 07.09.2023, received in revised form: 04.10.2023, accepted: 16.10.2023 published: 28.10.2023

necessary, thereby minimizing the overhead and conserving energy. Clustering helps in balancing the energy consumption among sensor nodes. By forming clusters, nodes can aggregate and merge their data at the cluster head, reducing the amount of data transmitted to the base station. Partial clustering further enhances energy efficiency by ensuring that clustering is initiated only when the cluster head has consumed a certain portion of its energy.This approach helps in prolonging the network's lifetime by efficiently utilizing the limited energy resources of sensor nodes. Sensor networks are often deployed on a large scale, consisting of hundreds or thousands of nodes. Partial clustering enables scalability by allowing the network to handle a large number of nodes. Instead of all nodes participating in the clustering process simultaneously, only a subset of nodes needs to be involved in each round of clustering. This reduces the computational and communication requirements, making it feasible to deploy and manage large-scale sensor networks [4]. Partial clustering allows for dynamic network reconfiguration based on changing conditions and requirements. As the network evolves over time, the clustering structure can be adjusted to accommodate new nodes, node failures, or changes in network topology. By continuously evaluating the energy levels and scores of nodes, the partial clustering algorithm can adaptively select new cluster heads and members, ensuring efficient utilization of resources and maximizing network performance. Partial clustering of sensor networks presents several challenges that need to be addressed. One of the key challenges in partial clustering is selecting appropriate cluster heads. The algorithm needs to consider factors such as residual energy, node scores, and network topology to identify the most suitable nodes to act as cluster heads. Designing an efficient and effective cluster head selection mechanism is crucial to ensure optimal cluster formation and energy balancing. Partial clustering algorithms operate in a distributed manner, where each sensor node independently decides whether to become a cluster head or join an existing cluster. Achieving consensus among nodes without global information or centralized control is challenging. The algorithm must incorporate mechanisms for nodes to exchange local information, make decisions based on that information, and coordinate their actions to form clusters in a decentralized manner. Sensor networks are dynamic in nature, with nodes joining or leaving the network over time. This poses a challenge for partial clustering algorithms, as the clustering structure needs to adapt to changes in node availability and network topology. Ensuring smooth transitions during cluster reformation and maintaining network stability in the face of dynamic events require careful algorithm design and coordination among nodes. Partial clustering aims

to reduce the overhead associated with complete clustering, but finding the right balance between overhead reduction and network efficiency is challenging. The algorithm should determine the optimal conditions for initiating partial clustering to minimize overhead while still maintaining efficient energy utilization and network performance. Striking the right trade-off is crucial to achieve energy savings without sacrificing data collection, communication reliability, or network lifetime. As sensor networks scale up in size, the challenges of partial clustering become more pronounced. The algorithm should be scalable to handle large numbers of nodes and should not be overly complex in terms of computational requirements or communication overhead. Ensuring scalability while maintaining the benefits of partial clustering, such as energy efficiency and adaptability, is a significant challenge in the design and implementation of clustering algorithms. Addressing these challenges requires a thorough understanding of the network characteristics, careful algorithm design, and performance evaluation. Researchers and engineers need to consider these challenges to develop robust and efficient partial clustering algorithms that can effectively support the requirements of diverse sensor network applications. To address the challenges of partial clustering in sensor networks, several solutions can be considered. Develop sophisticated algorithms for cluster head selection that take into account multiple factors such as residual energy, node scores, communication costs, and network topology. Machine learning techniques, optimization algorithms, or hybrid approaches can be employed to improve the accuracy and efficiency of cluster head selection. These advanced methods can provide better cluster formation and energy balancing. Design algorithms that adapt to changes in network dynamics, such as node mobility, failures, or additions. Introduce mechanisms for periodic re-evaluation of cluster formation to accommodate new nodes or changes in node characteristics. Incorporate dynamic decision-making processes that consider real-time information to adjust cluster configurations, ensuring the algorithm remains effective and efficient in dynamic environments. Develop efficient communication protocols to minimize the overhead associated with clustering messages. Utilize techniques such as data aggregation, compression, or hierarchical communication structures to reduce the amount of data transmission and clustering-related signalling. Efficient communication protocols can reduce energy consumption and improve network scalability. Combine partial clustering with other techniques such as data-driven approaches, machine learning, or mobility-aware algorithms [5]. Hybrid approaches can enhance the clustering process by leveraging additional information or context-specific factors. For example,

incorporating data similarity metrics or location-based information can improve the accuracy of cluster formation and enable better utilization of sensor resources. Implement energy-aware scheduling mechanisms that optimize the active and sleep periods of sensor nodes. Design protocols that allow nodes to coordinate their active periods to ensure data collection and communication while maximizing energy conservation. By synchronizing the active periods, nodes can effectively utilize available resources and reduce energy wastage. Conduct extensive simulations and performance evaluations to assess the efficiency and effectiveness of partial clustering algorithms. Consider various network scenarios, such as different network sizes, node densities, and traffic patterns, to evaluate the scalability, energy efficiency, network lifetime, and data reliability of the algorithms. This iterative process helps refine and optimize the algorithm design. Foster collaboration among researchers, practitioners, and standardization bodies to establish common guidelines and benchmarks for partial clustering algorithms. Encouraging the exchange of ideas, sharing of datasets, and promoting open research can accelerate advancements in partial clustering techniques and facilitate the adoption of standardized solutions. By exploring these possible solutions and continuously advancing research and development in partial clustering, it is possible to overcome the challenges and achieve more efficient and scalable sensor networks with prolonged network lifetime and improved energy utilization.

In summary, the scope of the proposed work is to estimate clusters in the presence of missing values and leverage data imputation techniques to fill in the gaps in the target and station device matrices. Therefore, clustering models can be performed.

The rest of this study is organized as follows. Section 2 shows related works. Section 3 is the proposed method with the experiment in Section 4. Section 5 is the conclusion.

2. Related Work

Non-Negative Matrix Factorization (NMF) is a matrix factorization technique that has been successfully applied in various domains, including signal processing, image processing, and clustering [6-16]. While NMF may not be directly applicable to the specific problem of partial clustering in sensor networks, it can be leveraged as a component or an optimization technique within a larger clustering framework [17, 18]. NMF can be employed to extract meaningful features from sensor data. By decomposing the sensor data matrix into non-

negative basis vectors, NMF can identify underlying patterns or components in the data. These extracted features can then be used as input for subsequent clustering algorithms. NMF can be used for data compression by approximating the original data matrix with a lower-rank NMF representation. This can help reduce the amount of data that needs to be transmitted or processed, leading to energy savings and improved efficiency in partial clustering algorithms. NMF can facilitate dimensionality reduction by identifying a reduced set of non-negative basis vectors that capture the most significant characteristics of the data. By reducing the dimensionality of the data, clustering algorithms can operate on a smaller feature space, which can enhance clustering performance and reduce computational complexity. NMF can be utilized to initialize the cluster centers or prototypes in clustering algorithms. By decomposing the data matrix into non-negative components, NMF can provide an initial set of representative patterns or cluster centers. These initial centers can then be refined and updated using traditional clustering algorithms to achieve better clustering results. NMF-based clustering algorithms can be developed by incorporating NMF directly into the clustering process. These algorithms aim to simultaneously cluster the data while decomposing it into non-negative components. By integrating NMF and clustering, it is possible to exploit the inherent non-negativity constraints of the data and potentially improve clustering accuracy and interpretability.

L_2-norm NMF (Nonnegative Matrix Factorization) is a variant of the traditional NMF algorithm that uses the L_2-norm (Euclidean norm) as the regularization term in the objective function [19, 20]. The objective of L_2-norm NMF is to find nonnegative factor matrices that minimize the reconstruction error while promoting sparsity and stability in the factorization. In the standard NMF formulation, the objective function is typically based on the Frobenius norm, which is the L_2-norm of the difference between the original data matrix and its approximation reconstructed from the factor matrices. However, L_2-norm NMF extends this by adding an L_2-norm regularization term to the objective function. The L_2-norm regularization term encourages sparsity in the factor matrices by penalizing large values and promoting small values. This helps in identifying a concise and meaningful set of components that contribute significantly to the data representation while suppressing the influence of less important or noisy components. By incorporating the L_2-norm regularization, L_2-norm NMF can help improve the robustness, generalization, and interpretability of the factorization results.

L_1-norm NMF (Nonnegative Matrix Factorization) is a variant of NMF that incorporates the L_1-norm (also known as the Manhattan norm or absolute norm) as the regularization term in the objective function. L_1-norm NMF promotes sparsity in the factor matrices, resulting in a more sparse and interpretable representation of the data. In the standard NMF formulation, the objective function is typically based on the Frobenius norm, which is the L_2-norm of the difference between the original data matrix and its approximation reconstructed from the factor matrices. However, L_1-norm NMF extends this by adding an L_1-norm regularization term to the objective function. The L_1-norm regularization term encourages sparsity by promoting coefficients or entries of the factor matrices to be exactly zero. This leads to a sparse representation, where only a subset of components is actively involved in explaining the data. By incorporating the L_1-norm regularization, L_1-norm NMF encourages sparse and concise representations, where only a few components are selected to explain the data, while the majority of components have zero or near-zero coefficients. This sparsity property can enhance interpretability, reduce overfitting, and improve the robustness of the factorization results. L_1-norm NMF has been widely used in various applications such as signal processing, image analysis, text mining, and feature selection, where sparsity and interpretability are desired.

$L_{2,1}$-norm NMF (Nonnegative Matrix Factorization) is an extension of NMF that utilizes the $L_{2,1}$-norm as a regularization term in the objective function [21, 22]. The $L_{2,1}$-norm promotes structured sparsity, which encourages groups of coefficients or entries in the factor matrices to be simultaneously zero or small. In the standard NMF formulation, the objective function is typically based on the Frobenius norm, which is the L_2-norm of the difference between the original data matrix and its approximation reconstructed from the factor matrices. However, $L_{2,1}$-norm NMF extends this by adding an $L_{2,1}$-norm regularization term to the objective function. The $L_{2,1}$-norm regularization term promotes structured sparsity by encouraging groups of coefficients or entries in the factor matrices to be jointly zero or small. This means that instead of having individual coefficients set to zero independently, entire groups or subsets of coefficients are simultaneously set to zero. This can lead to more interpretable and meaningful structure in the learned representations. The $L_{2,1}$-norm is computed as the sum of the L_2-norms of the rows of a matrix. In the context of NMF, the $L_{2,1}$-norm of the right factor matrix encourages sparsity across the columns of the right factor matrix, promoting group sparsity in the factorization. By incorporating the $L_{2,1}$-norm

regularization, $L_{2,1}$-norm NMF encourages structured sparsity, where entire groups of coefficients are simultaneously set to zero or small. This can be useful in scenarios where there is prior knowledge or assumption about the underlying structure of the data, and can lead to more meaningful and interpretable factorizations. $L_{2,1}$-norm NMF has been applied in various domains such as image processing, bioinformatics, and text mining, where structured sparsity and group-level interpretations are desired.

Robust NMF (Nonnegative Matrix Factorization) is an extension or variant of the traditional NMF algorithm that incorporates robustness mechanisms to handle outliers, noise, or corrupted data [23-31]. It aims to improve the stability and reliability of the factorization process in the presence of disturbances in the input data. The basic NMF algorithm assumes that the input data can be accurately represented as a linear combination of nonnegative components. However, in real-world scenarios, the data may contain outliers or noise that can significantly impact the factorization results. Robust NMF algorithms address this issue by introducing robust optimization techniques or incorporating additional constraints to enhance the resilience of the factorization process. There are different approaches to achieving robustness in NMF. Some common techniques include: Robust cost functions: Instead of using the traditional least squares error as the objective function, robust NMF algorithms employ robust cost functions that are less sensitive to outliers. Examples include L_1 norm, Huber loss, or Cauchy loss functions[32-35]. By imposing sparsity constraints on the factor matrices, robust NMF algorithms encourage the identification of a sparse set of meaningful components while suppressing the impact of outliers or noise. Robust NMF methods may include outlier detection and rejection mechanisms to identify and discard outliers or noisy samples before performing the factorization. This helps improve the accuracy of the factorization results. The goal of robust NMF is to achieve more reliable and interpretable factorizations, even in the presence of challenging data conditions. By considering the robustness aspect, these algorithms can be valuable in various domains, such as image processing, signal analysis, bioinformatics, and data mining, where data quality and integrity are critical factors.

3. Proposed Work

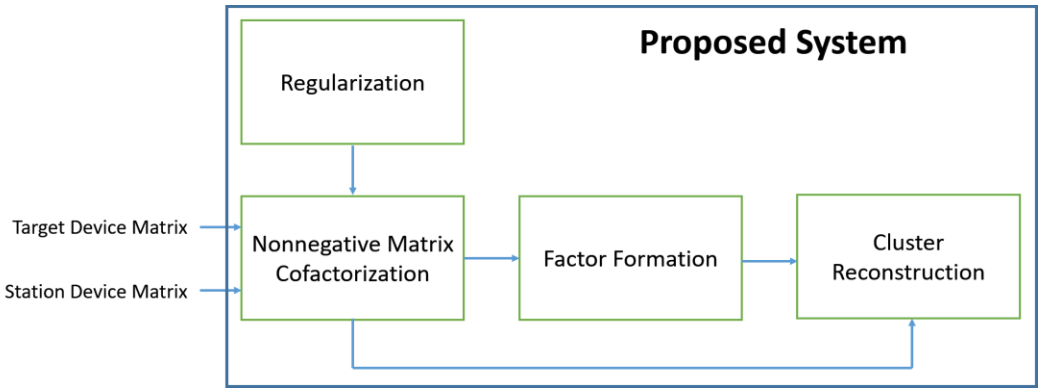

Figure 1. Framework of the Proposed System

The overall framework of the proposed system is illustrated in Figure 1. The inputs are target and station device matrices, which are analysed in the co-factorization module while regularization constraints are imposed. Then, the factor matrices are computed and used for cluster reconstruction. The details are as follows.

Let M be the number of feature dimensions, L mean the number of targets, and N signify the number of station devices. Target and station device matrices $\mathbf{X}_\mathrm{T}$ and $\mathbf{X}_\mathrm{D}$ are M-by-L and M-by-N, respectively. Assume that they contain missing value entries. Data imputation algorithms are used to fill in the entries with substituted values and obtain estimated $\hat{\mathbf{X}}_\mathrm{D}$ and $\hat{\mathbf{X}}_\mathrm{T}$. Then, the loss function is defined as (1) based on [15] with some modifications to fit the applications.

$$\begin{aligned}\min\Big\{&\left\|\hat{\mathbf{X}}_\mathrm{D}-\mathbf{W}\mathbf{H}_\mathrm{D}\right\|_F^2+\left\|\hat{\mathbf{X}}_\mathrm{T}-\mathbf{W}\mathbf{H}_\mathrm{T}\right\|_F^2+\\&\alpha\,\mathrm{Tr}\left(\mathbf{H}_\mathrm{D}\mathbf{A}\mathbf{H}_\mathrm{D}^{\cdot}\right)+\beta\,\mathrm{Tr}\left(\mathbf{H}_\mathrm{D}\mathbf{B}\mathbf{H}_\mathrm{T}^{\cdot}\right)+\gamma_2\left\|\mathbf{W}\right\|_2^2+\gamma_{2,1}\left\|\mathbf{W}\right\|_{2,1}\\&\delta\sum_{p=1}^{P}\left\|\mathbf{W}\left(\mathring{\mathrm{a}},p\right)\right\|_1^2+\lambda\sum_{n=1}^{N}\left\|\mathbf{H}_\mathrm{D}\left(\mathring{\mathrm{a}},n\right)\right\|_1^2+\lambda\sum_{l=1}^{L}\left\|\mathbf{H}_\mathrm{T}\left(\mathring{\mathrm{a}},l\right)\right\|_1^2\Big\},\end{aligned}\tag{1}$$

where $\mathbf{W}$, $\mathbf{H}_\mathrm{D}$, and $\mathbf{H}_\mathrm{T}$ are factor matrices, and their dimensions are M-by-P, P-by-N, and P-by-L. Additionally, $\|\cdot\|_F$, $\|\cdot\|_2$, $\|\cdot\|_1$, and $\|\cdot\|_{2,1}$ are the Frobenius norm, L_2 norm, L_1 norm, and $L_{2,1}$ norm, $\mathrm{Tr}(\cdot)$ is the trace operator, T is the transpose, and $\star$ is the slicing operator. Moreover, α, β, γ, δ, and λ are all scalar parameters to control penalty terms. Finally, $\mathbf{A}$ and $\mathbf{B}$ are a

similarity matrix and a connection bipartite matrix, and P is the hidden dimension. Cluster centers are formed in $\mathbf{W}$. To increase robustness, robust NMF algorithms are used to reshape cluster centers by using half quadratic loss functions $\phi(\cdot)$. Then,

$$\min\left\{\phi\left(\hat{\mathbf{X}}_{\mathrm{D}}-\mathbf{U}\mathbf{V}_{\mathrm{D}}\right)+\phi\left(\hat{\mathbf{X}}_{\mathrm{T}}-\mathbf{U}\mathbf{V}_{\mathrm{T}}\right)+\eta_2\left\|\mathbf{U}\right\|_2^2+\eta_1\left\|\mathbf{U}\right\|_1^2+\mathrm{Vol}\left(\mathbf{U}\right)\right\}, \tag{2}$$

where η is a scalar parameter, and Vol(·) is the volume penalty term. Cluster centers are formed in M-by-P $\mathbf{U}$. To fuse cluster center information, the weighted function is used.

$$\mathbf{C}=\mathbf{W}\%\left(\mathbf{W}+\mathbf{U}\right)+\mathbf{U}\%\left(\mathbf{W}+\mathbf{U}\right), \tag{3}$$

where % is elementwise division. The algorithm 1 below illustrates the robust NMF algorithms used in reshaping the cluster centers by using half quadratic loss functions.

	Input: Target and station device matrices $\mathbf{X}_{\mathrm{T}}$ and $\mathbf{X}_{\mathrm{D}}$
	Output: $\mathbf{C}$
1	Initialize $\mathbf{W}$, $\mathbf{H}_{\mathrm{D}}$, and $\mathbf{H}_{\mathrm{T}}$
2	Compute $\mathbf{A}$ and $\mathbf{B}$
3	Do While
4	Compute $\mathbf{W}$ using the multiplicative update rule
5	Compute $\mathbf{H}_{\mathrm{D}}$ using the multiplicative update rule
6	Compute $\mathbf{H}_{\mathrm{T}}$ using the multiplicative update rule
7	Until Convergence
8	Reconstruct $\hat{\mathbf{X}}_{\mathrm{D}}$ using $\mathbf{W}\mathbf{H}_{\mathrm{D}}$
9	Reconstruct $\hat{\mathbf{X}}_{\mathrm{T}}$ using $\mathbf{W}\mathbf{H}_{\mathrm{T}}$
10	Do While
11	Compute $\mathbf{U}$ using NNLS
12	Compute $\mathbf{V}_{\mathrm{D}}$ using NNLS
13	Compute $\mathbf{V}_{\mathrm{T}}$ using NNLS
14	Until Convergence
15	Compute $\mathbf{C}$ using $\mathbf{W}$ and $\mathbf{U}$

Algorithm .1 Cluster Centre Reshaping

The intuition behind the proposed method was that the factor matrices and coefficient matrices learned from the inputs (although they were incomplete) are used. The defect parts could be initialized and updated in iterations (like the Expectation-Minimization strategy). Therefore, one could estimate the correct content inside the incomplete part.

Data imputation plays a crucial role. Data imputation techniques are used to address the issue of missing values in the collected data. Missing data can arise due to various reasons, such as sensor failures, transmission errors, or limitations in data collection mechanisms. The absence of data points can significantly impact the quality of cluster analysis, making data imputation a necessary step to mitigate this problem. Necessity of data imputation is as follows: it's common to have missing data because not all sensors or devices report information simultaneously. Imputing missing values ensures that data integrity is maintained and that clusters are formed based on as much available information as possible. Data imputation enhancing cluster quality. Accurate and complete data are essential for producing meaningful clusters. Imputing missing data allows the clustering algorithms to consider all relevant information, leading to more accurate cluster assignments.

4. Results and Discussion

To evaluate the effectiveness of the proposed method, this study conducted experiments using real sensor networks to collect sensing data. The experiment involved a total of 10 targets and 20 stations, implemented on embedded systems comprising transmitter and receiver components. Specifically, this study used embedded systems “Raspberry Pi 4B + Waveshare RM502Q-AE 5G HAT” to collect data and MATLAB to simulate clustering reconstruction environments. Each target device transmitted labelled data to the corresponding station, with a total of two unique labels. It is important to note that each target device was associated with only one label type, ensuring clear distinctions. At lease one connection between a target and a station was established. To simulate real-world scenarios, this study introduced missing values into the collected data matrices. The missing values were randomly selected, resulting in a missing rate of 5.00%. Additionally, all labels were removed from the data matrices. To address the missing values, this study employed a data imputation algorithm, generating imputed matrices for further analysis. In the evaluation, this study varied the variable p, which represents a parameter range from two to 10, with a fixed separation of one. It is worth mentioning that the correct number of clusters in the data was known to be two. After the proposed method estimated the number of clusters, this study used accuracy rates to assess its effectiveness in producing accurate results.

The procedure for selecting favourable values of the model parameter was based on layers of loops, where one loop was designed for one parameter. Five parameters involved five layers. This study recorded the performance of the model with respect to each change in parameter. The step size was 0.50, and the range was within one and zero. When the loops ended, the system could find the best ones.

Table 1. Favourable Values of the Parameters

Parameter	Favourable Value
α	0.30
β	0.50
γ	0.25
δ	0.30
λ	0.50

Experiment 1: Assessment of the Need for Data Imputation

This study performed a test to assess the necessity of data imputation (with the use of Robust NMF [32], i.e., filling in missing data with initial values and iteratively updating replacements after matrix factorization and rebuilding as shown in Figure 2.using four distinct datasets collected with the devices.

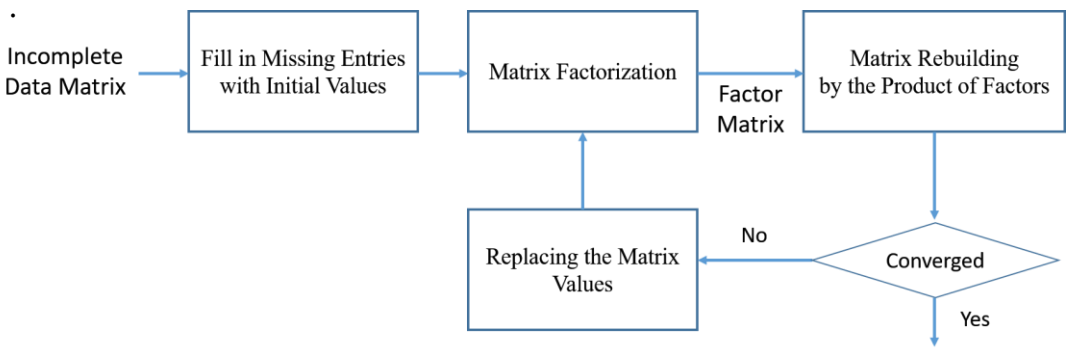

Figure 2. Data Imputation Flow

The procedure followed the same protocol mentioned at the beginning of this section. The baseline was the proposed method without data imputation. The experimental results are presented in Figure 4. The vertical axis represents clustering accuracy, while the horizontal axis displays different datasets. It is evident that the proposed algorithm with data imputation produced superior results. The figure.3 illustrates the accuracy comparison of the need for data imputation.

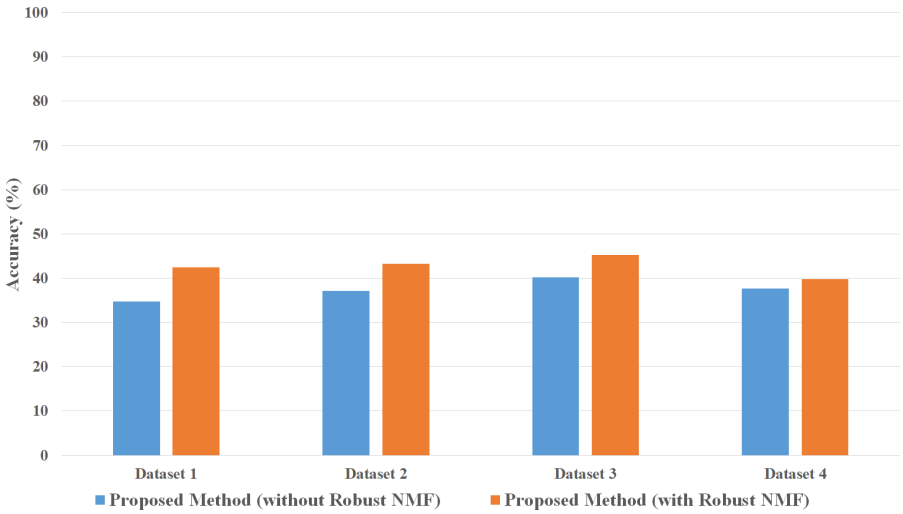

Figure 3. Accuracy Comparison of the Need for Data Imputation

Experiment 2: Assessment of Different Methods

This study conducted a performance evaluation test comparing the proposed method with several existing methods, using the same datasets and parameters as before. The methods for comparison included Comodule Discovery (CD) [15], Cooperative Comodule Discovery (CCD) [5], Versatile Clustering (VC) [14], and Alternate Least Squares (ALS) [17, 18]. Figure 4 displays the numerical results, with the horizontal axis representing different datasets and the vertical axis representing accuracy. As shown in the figure, this study observed that the proposed method improved accuracy by an average of 5.99% across all the datasets, demonstrating its effectiveness. The improvements were 7.24%, 4.58%, 6.05, and 6.11% for each dataset.

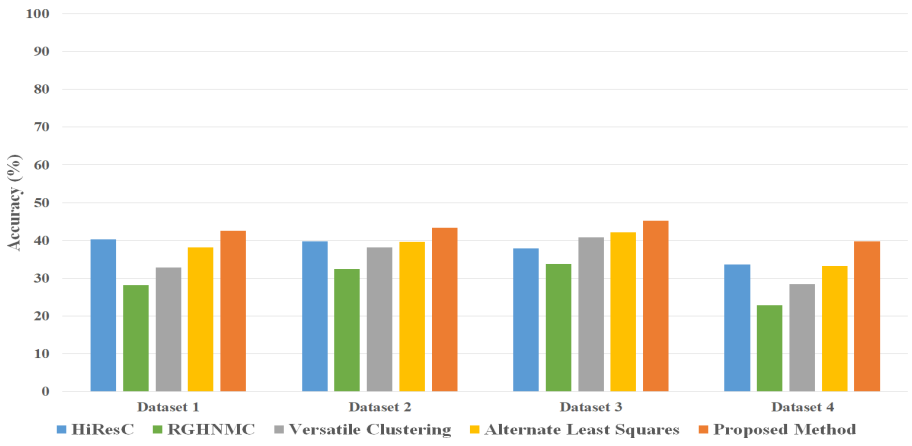

Figure 4. Accuracy Comparison of Different Methods.

Experiment 3: Testing of Cluster Prediction

The experimental results are depicted in Figure 5, with accuracy represented on the vertical axis and the number of clusters on the horizontal axis. It is evident from the graph that the highest accuracy aligns with the correct number of clusters. This clear correlation indicates the effectiveness of the proposed method in accurately reconstructing clusters.

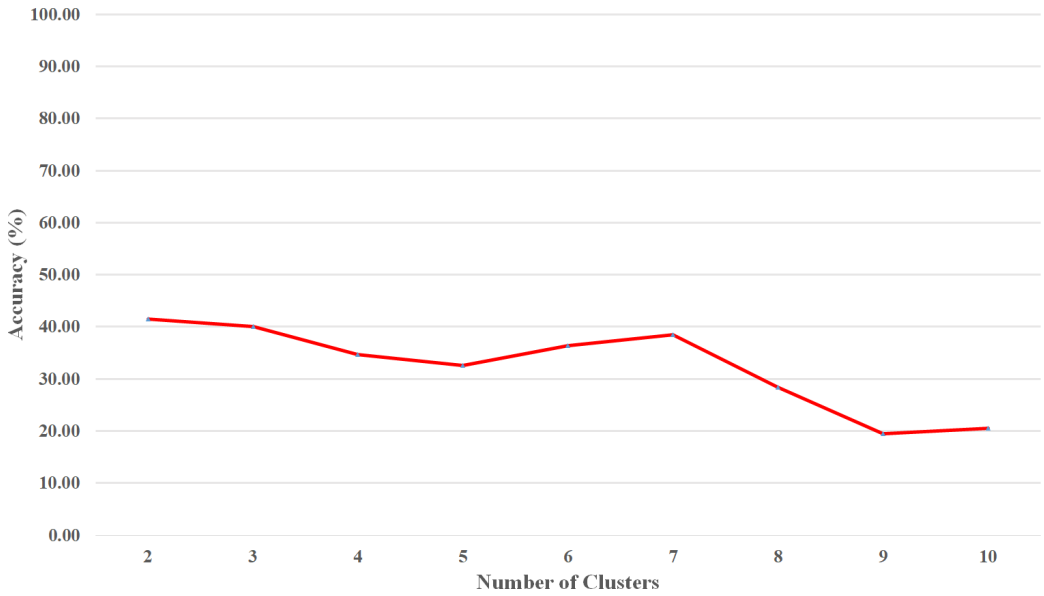

Figure 5. Experimental Results of Estimated the Number of Clusters

5. Conclusion

This study proposes a method for estimating clusters in cognitive sensor networks with partially observed data. The method involves utilizing target and station device matrices with missing value entries, which are filled using data imputation algorithms. A loss function is defined based on factor matrices and penalty terms to shape the cluster centers. Robust NMF algorithms are employed to increase robustness. To evaluate the proposed method, experiments were conducted using real sensor networks. The experiment involved 10 targets and 20 stations, and missing values were introduced into the data matrices with a 5.00% missing rate. Labels were removed from the matrices, and a data imputation algorithm was used to generate imputed matrices. The evaluation included varying a parameter range from two to 10, with the correct

number of clusters known to be two. The accuracy rates were used to assess the effectiveness of the method. The experimental results, illustrated in Figure 5, demonstrate that the highest accuracy corresponds to the correct number of clusters. This indicates the effectiveness of the proposed method in accurately reconstructing clusters.

References

[1]J. Liu, N. Kato, J. Ma, and N. Kadowaki, “Device-to-device communication in LTE-advanced networks: A survey,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 1923–1940, 2015.

[2]J. Liu, Y. Kawamoto, H. Nishiyama, N. Kato, and N. Kadowaki, “Device-to-device communications achieve efficient load balancing in LTE-advanced networks,” IEEE Wireless Communications, vol. 21, no. 2, pp. 57–65, Apr. 2014.

[3]J. Liu, S. Zhang, N. Kato, H. Ujikawa, and K. Suzuki, “Device-to-device communications for enhancing quality of experience in software defined multi-tier LTE-A networks,” IEEE Network, vol. 29, no. 4, pp. 46–52, Jul.–Au. 2015.

[4]B. Wei, “Novel kernel orthogonal partial least squares for dominant sensor data extraction,” IEEE Access, vol. 8, pp. 36131–36139, Feb. 2020.

[5]H. Chuang, K.-L. Hou, S. Rho, and B.-W. Chen, “Cooperative comodule discovery for swarm-intelligent drone arrays,” Computer Communications, vol. 154, pp. 528–533, Mar. 2020.

[6]Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki, “Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering,” IEEE Transactions on Neural Networks, vol. 22, no. 12, pp. 2117–2131, Dec. 2011.

[7]Y. Jia, H. Liu, J. Hou, and S. Kwong, “Semisupervised adaptive symmetric non-negative matrix factorization,” IEEE Transactions on Cybernetics, Feb. 2020.

[8]A. Vandaele, N. Gillis, Q. Lei, K. Zhong, and I. Dhillon, “Coordinate descent methods for symmetric nonnegative matrix factorization,” IEEE Transactions on Signal Processing, vol. 64, no. 21, pp. 5571–5584, May 2016.

[9] W. Wu, Y. Jia, S. Kwong, and J. Hou, "Pairwise constraint propagation-induced symmetric nonnegative matrix factorization," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 12, pp. 6348–6361, Dec. 2018.

[10] W. Yan, B. Zhang, Z. Yang, and S. Xie, "Similarity learning-induced symmetric nonnegative matrix factorization for image clustering," IEEE Access, vol. 7, pp. 166380–166389, Nov. 2019.

[11] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," IEEE Signal Processing Magazine, vol. 36, no. 2, pp. 59–80, Mar. 2019.

[12] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[13] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in Proc. 14th International Conference on Neural Information Processing Systems, Denver, Colorado, United States, 2000, Nov. 28–30, pp. 556–562.

[14] Y. Li and A. Ngom, "Versatile sparse matrix factorization and its applications in high-dimensional biological data analysis," in Proc. IAPR International Conference on Pattern Recognition in Bioinformatics, Nice, France, 2013, Jun. 17–20, pp. 91–101.

[15] S. Zhang, Q. Li, J. Liu, and X. J. Zhou, "A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules," Bioinformatics, vol. 27, no. 13, pp. i401-i409, Jul. 2011.

[16] J.-X. Liu, D. Wang, Y.-L. Gao, C.-H. Zheng, Y. Xu, and J. Yu, "Regularized nonnegative matrix factorization for identifying differentially expressed genes and clustering samples: A survey," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, no. 3, pp. 974–987, Feb. 2018.

[17] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent," IEEE Transactions on Image Processing, vol. 20, no. 7, pp. 2030–2048, Jul. 2011.

[18] N. Guan, D. Tao, Z. Luo, and B. Yuan, “Non-negative patch alignment framework,” IEEE Transactions on Neural Networks, vol. 22, no. 8, pp. 1218–1230, Aug. 2011.

[19] B.-W. Chen and W.-C. Ye, “Low-error data recovery based on collaborative filtering with nonlinear inequality constraints for manufacturing processes,” IEEE Transactions on Automation Science and Engineering, vol. 18, no. 4, pp. 1602–1614, Aug. 2020.

[20] B.-W. Chen, W. Ji, S. Rho, and Y. Gu, “Supervised collaborative filtering based on ridge alternating least squares and iterative projection pursuit,” IEEE Access, vol. 5, pp. 6600–6607, Mar. 2017.

[21] D. Kong, C. Ding, and H. Huang, “Robust nonnegative matrix factorization using L21-norm,” in Proc. 20th ACM International Conference on Information and Knowledge Management, Glasgow, Scotland, United Kingdom, 2011, Oct. 24–28, pp. 673–682.

[22] B. Wu, E. Wang, Z. Zhu, W. Chen, and P. Xiao, “Manifold NMF with L21 norm for clustering,” Neurocomputing, vol. 273, pp. 78–88, Jan. 2018.

[23] N. Guan, D. Tao, Z. Luo, and B. Yuan, “NeNMF: An optimal gradient method for nonnegative matrix factorization,” IEEE Transactions on Signal Processing, vol. 60, no. 6, pp. 2882–2898, Jun. 2012.

[24] Y. He, F. Wang, Y. Li, J. Qin, and B. Chen, “Robust matrix completion via maximum correntropy criterion and half-quadratic optimization,” IEEE Transactions on Signal Processing, vol. 68, pp. 181–195, 2020.

[25] W. Chang, “Symmetric nonnegative matrix factorization based on box-constrained half-quadratic optimization,” IEEE Access, vol. 8, pp. 170976–170990, Sep. 2020.

[26] S. Yang, C. Hou, C. Zhang, Y. Wu, and S. Weng, “Robust non-negative matrix factorization via joint sparse and graph regularization,” in Proc. 2013 International Joint Conference on Neural Networks, Dallas, Texas, United States, 2013, Aug. 04–09, pp. 1–5.

[27] L. Zhang, Q. Zhang, B. Du, D. Tao, and J. You, “Robust manifold matrix factorization for joint clustering and feature extraction,” in Proc. 31st AAAI Conference on Artificial

Intelligence, San Francisco, California, United States, 2017, Feb. 04–09, vol. 31, pp. 1662–1668.

[28] T. Liu, M. Gong, and D. Tao, "Large-cone nonnegative matrix factorization," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 9, pp. 2129–2142, Jun. 2016.

[29] N. Guan, T. Liu, Y. Zhang, D. Tao, and L. S. Davis, "Truncated Cauchy non-negative matrix factorization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 1, pp. 246–259, Jan. 2019.

[30] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 7, pp. 1087–1099, Jul. 2012.

[31] C. Bo and H. Kuang, "Half quadratic dual learning for fuzzy multiconcepts of partially-observed images," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 6, no. 4, pp. 994–1007, Aug. 2022.

[32] C. Zach, "Robust bundle adjustment revisited," in Proc. 13th European Conference on Computer Vision, Zurich, Switzerland, 2014, Sep. 06–12, pp. 772–787.

[33] C. Zach and G. Bourmaud, "Iterated lifting for robust cost optimization," in Proc. 28th British Machine Vision Conference, London, United Kingdom, 2017, Sep. 04–07.

[34] C. Zach and G. Bourmaud, "Descending, lifting or smoothing: Secrets of robust cost optimization," in Proc. 15th European Conference on Computer Vision, Munich, Germany, 2018, Sep. 08–14, pp. 558–574.

[35] C. Zac and G. Bourmaud, "Multiplicative vs. additive half-quadratic minimization for robust cost optimization," in Proc. 28th British Machine Vision Conference, Newcastle, United Kingdom, 2018, Sep. 03–06.

[36] J. Wang, F. Tian, C. H. Liu, H. Yu, X. Wang, and X. Tang, "Robust nonnegative matrix factorization with ordered structure constraints," in Proc. 2017 International Joint Conference on Neural Networks, Anchorage, Alaska, United States, 2017, May 14–19, pp. 478–485.