A Comparative Simulation Study of Cluster Ensemble Algorithms Integrated with Multiple Imputation for Clustering with Missing Data

Yui Tomo[1,2*], Funato Sato[2], and Mari Oba[2]

[1]Center for Surveillance, Immunization, and Epidemiologic Research,
National Institute of Infectious Diseases, 1-23-1 Toyama, Shinjuku-Ku,
Tokyo 162-0052, Japan
[2]Department of Clinical Data Science, National Center of Neurology and
Psychiatry, 4-1-1 Ogawa-Higashi, Kodaira, Tokyo 187-8551, Japan
[*]E-mail: tomoy@niid.go.jp

Abstract

Since cluster analysis methods usually cannot be applied directly to data with missing values, various approaches have been investigated to handle the problem. Multiple imputation is one of the standard procedures for addressing the problem of missing data. In cluster analysis, instead of Rubin's rule, cluster ensemble methods have been proposed to be combined with multiple imputation. However, it remains unrevealed which of the cluster ensemble algorithms leads to better performance when integrated with the procedure. Therefore, we conducted numerical comparisons of several algorithms to integrate the results from k-means++ clustering for multiply imputed datasets. Our results suggest that the non-negative matrix factorization algorithm may be suitable for scenarios with class balance, whereas the agglomerative cluster algorithm may be suitable for scenarios with class imbalance. Before application to actual datasets, we still recommend performing simulation experiments in scenarios reflecting the characteristics of the datasets and the assumption of missing value mechanisms.

Keywords: Cluster analysis, Consensus clustering, Hierarchical clustering, k-means, Non-negative matrix factorization

1 Introduction

Cluster analysis is the task of partitioning a series of observations into several clusters (groups) in a way that the observations in the same cluster tend to be similar. Cluster analysis has been applied in various fields, from medicine to social sciences. Numerous methods have been developed to perform cluster analysis, such as k-means and hierarchical clustering (Forgey, 1965; Ward Jr, 1963). However, most basic clustering methods cannot be directly applied to real-world datasets because they often include missing values.

Complete case analysis is the approach in which any observations with missing values are merely deleted from the analysis. Although this approach is one of the simplest ways to deal with missing values, it can influence the validity of results and impede subsequent analyses based on the assigned clusters to each observation. For example, if only a limited number of observations remain after the deletion of observations with missing values, the statistical power declines and the association between the assigned clusters and follow-up data might not be detectable. Therefore, we focus on the approaches that include all of the observations, even those with missing values. Multiple imputation is one of such methods (Rubin, 1976, 1987; Schafer, 1997). This approach consists of two main steps: first, multiple complete datasets by imputing the missing values using probabilistic models and second, the results from each complete dataset into the final result using Rubin's rule.

While multiple imputation has been used for regression analysis with incomplete data, the approach has recently been extended to cluster analysis. For cluster analysis, instead of Rubin's rule, cluster ensemble algorithms have been proposed to combine the results (Strehl and Ghosh, 2002; Ghaemi et al., 2009). Basagaña et al. (2013) employed a relabeling and voting algorithm to combine results from multiple complete datasets. Faucheux et al. (2021) used the MultiCons algorithm based on frequent item set mining (Al-Najdi et al., 2016). Bruckers et al. (2017) and Audigier and Niang (2023) focused on the formulation of a clustering ensemble algorithm as the solution to the mean partition problem. They applied a direct optimization approach of the target function and the non-negative matrix factorization ensemble algorithm (Li et al., 2007). Other proposed methods to combine multiple imputation results are the incorporation of distance matrices and the integration of cluster centroids (Lee and Harel, 2023; Aschenbruck et al., 2023).

Although various clustering ensemble methods and several comparison results have been presented, it remains unrevealed which algorithm is effective for integrating with multiple imputation to handle missing data (Kuncheva et al., 2006; Li et al., 2009). Therefore, the purpose of this study is to compare numerically the performance of several ensemble algorithms. We especially focus on the algorithms to offer approximate or locally optimal solutions to the problem formulated by the mean partition problem.

The remainder of this paper is organized as follows. In Section 2, we formulate the cluster ensemble methods and introduce several algorithms. In Section 3, we detail the settings of our numerical comparison. In Section 4, we show the results of the experiments. In Section 5, we discuss the implications and limitations. Finally, In Section 6, we conclude the study.

2 Methods

2.1 Cluster Ensemble Methods

Cluster ensemble methods integrate multiple results from various cluster analysis algorithms or techniques into a single result. These approaches have been developed to improve clustering quality, providing stability of results (Vega-Pons and Ruiz-Shulcloper, 2011). We first formulate the cluster ensemble method based on the mean partition problem. Let $\mathcal{X} := \{\mathbf{x}_u | 1 \leq u \leq n,\, \mathbf{x}_u \in \mathbb{R}^p\}$ be data points to be clustered. Suppose k clusters are assigned to $\mathcal{X}$, and the possible set of partitions is denoted by Ω_k^n. We assume that m clustering results $\{C_i | 1 \leq i \leq m,\, C_i \in \Omega_k^n\}$ are given. Let $d(C_i, C_j) \in \mathbb{R}_0^+$ denote the dissimilarity of two clustering results. We define $D(C) := \frac{1}{m} \sum_i^m d(C_i, C)$ as the target function. Then, the optimization problem

$$\hat{C} := \operatorname*{arg\,min}_{C \in \Omega_k^n} D(C) \tag{1}$$

is defined as the mean partition problem and we adopt the minimizer $\hat{C}$ as the integrated result (Topchy et al., 2004).

Because $d(\cdot,\cdot)$ is desirable for having the property of distance metric, we use the Mirkin distance and normalized mutual information (NMI) distance (Vinh et al., 2010). We define $L(C_i, u)$ as the cluster assigned to $\mathbf{x}_u$ in C_i. Then, the Mirkin distance is defined as

$$d(C_i, C_j) := \sum_{u=1}^{n} \sum_{v=1}^{n} d_{uv}\left(C_i, C_j\right),$$

where

$$d_{uv}\left(C_i, C_j\right) := \begin{cases} 1 & \text{if } L(C_i, u) = L(C_i, v) \text{ and } L(C_j, u) \neq L(C_j, v), \\ 1 & \text{if } L(C_j, u) = L(C_j, v) \text{ and } L(C_i, u) \neq L(C_i, v), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The NMI distance is based on the quantity of mutual information normalized using the entropy. The distance is defined as

$$d(C_i, C_j) := 1 - \mathrm{NMI}(C_i, C_j),$$

where

$$\mathrm{NMI}(C_i, C_j) = \sum_{s \in C_i} \sum_{t \in C_j} P(s,t) \log \left(\frac{P(s,t)}{P(s)P(t)} \right) / \max\{H(C_i), H(C_j)\}.$$

We note that $H(C) := -\sum_{s \in C} P(s) \log P(s)$ is the entropy, where $P(s,t)$ is the proportion of data points simultaneously assigned to cluster s and t, and $P(s)$ and $P(t)$ are the proportions of data points assigned to cluster s and t, respectively. Since the optimization problem (1) using these dissimilarities is known as NP-complete, several algorithms have been developed to offer approximate or locally optimal solutions to this problem.

2.2 Greedy Algorithm to Directly Optimize $D(C)$

Greedy algorithms are approaches that obtain local solutions via repeated optimization of partial problems. In this study, we focus on the algorithm by Strehl and Ghosh (2002), which directly optimizes $D(C)$. The procedures of this algorithm are as follows: (i) Provide the data points with the initial labels. (ii) Evaluate $D(C)$ repeatedly relabeling the data point x_i with labels different from the initial one. (iii) Replace the label of x_i such that it gives the minimal target function. (iv) Apply Steps ii and iii repeatedly until the label change does not occur. This algorithm is independent of the definition of $d(\cdot,\cdot)$, so it is applicable to both the Mirkin and NMI distance.

2.3 Non-negative Matrix Factorization

When using the Mirkin distance, the optimization problem of $D(C)$ is known to be attributed to the non-negative matrix factorization (NMF). That is, we obtain the following transformation:

$$\begin{aligned} D(C) &= \frac{1}{m} \sum_{i=1}^{m} \sum_{u=1}^{n} \sum_{v=1}^{n} \left[M_{uv}\left(C_i\right) - M_{uv}\left(C\right) \right]^2 \\ &= \Delta M^2 + \sum_{u=1}^{n} \sum_{v=1}^{n} \left(\tilde{M}_{uv} - U_{uv} \right)^2 , \end{aligned} \tag{3}$$

where

$$M_{uv}\left(C_i\right) := \begin{cases} 1 \text{ if } L\left(C_i, u\right) = L\left(C_i, v\right) \\ 0 \text{ otherwise} \end{cases} ,$$

$\mathbf{M}_i := \left(M_{u,v}(C_i)\right)_{1 \leq u \leq n, 1 \leq v \leq n}$, $\tilde{M}_{uv} := \frac{1}{m} \sum_{i=1}^{m} M_{uv}\left(C_i\right)$, $U_{uv} := M_{uv}(C)$, and $\Delta M^2 := \frac{1}{m} \sum_i \sum_{u,v} [M_{uv}\left(C_i\right) - \tilde{M}_{uv}]$. From the representation in (3), the optimization problem $\min_{C \in \Omega_k^n} D(C)$ can be reduced to the optimization problem $\min_U \|\tilde{\mathbf{M}} - \mathbf{U}\|_F^2$ under a certain constraint on $\mathbf{U}$, where $\tilde{\mathbf{M}} := \frac{1}{m} \sum_{i=1}^{m} \mathbf{M}_i$. It is known that the obtained matrix $\mathbf{H}$ from the symmetric NMF problem

$$\min_{\mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{D} \in \mathbb{R}^{k \times k} \mathbf{H} \geq 0, \mathbf{D} \geq 0} \left\| \tilde{\mathbf{M}} - \mathbf{H}\mathbf{D}\mathbf{H}^\top \right\|_F^2 , \text{ s.t. } \mathbf{H}^\top \mathbf{H} = \mathbf{I} \tag{4}$$

can be regarded as a cluster indicator matrix of the integrated results (Li et al., 2007; Ding et al., 2006). An iterative algorithm to solve the optimization problem (4) was proposed by Li et al. (2007).

2.4 Agglomerative Clustering

When using the Mirkin distance, the optimization problem of $D(C)$ is also known to be reduced to correlation clustering. That is, we obtain the following representation of $D(C)$:

$$D(C) = \sum_{u=1}^{n} \sum_{v=1}^{n} I\left(L(C, u) = L(C, v)\right) \tilde{M}_{uv} + \sum_{u=1}^{n} \sum_{v=1}^{n} I\left(L(C, u) \neq L(C, v)\right) (1 - \tilde{M}_{uv}),$$

where $I(\cdot)$ is the indicator function. The agglomerative clustering method using $\tilde{\mathbf{M}}$ as the input dissimilarity matrix provides an approximate solution to the correlation clustering (Gionis et al., 2007).

2.5 Combined Approach with Multiple Imputation

The integrated estimate after multiple imputation is obtained by Rubin's rule in the case of regression analysis. For cluster analysis, the results can be combined via cluster ensemble approaches. Assume that we have m results $\left\{C_1^M, C_2^M, \ldots, C_m^M\right\}$ for m imputed datasets. $\hat{C}^M$, defined as below can be obtained as the final result:

$$\hat{C}^M := \operatorname*{arg\,min}_{C \in \Omega_k^n} \frac{1}{m} \sum_{i}^{m} d(C_i^M, C).$$

The amount of uncertainty of the final result can be quantified via the integrated version of clustering instability defined in Audigier and Niang (2023), which is not considered in this study.

3 Numerical Experiments

3.1 Experiment 1: Comparison of Cluster Ensemble Algorithms Alone

We performed the comparison of the cluster ensemble algorithms alone to obtain some insight into the differences in performance of the algorithms introduced in the prior section. We designed the scenarios in terms of varying the sample size and class balance. While the data generation settings were based on the experiments by Audigier and Niang (2023), our settings were extended from 2 to 3 classes.

The data generation model in our experiments was the multivariate normal distribution with dimension $p = 10$ and three mixture components:

$$x \sim \pi_1 \mathcal{N}_p(\mu_1, \mathbf{\Sigma}) + \pi_2 \mathcal{N}_p(\mu_2, \mathbf{\Sigma}) + \pi_3 \mathcal{N}_p(\mu_3, \mathbf{\Sigma}), \quad \mathbf{\Sigma} = \sigma^2 \mathbf{I}_p,$$

where $\mathbf{I}_p$ denotes the $p-$dimensional identity matrix. The cluster centers were the mean vectors of each component set to $\mu_1 = (0,0,0,0,0,0,0,0,0,0)$, $\mu_2 = (0,0,0,0,0,2,2,2,2,2)$, and $\mu_3 = (2,2,2,2,2,2,2,2,2,2)$. The standard deviations of the components were set to $\sigma \in \{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$. The sample sizes of data points were $n \in \{30, 60, 120\}$. The mixture ratios were $(\pi_1, \pi_2, \pi_3) \in \{(1/3, 1/3, 1/3), (1/6, 1/3, 1/2)\}$, representing the balanced and imbalanced scenarios, respectively.

We compared three algorithms: the greedy algorithm for directly optimizing the NMI distance-based $D(C)$ (GNMI), non-negative matrix factorization (NMF), and agglomerative clustering (AClu) based on the Ward method.

The performance was evaluated in terms of recovering the true labels, which were determined according to which cluster center the data point was generated from. The evaluation metric was the Adjusted Rand Index (ARI), which is a measure of the degree of accordance between two cluster labels (Hubert and Arabie, 1985). The measure ranges from -1 to 1, and the values close to 1 indicate that the labels are similar. A detailed explanation of ARI is provided in Appendix A.

The evaluation process was as follows: (i) We obtained 30 results to be integrated by applying k-means++ to the 30 generated datasets. (ii) We integrated the 30 results into the final results via cluster ensemble algorithms and evaluate them. (iii) The procedures above were repeated 200 times. k-means++ is an improved version of the k-means algorithm to address sensitivity to initial cluster centers (Arthur and Vassilvitskii, 2007). The details of k-means++ are explained in Appendix B.

Additionally, to obtain a deeper insight, we introduced the instability measure of the clustering results to be integrated. The instability is defined as:

$$\text{Instability} := \sum_{i<j}^{100} d(C_i, C_j),$$

where $d(C_i, C_j)$ is the NMI distance.

3.2 Experiment 2: Evaluations of the Combined Approaches with Multiple Imputation

We conducted numerical experiments to compare the cluster ensemble methods combined with multiple imputation for addressing missing data. We also compared the k-pod algorithm, which optimizes the objective function defined over only observed values and has been one of the commonly used methods for clustering with missing data (Chi et al., 2016). The details of k-pod are provided in Appendix B.

We designed the scenarios with additional considerations of missing ratios and missing data mechanisms. The data generation model was the same as that in Experiment 1 except for the covariance matrix $\mathbf{\Sigma}$, which is defined as

$$\mathbf{\Sigma} = \left(\begin{array}{c|c} \mathbf{I}_5 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{R} \end{array}\right),$$

where $\mathbf{R}$ is a 5×5 matrix with its diagonal elements 1 and the other elements ρ. The correlation parameter ρ was set to $\rho \in \{0.3, 0.6\}$.

In addition, the missing values were brought to the generated data under the assumption of the missing completely at random (MCAR) or missing at random (MAR) mechanism. The MCAR is the mechanism in which the missingness is independent of both observed and unobserved values. The MAR is the mechanism in which the missingness depends only on the observed values. For the MCAR mechanism, the missing probability for each data point $\mathbf{x}_u$ and each dimension ℓ was set as

$$\begin{aligned} &\forall u \in \{1, \dots, n\}, \forall \ell \in \{2, \dots, p\}, \\ &\quad \text{Prob}\,(\text{missingness of the } \ell\text{-th component of data point } \mathbf{x}_u) = \tau, \end{aligned}$$

and for the MAR mechanism, the probability was set as

$$\begin{aligned} &\forall u \in \{1, \dots, n\}, \forall \ell \in \{2, \dots, p\}, \\ &\quad \text{Prob}\,(\text{missingness of the } \ell\text{-th component of data point } \mathbf{x}_u) = \Phi\left(a_\tau + x_{u1}\right), \end{aligned}$$

where the missing ratios were $\tau \in \{0.1, 0.3, 0.5\}$, and the constant a_τ was determined to adjust the ratios. The cumulative distribution function (CDF) of the standard normal distribution is denoted by Φ.

We compared the k-means++ on the complete data (k-means-full) for the benchmark results, the k-means++ on complete cases (k-means-CCA), the k-pod algorithm (k-pod), and each ensemble algorithm integrated with multiple imputation (MI-GNMI, MI-NMF, MI-AClu). Multiple imputation was conducted via the chained equation (MICE) to generate the imputed datasets. The imputation models were the predictive mean matching based on the nearest 5 points. For each imputed dataset, we applied k-means++ to obtain results with 3 clusters, and then the results were integrated through ensemble algorithms. The evaluation process, measure, and number of repetitions were the same as those in Experiment 1.

3.3 Software

Our experiments were performed using Python and R. For data generation, application of the k-means method, and implementation of each ensemble algorithm, we used Python 3.8.3, Numpy 1.23.3, scikit-learn 1.3.1, and Scipy 1.8.1(Pedregosa et al., 2011; Virtanen et al., 2020). The implementations of MICE and k-pod were executed using R 4.2.2, mice 3.16.0, and kpodclustr 1.1(van Buuren and Groothuis-Oudshoorn, 2011; Chi et al., 2016).

4 Results

4.1 Experiment 1: Comparison of the Cluster Ensemble Algorithms Alone

The ARI values for balanced data scenarios are presented in the box plot of the left column of Figure 1. From the results, NMF typically exhibited the highest ARI values over all the settings in terms of sample sizes. In the settings of $n = 30$ and $n = 60$, the ARI of GNMI and AClu was almost the same. However, in the $n = 120$ setting, GNMI was superior to AClu. As σ increased, the instability of the clustering results to be integrated also increased. It seems that in the larger instability, the differences in the performances became more distinct.

The results for the imbalanced data scenarios are shown in the right column. While the performance of the NMF and GNMI algorithms was not stable in relatively small instability settings, that of AClu was stable and relatively high. NMF outperformed other approaches except in scenarios of low instability and larger sample sizes. However, it is noted that the NMF may often have failed to recover the appropriate cluster labels in such scenarios. As the instability increased, the performance of NMF and GNMI improved.

4.2 Experiment 2: Evaluations of the Combined Approaches with Multiple Imputation

The results for balanced data and low correlation scenarios ($\rho = 0.3$) are shown in Figure 2, and those of high correlation scenario ($\rho = 0.6$) are exhibited in Figure 3. Because the results seem to be similar, we now focus on the low-correlation scenarios ($\rho = 0.3$). In the case of the MCAR missing data scenarios, there were hardly any differences in ARI among the cluster ensemble approaches and k-pod. However, in the scenarios with a high missing ratio ($\tau = 0.5$), the performance of k-pod declined. The CCA resulted in relatively low ARI or high variance. This tendency was particularly evident for the scenarios with higher missing ratios, where very few observations were complete cases. In the MAR missing data scenarios, the ARI values of k-pod were notably lower than those of the other methods. In both the MCAR and MAR scenarios, among the cluster ensemble approaches, the NMF algorithm had slightly higher ARI, followed by GNMI and AClu. These differences were more distinct in the settings with larger sample sizes. The CCA did not work well as in the MCAR scenarios.

The results of the imbalanced scenarios are shown in Figure 4 and Figure 5, which are low and high correlation scenarios, respectively. In the MCAR scenarios, the k-pod outperformed the other methods, while GNMI and AClu exhibited slightly higher ARI values than NMF. In the MAR scenarios, when the missing ratios were $\tau = 0.3$ and $\tau = 0.6$, the superiority of the cluster ensemble algorithm was notable. Although the performance of the three ensemble approaches was nearly identical, GNMI and AClu provided slightly higher ARI values than NMF. The ensemble approaches especially outperformed the k-pod in the high missing rate scenarios ($\tau = 0.5$).

5 Discussion

The experiments suggest that in terms of ARI between the true cluster labels, the NMF algorithm may tend to yield better performance in the class balance scenarios. In contrast, the AClu and GNMI approaches may perform better in scenarios with class imbalance. These results are consistent with the results of evaluating ensemble algorithms alone, which indicates that NMF tends to be more robust against the instability of the clustering results to be integrated than AClu and GNMI, and AClu typically demonstrates more stability in scenarios with class imbalance than other algorithms. Therefore, differences in performance when integrated with multiple imputation procedures may be attributable to the characteristics of the type of cluster ensemble algorithms.

The NMF algorithm tends to show lower performance in scenarios with large sample sizes and low instability. This could be because the symmetric non-negative factorization method falls into inappropriate local minima, which leads to the failure to reconstruct the original matrix. Figure 6 shows an example of failure of the NMF algorithm to reconstruct the original matrix in a dataset with $n = 120$ and class imbalanced scenarios. It seems that the GNMI algorithm also falls into inappropriate local minima in scenarios with class imbalance. In our simulation study, we adopted the best solution of the NMF among five implementations for each iteration to address the issue of the selection of initial values. Although this approach may have led to better solutions than NMF with a single initial value, it did not completely overcome the issue.

The k-pod algorithm is commonly used for cluster analysis with incomplete data. Since multiple imputation procedures are computationally intensive, the k-pod algorithm may be better in terms of computational efficiency. However, our results suggest that the performance of k-pod may decline compared to those of the multiple imputation and ensemble approaches in high missing ratio scenarios or MAR missing scenarios. The additional advantage of multiple imputation and ensemble approaches over k-pod is the ability to select clustering algorithms and missing value imputation models in a way that is suitable for the data.

In this study, we compared three ensemble algorithms. However, there are many variations of ensemble algorithms (Strehl and Ghosh, 2002; Ghaemi et al., 2009). Among the algorithms proposed for combination with multiple imputation, we did not adopt the voting and relabeling algorithm and the MultiCons algorithm. The former involves the manual process of relabeling data points so that the interpretations of the assigned labels are the same across results. This manual process is so labour-intensive that it is impractical to conduct it on many clustering results. The MultiCons algorithm is an ensemble approach based on frequent pattern mining. This algorithm requires substantial computational time. In our preliminary studies, even with smaller sample sizes, such as $n = 30$ or $n = 60$, a single run required hours. Thus, the practicality of these algorithms was deemed limited.

In this study, we used MICE with predictive mean matching, a popular multiple imputation method in various applications. However, it is not always the best choice for cluster analysis. DPImputeCont is a method based on the Dirichlet Process Gaussian mixture model and it may be more suitable (Kim et al., 2014). Other imputation approaches can be implemented using the clusterMI package in R, and further simulation experiments on their combination with cluster ensemble methods should be conducted in the future (Audigier et al., 2021; Audigier and Niang, 2023).

Various measures can be used to evaluate clustering results. While we focused on the accuracy of recovered labels through the ARI measure, other metrics such as the silhouette coefficient, normalized mutual information, and purity assess different aspects of clustering results (Rousseeuw, 1987). Even when these measures provide similar values, the patterns of clusters may differ and thus, considering multiple measures in practice is important.

When applying cluster analysis to real-world datasets, it is important to determine the number of clusters. This remains to be explored in this study. One possible approach is using information criteria to select the number of clusters for k-means to be integrated or other automatic cluster number selection such as X-means (Ishioka, 2000). Then, the final number of clusters could be determined by the most frequent number across the imputed datasets. However, in actual studies in the applied field, the number of clusters may be predetermined based on the prior knowledge of researchers. Therefore our investigation is still relevant.

Finally, in actual applications, it is important to consider the missing data mechanisms and select the appropriate methods to address them. The missing data mechanisms investigated in this study are limited, and our scenarios do not cover all possible data scenarios. Therefore, before applying the algorithms to actual data, we recommend assessing their performance via simulation experiments under some data generation scenarios that reflect both the characteristics of the actual data and the assumptions of missing mechanisms.

6 Conclusion

In conclusion, our numerical comparison of cluster ensemble algorithms provided us with some practical guides for combination with multiple imputation. Our results suggest that the non-negative matrix factorization algorithm may be suitable for the scenarios with class balance, whereas agglomerative clustering may be suitable in scenarios with class imbalance. Although the performance of the three ensemble algorithms was nearly identical in the scenarios with class imbalance, we should consider that the non-negative matrix factorization and greedy algorithm may fall into inappropriate local minima and provide low performance. Before application to actual data, we still recommend performing simulation experiments in scenarios reflecting the characteristics of the datasets and the assumption of missing data mechanisms.

Acknowledgement

We gratefully thank Editage (www.editage.jp) for the English language editing.

Disclosure statement

This work was not supported by any funding agency in the public, commercial, or not-for-profit sectors.

The authors declare that there are no competing interests.

Code and Data Availability

The Python and R scripts to generate data and perform experiments in this study are available from `https://github.com/t-yui/MI-ClusterEnsembles-Comparison`.

Appendix A Adjusted Rand Index

Rand (1971) proposed the Rand Index as a measure of the correspondence between two clustering results. When we define $d_{uv}(C_i, C_j)$ as (2), we have

$$1 - d_{uv}(C_i, C_j) = \begin{cases} 1 & \text{if } \{L(C_i, u) = L(C_i, v) \text{ and } L(C_j, u) = L(C_j, v)\}, \\ 1 & \text{if } \{L(C_i, u) \neq L(C_i, v) \text{ and } L(C_j, u) \neq L(C_j, v)\}, \\ 0 & \text{otherwise,} \end{cases}$$

which indicates the agreement of C_i and C_j on the pair $\{\mathbf{x}_u, \mathbf{x}_v\}$. The Rand index $\mathrm{RI}(C_i, C_j)$ is the proportion of pairs $\{\mathbf{x}_u, \mathbf{x}_v\}$ on which C_i and C_j agrees, that is,

$$\mathrm{RI}(C_i, C_j) := \frac{\sum_{1 \le u < v \le n} \{1 - d_{uv}(C_i, C_j)\}}{\binom{n}{2}}.$$

The Adjusted Rand Index proposed by Hubert and Arabie (1985) is the normalized version of the Rand Index considering the expected agreement under a random clustering model. We define a probability space $(\Omega_k^n, \mathcal{F}, \mathbb{P})$, where $\mathcal{F}$ is a σ-algebra on Ω_k^n, and $\mathbb{P}$ is a probability measure that models a random clustering process. Then, the Adjusted Rand Index is defined as

$$\mathrm{ARI}(C_i, C_j) = \frac{\mathrm{RI}(C_i, C_j) - \mathbb{E}_{\mathbb{P}}[\mathrm{RI}(C_i, C_j)]}{\max\{\mathrm{RI}(C_i, C_j)\} - \mathbb{E}_{\mathbb{P}}[\mathrm{RI}(C_i, C_j)]}.$$

When we define $a_j := |C_i^{(j)}|$ and $b_i := |C_j^{(i)}|$, the Adjusted Rand Index can be expressed as

$$\mathrm{ARI}(C_i, C_j) = \frac{\mathrm{RI}(C_i, C_j) - \sum_j \binom{a_j}{2} \sum_i \binom{b_i}{2} / \binom{n}{2}}{\left\{\sum_j \binom{a_j}{2} + \sum_i \binom{b_i}{2}\right\}/2 - \sum_j \binom{a_j}{2} \sum_i \binom{b_i}{2} / \binom{n}{2}}.$$

Appendix B k-means and Its Related Methods

Let $\mathbf{b}_j \in \mathbb{R}^p$ denote the centroid of the j-th cluster and $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_k)^\top \in \mathbb{R}^{k \times p}$. Let $\mathcal{I}(C) = \{C^{(1)}, \dots, C^{(k)}\}$ denote the set of index set of clustering result $C \in \Omega_k^n$, where $C^{(j)}$ corresponds to the j-th cluster, that is, the sets in $\mathcal{I}(C)$ are disjoint and $\cup_{j=1}^k C^{(j)} = \{1, \dots, n\}$. The k-means problem is defined as the following optimization problem:

$$\min_{C \in \Omega_k^n, \mathbf{B} \in \mathbb{R}^{k \times p}} \sum_{i=1}^{k} \sum_{j \in C^{(i)}} \|\mathbf{x}_j - \mathbf{b}_i\|_2^2 . \tag{5}$$

The k-means algorithm finds a solution of (5) by alternately repeating the following steps: (i) Assigning each data point to the cluster whose centroid is closest in terms of

Euclidean distance. (ii) Updating each cluster centroid as the mean of the data points assigned to the cluster. This algorithm requires initial centroids as input and is sensitive to their selection.

Arthur and Vassilvitskii (2007) proposed an initialization method to address this problem. It consists of the following steps: (i) Taking the first point $\mathbf{b}_1$ uniformly randomly from $\mathcal{X}$. (ii) Taking the next point from $\mathcal{X}$ with the probability

$$\frac{\mathrm{Dist}^2(\mathbf{b})}{\sum_{\mathbf{x}\in\mathcal{X}} \mathrm{Dist}^2(\mathbf{x})},$$

where $\mathrm{Dist}(\mathbf{b})$ is the Euclidean distance from the nearest previously chosen point. (iii) Repeating the second step until k points have been taken in total. This initialization method followed by the standard k-means algorithm is called k-means++.

Let $\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \in \mathbb{R}^{n\times p}$. Then, the optimization problem (5) can be rewritten as

$$\min_{A\in\mathcal{H},\mathbf{B}\in\mathbb{R}^{k\times p}} \|\mathbf{X} - \mathbf{AB}\|_F^2 ,$$

where $\mathcal{H} := \left\{\mathbf{A} | \mathbf{A} \in \{0,1\}^{n\times k}, \mathbf{A1} = \mathbf{1}\right\}$. If the (i,j)-th element of $\mathbf{A} \in \mathcal{H}$ is 1, it indicates $i \in C^{(j)}$. Furthermore, let $\Lambda \subseteq \{1,\dots,n\} \times \{1,\dots,p\}$ denote a subset of the indices corresponding to the observed values of data points, where $(i,j) \in \Lambda$ indicates the j-th element of $\mathbf{x}_i$ is observed. We then define the projection operator P_Λ on $\mathbb{R}^{n\times p}$ as

$$[P_\Lambda(\mathbf{Y})]_{ij} = \begin{cases} y_{ij} & \text{if } (i,j) \in \Lambda, \\ 0 & \text{if } (i,j) \in \Lambda^c. \end{cases}$$

Chi et al. (2016) proposed the following optimization problem as k-means-type problem in the presence of missing values:

$$\min_{A\in\mathcal{H},\mathbf{B}\in\mathbb{R}^{k\times p}} \|P_\Lambda(\mathbf{X}) - P_\Lambda(\mathbf{AB})\|_F^2 .$$

They also developed a majorization-minimization algorithm to solve this problem and named it k-pod.

References

Al-Najdi, A., Pasquier, N., and Precioso, F. (2016). Frequent closed patterns based multiple consensus clustering. In *Artificial Intelligence and Soft Computing*, pages 14–26. Springer International Publishing.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Aschenbruck, R., Szepannek, G., and Wilhelm, A. F. (2023). Imputation strategies for clustering mixed-type data with missing values. *Journal of Classification*, 40(1):2–24.

Audigier, V. and Niang, N. (2023). Clustering with missing data: Which equivalent for rubin's rules? *Advances in Data Analysis and Classification*, 17:623–657.

Audigier, V., Niang, N., and Resche-Rigon, M. (2021). Clustering with missing data: which imputation model for which cluster analysis method? *arXiv 2106.04424*.

Basagaña, X., Barrera-Gómez, J., Benet, M., Antó, J., and Garcia-Aymerich, J. (2013). A framework for multiple imputation in cluster analysis. *American Journal of Epidemiology*, 177(7):718–725. Epub 2013 Feb 27.

Bruckers, L., Molenberghs, G., and Dendale, P. (2017). Clustering multiply imputed multivariate high-dimensional longitudinal profiles. *Biometrical Journal*, 59(5):998–1015.

Chi, J. T., Chi, E. C., and Baraniuk, R. G. (2016). k-pod: A method for k-means clustering of missing data. *The American Statistician*, 70(1):91–99.

Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 126–135.

Faucheux, L., Resche-Rigon, M., Curis, E., Soumelis, V., and Chevret, S. (2021). Clustering with missing and left-censored data: A simulation study comparing multiple-imputation-based procedures. *Biometrical Journal*, 63(2):372–393.

Forgey, E. (1965). Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics*, 21(3):768–769.

Ghaemi, R., Sulaiman, M. N., Ibrahim, H., and Mustapha, N. (2009). A survey: Clustering ensembles techniques. *International Journal of Computer and Information Engineering*, 3(2):365–374.

Gionis, A., Mannila, H., and Tsaparas, P. (2007). Clustering aggregation. *Acm Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):4–es.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Ishioka, T. (2000). Extended k-means with an efficient estimation of the number of clusters. *Japanese Journal of Applied Statistics*, 29(3):141–149.

Kim, H. J., Reiter, J. P., Wang, Q., Cox, L. H., and Karr, A. F. (2014). Multiple imputation of missing or faulty values under linear constraints. *Journal of Business & Economic Statistics*, 32(3):375–386.

Kuncheva, L., Hadjitodorov, S., and Todorova, L. (2006). Experimental comparison of cluster ensemble methods. In *2006 9th International Conference on Information Fusion*, pages 1–7.

Lee, J. W. and Harel, O. (2023). Incomplete clustering analysis via multiple imputation. *Journal of Applied Statistics*, 50(9):1962–1979.

Li, K., Wang, L., and Hao, L. (2009). Comparison of cluster ensembles methods based on hierarchical clustering. In *2009 International Conference on Computational Intelligence and Natural Computing*, volume 1, pages 499–502.

Li, T., Ding, C., and Jordan, M. I. (2007). Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 577–582. IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data.* CRC press.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(3):583–617.

Topchy, A. P., Law, M. H., Jain, A. K., and Fred, A. L. (2004). Analysis of consensus partition in cluster ensemble. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 225–232. IEEE.

van Buuren, S. and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67.

Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(3):337–372.

Vinh, N., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

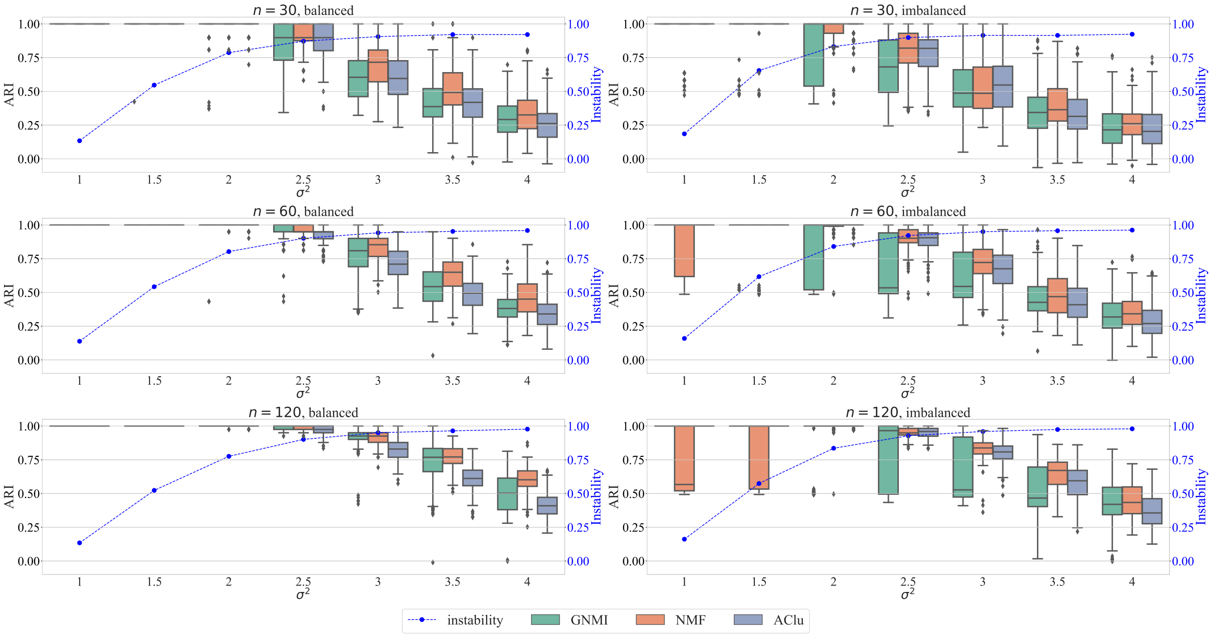

Figure 1: Adjusted rand index (ARI) of clustering ensemble algorithms and instability of clustering results to be combined in each data setting

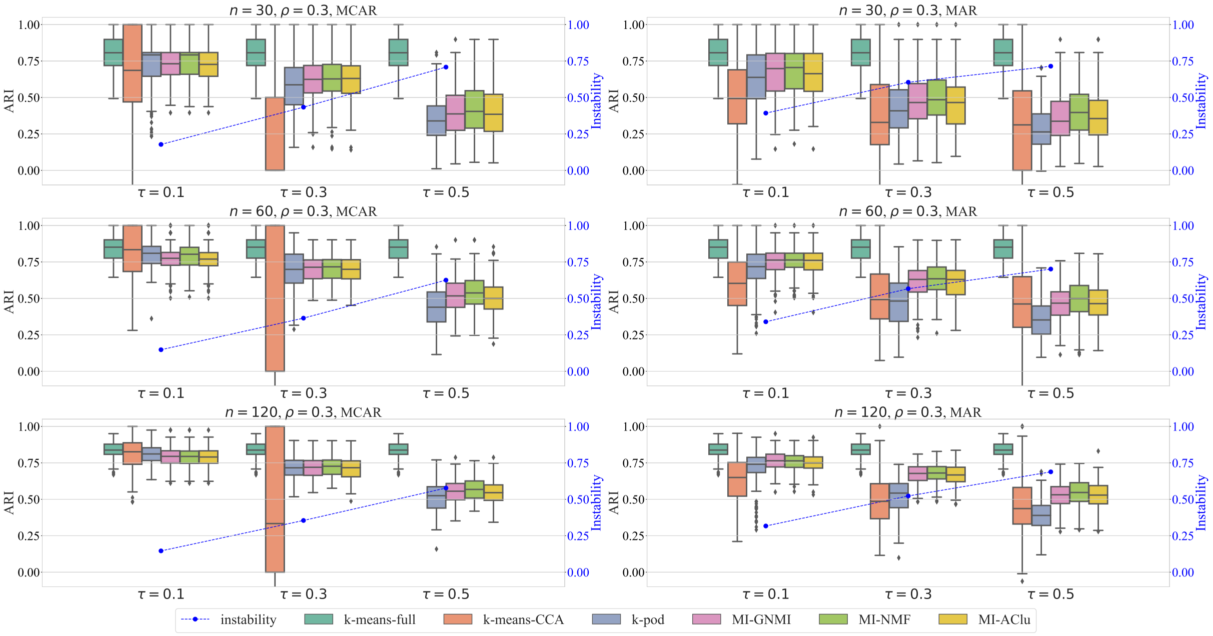

Figure 2: ARI of clustering algorithms for missing data in balanced settings with $\rho = 0.3$

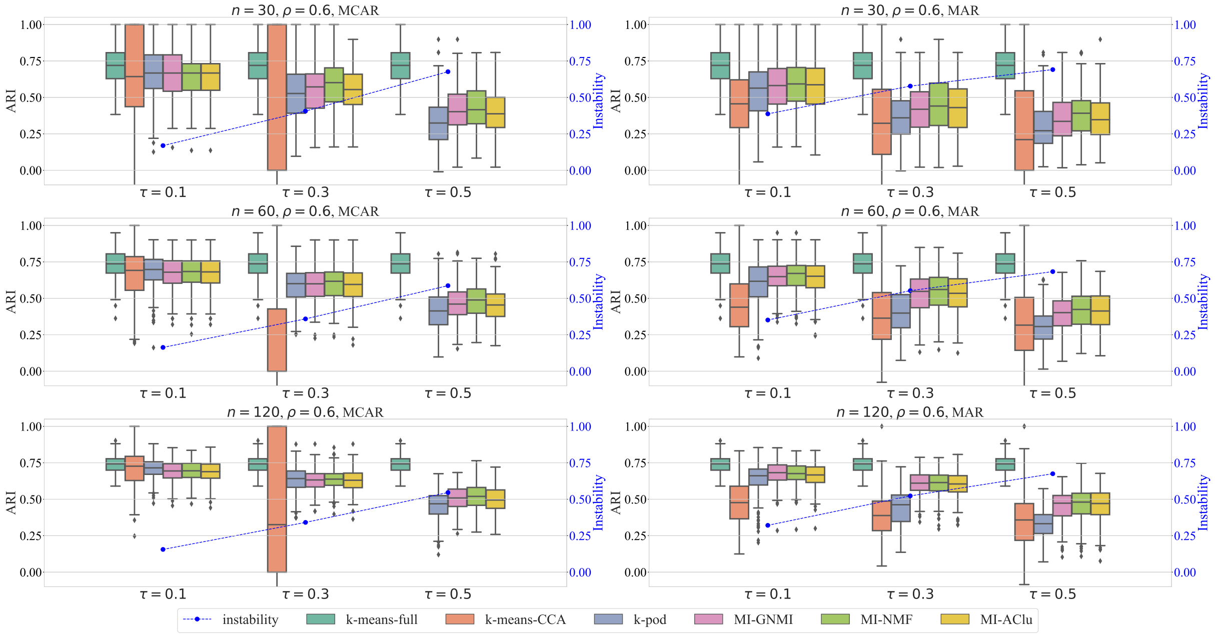

Figure 3: ARI of clustering algorithms for missing data in balanced settings with $\rho = 0.6$

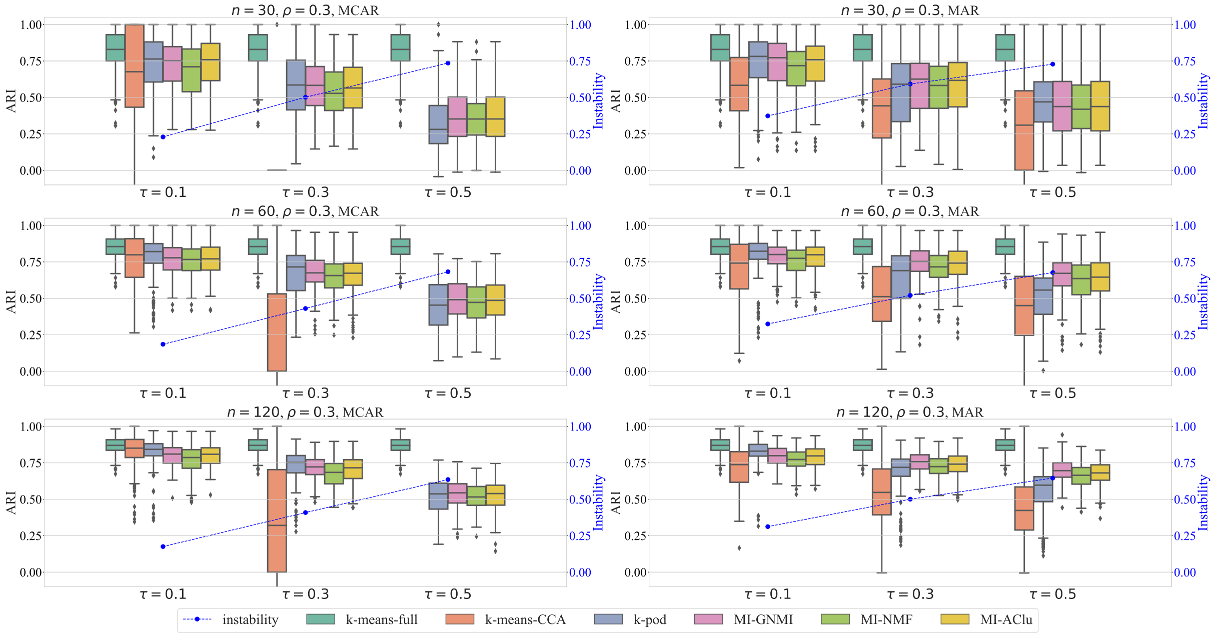

Figure 4: ARI of clustering algorithms for missing data in imbalanced settings with $\rho = 0.3$

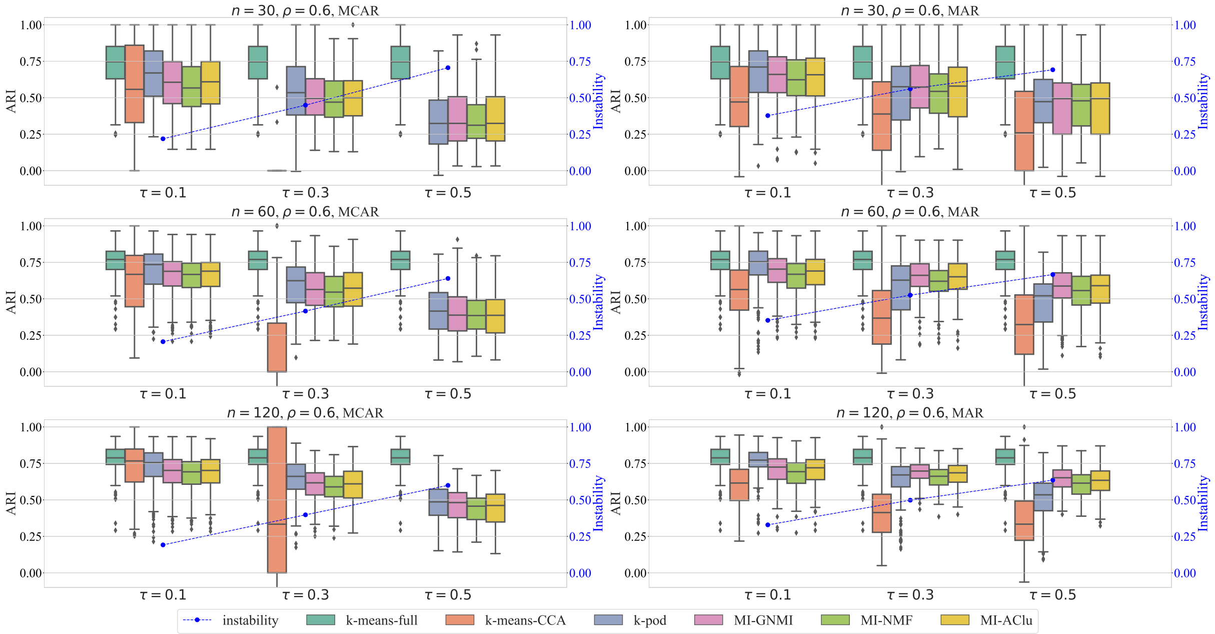

Figure 5: ARI of clustering algorithms for missing data in imbalanced settings with $\rho = 0.6$

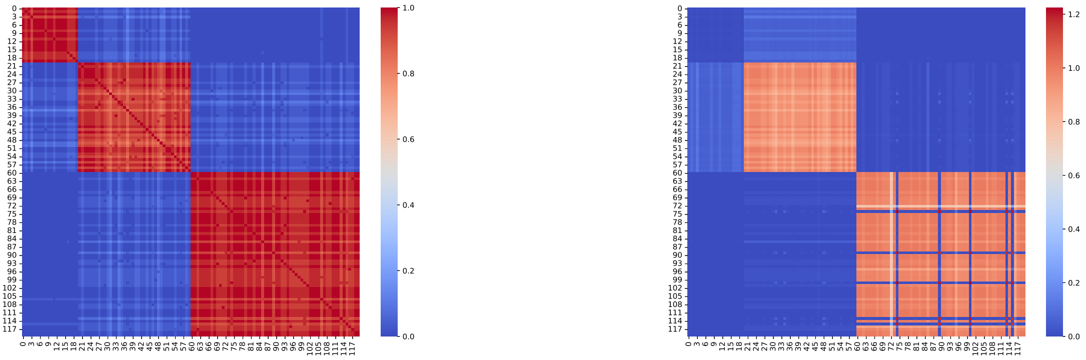

Figure 6: Example of matrix restoration failure via symmetric non-negative matrix factorization (NMF): Original matrix (left) and recovered matrix (right)