IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 4, 2019, accepted March 22, 2019, date of publication April 19, 2019, date of current version June 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2910287

K-Means Clustering With Incomplete Data

SIWEI WANG[1], MIAOMIAO LI[2], NING HU[3], EN ZHU[1], JINGTAO HU[1], XINWANG LIU[1], (Member, IEEE), AND JIANPING YIN[4]

[1]School of Computer, National University of Defense Technology, Changsha 410073, China
[2]Department of Computer, Changsha College, Changsha 410073, China
[3]Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China
[4]Dongguan University of Technology, Guangdong 523808, China

Corresponding authors: Ning Hu (huning@gzhu.edu.cn) and Miaomiao Li (miaomiaolinudt@gmail.com)

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1003203, and in part by the National Natural Science Foundation of China under Grant 61672528 and Grant 61773392.

ABSTRACT Clustering has been intensively studied in machine learning and data mining communities. Although demonstrating promising performance in various applications, most of the existing clustering algorithms cannot efficiently handle clustering tasks with incomplete features which is common in practical applications. To address this issue, we propose a novel K-means based clustering algorithm which unifies the clustering and imputation into one single objective function. It makes these two processes be negotiable with each other to achieve optimality. Furthermore, we design an alternate optimization algorithm to solve the resultant optimization problem and theoretically prove its convergence. The comprehensive experimental study has been conducted on nine UCI benchmark datasets and real-world applications to evaluate the performance of the proposed algorithm, and the experimental results have clearly demonstrated the effectiveness of our algorithm which outperforms several commonly-used methods for incomplete data clustering.

INDEX TERMS K-means clustering, incomplete data, imputing method.

I. INTRODUCTION

Clustering has been intensively studied in machine learning and data mining communities [1]–[5]. It aims to find the underlying intrinsic structure of each cluster from given data. Various clustering algorithms have been proposed in practical applications, including K-means [6]–[9], Fuzzy cmeans [10], Dbscan [11], [12], Hierarchical clustering [13] and Gaussian Mixture Model (GMM) [14], [15], to name just a few. These aforementioned clustering algorithms have shown promising clustering performance and are widely applied into various applications. For example, the clustering algorithm has been proposed to discover the latent factors for community identification and summarization [16].

Although achieving great success, existing clustering algorithms cannot efficiently cope with the situation when the data has incomplete features [17], [18]. In real world applications, incompleteness and uncertainty in th data can be formed by various factors: sensor failure, measurement errors and unreliable features [19]–[21]. Many approaches have been proposed to handle incompleteness with supervised tasks [22]–[26]. They normally impute the incomplete features firstly and learn with complete data matrix. However very few methods are under unsupervised settings [27]. For example, the expectation-maximization (EM) algorithm has been applied to incomplete data clustering as well as zero-filling and mean filling [28]–[30]. Though demonstrating promising performance, we observe that these imputation methods treat the imputation and clustering processes separately and therefore the imputed entries may not be served for clustering. Hence the imputing quality of uncertain values plays an essential role in the success of clustering task while the existing methods usually lead to poor performances.

To address this issue, we propose a novel k-means based clustering algorithm to handle incomplete data which unifies the clustering and imputation into one objective function. We integrate the imputation and clustering steps into one process and these two processes are guided by each other serving for better clustering. The missing entries of the data matrix have been alternately optimized in order to better serve for the clustering task and reveal the inner structures in each cluster. After that, we propose a three-step alternate optimization algorithm with proved convergence to solve the resultant optimization problem. Extensive experimental study has been conducted on nine widely used UCI benchmark datasets and real-world applications to evaluate clustering

The associate editor coordinating the review of this manuscript and approving it for publication was Anand Paul.

performance of the proposed algorithm. As indicated, our algorithm consistently achieves state-of-the-art performance comparing to other imputing methods.

We summarize our main contributions of this paper as follows:

(i) Different from existing algorithms where the imputation and clustering are separately performed, we unify both precesses into a single optimization objective. The missing features are alternately imputed with better serving for clustering while the existing observed entries remain unchanged during the whole process.

(ii) A novel adaptive approach termed (*K-means Clustering with Incomplete Data*) is proposed to fulfill the aforementioned idea. Besides, we design an alternate algorithm to solve the resultant optimization problem in incomplete data clustering with fast convergence.

(iii) Extensive experimental study has been conducted on several UCI benchmark datasets and large real-world applications. As indicated, our algorithm consistently achieves state-of-the-art performance when compared to other imputation methods. The experimental results verify the effectiveness and superiority of our algorithm.

The rest of this paper is organized as follows. Section II outlines the related work of several imputing methods and k-means clustering method. Section III presents the proposed optimization objective function and the three-step alternate algorithm. Section IV shows the experiment results with evaluation. Section V concludes the paper.

II. RELATED WORKS

A. IMPUTING METHODS

Many prior methods have been proposed to impute the missing entries of the data matrix in the literature. These algorithms can be grouped into two categories: heuristic methods and statistical ones. In the following section, we will briefly introduce these widely used filling approaches and discuss their differences.

1) HEURISTIC METHODS

The foundation of heuristic approaches on dealing with incompleteness is intuitive where heuristic information is used to reduce the missing values, and then existing clustering algorithms can be applied into the imputed data matrix $\mathbf{X}$.

One natural idea for handling with incomplete data is to remove the data samples which have missing entries. In other words, this method generates a fully-observable new data set from the original incomplete data matrix. Its clustering performance could be acceptable in the case that the incompleteness rate is relatively small (for example, less than 10%). This technology has been widely used into the medical field and can be very easily accomplished.

However, they ignore the information of missing values better serving for seeking underlying patterns and could result in a severe reduction of original data matrix when the missing rate is high. Moreover, the previous technology could not handle the learning tasks of incomplete sample vectors. Although easy and simple to implement, heuristic approaches are always unsatisfactory as it is extremely hard to predict their performance for a learning task.

2) STATISTICAL METHODS

Different from the aforementioned heuristic methods, the statistical methods seek more useful information from the missing data. The majority of them impute the missing values by statistical properties and do not discard the incomplete information. They fill the incomplete entries with constant number to take the complete data sample and applied to learning tasks.

The simplest filling values and also the most commonly-used ones are zero, conditional mean number, median and modal number in that dimension. Moreover, the missing entries could be imputed by doing regression on the complete features. Specially, the KNN-filling method has been proposed to impute the missing entries with the the mean feature on the K-closest neighbors in that dimension [31], [32]. The work in [33] proposes a neural network which is able to deal with incomplete data both in training and testing stages. Another popular approach, i.e., fuzzy c-means imputation (FCMI) utilizes the property of fuzzy degree to fill in the missing values [34]. The algorithm replaces the incomplete entries with the mean of complete data on that dimension during each iteration. All of the aforementioned statistical approaches suffer from the selection of hyper-parameters in their algorithms, e.g., fuzzy coefficient, neighbor rate and regularization coefficient, which significantly limits the usage of algorithms to real-world applications.

Different from the aforementioned methods, another statistical class to deal with incomplete features is followed by Bayesian frameworks [35], [36]. These frameworks are often shown in a maximum-likelihood manner, which imputes the missing values with the most-likely estimated numbers. The most followed or popular method is expectation-maximization (EM) algorithm was put forward in [37]. Suppose the given data set is $\mathbf{X} = \{\mathbf{x}\}_i^n$ consisting n samples. The assumption follows that the data vectors are sampled from the parametric model of a density function $p(\mathbf{x}|\theta)$ with latent parameter θ. Further we define that $\mathbf{X}^o$ represents the observable part of data matrix (complete patterns) and $\mathbf{X}^m$ represents the missing entries of incomplete data.

The EM algorithm is a two-step alternate optimization approach, which consists of expectation step and maximum steps [38]. In the E-step, the latent parameter θ is estimated by the complete data $\mathbf{X}^o$ and the $\mathbf{X}^m$ is filled with conditional expectations. The next M-step, the method calculates the maximum-likelihood estimation of θ. These two steps are repeated until convergence. The EM algorithm usually demonstrates good estimation quality of missing values at the cost of long training time.

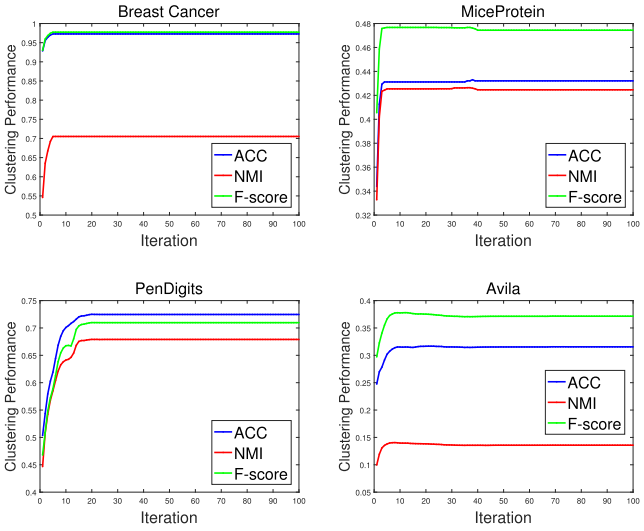

FIGURE 1. The ACC, NMI and F-score of the imputed data matrix at each iterations on nine Breast Cancer, MiceProtein, PenDigits and Avila.

B. K-MEANS ALGORITHM

K-means algorithm is the most widely-applied clustering algorithm in real-world pattern recognition applications [9]. The algorithm seeks a well-defined partition such that the squared distance of with-in clusters is minimized. Suppose $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ is a collection of n samples taken from data matrix. The objective of k-means clustering is to minimize the sum of the square of the within-cluster distance. By taking the assignment matrix $\mathbf{H} \in \{0,1\}^{n\times k}$, the optimization objective of K-means algorithm could be written as follows:

$$\min_{\mathbf{H}\in\{0,1\}^{n\times k}} \sum_{i=1}^{n}\sum_{c=1}^{k} \mathbf{H}_{ic}\|\mathbf{x}_i - \mu_c\|^2 \; s.t. \sum_{c=1}^{k} \mathbf{H}_{ic} = 1. \quad (1)$$

where $n_c = \sum_{i=1}^{n} \mathbf{H}_{ic}$ and $\mu_c = \frac{1}{n_c}\sum_{i=1}^{n} \mathbf{H}_{ic}\mathbf{x}_i$ are the number and centroid of the $c-th$ $(1 \le c \le k)$ cluster respectively.

Directly solving the optimization problem in Eq. (1) is difficult since minimizing the squared from of with-in distance is proved to an NP-hard problem [39]. As a result, many methods have been proposed to solve Eq. (1) and the most popular algorithm is as follows [40]:

(i) Setting k cluster centers as an initialization;
(ii) According to the k centers, generating a new partition which assigns each sample to its closest center;
(iii) Computing the new cluster centers from assignment matrix and data matrix.

By taking the aforementioned steps, K-means algorithm converges to a local minimum. Serving for better clustering performance, the K-means method could be run in a couple of times to approximate the global minimal value. As seen, it is a hard assignment algorithm as each sample is assigned into a single cluster.

Although widely used in practical applications, K-means and its variants can not efficiently deal with clustering tasks with incomplete features, which is not uncommon in practical applications. In the following, we design a novel algorithm termed K-means Clustering with Incomplete Data to address this issue.

III. K-MEANS CLUSTERING WITH INCOMPLETE DATA

Different from previous work which separates the imputing and clustering into two independent processes, we decide to dynamically fill the missing values with considerations of serving better clustering performance. For incomplete data, each sample $\mathbf{x}_i\,(1 \le i \le n)$ can be divided into two parts: the observable features $\mathbf{x}_i(o_i)$ and missing features $\mathbf{x}_i(m_i)$. Besides optimizing the assignment matrix $\mathbf{H}$ and cluster centers $\mu_\mathbf{c}\,(1 \le c \le k)$ in the commonly-used K-means algorithms, we propose to optimize additional variables $\mathbf{x}_i(m_i)$. Meanwhile, the observed features $\mathbf{x}_i(o_i)$ are kept unchanged during the optimization process.

A. PROPOSED FORMULATION

Based on the aforementioned discussion, we redefine the objective of K-means algorithm to cope with incomplete data clustering. Given data matrix $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and the number of clusters k, we have three variables to be optimized: the data matrix $\mathbf{X}$, assignment matrix $\mathbf{H}$ and the clusters' centers $\boldsymbol{\mu}_c\,(1 \le c \le k)$. By imposing the constraint on $\mathbf{X}$, we set our K-means clustering with incomplete data as follows:

$$\begin{aligned} &\min_{\mathbf{H},\{\mu_c\}_{c=1}^k,X} \sum_{i=1}^{n}\sum_{c=1}^{k} \mathbf{H}_{ic}\|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2 \\ &s.t. \sum_{c=1}^{k} \mathbf{H}_{ic} = 1, \quad \mathbf{x}_i(o_i) = \mathbf{x}_i^o, \; \forall i, \end{aligned} \quad (2)$$

where $n_c = \sum_{i=1}^{n} \mathbf{H}_{ic}$ and $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{n} \mathbf{H}_{ic}\mathbf{x}_i$ are the number and centroid of the $c-th$ $(1 \le c \le k)$ cluster and $\mathbf{x}_i(o_i)$ represents the complete elements of the i-th sample $\mathbf{x}_i$. Moreover, we also impose constraints on the observable part of data matrix $\mathbf{x}_i(o_i)$ to ensure their values are kept unchanged during optimization process.

B. OPTIMIZATION

As seen, the additional constraint $\mathbf{x}_i(o_i) = \mathbf{x}_i^o$ makes the whole optimization problem difficult to solve. In order to solve it, we design a three-step alternate optimization algorithm with a fast convergence rate, where each step can be easily solved by applying the existing off-the-shelf packages.

Optimizing $\mathbf{H}$ *with fixed* $\mathbf{X}$ *and* $\boldsymbol{\mu}$: With data matrix $\mathbf{X}$ and the clusters' centers $\boldsymbol{\mu}_c$ being fixed, the optimization Eq. (2) can be equivalently rewritten as follows,

$$\min_{\mathbf{H}} \sum_{i=1}^{n}\sum_{c=1}^{k} \mathbf{H}_{ic}U_{ic}^2 \; s.t. \sum_{c=1}^{k} \mathbf{H}_{ic} = 1, \quad \mathbf{H} \in 0, 1^{n\times k} \quad (3)$$

where $U_{ic}^2 = \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2$.

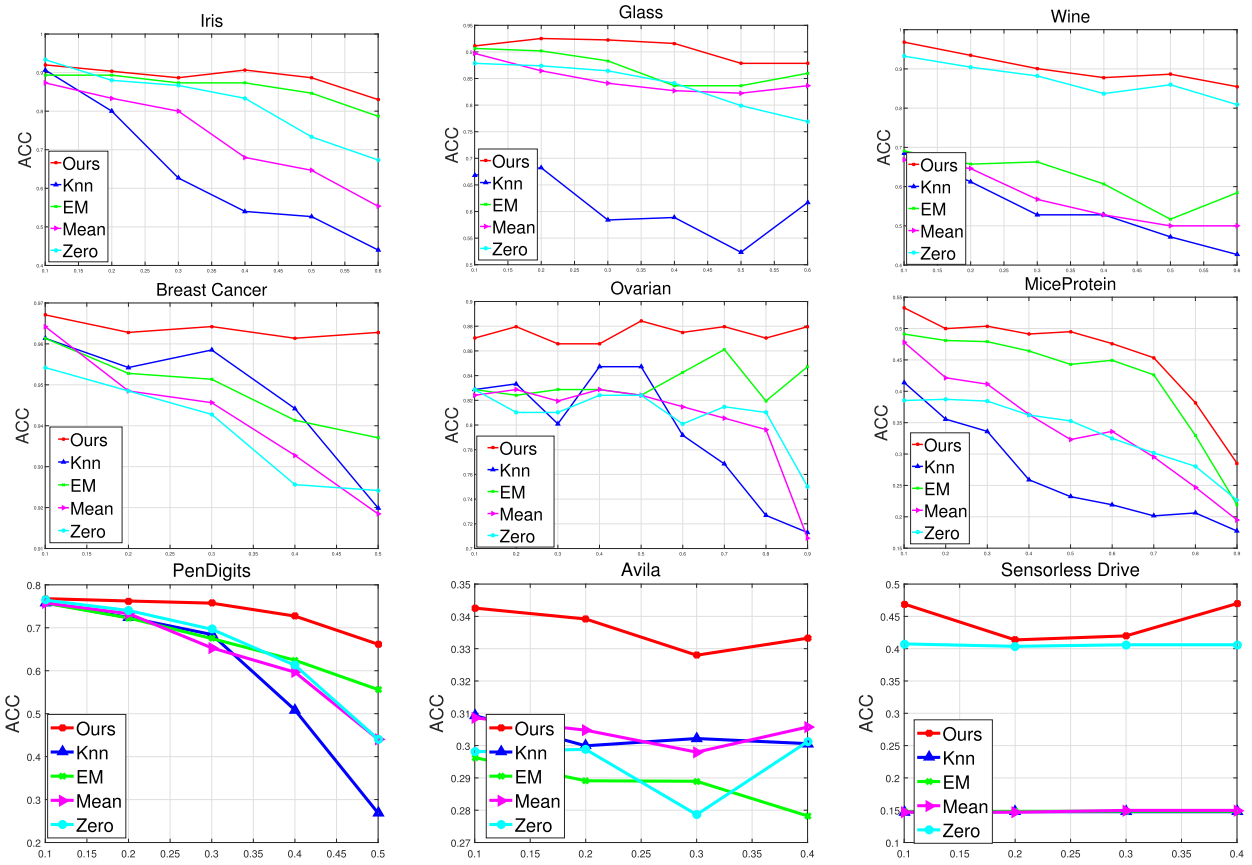

FIGURE 2. The ACC of the compared algorithms with the variation of missing ratios on nine benchmark datasets.

The problem in Eq. (3) can be divided into n sub-problems with consideration to each sample $\mathbf{x}_i$,

$$\sum_{i=1}^{n}\sum_{c=1}^{k}\mathbf{H}_{ic}U_{ic}^2 = \sum_{i=1}^{n}(\mathbf{H}_{i1}U_{i1}^2 + \mathbf{H}_{i2}U_{i2}^2 + \cdots + \mathbf{H}_{ik}U_{ik}^2) \tag{4}$$

Each row of assignment matrix $\mathbf{H}$ has only one entry of 1 while others are 0. To minimize Eq. (4) with the hard assignment constraint, each sample $\mathbf{x}_i$ should be assigned into its closest cluster center as their distance will be minimized. As a result, the assignment matrix $\mathbf{H}$ could be updated by calculating the distances.

Optimizing μ *with fixed* $\mathbf{X}$ *and* $\mathbf{H}$: With $\mathbf{X}$ and $\mathbf{H}$ being fixed, the data matrix is complete and assignment matrix is given. Therefore, the optimization problem in Eq. (2) is equivalent to Eq. (5) as follows,

$$\min_{\mu_c\ (1\le c\le k)}\sum_{i=1}^{n}\sum_{c=1}^{k}\mathbf{H}_{ic}\|\mathbf{x}_i-\boldsymbol{\mu}_c\|^2 \tag{5}$$

For each cluster center $\boldsymbol{\mu}_c\ (1\le c\le k)$, the above equation can be rewritten into k sub-problems,

$$\sum_{i=1}^{n}\sum_{c=1}^{k}\mathbf{H}_{ic}\|\mathbf{x}_i-\boldsymbol{\mu}_c\|^2 = \sum_{c=1}^{k}(\mathbf{H}_{1c}\|\mathbf{x}_1-\boldsymbol{\mu}_c\|^2 + \mathbf{H}_{2c}\|\mathbf{x}_2-\boldsymbol{\mu}_c\|^2+\cdots+\mathbf{H}_{nc}\|\mathbf{x}_n-\boldsymbol{\mu}_c\|^2) \tag{6}$$

For every single cluster center $\boldsymbol{\mu}_c\ (1\le c\le k)$, the sub-problem is a simple quadratic function and has a closed-form solution as follows,

$$\boldsymbol{\mu}_c = \frac{\sum_{i=1}^{n}H_{ic}\mathbf{x}_i}{\sum_{i=1}^{n}H_{ic}}. \tag{7}$$

Optimizing $\mathbf{X}$ *with fixed* $\mathbf{H}$ *and* $\boldsymbol{\mu}$: As aforementioned, the sample $\mathbf{x}_i$ is divided into two parts: the observable features $\mathbf{x}_i(o_i)$ and missing features $\mathbf{x}_i(m_i)$. The $\mathbf{x}_i(o_i)$ is enforced to keep unchanged during process. With the assignment matrix $\mathbf{H}$ and cluster centers $\boldsymbol{\mu}$ being fixed, the optimization problem in Eq. (2) is equivalent to the optimization problem as follows,

$$\begin{aligned}&\min_{\{\mathbf{x}_i\}_{i=1}^n}\ \sum_{i=1}^{n}\sum_{c=1}^{k}\mathbf{H}_{ic}\|\mathbf{x}_i-\boldsymbol{\mu}_c\|^2\\ &s.t.\ \mathbf{x}_i(o_i)=\mathbf{x}_i^o,\quad \forall i,\end{aligned} \tag{8}$$

where $\mathbf{x}_i = [\mathbf{x}_i(o_i), \mathbf{x}_i(m_i)]$, and $\mathbf{x}_i(o_i)$ and $\mathbf{x}_i(m_i)$ represent the complete and missing entries of the i-th sample $\mathbf{x}_i$, respectively.

At a first glance, the optimization problem in Eq. (8) is difficult to solve due to the equality constraint. However, we observe that the objective can be divided into the sum of n samples and the equality constraints are independent. Therefore, we can equivalently solve Eq. (8) by solving n

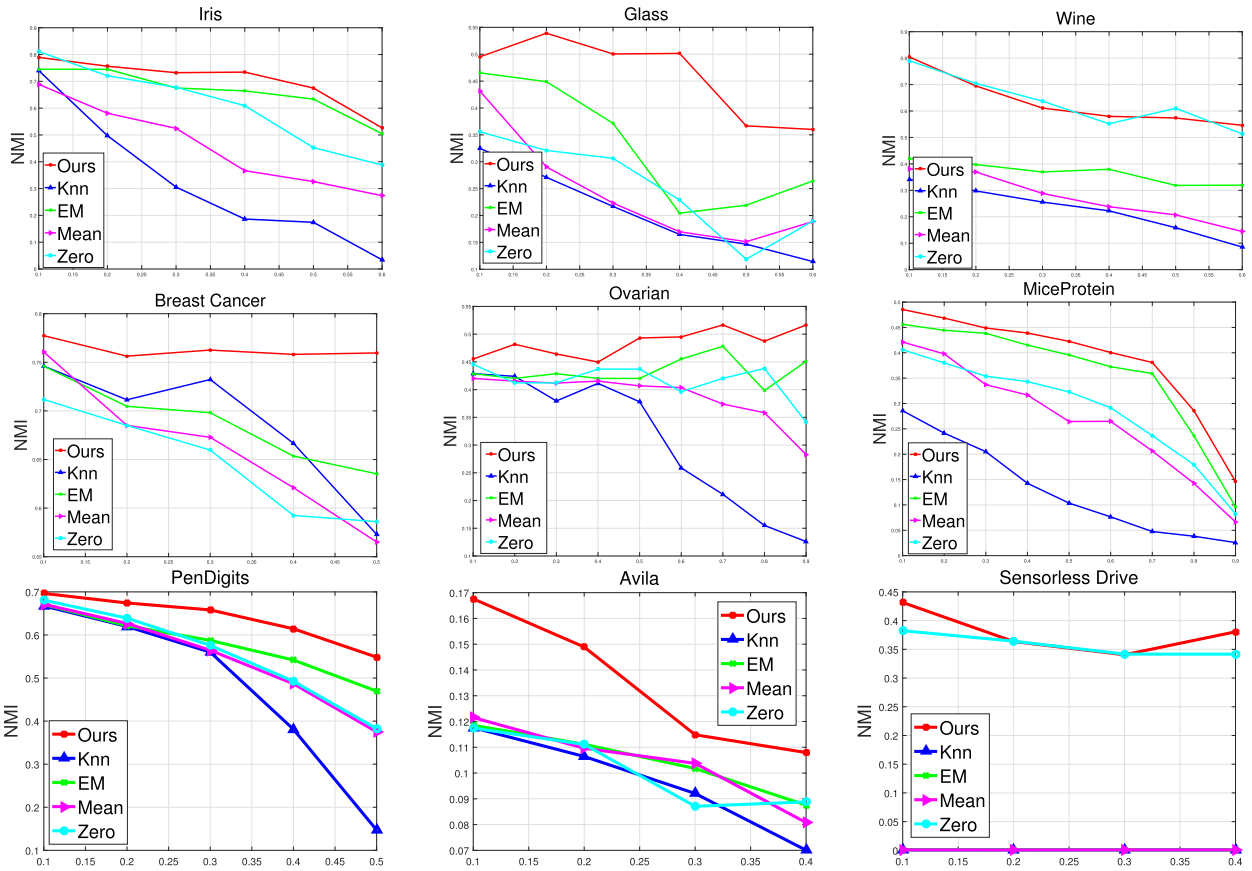

FIGURE 3. The NMI of the compared algorithms with the variation of missing ratios on nine benchmark datasets.

sub-problem as follows,

$$\min_{\mathbf{x}_i} \sum_{c=1}^{k} \mathbf{H}_{ic} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2 \quad s.t.\ \mathbf{x}_i(o_i) = \mathbf{x}_i^o, \tag{9}$$

Further, based on $\mathbf{x}_i = [\mathbf{x}_i(o_i), \mathbf{x}_i(m_i)]$, the Eq. (9) can be rewritten as

$$\min_{\mathbf{x}_i} \sum_{c=1}^{k} \mathbf{H}_{ic} \left(\|\mathbf{x}_i(o_i) - \boldsymbol{\mu}_c(o_i)\|^2 + \|\mathbf{x}_i(m_i) - \boldsymbol{\mu}_c(m_i)\|^2 \right) \quad s.t.\ \mathbf{x}_i(o_i) = \mathbf{x}_i^o, \tag{10}$$

which can be further rewritten as a unconstrained optimization problem as follows,

$$\min_{\mathbf{x}_i} \quad \textstyle\sum_{c=1}^{k} \mathbf{H}_{ic} \|\mathbf{x}_i(m_i) - \boldsymbol{\mu}_c(m_i)\|^2. \tag{11}$$

This is because the first term in Eq. (10) is a constant. As a result, the optimum of Eq. (11) can be analytically expressed as

$$\mathbf{x}_i(m_i) = \sum_{c=1}^{k} \mathbf{H}_{ic} \boldsymbol{\mu}_c(m_i). \tag{12}$$

As seen from Eq. (12), the missing elements of each sample $\mathbf{x}_i$ is imputed with the corresponding dimension of its cluster center.

Algorithm 1 K-Means Clustering With Incomplete Data

1: **Input**: incomplete data matrix $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, number of cluster k, convergence tolerance ϵ_0 and missing index m.
2: **Output**: complete data matrix $\mathbf{X}$ and assignment matrix $\mathbf{H}$.
3: Initialize the missing values by mean values.
4: **repeat**
5: Update $\mathbf{H}$ by solving Eq. (4) with fixed data matrix $\mathbf{X}$ and cluster centers $\boldsymbol{\mu}_c$ $(1 \le c \le k)$.
6: Update cluster centers μ_c with fixed $\mathbf{H}$ and $\mathbf{X}$ by Eq. (7).
7: Update each data sample $\mathbf{x}_i$ by solving Eq. (12) with fixed $\mathbf{H}$ and cluster centers $\boldsymbol{\mu}_c$ $(1 \le c \le k)$.
8: $t = t + 1$.
9: **until** $\left(\text{obj}^{(t-1)} - \text{obj}^{(t)}\right)/\text{obj}^{(t)} \le \epsilon_0$

C. CONVERGENCE AND COMPLEXITY ANALYSIS

Convergence: Our algorithm is outlined in Algorithm 1, where $\text{obj}^{(t)}$ denotes the objective value at the t-th iteration. At each iteration, the objective of Algorithm 1 is monotonically decreased when optimizing one variable with others fixed. At the same time, the whole optimization problem is lower-bounded by zero. As a result, the proposed algorithm is theoretically guaranteed to converge to a local minimum.

TABLE 1. Datasets used in our experiments.

Dataset	#Samples	#Dimensions	#Number of Clusters
Iris	150	4	3
Wine	178	13	3
Glass	214	9	2
Ovarian	216	100	2
Breast Cancer	699	9	2
MiceProtein	1077	77	8
PenDigits	10992	16	10
Avila	20871	10	12
Sensorless Drive	37715	48	8

TABLE 2. Datasets used in our experiments.

Dataset	Missing Ratio
Iris	$10\% - 60\%$
Wine	$10\% - 60\%$
Glass	$10\% - 60\%$
Ovarian	$10\% - 90\%$
Breast Cancer	$10\% - 50\%$
MiceProtein	$10\% - 90\%$
PenDigits	$10\% - 50\%$
Avila	$10\% - 40\%$
Sensorless Drive	$10\% - 40\%$

We also record the objective at each iteration and the results validate the convergence. In addition, we observe that the proposed algorithm usually converges in less than ten iterations in our experiments.

Complexity: Comparing to the k-means algorithm, our Algorithm 1 considers the data matrix $\mathbf{X}$ as another variable to be optimized. In Eq. (12), we replace the missing values with the related cluster centers. Therefore, the time complexity of our algorithm is $\mathcal{O}(tkmd)$, where t, k, n, d represents the number of iterations, clusters, samples and dimensions respectively.

IV. EXPERIMENTS

A. DATASETS

We evaluate the proposed algorithm on several UCI and several large benchmark dataset. They are Iris, Wine, Glass, Breast Cancer, Mice Protein,[1] Ovarian Cancer Dataset, PenDigits [41], Avila and Sensorless Drive. The detailed information of these datasets is listed in Table 1.

For all the datasets provided in Table 1, we randomly generate the incompleteness by the original complete data matrix. The incompleteness of the used datasets are listed in Table 2. We have also uploaded the incomplete datasets at Github.[2]

The first five datasets, including Iris, Wine, Glass, Ovarian and Breast Cancer [3] are the most commonly-used benchmarks for incomplete data clustering. **Mice Protein** is a dataset that consists of the expression levels of 77 proteins measured in the cerebral cortex of eight classes of control and trisomic mice. Differently, **PenDigits** is a hand-written digits with 10992 samples for 10 classes. Moreover, Avila and Sensorless Drive are downloaded from the UCI Machine Learning Repository. The **Avila** dataset[4] has been extracted from 800 images of the 'Avila Bible', an XII century giant Latin copy of the Bible, which has 20871 samples in 12 classes. The **Sensorless Drive** dataset [5] extracts the features from electric current drive signals, resulting in 37715 samples with 8 classes.

TABLE 3. The aggregated ACC, NMI, and F-score comparison of different imputing algorithms on nine benchmark dataset.

Datasets \ Methods	KNN	EM	MF	ZF	Ours
ACC					
Iris	64.00	86.11	73.11	82.00	88.89
Glass	61.06	87.07	84.81	83.77	90.53
Wine	54.21	61.99	56.84	87.08	90.37
Breast Cancer	94.76	94.88	94.19	93.91	96.37
Ovarian	83.15	82.69	82.50	81.94	87.31
MiceProtein	31.94	47.17	39.94	37.44	50.46
PenDigits	58.88	66.72	63.62	65.09	73.53
Avila	30.30	28.82	30.43	29.42	33.57
Sensorless Drive	14.76	14.83	14.84	40.58	44.31
NMI					
Iris	32.30	66.13	46.02	61.00	70.23
Glass	20.65	32.89	24.25	25.34	46.05
Wine	22.69	36.75	27.13	63.46	63.50
Breast Cancer	68.59	68.76	66.09	64.70	76.28
Ovarian	40.43	42.36	41.39	42.88	46.88
Mice Protein	19.57	43.00	34.75	36.12	45.28
PenDigits	47.51	57.82	54.49	55.42	63.83
Avila	9.65	10.47	10.39	10.11	13.48
Sensorless Drive	0.07	0.08	0.07	35.75	37.90
F-score					
Iris	65.68	86.38	73.64	81.95	88.57
Glass	63.78	86.62	83.55	81.47	90.37
Wine	57.02	65.89	59.15	87.03	89.01
Breast Cancer	94.70	94.84	94.11	93.83	96.38
Ovarian	83.09	82.60	82.42	81.78	87.36
Mice Protein	33.71	49.67	42.87	40.78	51.57
PenDigits	59.04	66.88	63.91	64.81	72.80
Avila	32.64	31.75	33.89	32.74	36.10
Sensorless Drive	23.41	23.45	23.77	43.03	45.95

B. COMPARED ALGORITHM

In literature, the missing elements among data are firstly imputed, and then the traditional K-means algorithm is applied into the imputed dataset. The widely used imputation algorithms include,

- KNN Filling (KNN-Filling) [32]: The missing values are filled with the mean feature of the K-nearest neighbors.
- Expectation Maximum (EM) [37]: The algorithm estimates the model parameters for filling incomplete data.
- Mean Filling (MF) [30]: The algorithm fills the missing values with mean values, as introduced in the related work.
- Zero Filling (ZF): The algorithm firstly standardizes the data matrix and imputes zeros on the missing values.

[1] http://archive.ics.uci.edu/ml/datasets.html
[2] https://github.com/wangsiwei2010/k-means-filling /tree/master/dataset
[3] http://archive.ics.uci.edu/ml/datasets.html
[4] http://archive.ics.uci.edu/ml/datasets/Avila
[5] https://archive.ics.uci.edu/ml/datasets/dataset+for+sensorless+drive+diagnosis

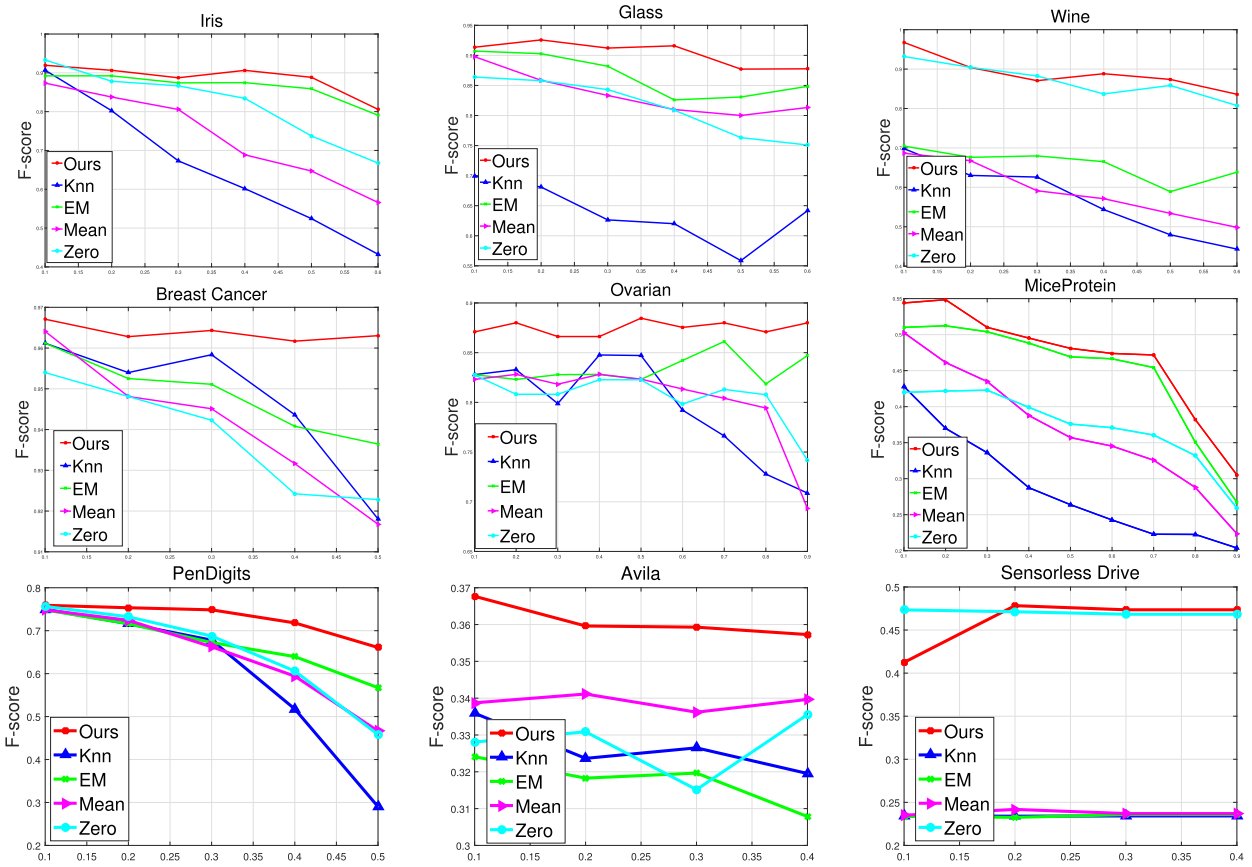

FIGURE 4. The F-score of the compared algorithms with the variation of missing ratios on nine benchmark datasets.

C. EXPERIMENT SETTINGS

In all our experiments, it is assumed that the true number of clusters is pre-specified. The widely used clustering accuracy (ACC), normalized mutual information (NMI) [42] and F-score are applied to evaluate the clustering performance of each algorithm. For all algorithms, we repeat each experiment for 100 times with random initialization to reduce the effect of randomness caused by K-means, and report the best result. All of the code is implemented in Matlab and available at Github.[6]

D. EXPERIMENTAL RESULTS

The aggregated ACC, NMI and F-score of the compared algorithms on the seven benchmark datasets are reported in Table 3, where the best results are painted in red. We also plot the three evaluation metrics by the mentioned algorithms on each dataset in Figure 2, 3 and 4. From these results, we have the following observations:

- Our proposed algorithm always achieves the state-of-the-art on all the nine benchmarks datasets. Meanwhile, it is much more robust compared to other algorithms when the missing rate is relatively high. Taking the aggregated results on the several large datasets PenDigits, Avila and Sensorless Drive as examples, our algorithm outperforms the second best algorithm by **10.2**%, **10.8**% and **9.2**% on terms of ACC, **10.4**%, **29.3**% and **6.0**% on terms of NMI and **8.9**%, **10.6**% and **6.8**% on terms of F-score. Comparing to the small datasets like Iris, Wine and Glass, our algorithm significantly improves the clustering performances on the large incomplete datasets.
- As a strong baseline, the EM algorithm has been considered to be a popular choice for incomplete estimation. It indeed achieves comparable performance with the proposed algorithm on four datasets. As the experimental results show, the EM usually obtains poor performance due to the lack of sufficient information when the missing ratio is significantly high. Similarly, the KNN-filling always leads to poor estimation of missing patterns due to the lack of enough complete samples.
- As our algorithm indicates, we first cover the missing entries with the mean values of those dimensions the same as Mean-filling. However, the difference between our proposed method and Mean-filling is that we dynamically optimize the incomplete patterns at each iteration. The proposed algorithm significantly outperforms the mean-filling method, which demonstrates the effectiveness of dynamic optimization.

The clustering results of compared algorithms illustrate the effect of applying the dynamic-filling into our

[6]https://github.com/wangsiwei2010/k-means-filling /tree/master

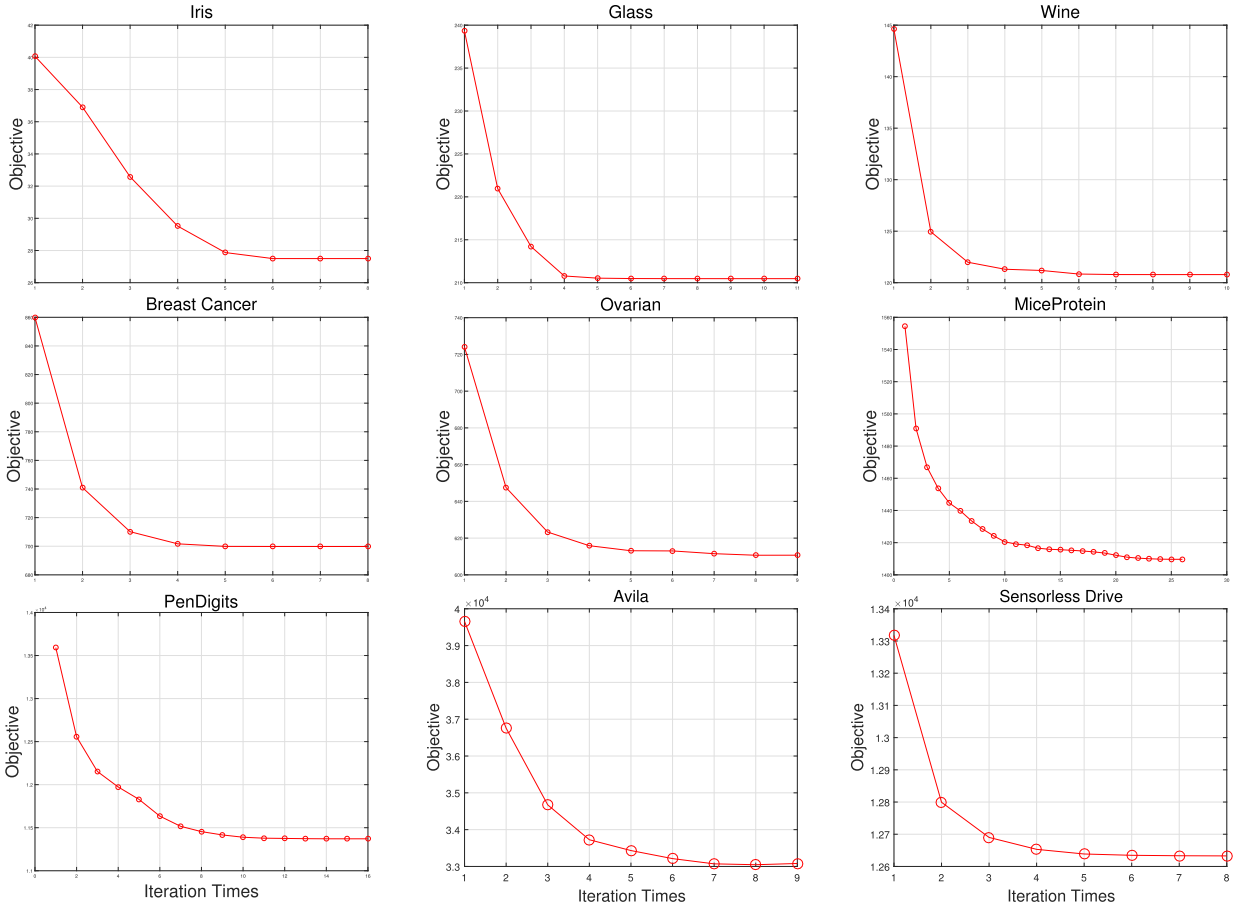

FIGURE 5. The variation of the objective function values with iterations on nine benchmark datasets.

clustering tasks. While keeping the observable part of data matrix unchangeable, the missing entries are able to be optimized during the process. The samples themselves should get closer to their centers and stay away from other clusters.

We also plot the objective value of our algorithm at each iteration in Figure 5. As observed, this value is monotonically decreased and the algorithm usually converges in very few iterations.

E. EFFECTIVENESS OF IMPUTED VALUES FOR MISSING ENTRIES

To further evaluate the effectiveness of our proposed algorithm, we conduct experiments to show the evolution of the imputed patterns during the learning procedure. Specifically, we evaluate the ACC, NMI and F-score of our algorithm based on the data matrix $\mathbf{X}$ at each iteration on Breast Cancer, MiceProtein, PenDigits and Avila datasets and plot them in Figure 1.

From these figures, we observe that the clustering performance on the four large datasets gradually increases and then maintains a stable maximum with the increasing iterations. These experiments have clearly demonstrated the effectiveness of our k-means filling, indicating better serving for clustering and inner cluster structures.

F. DISCUSSIONS AND EXTENSIONS

In this section, we discuss the proposed method for incomplete data clustering and offer some extensions of our algorithm.

Discussions: From the above experiments, we can conclude that our proposed algorithm has the following advantages: i) dynamically estimates the missing entries of the given data matrix serving for better clustering performance; and ii) is more robust than several popular incomplete approaches across a wide range of missing ratios.

It is worthy to notice that comparing to those two-imputed algorithms(mean, zero and KNN), our algorithm adopt the one-stage process and jointly optimize the clustering loss and imputing qualities into one problem. Therefore, the clustering-guided manner leads better imputation of missing entries and in return beneficial serving for clustering.

Extensions: Our methods can be further improved by the following aspects. Firstly, the initialization values for the missing entries can be readily extended to other

statistical values. Future work of exploiting different initialization values would be an interesting work. Secondly, for high-dimensional data representations, the metric for clustering is far more complicated than the normal k-means. The performance of our method on high-dimensional dataset could be further improved by adjusting appropriate metrics.

V. CONCLUSION

Data incompleteness is common in real-world applications and many efforts have been devoted to deal with incomplete data clustering. Existing methods firstly estimate the missing values and then the imputed data is then fed into traditional clustering algorithms. Different from those approaches, we have proposed a k-means filling method to dynamically optimize the missing values. The incomplete entries are filled with the value of their belonging centers in order to better serve for performance. The proposed algorithm, i.e., K-means Clustering with Incomplete Data, demonstrates the state-of-the-art performance on seven benchmark datasets and is robust across a wide range of incompleteness, underlying the strength of dynamic filling in incomplete data clustering.

In the future, we try to apply the one-stage framework to other clustering tasks. Most of the real-world applications are unfriendly to K-means algorithm due to the high-dimension patterns. This limits the usage of K-means algorithm. Moreover, the estimation of missing entries is still an interesting research field to be explored.

REFERENCES

[1] Z. Wang, "Determining the clustering centers by slope difference distribution," *IEEE Access*, vol. 5, pp. 10995–11002, 2017.
[2] A. Dutt, M. A. Ismail, and T. Herawan, "A systematic review on educational data mining," *IEEE Access*, vol. 5, pp. 15991–16005, 2017.
[3] D. Puiu *et al.*, "CityPulse: Large scale data analytics framework for smart cities," *IEEE Access*, vol. 4, pp. 1086–1108, 2017.
[4] W. Yan, Z. Wu, L. Qian, and Y. Zhu, "A model of telecommunication network performance anomaly detection based on service features clustering," *IEEE Access*, vol. 5, pp. 17589–17596, 2017.
[5] T. Zhang and Y. Bo, "Density-based multiscale analysis for clustering in strong noise settings with varying densities," *IEEE Access*, vol. 6, pp. 25861–25873, 2018.
[6] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
[7] M. U. Munir, M. Y. Javed, and S. A. Khan, "A hierarchical k-means clustering based fingerprint quality classification," *Neurocomputing*, vol. 85, pp. 62–67, May 2012.
[8] P. Kai, V. C. M. Leung, and Q. Huang, "Clustering approach based on mini batch kmeans for intrusion detection system over big data," *IEEE Access*, vol. 6, pp. 11897–11906, 2018.
[9] J. Feng, Y. Zhang, G. Yue, X. Liu, H. Su, and P.-F. Zhang, "Atherosclerotic plaque pathological analysis by unsupervised K-means clustering," *IEEE Access*, vol. 6, pp. 21530–21535, 2018.
[10] R. J. Hathaway and J. C. Bezdek, "Fuzzy c-means clustering of incomplete data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 5, pp. 735–744, Oct. 2001.
[11] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, Jan. 2007.
[12] J. Lu and Q. Zhu, "An effective algorithm based on density clustering framework," *IEEE Access*, vol. 5, pp. 4991–5000, 2017.
[13] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
[14] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2, 2004, pp. 28–31.
[15] D. Ren, Z. Jia, Y. Jie, and N. K. Kasabov, "A practical GrabCut color image segmentation based on Bayes classification and simple linear iterative clustering," *IEEE Access*, vol. 5, pp. 18480–18487, 2017.
[16] T. He, H. Lun, K. C. C. Chan, and P. Hu, "Learning latent factors for community identification and summarization," *IEEE Access*, vol. 6, pp. 30137–30148, 2018.
[17] A. Trivedi, P. Rai, H. Daumé, III, and S. L. DuVall, "Multiview clustering with incomplete views," in *Proc. NIPS Workshop*, 2010.
[18] P. Ezatpoor, J. Zhan, J. M.-T. Wu, and C. Chiu, "Finding top-k dominance on incomplete big data using MapReduce framework," *IEEE Access*, vol. 6, pp. 7872–7887, 2018.
[19] Y. Wang, Z. Shi, J. Wang, L. Sun, and B. Song, "Skyline preference query based on massive and incomplete dataset," *IEEE Access*, vol. 5, pp. 3183–3192, 2017.
[20] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.
[21] E. Pan, P. Miao, and H. Zhu, "Tensor voting techniques and applications in mobile trace inference," *IEEE Access*, vol. 3, pp. 3000–3009, 2015.
[22] J. Shen, E. Zheng, Z. Cheng, and D. Cheng, "Assisting attraction classification by harvesting Web data," *IEEE Access*, vol. 5, pp. 1600–1608, 2017.
[23] J. Li, Z. Struzik, L. Zhang, and A. Cichocki, "Feature learning from incomplete EEG with denoising autoencoder," *Neurocomputing*, vol. 165, pp. 23–31, Oct. 2015.
[24] D. P. P. Mesquita, J. P. P. Gomes, A. H. S. Junior, and J. S. Nobre, "Euclidean distance estimation in incomplete datasets," *Neurocomputing*, vol. 248, pp. 11–18, Jul. 2017.
[25] H. Timm, C. Döring, and R. Kruse, "Different approaches to fuzzy clustering of incomplete datasets," *Int. J. Approx. Reasoning*, vol. 35, no. 3, pp. 239–249, 2004.
[26] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Towards missing data imputation: A study of fuzzy k-means clustering method," in *Proc. Int. Conf. Rough Sets Current Trends Comput.* Springer, 2004, pp. 573–579.
[27] D. Lam, M. Wei, and D. Wunsch, "Clustering data of mixed categorical and numerical type with unsupervised feature learning," *IEEE Access*, vol. 3, pp. 1605–1613, 2015.
[28] I. A. Gheyas and L. S. Smith, "A neural network-based framework for the reconstruction of incomplete data sets," *Neurocomputing*, vol. 73, nos. 16–18, pp. 3039–3065, 2010.
[29] T. Li *et al.*, "Interval kernel fuzzy C-means clustering of incomplete data," *Neurocomputing*, vol. 237, pp. 316–331, May 2017.
[30] M. Aste, M. Boninsegna, A. Freno, and E. Trentin, "Techniques for dealing with incomplete data: A tutorial and survey," *Pattern Anal. Appl.*, vol. 18, no. 1, pp. 1–29, 2015.
[31] R. C. Lee, J. R. Slagle, and C. Mong, "Application of clustering to estimate missing data and improve data integrity," in *Proc. 2nd Int. Conf. Softw. Eng.*, 1976, pp. 539–544.
[32] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, and M. Verleysen, "K nearest neighbours with mutual information for simultaneous classification and missing data imputation," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1483–1493, 2009.
[33] C.-P. Lim, J.-H. Leong, and M.-M. Kuan, "A hybrid neural network system for pattern classification tasks with missing features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 648–653, Apr. 2005.
[34] M. Sarkar and T.-Y. Leong, "Fuzzy k-means clustering with missing values," in *Proc. Amer. Med. Inform. Assoc. Symp. (AMIA)*, 2001, pp. 588–592.
[35] R. J. Little and D. B. Rubin, *Statistical Analysis With Missing Data*, vol. 333. Hoboken, NJ, USA: Wiley, 2014.
[36] H.-C. Lin and C.-T. Su, "A selective Bayes classifier with meta-heuristics for incomplete data," *Neurocomputing*, vol. 106, pp. 95–102, Apr. 2013.
[37] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the *EM* algorithm," *J. Roy. Stat. Soc., B (Methodol.)*, 1977, pp. 1–22.
[38] M. Zhong, H. Tang, H. Chen, and Y. Tang, "An EM algorithm for learning sparse and overcomplete representations," *Neurocomputing*, vol. 57, pp. 469–476, Mar. 2004.
[39] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Mach. Learn.*, vol. 56, nos. 1–3, pp. 9–33, 2004.

[40] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. 1988.
[41] F. Alimoglu and E. Alpaydin, "Combining multiple representations and classifiers for pen-based handwritten digit recognition," in *Proc. 4th Int. Conf. Document Anal. Recognit.*, vol. 2, Aug. 1997, pp. 637–640.
[42] W. Qian and W. Shu, "Mutual information criterion for feature selection from incomplete data," *Neurocomputing*, vol. 168, pp. 210–220, Nov. 2015.

SIWEI WANG is currently pursuing the degree with the National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning, scalable kernel k-means, and deep neutral networks.

MIAOMIAO LI is currently pursuing the Ph.D. degree with the National University of Defense Technology, China. She is also a Lecturer with the Changsha College, Changsha, China. She has published several peer-reviewed papers such as AAAI, IJCAI, and neurocomputing. Her current research interests include kernel learning and multi-view clustering. She served on the Technical Program Committees of IJCAI 2017 and 2018.

NING HU received the Ph.D. degree from the National University of Defense Technology (NUDT), China. He is currently a Professor with the Cyberspace Institute of Advanced Technology, Guangzhou University. He has published over 30 papers in journals and conferences. His current research interests include artificial intelligence safety and security. He has also achieved the Second Class Prize of the Chinese State Scientific and Technological Progress Award.

EN ZHU received the Ph.D. degree from the National University of Defense Technology (NUDT), China, where he is currently a Professor with the School of Computer Science. He has published over 60 peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, and IJCAI. His main research interests include pattern recognition, image processing, machine vision, and machine learning. He received the China National Excellence Doctoral Dissertation.

JINGTAO HU is currently pursuing the degree with the National University of Defense Technology (NUDT), China. Her current research interests include unsupervised abnormal detection, outlier detection, and neutral networks.

XINWANG LIU (M'13) received the Ph.D. degree from the National University of Defense Technology (NUDT), China, where he is currently an Assistant Researcher with the School of Computer Science. He has published over 40 peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-IP, IEEE T-NNLS, ICCV, AAAI, and IJCAI. His current research interests include kernel learning and unsupervised feature learning. He served on the Technical Program Committees of IJCAI 2016/2017/2018 and AAAI 2016/2017/2018.

JIANPING YIN received the Ph.D. degree from the National University of Defense Technology (NUDT), China. He is currently a Distinguished Professor with the Dongguan University of Technology. He has published over 100 peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, and IJCAI. His research interests include pattern recognition and machine learning. He received the China National Excellence Doctoral Dissertation' Supervisor and National Excellence Teacher. He served on the Technical Program Committees of over 30 international conferences and workshops.