Robust Ensemble Clustering by Matrix Completion

Jinfeng Yi†, Tianbao Yang†, Rong Jin‡, Anil K. Jain†, Mehrdad Mahdavi†
†Department of Computer Science and Engineering, Michigan State University
‡Machine Learning Lab, GE Global Research
†{yijinfen, rongjin, jain, mahdavim}@cse.msu.edu, ‡tyang@ge.com

Abstract—Data clustering is an important task and has found applications in numerous real-world problems. Since no single clustering algorithm is able to identify all different types of cluster shapes and structures, ensemble clustering was proposed to combine different partitions of the same data generated by multiple clustering algorithms. The key idea of most ensemble clustering algorithms is to find a partition that is consistent with most of the available partitions of the input data. One problem with these algorithms is their inability to handle uncertain data pairs, i.e., data pairs for which about half of the partitions put them into the same cluster and the other half do the opposite. When the number of uncertain data pairs is large, they can mislead the ensemble clustering algorithm in generating the final partition. To overcome this limitation, we propose an ensemble clustering approach based on the technique of matrix completion. The proposed algorithm constructs a partially observed similarity matrix based on the data pairs whose cluster memberships are agreed upon by most of the clustering algorithms in the ensemble. It then deploys the matrix completion algorithm to complete the similarity matrix. The final data partition is computed by applying an efficient spectral clustering algorithm to the completed matrix. Our empirical studies with multiple real-world datasets show that the proposed algorithm performs significantly better than the state-of-the-art algorithms for ensemble clustering.

Keywords—Ensemble Clustering, Matrix Completion, Low Rank, Sparse

I. INTRODUCTION

Although data clustering techniques have been successfully applied to many domains [14], [3], it still remains as a challenging problem. Different clustering algorithms may produce different results on the same data set, and no single clustering algorithm is universally better than others for all types of data [18]. Each clustering algorithm has its own merits, as well as its own limitations. It is this observation that motivated the development of ensemble clustering [34], [12], also known as consensus clustering.

Many algorithms have been proposed for ensemble clustering ([40] and references therein). Among them, one popular group of approaches is based on the similarity (or co-association) matrix. Approaches in this category first compute a similarity (or co-association) matrix based on multiple data partitions, where the similarity between any two data points is measured by the percentage of partitions in the ensemble that assign the two data points to the same cluster. A similarity-based clustering algorithm (e.g., single link and normalized cut) is then applied to the similarity (or co-association) matrix to obtain the final partition of the data. One issue that is often overlooked by the similarity-based approaches is how to handle the uncertain data pairs, i.e., the pairs of data points which have been assigned to the same cluster by approximately half of the partitions in the ensemble and assigned to different clusters by the other half. When the number of the uncertain data pairs is large, they can collectively mislead the ensemble clustering algorithm to output an inappropriate partition of the data.

To address the issue of uncertain data pairs, we propose a novel ensemble clustering approach based on the theory of matrix completion [4]. Instead of assigning similarity values to the uncertain data pairs, we construct a partially observed similarity matrix that only includes reliable data pairs whose similarities are agreed upon by most of the partitions in the ensemble. We then deploy the matrix completion method to complete the partially observed similarity matrix, and generate the final data partition by applying a spectral clustering [5] to the completed similarity matrix. By filtering out the uncertain data pairs, the proposed algorithm is resilient to the noise in the data partitions, leading to a more robust performance. To verify the effectiveness of the proposed algorithm, we conduct studies on multiple real-world datasets. Our results show that the proposed algorithm outperforms both individual clustering algorithms and the state-of-the-art ensemble clustering algorithms.

II. RELATED WORK

There are two key issues in developing an ensemble clustering algorithm: (a) how to generate multiple partitions of the data, and (b) how to combine multiple partitions/clusterings into a single data partition (i.e., consensus partition). According to [13], multiple partitions can be generated by (i) using different clustering algorithms [8], [30], (ii) repeatedly sampling data points and creating partitions for each sample [29], [38], and (iii) running the same clustering algorithm several times with different parameters or initialization, such as k-means [13], mixture models [36] and hyper-planes [35], [7].

Many approaches have been developed to combine multiple partitions into a consensus partition. These approaches can be classified into two categories [40]: median partition
based approaches and object co-occurrence based approaches. In the median partition based approaches, ensemble clustering is cast into an optimization problem that finds the best partition by maximizing the within-cluster similarity, using similarity measures, such as Jaccard coefficient [1], utility function [37] and normalized mutual information [34]. Since the median partition based approaches rely on the similarity measure to determine the final partition, they can be affected significantly by the problem of uncertain data pairs as fined earlier.

The object co-occurrence ensemble clustering approaches can be further divided into three groups. The first group is the relabeling/voting based methods [6], [10], [39]. The basic idea is to first find the corresponding cluster labels between different partitions, and then obtain the consensus partition through a voting process. The second group of approaches in this category is based on co-association/similarity matrix [13], [25], [41]. They use the similarity measure to combine multiple partitions, thus avoiding the label correspondence problem. The third group of approaches in this category is the graph based methods [34], [9]. They construct a weighted graph to represent multiple partitions from the ensemble and find the optimal partition of data by minimizing the graph cut. Similar to the co-association/similarity matrix based approaches, these approaches have to assign weights to the edges connecting the uncertain data pairs, which could mislead the ensemble clustering algorithms.

III. Ensemble Clustering by Matrix Completion (ECMC)

The key idea of the proposed algorithm is to first construct a partially observed similarity matrix, where the entries associated with the uncertain data pairs are marked as unobserved. We then apply the matrix completion algorithm to complete the partially observed similarity matrix by filling in the unobserved entries. Finally, an efficient spectral clustering algorithm is applied to the completed similarity matrix to obtain the final clustering result. Below, we describe in detail the two key steps of the proposed algorithm, i.e., the filtering step that removes the entries associated with the uncertain data pairs from the similarity matrix, and the matrix completion step that completes the partially observed similarity matrix.

The notations described below will be used throughout the paper. Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ be a set of $n$ data points to be clustered, where each data point $x_i \in \mathbb{R}^d$, $i \in [n]$ is a vector of $d$ dimensions. Let $P = \{P_1, P_2, \ldots, P_m\}$ denote a set of $m$ individual partitions or clusterings (clustering ensemble) for the dataset $\mathcal{X}$. Each partition $P_i$, $i \in [m]$, is a component partition, divides $\mathcal{X}$ into $r$ disjoint subsets, where $r$ is the number of clusters. For each $P_i$, we define a similarity matrix $M_i \in \mathbb{R}^{n \times n}$ such that $M_{ij} = 1$ if data points $x_i$ and $x_j$ are assigned to the same cluster in $P_i$, and 0 otherwise. Finally, given a subset $\Delta \subset \{(i, j), i, j = 1, \ldots, n\}$, we define a matrix projection operator $P_\Delta : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ that takes an matrix $E$ as the input and outputs a new matrix $P_\Delta(E) \in \mathbb{R}^{n \times n}$ as

$$[P_\Delta(E)]_{ij} = \begin{cases} E_{ij}, & (i, j) \in \Delta \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

A. Filtering Entries with Uncertain Data Pairs

The purpose of the filtering step is to identify the uncertain data pairs, i.e., data pairs that are assigned to the same cluster by close to half of the component partitions and to different clusters by another half. Thus, the key to the filtering step is to design an appropriate uncertainty measure. Given the similarity matrix $M^1$ obtained from partition $P_i, i \in [m]$, we compute matrix $A = [A_{ij}] \in \mathbb{R}^{n \times n}$ as the average of $(M^l)^m_{i=1}$, i.e., $A_{ij} = \frac{1}{m} \sum_{l=1}^{m} M^l_{ij}$. Since $A_{ij} \in [0, 1]$ essentially measures the probability of assigning data points $x_i$ and $x_j$ to the same cluster by the ensemble of $m$ partitions in $P$, it can be used as the basis for the uncertainty measure. In particular, we define the set of reliable data pairs whose labelings are agreed upon by the percentage of the partitions in the ensemble as $\Delta = \{ (i, j) \in [n] \times [n] : A_{ij} \notin (d_0, d_1) \}$, where $d_0 < d_1 \in [0, 1]$ are two thresholds that are determined empirically. We then construct the partially observed similarity matrix $\hat{A}$ as follows

$$\hat{A}_{ij} = \begin{cases} 1, & (i, j) \in \Delta, A_{ij} \geq d_1 \\ 0, & (i, j) \in \Delta, A_{ij} \leq d_0 \\ \text{unobserved}, & (i, j) \notin \Delta \end{cases} \quad (2)$$

By choosing a sufficiently large value for threshold $d_1$ and a sufficiently small value for $d_0$, we ensure that most of the observed entries in $\hat{A}$ are consistent with the cluster assignments for the corresponding data pairs.

B. Completing Partially Observed Matrix

After the filtering step, the partially observed similarity matrix $\hat{A}$ contains three types of entries: 0, 1 and unobserved. See Eq. 2. The second step of the algorithm is to reconstruct the full similarity matrix $M^* \in \mathbb{R}^{n \times n}$ based on the partially observed matrix $\hat{A}$ to fill in the unobserved entries. For the matrix completion step, we need to make several assumptions about the relationship between $\hat{A}$ and $M^*$.

Our first assumption is about the observed entries in $\hat{A}$. It may appear to be reasonable to assume $\hat{A}_{ij} = M^*_{ij}$ for every observed entry $(i, j) \in \Delta$. However, this assumption may not be satisfied since $\hat{A}$ is constructed from the clustering ensemble; due to errors in component partitions in the ensemble, we expect $\hat{A}$ and $M^*$ to be different for a small number of the observed entries in $\Delta$. Thus, a more realistic assumption to make is that $\hat{A}_{ij} = M^*_{ij}$ for most of the observed entries in $\Delta$. We introduce the matrix $N \in \mathbb{R}^{n \times n}$ to capture the noise in $\hat{A}$, i.e.,

$$P_\Delta(M^* + N) = P_\Delta(\hat{A}) \quad (3)$$
where $\mathcal{P}_\Delta$ is a matrix projection operator defined in (1). Under this assumption, we expect $N$ to be a sparse matrix with most of its entries being zero.

The assumption specified in condition (3) is not sufficient to recover the full similarity $M^*$ as we can fill the unobserved entries in $M^*$ with arbitrary values. An additional assumption is needed to make it possible to recover the full matrix from a partially observed one. To this end, we follow the theory of matrix completion [4] by assuming the full similarity matrix $M^*$ to be of low rank. Since, according to [19], the rank of similarity matrix $M^*$ is the same as the number of clusters, this is a natural assumption provided the number of clusters is not too large.

Under these two assumptions, to recover the full similarity matrix $M^*$ from the partially observed matrix $\tilde{A}$, we need to decompose $\tilde{A}$ into the sum of matrices $N$ and $M^*$, where $N$ is a sparse matrix that captures the noise in $\tilde{A}$ and $M^*$ is a low rank matrix that gives the similarity between any two data points. Thus, we can recover the true similarity matrix $M_*$ by solving the following optimization problem

$$\min_{M,N} |M|_* + C\|N\|_1 \quad \text{s.t.} \quad \mathcal{P}_\Delta (M + N) = \mathcal{P}_\Delta (\tilde{A}),$$

where $|\cdot|_*$ is trace norm and $|\cdot|_1$ is element-wise $\ell_1$ norm. We use inexact augmented Lagrangian algorithm [26] for solving above optimization problem.

C. Parameter Selection

Two sets of parameters are used by the proposed algorithm: thresholds $d_0$ and $d_1$ for constructing the partially observed similarity matrix, and parameter $C$ used in (5) to complete the partially observed similarity matrix.

Two criteria are used in determining the values for $d_0$ and $d_1$ in (2). First, $d_0$ ($d_1$) should be small (large) enough to ensure that most of the retained pairwise similarities are consistent with the cluster assignments. Second, $d_0$ ($d_1$) should be reasonably large (small) to obtain a sufficient number of observed entries in the partially observed matrix $\tilde{A}$. For all the data sets used in our empirical study, we set $d_0$ to 0.2 and $d_1$ to 0.8.

Parameter $C$ in (4) plays an important role in deciding the final similarity matrix. Unlike supervised learning, where parameters can be determined by cross-validation, no supervised information is available for tuning the parameter $C$, making it a challenging problem. Here, we present a heuristic for determining $C$.

We assume that for most clustering problems, the data points are roughly evenly distributed across clusters. Note that a similar assumption was made in the normalized cut algorithm [32]. Based on this assumption, we propose to choose $C$ that leads to a balanced distribution of data points over different clusters. To this end, we measure the imbalance of data distribution over clusters by computing

$$\sum_{i,j=1}^n M_{i,j} = 1^T M 1,$$

where 1 is a vector of all ones. Our heuristic is to choose $C$ that minimizes $1^T M 1$. The rationale behind the imbalance measurement $1^T M 1$ is the following: Let $n_1, \ldots, n_r$ be the number of data points in the $r$ clusters, where $\sum_{k=1}^r n_k = n$. Since $1^T M 1 = \sum_{k=1}^r n_k^2$, within any further constraint, the optimal solution that minimizes $1^T M 1$ is $n_i = n/r, i = 1, \ldots, r$, the most balanced data distribution. Hence, $1^T M 1$, to a degree, measures the imbalance of data distribution over clusters.

IV. EXPERIMENTS

In this section, we present an empirical evaluation of the proposed ensemble clustering algorithm, i.e., Ensemble Clustering by Matrix Completion (ECMC for short) on several different data sets. In particular, we demonstrate that our method performs consistently better than individual clustering algorithms and it outperforms the state of the art ensemble clustering algorithms.

A. Experimental Setup

Construction of the ensemble Our ensemble clustering algorithm does not place requirements on how to generate multiple partitions. However, to compare the performance of our method with individual clustering algorithms, we use the following seven well known clustering algorithms, to generate component partitions (ensembles): K-means algorithm (KM) [17], K-medoids algorithm (KMD) [21], Single Link Hierarchical Clustering (SL) [33], Complete Link Hierarchical Clustering (CL) [22], Average Link Hierarchical Clustering (AL) [31], Fuzzy C-means algorithm (FCM) [2], and Normalized Cut algorithm (NC) [32].

Datasets In order to examine the effectiveness of the proposed ensemble clustering algorithm, six public domain datasets are used in our evaluation. Details of these datasets are given in Table I. Due to the high computational cost of some of the component clustering algorithms, for several large datasets (i.e. RCV1 and MNIST), only their subsets are used in our study. Below, we briefly describe these datasets:

- **USPS M5** We extract subsets of images from the USPS handwritten dataset [15] to form the subset “USPS M5”, which consists of the first five categories of USPS dataset and has a total of 5,427 images.
- **RCV1 M2**. We extract a subset of text documents from the RCV1 corpus [24] to form a subset “RCV1 M2”. It consists of 4,923 instances which belongs to 2 categories “C15” and “GCAT”.
- **MNIST4k**. It contains 4,000 images randomly selected from the MNIST handwritten digits data set [23]. Each image is a 784-dimensional vector that belongs to one of 10 digit classes.
• **Breast-Cancer**, **Segment**, and **Yeast** datasets come from the UCI data repository [11]. Among them, **Breast-Cancer** is a two-class dataset consisting of 683 instances. **Segment** contains 2,310 instances that belong to seven classes. **Yeast** has 1,484 examples that are grouped into ten classes.

<table>
<thead>
<tr>
<th>Name</th>
<th>#Instances</th>
<th>#Features</th>
<th>#Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>USPS M5</td>
<td>5,427</td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td>RCV1 M2</td>
<td>4,923</td>
<td>29,992</td>
<td>2</td>
</tr>
<tr>
<td>MNIST4k</td>
<td>4,000</td>
<td>784</td>
<td>10</td>
</tr>
<tr>
<td>Breast-Cancer</td>
<td>683</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>Segment</td>
<td>2,310</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>Yeast</td>
<td>1,484</td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

### Evaluation metrics
Two well known metrics are used to evaluate the clustering performance, namely normalized mutual information (NMI) and pairwise F-measure. See [42] for details about these two metrics.

All the experiments were conducted on a computer with Intel Xeon 2.4 GHz processor and 8.0 GB of main memory, running on Linux Ubuntu Operating System.

### B. Comparison with Component Clustering Algorithms

We applied the proposed ECMC algorithm to construct the consensus partition based on the partitions generated by the seven clustering algorithms listed in Section IV-A. Since some of the individual clustering algorithms (eg., K-means, K-medoids, and Fuzzy C-means) involve random initializations, we repeat the clustering algorithm five times. The clustering performance averaged over these five trials are reported in Table II.

The proposed ECMC algorithm outperforms every component clustering algorithms on at least five of the six datasets in Table I. Further, the proposed ensemble method performs significantly better than six of the seven component clustering algorithms, including the K-means (KM), K-medoids (KMD), single link (SL) clustering, complete link (CL) clustering, average link (AL) clustering, and Fuzzy C-means (FCM) algorithm on all the test datasets. For the data set RCV1 M2, the normalized cut (NC) performs slightly better than the proposed clustering algorithm. We believe that this is because most of the clustering algorithms in the ensemble perform poorly for this dataset. Since the objective of ensemble clustering algorithm is to produce a consensus partition that is consistent with most of the partitions, the ensemble clustering algorithm performs poorly on this dataset. Still, the performance of the proposed ensemble clustering algorithm is only second to the best individual clustering algorithm, namely NC, for the RCV1 M2 dataset, indicating the robustness of the proposed algorithm in the presence of poor data partitions in the ensemble.

### C. Comparison with Other Ensemble Clustering Algorithms

Seven ensemble clustering algorithms are used as the baseline in our study. They are:

- **MCLA**, **CSPA** and **HPGA** [34]. These algorithms first transform multiple clusterings into a hypergraph representation, and then apply a METIS [20] algorithm to find the consensus partition.
- **EAC-SL** and **EAC-AL** [13]. These two algorithms first construct a similarity matrix, and then apply single link (SL) and average link (AL) algorithm, respectively, to the similarity matrix to obtain the consensus partition.
- Ensemble Clustering based on Quadratic Mutual Information (QMI) [35]. It searches for the consensus partition by maximizing the mutual information between the consensus partition and the ensemble partitions via an EM algorithm.
- Divisive Clustering Ensemble with Automatic Cluster Number (DiCLENS) [27]. It first computes the relationship between data partitions in the ensemble and then derives the consensus partition by a minimum spanning tree. The study in [27] showed that DiCLENS delivers state-of-the-art performance in comparison to several ensemble clustering algorithms, including several recently developed algorithms (e.g., link-based cluster ensemble method (LCE) [16] and combining multiple clusterings via similarity graph (COMUSA) [28]).

The code for all the baseline ensemble clustering algorithms were provided by their respective authors and the reported results are the average over five runs.

Table III summarizes the performance of the proposed algorithm and the baseline algorithms for ensemble clustering. Compared to all the baseline algorithms, the proposed algorithm yields the best performance on all the test datasets, indicating that the proposed ensemble clustering algorithm delivers state-of-the-art performance. Finally, by comparing the results in Tables II and III, we observe that for every baseline ensemble clustering algorithm, there exists at least one dataset on which it performs dramatically worse than the component clustering algorithms. For example, although DiCLENS works well on several datasets, its clustering accuracy for the Yeast dataset is worse than five out of seven component clustering algorithms. This observation indicates that the proposed algorithm is more robust compared to state of the art ensemble clustering algorithms.

### V. Conclusions

We have proposed a robust ensemble clustering algorithm. The key idea is to filter out the data pairs whose co-cluster memberships computed based on the component partitions are not reliable. We only use the reliable data pairs in constructing a partially observed similarity matrix. A matrix completion algorithm is employed to complete the partially observed similarity matrix, and a spectral clustering
algorithm is applied to the completed similarity matrix to obtain the final partition or the clustering result. Empirical studies show that the proposed method (i) performs better than both component (individual) clustering algorithms and the state-of-the-art algorithms for ensemble clustering. Our ongoing effort is to improve the efficiency of our algorithm to make it scalable to large datasets.

VI. ACKNOWLEDGEMENT:

This work was supported in part by National Science Foundation (Award nos. N00014-12-1-0431, N00014-11-1-0100, and N00014-09-1-0663). and Office of Navy Research (IIS-0643494) and Office of Naval Research.

REFERENCES


<table>
<thead>
<tr>
<th>Datasets</th>
<th>ECMC</th>
<th>KM</th>
<th>KMD</th>
<th>SL</th>
<th>CL</th>
<th>AL</th>
<th>FCM</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>USPS M5</td>
<td>NMI</td>
<td>0.693</td>
<td>0.687</td>
<td>0.485</td>
<td>0.017</td>
<td>0.445</td>
<td>0.024</td>
<td>0.663</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.704</td>
<td>0.697</td>
<td>0.526</td>
<td>0.351</td>
<td>0.499</td>
<td>0.352</td>
<td>0.685</td>
</tr>
<tr>
<td>RCV1 M2</td>
<td>NMI</td>
<td>0.731</td>
<td>0.029</td>
<td>0.541</td>
<td>0.010</td>
<td>0.063</td>
<td>0.013</td>
<td>0.651</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.707</td>
<td>0.868</td>
<td>0.835</td>
<td>0.681</td>
<td>0.647</td>
<td>0.680</td>
<td>0.865</td>
</tr>
<tr>
<td>MNIST4k</td>
<td>NMI</td>
<td>0.518</td>
<td>0.480</td>
<td>0.310</td>
<td>0.025</td>
<td>0.279</td>
<td>0.013</td>
<td>0.273</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.496</td>
<td>0.437</td>
<td>0.296</td>
<td>0.182</td>
<td>0.220</td>
<td>0.182</td>
<td>0.290</td>
</tr>
<tr>
<td>Breast-Cancer</td>
<td>NMI</td>
<td>0.519</td>
<td>0.045</td>
<td>0.511</td>
<td>0.056</td>
<td>0.467</td>
<td>0.024</td>
<td>0.503</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.794</td>
<td>0.701</td>
<td>0.789</td>
<td>0.702</td>
<td>0.775</td>
<td>0.704</td>
<td>0.778</td>
</tr>
<tr>
<td>Segment</td>
<td>NMI</td>
<td>0.540</td>
<td>0.025</td>
<td>0.485</td>
<td>0.039</td>
<td>0.350</td>
<td>0.031</td>
<td>0.515</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.498</td>
<td>0.249</td>
<td>0.457</td>
<td>0.249</td>
<td>0.317</td>
<td>0.249</td>
<td>0.437</td>
</tr>
<tr>
<td>Yeast</td>
<td>NMI</td>
<td>0.277</td>
<td>0.223</td>
<td>0.210</td>
<td>0.067</td>
<td>0.189</td>
<td>0.079</td>
<td>0.247</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.334</td>
<td>0.030</td>
<td>0.243</td>
<td>0.311</td>
<td>0.322</td>
<td>0.327</td>
<td>0.259</td>
</tr>
</tbody>
</table>

Table II

AVERAGE CLUSTERING PERFORMANCE OF THE PROPOSED ENSEMBLE CLUSTERING ALGORITHM (ECMC) AND THE COMPONENT CLUSTERING ALGORITHMS K-MEDOIDS (KMD), SINGLE-LINK (SL), COMPLETE-LINK (CL), AVERAGE-LINK (AL), FUZZY C-MEANS (FCM), AND NORMALIZED CUT (NC)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>ECMC</th>
<th>MCLA</th>
<th>CSPA</th>
<th>HPGA</th>
<th>EAC-SL</th>
<th>EAC-AL</th>
<th>QMI</th>
<th>DiCLENS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USPS M5</td>
<td>NMI</td>
<td>0.693</td>
<td>0.685</td>
<td>0.591</td>
<td>0.363</td>
<td>0.175</td>
<td>0.612</td>
<td>0.678</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.704</td>
<td>0.696</td>
<td>0.623</td>
<td>0.352</td>
<td>0.328</td>
<td>0.619</td>
<td>0.693</td>
</tr>
<tr>
<td>RCV1 M2</td>
<td>NMI</td>
<td>0.731</td>
<td>0.288</td>
<td>0.618</td>
<td>0.467</td>
<td>0.595</td>
<td>0.589</td>
<td>0.717</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.907</td>
<td>0.620</td>
<td>0.832</td>
<td>0.680</td>
<td>0.801</td>
<td>0.795</td>
<td>0.899</td>
</tr>
<tr>
<td>MNIST4k</td>
<td>NMI</td>
<td>0.518</td>
<td>0.025</td>
<td>0.475</td>
<td>0.172</td>
<td>0.035</td>
<td>0.504</td>
<td>0.469</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.496</td>
<td>0.182</td>
<td>0.445</td>
<td>0.183</td>
<td>0.188</td>
<td>0.449</td>
<td>0.436</td>
</tr>
<tr>
<td>Breast-Cancer</td>
<td>NMI</td>
<td>0.519</td>
<td>0.301</td>
<td>0.489</td>
<td>0.382</td>
<td>0.442</td>
<td>0.418</td>
<td>0.510</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.794</td>
<td>0.775</td>
<td>0.748</td>
<td>0.705</td>
<td>0.735</td>
<td>0.697</td>
<td>0.782</td>
</tr>
<tr>
<td>Segment</td>
<td>NMI</td>
<td>0.540</td>
<td>0.024</td>
<td>0.502</td>
<td>0.387</td>
<td>0.415</td>
<td>0.530</td>
<td>0.537</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.498</td>
<td>0.198</td>
<td>0.480</td>
<td>0.249</td>
<td>0.312</td>
<td>0.418</td>
<td>0.479</td>
</tr>
<tr>
<td>Yeast</td>
<td>NMI</td>
<td>0.277</td>
<td>0.113</td>
<td>0.161</td>
<td>0.106</td>
<td>0.092</td>
<td>0.271</td>
<td>0.202</td>
</tr>
<tr>
<td></td>
<td>PWF</td>
<td>0.334</td>
<td>0.315</td>
<td>0.191</td>
<td>0.304</td>
<td>0.306</td>
<td>0.203</td>
<td>0.218</td>
</tr>
</tbody>
</table>

Table III

AVERAGE CLUSTERING PERFORMANCE OF THE PROPOSED ENSEMBLE CLUSTERING ALGORITHM (ECMC) AND THE BASELINE ENSEMBLE CLUSTERING ALGORITHMS (MCLA, CSPA, HPGA, EAC-SL, EAC-AL, QMI, DiCLENS) ON NINE DATASETS


