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 $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a set of n data points to be clustered, where each data point $\mathbf{x}_i \in \mathbb{R}^d,\ i \in [n]$ is a vector of d dimensions. Let $\mathrm{P} = \{P_1, P_2, \ldots, P_m\}$ denote a set of m individual partitions or clusterings (clustering ensemble) for the dataset $\mathcal{X}$. Each partition $P_l,\ l \in [m]$, called a component partition, divides $\mathcal{X}$ into r disjoint subsets, where r is the number of clusters. For each P_l, we define a similarity matrix $M^l \in \mathbb{R}^{n\times n}$ such that $M^l_{ij} = 1$ if data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned to the same cluster in P_l, and 0 otherwise. Finally, given a subset $\Delta \subset \{(i,j),\ i,j = 1,\ldots n\}$, we define a matrix projection operator $\mathcal{P}_\Delta : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$ that takes an matrix E as the input and outputs a new matrix $\mathcal{P}_\Delta(E) \in \mathbb{R}^{n\times n}$ as

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

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^l obtained from partition $P_l, l \in [m]$, we compute matrix $A = [A_{ij}] \in \mathbb{R}^{n\times n}$ as the average of $\{M^l\}_{l=1}^m$, 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 $\mathbf{x}_i$ and $\mathbf{x}_j$ 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 $\tilde{A}$ as follows

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

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 $\tilde{A}$ 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 $\tilde{A}$ 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 $\tilde{A}$ to fill in the unobserved entries. For the matrix completion step, we need to make several assumptions about the relationship between $\tilde{A}$ and M^*.

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

$$\mathcal{P}_\Delta(M^* + N) = \mathcal{P}_\Delta(\tilde{A}) \tag{3}$$

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

$$\begin{aligned} \min_{M,N} \quad & |M|_* + C\|N\|_1 \\ \text{s. t.} \quad & \mathcal{P}_\Delta(M+N) = \mathcal{P}_\Delta(\tilde{A}). \end{aligned} \tag{4}$$

where $|\cdot|_*$ is trace norm and $|\cdot|_1$ is element-wise ℓ_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

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

where $\mathbf{1}$ is a vector of all ones. Our heuristic is to choose C that minimizes $\mathbf{1}^\top M \mathbf{1}$. The rationale behind the imbalance measurement $\mathbf{1}^\top M \mathbf{1}$ is the following: Let $n_1, \cdots, n_r$ be the number of data points in the r clusters, where $\sum_{k=1}^{r} n_k = n$. Since $\mathbf{1}^\top M \mathbf{1} = \sum_{k=1}^{r} n_k^2$, without any further constraint, the optimal solution that minimizes $\mathbf{1}^\top M \mathbf{1}$ is $n_i = n/r, i = 1, \ldots, r$, the most balanced data distribution. Hence, $\mathbf{1}^\top M \mathbf{1}$, 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 I
DESCRIPTION OF DATASETS

Name	#Instances	#Features	#Clusters
USPS M5	5,427	256	5
RCV1 M2	4,923	29,992	2
MNIST4k	4,000	784	10
Breast-Cancer	683	10	2
Segment	2,310	19	7
Yeast	1,484	8	10

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 1. 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

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

Datasets		ECMC	KM	KMD	SL	CL	AL	FCM	NC
USPS M5	NMI	**0.693**	0.687	0.485	0.017	0.445	0.024	0.663	0.690
	PWF	**0.704**	0.697	0.526	0.351	0.499	0.352	0.685	0.700
RCV1 M2	NMI	0.731	0.029	0.541	0.010	0.063	0.013	0.651	**0.778**
	PWF	0.907	0.686	0.835	0.681	0.647	0.680	0.865	**0.933**
MNIST4k	NMI	**0.518**	0.480	0.310	0.025	0.279	0.013	0.273	0.505
	PWF	**0.496**	0.437	0.296	0.182	0.220	0.182	0.290	0.453
Breast-Cancer	NMI	**0.519**	0.045	0.511	0.056	0.467	0.024	0.503	0.510
	PWF	**0.794**	0.701	0.789	0.702	0.775	0.704	0.778	0.788
Segment	NMI	**0.540**	0.025	0.485	0.039	0.350	0.031	0.515	0.528
	PWF	**0.498**	0.249	0.457	0.249	0.317	0.249	0.437	0.486
Yeast	NMI	**0.277**	0.223	0.210	0.067	0.189	0.079	0.247	0.268
	PWF	**0.334**	0.301	0.243	0.311	0.322	0.327	0.259	0.266

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

Datasets		ECMC	MCLA	CSPA	HPGA	EAC-SL	EAC-AL	QMI	DiCLENS
USPS M5	NMI	**0.693**	0.685	0.591	0.363	0.175	0.612	0.678	0.688
	PWF	**0.704**	0.696	0.623	0.352	0.328	0.619	0.693	0.694
RCV1 M2	NMI	**0.731**	0.288	0.618	0.467	0.595	0.589	0.717	0.705
	PWF	**0.907**	0.620	0.832	0.680	0.801	0.795	0.899	0.806
MNIST4k	NMI	**0.518**	0.025	0.475	0.172	0.035	0.504	0.469	0.459
	PWF	**0.496**	0.182	0.443	0.183	0.188	0.449	0.436	0.424
Breast-Cancer	NMI	**0.519**	0.501	0.489	0.382	0.442	0.418	0.510	0.512
	PWF	**0.794**	0.775	0.748	0.705	0.735	0.697	0.782	0.789
Segment	NMI	**0.540**	0.024	0.502	0.387	0.415	0.530	0.537	0.349
	PWF	**0.498**	0.198	0.480	0.249	0.312	0.418	0.479	0.317
Yeast	NMI	**0.277**	0.113	0.161	0.106	0.092	0.271	0.202	0.089
	PWF	**0.334**	0.315	0.191	0.304	0.306	0.203	0.218	0.261

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 (IIS-0643494) and Office of Navy Research (Award nos. N00014-12-1-0431, N00014-11-1-0100, and N00014-09-1-0663).

REFERENCES

[1] Asa Ben-Hur, André Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[3] Sanjiv K. Bhatia and Jitender S. Deogun. Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(3):427–436, 1998.

[4] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[5] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):568–586, 2011.

[6] Sandrine Dudoit and Jane Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

[7] Xiaoli Zhang Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *ICML*, pages 186–193, 2003.

[8] Xiaoli Zhang Fern and Carla E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, 2004.

[9] X.Z. Fern and C.E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, page 36. ACM, 2004.

[10] Bernd Fischer and Joachim M. Buhmann. Bagging for path-based clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(11):1411–1415, 2003.

[11] A. Frank and A. Asuncion. UCI machine learning repository. *URL http://archive. ics. uci. edu/ml*, 10, 2010.

[12] A. L. N. Fred and A. K. Jain. Data clustering using evidence accumulation. *Proc International Conf. on Pattern Recognition - ICPR*, I(6):276–280, 2002.

[13] Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):835–850, 2005.

[14] Hichem Frigui and Raghu Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):450–465, 1999.

[15] J.J. Hull. A database for handwritten text recognition research. *PAMI*, 16(5):550–554, 1994.

[16] N. Iam-On, T. Boongoen, and S. Garrett. Lce: a link-based cluster ensemble method for improved gene expression data analysis. *Bioinformatics*, 26(12):1513–1519, 2010.

[17] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., 1988.

[18] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[19] Ali Jalali, Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. In *ICML*, pages 1001–1008, 2011.

[20] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[21] L. Kaufman and P. Rousseeuw. *Clustering by means of medoids*. Reports of the Faculty of Mathematics and Informatics. Fac., Univ., 1987.

[22] B. King. Step-wise clustering procedures. *Journal of the American Statistical Association*, pages 86–101, 1967.

[23] Y. LeCun and C. Cortes. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 1998.

[24] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[25] Yan Li, Jian Yu, Pengwei Hao, and Zhulin Li. Clustering ensembles based on normalized edges. In *PAKDD*, pages 664–671, 2007.

[26] Zhouchen Lin, Minming Chen, Leqin Wu, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, 2010.

[27] S. Mimaroglu and E. Aksehirli. Diclens: Divisive clustering ensemble with automatic cluster number. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (99):1–1, 2011.

[28] S. Mimaroglu and E. Erdil. Combining multiple clusterings using similarity graph. *Pattern Recognition*, 44(3):694–703, 2011.

[29] Behrouz Minaei-Bidgoli, Alexander P. Topchy, and William F. Punch. Ensembles of partitions via data resampling. In *ITCC*, pages 188–192, 2004.

[30] Stefano Monti, Pablo Tamayo, Jill P. Mesirov, and Todd R. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.

[31] F. Murtagh. Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistics Quarterly*, 1(2):101–113, 1984.

[32] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[33] P.H.A. Sneath. The application of computers to taxonomy. *Journal of general microbiology*, 17(1):201, 1957.

[34] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *JMLR*, 3:583–617, 2002.

[35] Alexander P. Topchy, Anil K. Jain, and William F. Punch. Combining multiple weak clusterings. In *ICDM*, pages 331–338, 2003.

[36] Alexander P. Topchy, Anil K. Jain, and William F. Punch. A mixture model for clustering ensembles. In *SDM*, 2004.

[37] Alexander P. Topchy, Anil K. Jain, and William F. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1866–1881, 2005.

[38] Alexander P. Topchy, Behrouz Minaei-Bidgoli, Anil K. Jain, and William F. Punch. Adaptive clustering ensembles. In *ICPR*, pages 272–275, 2004.

[39] Kagan Tumer and Adrian K. Agogino. Ensemble clustering with voting active clusters. *Pattern Recognition Letters*, 29(14):1947–1953, 2008.

[40] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *IJPRAI*, 25(3):337–372, 2011.

[41] X. Wang, C. Yang, and J. Zhou. Clustering aggregation by probability accumulation. *Pattern Recognition*, 42(5):668–675, 2009.

[42] Jinfeng Yi, Rong Jin, Anil K. Jain, Shaili Jain, and Tianbao Yang. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *NIPS*, 2012.