INTERPRETABLE CLUSTERING ENSEMBLES USING BINARY MATRIX FACTORIZATION

S. Sukhanov, C. Debes
AGT International, Darmstadt, Germany
{ssukhanov, cdebes}@agtinternational.com

A.M. Zoubir
Signal Processing Group, TU Darmstadt, Germany
zoubir@spg.tu-darmstadt.de

ABSTRACT

The combination of multiple clustering solutions used to obtain accurate and novel output has attracted attention in data clustering research. Despite the success of clustering ensembles, there are still several fundamental limiting issues including the lack of a unified formalized problem formulation and an intuitive interpretation of the resulting solution. We formulate the clustering ensemble problem as a binary matrix factorization imposing assumptions of a binary structure on the resulting matrices. In such a framework, every data object is assigned to its representative ensemble centroid allowing for interpretation and validation of the consensus clustering results. We demonstrate that the formulated problem can be efficiently solved by means of iterative rank-one binary matrix approximation and apply the Proximus algorithm proposing an effective initialization scheme. The evaluation of the proposed clustering ensemble method demonstrates its efficacy on synthetic and real problems.

***Index Terms*—** clustering ensembles, consensus clustering, binary matrix factorization

1. INTRODUCTION

Data clustering is considered to be an important problem in many fields of data analysis and exploration. Due to its fully unsupervised nature and often unknown details about the underlying structure of data distributions, clustering can be extremely challenging. Clustering ensemble methods (also known as consensus clustering) have emerged as a tool to achieve more stable, quality and accurate clustering results [1]. They often allow obtaining novel solutions [2, 3] that are not feasible by any of the single clustering methods.

The majority of consensus functions are based on either object co-occurrence approaches [4] or median partition [5]. Methods based on object co-occurrence consider pairs of objects, analyzing their cluster membership in every partition. While usually providing reasonable and quality consensus solutions these methods exhibit high memory and computational complexity which limit their practical application to even moderate-sized datasets [6]. Median partition-based methods search for solutions that maximize the sum of similarities between partitions of an ensemble [5]. Such methods usually lead to reasonable consensus, however, the choice of the optimal similarity function is still an open question.

Besides the consensus function, the ensemble generation mechanism is also considered to be an important part of clustering ensembles [5]. Typically, the ensemble generation and consensus steps are studied separately. In order to provide unbiased comparison of consensus functions we focus entirely on them, assuming diverse and quality partitions as available ensemble members.

Most existing consensus functions offer a trade-off between accuracy and scalability [6, 5]. In addition, while being complex and sometimes lacking a clear objective formulation, these methods do not offer a straightforward interpretation of the solution, thus providing no guarantee on its quality. Moreover, there are usually many parameters that should be carefully optimized for every particular task, including the target number of clusters that in practice is often an unknown value.

To address the above mentioned limitations, we propose a clustering ensemble framework that allows us to accurately combine multiple input clusterings while providing descriptive results interpretation. We formulate a clustering ensemble problem as a Binary Matrix Factorization (BMF) [7] and efficiently solve it by means of a recursive rank-one binary matrix approximation based on the Proximus algorithm and introducing an effective initialization strategy. Besides providing an accurate and stable consensus solution, the proposed framework requires no information about the target number or size of clusters that offers intuitive result treatment and is suited for large-scale datasets and high amount of ensemble members.

The paper is structured as follows. In Section 2 we formulate the clustering ensemble problem and highlight the limitations of median partition and object co-occurance-based problem formulations. In Section 3, we turn the clustering ensemble formulation into a BMF and discuss the arising constraints and the way to account for them. We evaluate the proposed framework in Section 4, study its susceptibility to ensemble quality and report the consensus performance on synthetic and real-world datasets. We conclude our paper in Section 5.

2. PROBLEM FORMULATION

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of N objects, where x_i is a vector in a d-dimensional feature space $\mathbb{R}^d$. $\mathbb{P} = \{P_1, P_2, \ldots, P_H\}$ is a set of H partitions (or clusterings) of X, where each $P_h = \{C_1^h, C_2^h, \ldots, C_{K_h}^h\}$ is a single partition of X with K_h clusters satisfying

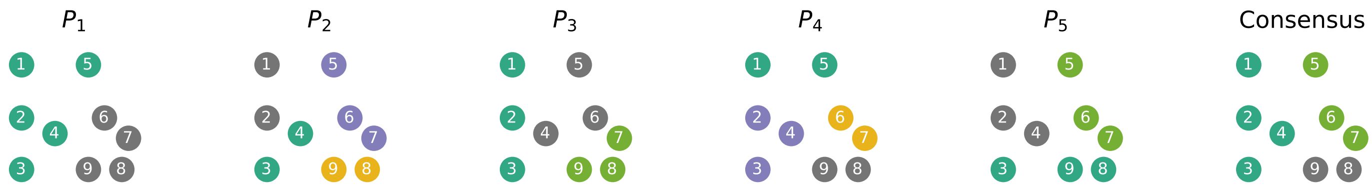

Fig. 1: Example of five distinct partitions of a toy dataset and a consensus solution

- $C_j^h \neq \emptyset, \forall\, j = 1, \ldots, K_h$
- $C_j^h \cap C_p^h = \emptyset, \forall\, j \neq p$
- $\cup_{k=1}^{K_h} C_k^h = X$.

For each x_i, we define a H-dimensional label vector y_i:

$$y_i = [P_1(x_i), P_2(x_i), \ldots, P_H(x_i)] \tag{1}$$

where $P_h(x_i)$ is the cluster label of x_i in partition P_h. Since every clustering P_h assigns symbolic labels to objects x_i the vectors y_i consist of categorical values. In addition, all vectors y_i are forming a matrix $Y = [y_1^\top, y_2^\top, \ldots, y_N^\top]^\top$ that represents data objects in the ensemble label space. Figure 1 depicts an example of a clustering ensemble over $N = 9$ objects clustered differently $H = 5$ times (color encodes cluster label). The consensus solution is depicted on the same Figure 1 and has $K = 3$ resulting clusters.

For the object co-occurrence-based methods [8] the consensus function G maps $\mathbb{P}$ to a consensus solution as:

$$G : \{P_h | h \in \{1, \ldots, H\}\} \to P^* \tag{2}$$

The consensus solution for median partition-based problems provides the maximum cumulative similarity with respect to all clusterings in the ensemble:

$$P^* = \operatorname*{argmax}_{P \in \mathcal{P}_x} \sum_{h=1}^{H} S(P, P_h) \tag{3}$$

where S is a similarity measure between partitions, $\mathcal{P}_x$ is a search space with all possible clusterings of Y. The median partition-based problem was proven to be NP-hard for the Mirkin distance (symmetric difference distance) [5] while the object co-occurrence-based formulation is rather arbitrary. Moreover, both problems are formulated in a way that does not make use of dominant objects (or centroids) and most of the consensus functions provide final labels only.

3. PROPOSED CLUSTER ENSEMBLE APPROACH

To account for the drawbacks of the object co-occurrence- and median partition-based problem formulation we formulate the clustering ensemble problem as a Non-negative Matrix Factorization (NMF) [9] problem where Y is factorized into a membership matrix M and a pattern matrix Q as $Y \approx MQ^\top$, $Y, M, Q > 0$ to minimize the approximation error that is the squared Frobenius norm [10] of the residual:

$$\operatorname*{argmin}_{M,Q} \left\| Y - MQ^\top \right\|_F^2 \tag{4}$$

The main issue with NMF on the Y matrix is that Y is formed from categorical data vectors since every partition within the ensemble represents a symbolic assignment of a point to a cluster. To account for that, we transform Y to a matrix of indicator variables (also known as one-hot or dummy encoding) obtaining $Y_b \in \{0,1\}^{N \times \sum_{h=1}^{H} K_h}$ that represents every partition P_h as a binary matrix of size $N \times K_h$. For further convenience we define $T = \sum_{h=1}^{H} K_h$. Due to the nature of Y_b the problem transforms to the Binary Matrix Factorization (BMF) [7] problem where Y_b is decomposed to a consensus membership matrix M and a matrix of consensus representations Q that both have an additional constraint to be binary. The constraint comes from the fact that the consensus clustering solution has to be crisp (i.e. a non-overlapping solution) and provide interpretable results to be able to evaluate the consensus quality. According to the described transformations, Problem 4 is now reformulated to a BMF as

$$\operatorname*{argmin}_{M_b,Q_b} \left\| Y_b - M_b Q_b^\top \right\|_F^2 \tag{5}$$

where $M_b \in \{0,1\}^{N \times K}$ and $Q_b \in \{0,1\}^{K \times T}$ are restricted to be binary. Matrix M_b consists of presence vectors specifying the consensus clustering membership of every object x_i while Q_b contains dominant binary patterns of Y_b that can be interpreted as centroids in the ensemble label space. An additional property of BMF that we would like to achieve is based on the fact that the target number of clusters is an unknown value and we have to induce it based on the ensemble structure. For this we impose a constraint on matrix Q_b of being able to reconstruct Y_b with a desired error ε providing a minimum number of centroids K:

$$\forall y_i \in Y\ \exists q_k : ||y_i - q_k||_2^2 \leq \varepsilon, k = \{1, \ldots, K\} \tag{6}$$

The error ε implicitly controls the target number of clusters K. Note that the error ε and the resulting number of clusters K are directly linked to ensemble diversity.

To preserve the discrete properties of the data, an efficient and elegant solution for BMF can be found by solving a rank-one binary matrix approximation [11] that searches for two binary vectors m_b and q_b whose outer product provides the minimum distance (that is the Hamming distance for binary vectors) from the matrix to factorize:

$$\min_{m_b,q_b} \left\| Y_b - m_b q_b^\top \right\|_F^2 = \min_{m_b,q_b} \sum_{n,t=1}^{N,T} |(Y_b - m_b q_b^\top)_{n,t}| \tag{7}$$

Since the rank-one binary matrix approximation minimizes the number of nonzero elements in the residual matrix it provides a useful framework to implicitly assess the quality of the consensus solution. When having such a setting there is the possibility to solve BMF iteratively (for $K = 1$) without specifying the number of clusters but relying on the aggregation of those solutions providing an error ε that would determine the optimal number of centroids. All the data objects then would be centered around their respective centroid that provides the minimum distance with each of them. Unfortunately, it was shown that Problem 7 is NP-hard [11] and that only approximate solutions might be reasonably found. For that several algorithm were proposed [12, 10], some of them with guaranteed approximation error bound [11, 13]. However, many of them require high computational resources and deliver difficulties for high N that is common for current practical clustering tasks. Moreover, since we allow any number of ensemble members with any number of clusters the other dimension of the matrix Y_b can be large as well.

To overcome this limitation, we consider the Proximus algorithm [14] that performs a non-orthogonal binary matrix factorization. The idea of Proximus is to recursively grow a tree by employing a rank-one binary matrix approximation that splits the matrix in each node into two sub-matrices based on their distance to a dominant pattern. The splitting is stopped when the distance becomes less than the prescribed bound. Since Proximus is an iterative heuristic it handles large N and H and performs nearly in linear time.The downside of this is that the solution it provides is sensitive to the initialization. For that we propose to choose an initialization based on the maximum count of repetitive vectors in Y. Such vectors known as data fragments [15] constitute stable groups of objects in the label space across all clusterings and serve as centroid candidates.

Based on Figure 1 the BMF on matrix Y solved by rank-one binary matrix approximation provides the following membership and centroid matrices M_b and Q_b, respectively (for convenience we converted Q_b back to label representations using inverse one-hot encoding, and defined it as Q_{cat}).

$$Y = \begin{bmatrix} 1&1&1&1&1&2&2&2&2 \\ 4&4&1&1&2&2&2&3&3 \\ 1&1&1&3&3&3&2&2&2 \\ 1&2&2&2&1&3&3&4&4 \\ 3&3&1&3&2&2&2&1&1 \end{bmatrix}^\top \quad M_b = \begin{bmatrix} 1&1&1&1&0&0&0&0&0 \\ 0&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&0&0&1&1 \end{bmatrix}^\top$$

$$Q_{cat} = \begin{bmatrix} 1&4&1&2&3 \\ 2&2&3&3&2 \\ 2&3&2&4&1 \end{bmatrix}$$

Three clusters with their respective centroids are identified. Using matrix M_b it is possible to find out which object belongs to which centroid and perform further quality analysis. According to the proposed initialization strategy, vector y_8 or y_9 would be chosen to start the decomposition since their respective data fragment shows the largest cardinality.

4. EXPERIMENTAL RESULTS

In this Section, we evaluate the properties and the performance of the proposed BMF-based consensus function that employs the Proximus algorithm (we call it further BMFC) and compare it with other state-of-the-art consensus functions. In all experiments we generate ensembles that consist of 12 partitions. The way these partitions were generated is described in every subsection individually. The synthetic *cone torus, checker board, halfring, boat, petals, aggregation* and real-world *ionosphere, thyroid, wine, glass, wisconsin* datasets that we use in the experiments are obtained from [16] and UCI repository [17], respectively and are commonly used in clustering research. In addition, we use a recent dataset *tiselac* provided by the ECML-PKDD 2017 TiSeLaC challenge [18]. For every dataset the ground truth labels are available and the number of clusters K_t is known.

4.1. Effect of error ε on number of clusters

In the first experiment, we study the effect of the error ε on the number of clusters of the consensus solution. For that we apply BMFC on synthetic datasets while varying the normalized error ε_n in the interval $[0, 1]$ and report the number of clusters on the solution on Figure 2. The normalized error ε_n is defined as $\lceil \frac{\varepsilon}{T} \rceil$. To generate ensemble partitions we standardize the data and run four instances of k-means, BIRCH [19] and mini-batch k-means [20] each. For every clustering instance the target number of clusters is drawn uniformly from the interval $[\max(2, K_t - 2), K_t + 2]$, for BIRCH the values for the branching factor and the subcluster threshold are drawn uniformly from the interval $[40, 60]$ and $[0.35, 0.65]$ correspondingly. For mini-batch k-means the batch size is $\lceil N \times 10^{-3} + 1 \rceil$.

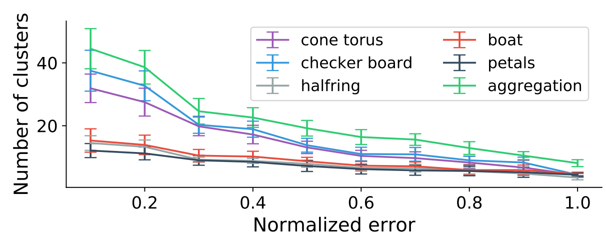

Fig. 2: Number of clusters K for BMFC for various values of normalized error ε over clustered partitions

4.2. Effect of ensemble quality on the performance of BMFC

In this experiment, to understand how ensemble quality affects the performance of the proposed BMFC we apply random noise that follows a Bernoulli distribution with probability p. The noise is applied to each object label so that the label is flipped to a random cluster label with equal probability $q = \frac{p}{K_t - 1}$. In addition, we perform random permutations of resulting labels with uniform probability. We vary the probability of noise p in the interval $[0.05, 0.5]$ with step size 0.05 and report the average over 100 Monte Carlo iterations Adjusted Rand Index (ARI) [21] (Figure 3) for BMFC and four other state-of-the-art consensus functions CTS [22], Knowledege Based (KB) [23], CSPA [8], CAtree [6] (we evaluate

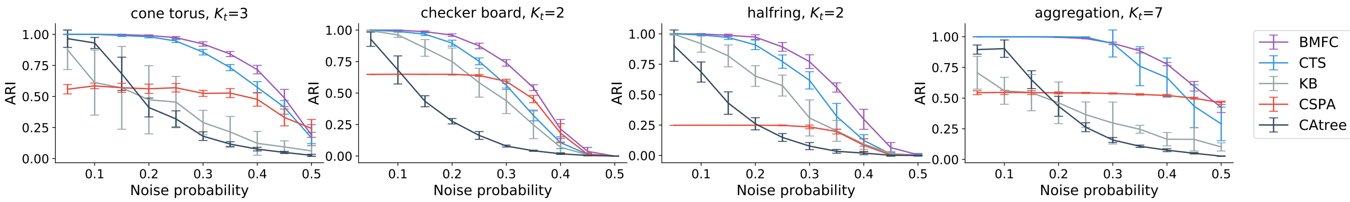

Fig. 3: ARI on four synthetic datasets for several consensus functions over noisy partitions

here only four methods in order not to overload the plots). ARI shows the similarity between resulting clustering and the ground truth labels providing 1.0 when they are identical independently on the cluster symbolic labels. For BMFC we set $\varepsilon = 0.9$, for the other methods we set the target cluster number $K = K_t$.

4.3. Performance on synthetic and real datasets

In the third experiment, we evaluate the performance of the BMFC and the other state-of-the-art methods both on synthetic and real-world datasets. In addition to methods evaluated in the previous experiment, we evaluate HBGF [24], HGPA, MCLA [8], SRC and ASRS [22]. The parameters for BMFC and other methods are set as in the previous experiment. The ensemble partitions are generated as in the first experiment. The results averaged over 100 Monte Carlo iterations provide ARI with their standard deviation in Table 1. In Table 2 we report ARI and Impurity Index (IMP) [25] for dataset *tiselac* for BMFC, HBGF, CAtree, HGPA, MCLA. IMP indicates the number of differently labeled objects in clusters and equals 0.0 for pure clusters in the resulting partition. Because of the large size of *tiselac* dataset ($N = 81715$) several methods that rely on the object co-occurrence matrix failed during execution due to the lack of memory (for evaluation we used a working station with 48GB RAM). Additionally, for this dataset we report average execution time.

Table 2: Evaluation results on tiselac dataset

	BMFC	HBGF	CAtree	HGPA	MCLA
ARI	$\mathbf{0.35 \pm 0.02}$	0.32 ± 0.01	0.31 ± 0.02	0.13 ± 0.09	0.29 ± 0.03
IMP	$\mathbf{0.43 \pm 0.03}$	0.48 ± 0.02	0.49 ± 0.03	0.46 ± 0.31	0.50 ± 0.06
Time, s	**2.1**	24.8	23.2	35.7	24.1

4.4. Discussion

By analyzing the evaluation results we observe that the proposed BMFC demonstrates high operational characteristics. Figure 2 confirms the expected behavior of BMFC with different allowable error bounds showing that for large errors the number of discovered clusters is decreasing. This provides a useful mechanism to affect the number of clusters of the final solution when it is required. Figure 3 indicates that BMFC as well as the other methods are sensitive to the ensemble quality, however, for moderate noise level, BMFC demonstrates resistance to noise and provides acceptable results. From the all three experiments we observe that the proposed BMFC along with providing high operational performance yields low variance. This property indicates the proper choice of the proposed initialization technique that is able to bring the algorithm to the representative objects as it starts. An interesting observation on BMFC can be also done from Tables 1 and 2. While showing good results in terms of ARI on commonly used real-world and synthetic datasets, on large datasets *tiselac* BMFC clearly outperforms other methods both with respect to solution quality and execution time. This shows the potential of the proposed BMFC to be used on large-scale consensus clustering problems without performance degradation.

5. CONCLUSIONS

We addressed the clustering ensembles problem formulating it as the binary matrix factorization that does not require the target number of clusters to be provided while delivering ensemble centroids for better result interpretation. We demonstrated an effective way to solve the problem with the help of one of the existing rank-one binary matrix approximation heuristics. Additionally, we proposed an effective initialization scheme allowing convergence to a stable and quality solution. We experimentally studied the properties of the proposed framework and compared it with the other state-of-the-art consensus functions, demonstrating its efficacy and stability across multiple synthetic and real-world problems.

Table 1: ARI on real-world and synthetic data sets for multiple consensus functions

	BMFC	HBGF	CAtree	HGPA	MCLA	CSPA	KB	CTS	SRS	ASRS
ionosphere	$\mathbf{0.21 \pm 0.03}$	0.18 ± 0.05	0.16 ± 0.05	0.02 ± 0.02	0.17 ± 0.02	0.17 ± 0.02	0.03 ± 0.06	0.18 ± 0.08	0.18 ± 0.01	0.17 ± 0.02
thyroid	$\mathbf{0.24 \pm 0.04}$	0.14 ± 0.13	0.11 ± 0.04	0.05 ± 0.03	0.13 ± 0.10	0.13 ± 0.09	0.12 ± 0.05	0.23 ± 0.01	0.23 ± 0.03	0.12 ± 0.02
wine	0.80 ± 0.07	$\mathbf{0.81 \pm 0.04}$	0.74 ± 0.17	0.40 ± 0.24	0.79 ± 0.04	0.79 ± 0.02	0.17 ± 0.19	$\mathbf{0.81 \pm 0.04}$	0.79 ± 0.04	0.72 ± 0.10
glass	$\mathbf{0.26 \pm 0.03}$	0.19 ± 0.03	0.15 ± 0.03	0.14 ± 0.04	0.18 ± 0.04	0.15 ± 0.03	0.08 ± 0.05	0.24 ± 0.03	0.24 ± 0.02	0.24 ± 0.02
wisconsin	$\mathbf{0.89 \pm 0.04}$	0.87 ± 0.03	0.84 ± 0.03	0.01 ± 0.01	0.87 ± 0.01	0.47 ± 0.01	0.12 ± 0.10	0.87 ± 0.03	0.87 ± 0.03	0.85 ± 0.02
boat	0.44 ± 0.05	0.41 ± 0.01	0.35 ± 0.06	0.33 ± 0.16	0.43 ± 0.05	$\mathbf{0.45 \pm 0.11}$	0.14 ± 0.13	0.41 ± 0.04	0.40 ± 0.03	0.43 ± 0.06
petals	0.95 ± 0.11	0.91 ± 0.13	0.91 ± 0.10	0.85 ± 0.20	0.96 ± 0.03	$\mathbf{0.97 \pm 0.08}$	0.21 ± 0.20	0.89 ± 0.13	0.95 ± 0.07	0.92 ± 0.10
aggregation	$\mathbf{0.83 \pm 0.06}$	0.74 ± 0.04	0.65 ± 0.06	0.51 ± 0.13	0.72 ± 0.04	0.55 ± 0.01	0.29 ± 0.19	0.80 ± 0.06	0.79 ± 0.03	0.81 ± 0.08
cone torus	$\mathbf{0.37 \pm 0.03}$	0.35 ± 0.02	0.35 ± 0.04	0.18 ± 0.10	0.35 ± 0.04	$\mathbf{0.37 \pm 0.03}$	0.15 ± 0.13	$\mathbf{0.37 \pm 0.04}$	0.36 ± 0.06	$\mathbf{0.37 \pm 0.06}$
checker board	0.12 ± 0.08	0.14 ± 0.09	0.06 ± 0.04	0.13 ± 0.10	0.09 ± 0.04	0.12 ± 0.10	0.08 ± 0.05	0.14 ± 0.07	0.13 ± 0.06	$\mathbf{0.15 \pm 0.09}$
halfring	$\mathbf{0.58 \pm 0.06}$	0.54 ± 0.02	0.34 ± 0.12	0.04 ± 0.01	0.47 ± 0.10	0.25 ± 0.01	0.24 ± 0.22	0.56 ± 0.03	$\mathbf{0.58 \pm 0.07}$	0.56 ± 0.03

6. REFERENCES

[1] A. Goder and V. Filkov, "Consensus clustering algorithms: Comparison and refinement," in *Proceedings of the Meeting on Algorithm Engineering & Expermiments*. Society for Industrial and Applied Mathematics, 2008, pp. 109–117.

[2] I. T. Christou, "Coordination of cluster ensembles via exact methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 279–293, 2011.

[3] A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. Fred, "Analysis of consensus partition in cluster ensemble," in *Proceedings of the Fourth IEEE International Conference on Data Mining*, 2004, ICDM '04, pp. 225–232.

[4] A. Lourenço, S. R. Bulò, A. L. N. Fred, and M. Pelillo, "Consensus clustering with robust evidence accumulation," in *Energy Minimization Methods in Computer Vision and Pattern Recognition - 9th International Conference, EMMCVPR 2013, Lund, Sweden, August 19-21, 2013. Proceedings*, 2013, pp. 307–320.

[5] S. V. Pons and J. R. Shulcloper, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artifical Intelligence*, vol. 25, no. 3, pp. 337–372, 2011.

[6] T. Wang, "Ca-tree: A hierarchical structure for efficient and scalable coassociation-based cluster ensembles," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 686–698, 2011.

[7] Z. Zhang, T. Li, C. Ding, and X. Zhang, "Binary matrix factorization with applications," in *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, Washington, DC, USA, 2007, ICDM '07, pp. 391–400, IEEE Computer Society.

[8] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2003.

[9] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[10] T. Li, "A general model for clustering binary data," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, New York, NY, USA, 2005, KDD '05, pp. 188–197, ACM.

[11] B.-H. Shen, S. Ji, and J. Ye, "Mining discrete patterns via binary matrix factorization," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2009, KDD '09, pp. 757–766, ACM.

[12] Z. Shi, L. Wang, and L. Shi, "Approximation method to rank-one binary matrix factorization," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014, pp. 800–805.

[13] H. Lu, J. Vaidya, V. Atluri, H.Shin, and L. Jiang, *Weighted Rank-One Binary Matrix Factorization*, pp. 283–294.

[14] M. Koyutürk and A. Grama, "Proximus: A framework for analyzing very high dimensional discrete-attributed datasets," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2003, KDD '03, pp. 147–156, ACM.

[15] S. Sukhanov, V. Gupta, C. Debes, and A. M. Zoubir, "Consensus clustering on data fragments," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4631–4635.

[16] L. Kuncheva, "Artificial data sets [online]," http://pages.bangor.ac.uk/m̃as00a/activities/artificial_data.htm, 2016.

[17] M. Lichman, "UCI machine learning repository," 2013.

[18] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel, "Land cover classification via multitemporal spatial data by deep recurrent neural networks," *IEEE Geosci. Remote Sensing Lett.*, vol. 14, no. 10, pp. 1685–1689, 2017.

[19] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 1996, SIGMOD '96, pp. 103–114, ACM.

[20] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th International Conference on World Wide Web*, New York, NY, USA, 2010, WWW '10, pp. 1177–1178, ACM.

[21] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[22] I. Natthakan and S. Garrett, "Linkclue: A matlab package for link-based cluster ensembles," *Journal of Statistical Software*, vol. 36, no. 1, pp. 1–36, 2010.

[23] Z. Zhi-Hua and T. Wei, "Clusterer ensemble," *Knowl.-Based Syst.*, vol. 19, no. 1, pp. 77–83, 2006.

[24] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proceedings of the International Conference on Machine Learning*, 2004.

[25] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, 2007.