Instance-Wise Weighted Nonnegative Matrix Factorization for Aggregating Partitions with Locally Reliable Clusters

Xiaodong Zheng, Shanfeng Zhu, Junning Gao
Fudan University*
Shanghai, China
{10210240049,zhusf,13110240002}@fudan.edu.cn

Hiroshi Mamitsuka
Kyoto University†
Uji, Japan
mami@kuicr.kyoto-u.ac.jp

Abstract

We address an ensemble clustering problem, where reliable clusters are locally embedded in given multiple partitions. We propose a new nonnegative matrix factorization (NMF)-based method, in which locally reliable clusters are explicitly considered by using instance-wise weights over clusters. Our method factorizes the input cluster assignment matrix into two matrices $\boldsymbol{H}$ and $\boldsymbol{W}$, which are optimized by iteratively 1) updating $\boldsymbol{H}$ and $\boldsymbol{W}$ while keeping the weight matrix constant and 2) updating the weight matrix while keeping $\boldsymbol{H}$ and $\boldsymbol{W}$ constant, alternatively. The weights in the second step were updated by solving a convex problem, which makes our algorithm significantly faster than existing NMF-based ensemble clustering methods. We empirically proved that our method outperformed a lot of cutting-edge ensemble clustering methods by using a variety of datasets.

1 Introduction

We address the problem of combining multiple partitions (or clusterings) into a single consolidated partition, where each partition is a set of clusters made over the same set of instances [Li *et al.*, 2010; Ghosh and Acharya, 2011]. In the literature, this problem has been called in various ways: *ensemble clustering*, *clustering aggregation*, *consensus clustering* and so on. The objective of this problem is to improve the results given by single clustering algorithms [Strehl and Ghosh, 2003]. There already exist a lot of different types of ensemble clustering approaches, which are based on graphs [Strehl and Ghosh, 2003; Fern and Brodley, 2004; Gionis *et al.*, 2005], similarities [Strehl and Ghosh, 2003], probabilistic models [Topchy *et al.*, 2005; Wang *et al.*, 2011], and consensus functions [Nguyen and Caruana, 2007; Li *et al.*, 2007]. See the Related Work section for detail on these methods, particularly graph-based and consensus functions-based methods.

A key point of ensemble clustering is local cluster reliability, which can be measured by how often the same instance set shares clusters over multiple partitions. That is, if a cluster is shared by a certain size of same instance set a larger number of times, this cluster is more reliable, meaning that cluster reliability can be estimated from input partitions, if input partitions are diverse enough[1]. Furthermore we can say that clusters can be weighted according to reliability through instances. All existing methods have neither considered such intrinsic local properties of input partitions nor realized this idea. The key idea of our method is to explicitly consider the cluster reliability by using cluster weights to enhance the performance of ensemble clustering.

We first present a simple NMF-based ensemble clustering (NMFE), which is a rather straightforward application of typical nonnegative matrix bi-factorization to ensemble clustering. We then extend NMFE by the idea of cluster reliability. That is, if a cluster has particular (coherent) instances, which are always in the same cluster a larger number of times, this cluster must be more reliable. We propose a method, which we call Instance-wise weighted NMF-based Aggregation (INA), which formulates the idea of cluster reliability by incorporating weights over clusters to be estimated from given partitions. More concretely, the point is to capture the shared clusters (and the shared instance set) through the matrix factors, by which the shared (reliable) clusters will be weighted more. To estimate the factorized matrices and weights, INA uses an iterative algorithm which repeats the following two steps alternately: 1) $\boldsymbol{H}$ and $\boldsymbol{W}$ are estimated using the same way as NMFE while cluster weights are fixed, and 2) the (globally) optimal weights are estimated analytically while $\boldsymbol{H}$ and $\boldsymbol{W}$ are fixed. Thus, INA is time-efficient as long as NMFE is fast.

We empirically evaluated our proposed methods, NMFE and INA, comparing with several cutting-edge ensemble clustering methods by using three different scenarios on experimental data: 1) a synthetic dataset with seven different types of clusters, such as the normal distribution, scroll-shaped and circle-shaped clusters which was introduced in [Jain, 2010] as

*Shanghai Key Laboratory of Intelligent Information Processing and School of Computer Science, Fudan University

†Institute for Chemical Research, Kyoto University

[1]Reducing redundant partitions is one important research topic in ensemble clustering, resulting in many good work already [Hadjitodorov *et al.*, 2006; Li and Ding, 2008; Fern and Lin, 2008; Azimi and Fern, 2009]. Thus we can reasonably assume that input partitions are diverse enough.

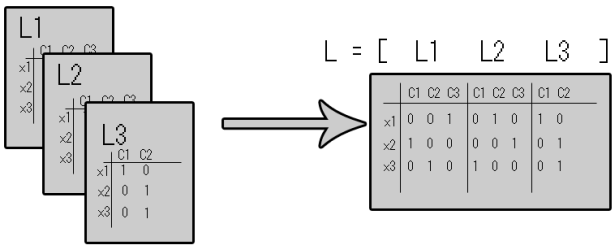

Figure 1: Example of $\boldsymbol{L}$ ($N = 3, K_1 = 3, K_2 = 3$ and $K_3 = 2$)

a challenging problem for developing new clustering methods; 2) real datasets from UCI [Newman *et al.*, 1998] and CLUTO [Karypis, 2002] data repositories, from which partitions are generated to have locally reliable clusters; and 3) real dataset of multiview using partitions generated over each view. INA outperformed other competing methods throughout all three experiments, confirming the validity of our idea on capturing locally reliable clusters.

2 Method

2.1 Notation and Problem Formulation

Let $P = \{P^1, P^2, \ldots, P^M\}$ be M input partitions, sharing the same set of N instances $X = \{x_1, x_2, \ldots, x_N\}$. Partition P^m $(m = 1, 2, \ldots, M)$ consists of a set of clusters $C^m = \{C_1^m, C_2^m, \ldots, C_{K_m}^m\}$, where K_m is the number of clusters for partition P^m and $X = \bigcup_{k=1}^{K_m} C_k^m$. $K = \sum_{m=1}^{M} K_m$. We can further define binary indicator matrix $\boldsymbol{L}^m$, showing the clusters to which each instance belongs, as follows:

$$\boldsymbol{L}_{ij}^m = \begin{cases} 1 & \text{if } x_i \in C_j^m \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Note that for each m, C_j^m can be reordered without loss of generality. The input is $N \times K$ binary matrix $\boldsymbol{L} = [\boldsymbol{L}^1\ \boldsymbol{L}^2\ \ldots\ \boldsymbol{L}^M]$. Fig. 1 shows an example of $\boldsymbol{L}$.

2.2 Nonnegative Matrix Factorization for Ensemble Clustering (NMFE)

Our objective is to find a consolidated partition P^* which can be shown by indicator matrix $\boldsymbol{H} = \{0,1\}^{N \times K^*}$, where K^* is the number of clusters in P^*. For each P^m, we introduce a column permutation matrix $\boldsymbol{W}_m = \{0,1\}^{K_m \times K^*}$ to align columns of $\boldsymbol{H}$ to those of P^m so that $\boldsymbol{L}^m$ should be approximated by $\boldsymbol{H}\boldsymbol{W}_m^T$. Thus the problem can be defined as follows:

$$\min_{\boldsymbol{H}, \boldsymbol{W}_m, m=1,2,\ldots,M} \sum_{m=1}^{M} \|\boldsymbol{L}^m - \boldsymbol{H}\boldsymbol{W}_m^T\|^2, \tag{2}$$

where $\|\cdot\|$ is the Frobenius norm. Letting $\boldsymbol{W} = [\boldsymbol{W}_1^T\ \boldsymbol{W}_2^T\ \ldots\ \boldsymbol{W}_M^T]^T$, Eq. (2) is equivalent to the following general NMF problem:

$$\min_{\boldsymbol{H}, \boldsymbol{W}} \|\boldsymbol{L} - \boldsymbol{H}\boldsymbol{W}^T\|^2. \tag{3}$$

where $\boldsymbol{H}$ and $\boldsymbol{W}$ are binary matrices.

Eq. (3) is an integer optimization problem, which is NP-hard [Moret, 1997; Brucker, 1978; Cohen, 1960]. A general solution of this problem is to relax $\boldsymbol{H}$ and $\boldsymbol{W}$ into a nonnegative continuous domain:

$$\min_{\boldsymbol{H} \geq 0, \boldsymbol{W} \geq 0} \|\boldsymbol{L} - \boldsymbol{H}\boldsymbol{W}^T\|^2. \tag{4}$$

A locally optimal solution of Eq. (4) can be obtained easier than solving the optimization of Eq. (3) by using a standard multiplicative updating procedure [Lee and Seung, 2001] as follows:

$$\boldsymbol{H}_{ik} \leftarrow \boldsymbol{H}_{ik} \frac{(\boldsymbol{L}\boldsymbol{W})_{ik}}{(\boldsymbol{H}\boldsymbol{W}^T\boldsymbol{W})_{ik}} \tag{5}$$

$$\boldsymbol{W}_{jk} \leftarrow \boldsymbol{W}_{jk} \frac{(\boldsymbol{L}^T\boldsymbol{H})_{jk}}{(\boldsymbol{W}\boldsymbol{H}^T\boldsymbol{H})_{jk}} \tag{6}$$

Finally, instance i is assigned to cluster x if $x = \arg\max_j \boldsymbol{H}_{ij}$.

2.3 Instance-wise Weighted NMF-based Aggregation (INA)

Formulation

The key idea of INA is that, instead of considering input partitions (and clusters) uniformly, we introduce weights over clusters (κ_{ij} for cluster j and instance i of $\boldsymbol{L}$), indicating the reliability of cluster j per instance i. Specifically, with the regularization term on the cluster weights, the objective function can be as follows:

$$\begin{aligned} \min_{\boldsymbol{H} \geq 0, \boldsymbol{W} \geq 0, \boldsymbol{\Psi} \geq 0} \quad & \|\boldsymbol{L} \odot \boldsymbol{\Psi} - \boldsymbol{H}\boldsymbol{W}^T\|^2 + \lambda \|\boldsymbol{\Psi}\|^2. \\ & \text{s.t. } (\boldsymbol{L} \odot \boldsymbol{\Psi})\mathbf{1}^{K \times 1} = \mathbf{1}^{N \times 1}. \end{aligned} \tag{7}$$

where $\boldsymbol{\Psi} = (\kappa_{ij})^{N \times K}$ is a weighting matrix, $\boldsymbol{L} \odot \boldsymbol{\Psi}$ denote the element-wise product of matrices $\boldsymbol{L}$ and $\boldsymbol{\Psi}$, and λ is a regularization coefficient.

Optimization Algorithm

We can solve this problem by a locally optimal solution, which uses an iterative algorithm that first updates $\boldsymbol{H}$ and $\boldsymbol{W}$ and then updates $\boldsymbol{\Psi}$ alternately:

Step 1: Fix $\boldsymbol{\Psi}$ and optimize $\boldsymbol{H}$ and $\boldsymbol{W}$ according to NMFE.
Step 2: Fix $\boldsymbol{H}$ and $\boldsymbol{W}$ and optimize $\boldsymbol{\Psi}$.

We can rewrite Eq. (7) by using scalars as follows:

$$\begin{aligned} & \|\boldsymbol{L} \odot \boldsymbol{\Psi} - \boldsymbol{L}^*\|^2 + \lambda \|\boldsymbol{\Psi}\|^2 \\ = & \sum_{i,j} (E_{ij}\kappa_{ij}^2 + F_{ij}\kappa_{ij} + G_{ij}) \\ & \text{s.t.} \sum_j \boldsymbol{L}_{ij}\kappa_{ij} = 1. \end{aligned}$$

where $\boldsymbol{L}^* = \boldsymbol{H}\boldsymbol{W}^T$, $E_{ij} = \boldsymbol{L}_{ij}^2 + \lambda$, $F_{ij} = -2\boldsymbol{L}_{ij}\boldsymbol{L}_{ij}^*$ and $G_{ij} = (\boldsymbol{L}_{ij}^*)^2$.

Thus Eq. (7) is a convex problem, which can be solved analytically through a usual derivation of taking the derivative of

Input: $\boldsymbol{L}, \lambda$
Output: $\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{\Psi}$

INA $(\boldsymbol{L}, \lambda, \boldsymbol{H}, \boldsymbol{W}, \boldsymbol{\Psi})$
1: Initialize $\boldsymbol{\Psi}$ (e.g., $\kappa_{ij} \leftarrow \frac{1}{M}$)
2: **while** until convergence **do**
3: Optimize $\boldsymbol{H}$ and $\boldsymbol{W}$ according to Eqs. (5) and (6)
4: Optimize $\boldsymbol{\Psi}$ according to Eq. (8)
5: **end while**

Figure 2: Pseudocode of the optimized algorithm of Instance-wise weighted NMF-based Aggregation (INA)

Lagrange multiplier and satisfying the Karush–Kuhn–Tucker (KKT) conditions. Then $\boldsymbol{\Psi}$ is given as follows:

$$\kappa_{ij} = \begin{cases} \frac{\boldsymbol{L}_{ij}\boldsymbol{L}^*_{ij} - \mu_i \boldsymbol{L}_{ij}}{\boldsymbol{L}^2_{ij} + \lambda} & \boldsymbol{L}_{ij} \neq 0. \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where

$$\mu_i = \frac{\sum_j \frac{\boldsymbol{L}^2_{ij}\boldsymbol{L}^*_{ij}}{\boldsymbol{L}^2_{ij}+\lambda} - 1}{\sum_j \frac{\boldsymbol{L}^2_{ij}}{\boldsymbol{L}^2_{ij}+\lambda}}$$

After obtaining $\boldsymbol{H}$, $\boldsymbol{W}$ and $\boldsymbol{\Psi}$, we assign cluster x to instance i, if $x = \arg\max_j \boldsymbol{H}_{ij}$. Fig. 2 shows a pseudocode of the optimization algorithm of INA. For the validity of this algorithm, the time complexity and convergence proof are shown in appendices. Finally we can briefly show the validity of our formulation from Eq. (8): If we remove the regularization term (i.e. $\lambda = 0$), κ_{ij} can be approximated as follows:

$$\kappa_{ij} \quad \sim \quad L^*_{ij} - \frac{\sum_j L^*_{ij} - 1}{K} \tag{9}$$

We note that this indicates that the prediction result, i.e. L^*_{ij}, is directly connected to reliability weight κ_{ij}, being consistent with our motivation of incorporating the cluster weights.

3 Related Work

Graph-based methods: Three representative ensemble clustering methods are all graph-based methods [Strehl and Ghosh, 2003]: Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph-Partitioning Algorithm (HGPA) and Meta-CLustering Algorithm (MCLA). CSPA has two steps: 1) a similarity matrix over instances is generated by how many times each pair of two instances is in the same cluster and 2) similarity-based graph partitioning is applied to the similarity matrix. HGPA and MCLA use hypergraphs, in which a node is an instance and a hyperedge is a cluster of instances in a partition. HGPA applies graph-cut partitioning to the nodes in the hypergraph, while in MCLA hyperedges are clustered and then each instance is assigned to its mostly associated cluster (i.e. hyperedge).

Another graph-based ensemble clustering method, hybrid bipartite graph formation (HBGF) [Fern and Brodley, 2004] generates a "meta-graph" (in which both input instances and input clusters are nodes) which is a bipartite graph, in which nodes are clustered. This is a one-step procedure which partitions given clusters into "meta-clusters" and assigns each instance to one meta-cluster simultaneously. HBGF always works well, comparing to a variety of graph-based ensemble clustering methods [Huang *et al.*, 2011], including CSPA, HGPA and MCLA (this was confirmed in our experiment).

Nonnegative matrix factorization (NMF): The first, representative NMF-based method is NMF-based consensus clustering (NMFC) [Li *et al.*, 2007], which uses orthogonal nonnegative matrix tri-factorization (tri-NMF) [Ding *et al.*, 2006]. NMFC was extended to weighted consensus clustering (WCC) by considering weights over input partitions [Li and Ding, 2008].

The input of NMFC is a connectivity (similarity) matrix $\boldsymbol{W}$, in which $\boldsymbol{W}^m_{ij}$ is 1 if instances i and j are in the same cluster of the m-th partition, zero otherwise. Then to obtain the consensus clustering, which is closest to all partitions, NMFC formulates the following optimization problem:

$$\min_{\tilde{\boldsymbol{H}}^T\tilde{\boldsymbol{H}}=I, \tilde{\boldsymbol{H}}, \boldsymbol{Q}>0} \|\tilde{\boldsymbol{W}} - \tilde{\boldsymbol{H}}\boldsymbol{Q}\tilde{\boldsymbol{H}}^T\|^2, \text{ s.t. } \boldsymbol{Q} \text{ is a diagonal.}$$

where $\tilde{\boldsymbol{W}}_{ij} = \frac{1}{M}\sum_{i=1}^M \boldsymbol{W}^m_{ij}$. NMFC solves this problem by a multiplicative updating rule which is very similar to most standard optimization method of NMF. NMFC deals with the input multiple partitions equally and also clusters in one partition uniformly, by which important information of the input partitions might be lost.

To overcome this issue, WCC [Li and Ding, 2008] introduces weights over input partitions to formulate connectivity matrix $\tilde{\boldsymbol{W}}$ as follows:

$$\tilde{\boldsymbol{W}} = \frac{1}{M}\sum_{m=1}^{M} w_m \boldsymbol{W}^m, \tag{10}$$

where $\boldsymbol{w} = (w_1, w_2, \ldots, w_M)^T$, $w_m \geq 0$, $\|w\|_1 = 1$. The problem formulation is tri-NMF again, and the optimization algorithm repeats the following two steps alternately: 1) weights $\boldsymbol{w}$ are fixed and $\boldsymbol{H}$ is estimated by a standard multiplicative updating for tri-NMF, and 2) $\boldsymbol{H}$ is fixed and weights $\boldsymbol{w}$ are estimated by using quadratic programming (QP). This procedure is similar to INA, while QP may cause the following two serious problems, which are not in INA: 1) QP leads to a relatively sparse solution, like choosing only the best partition, and so it may not work if there are no good partitions, 2) QP may be computationally inefficient.

In WCC, weights are over input partitions only, while weights of INA are over all clusters of the input partitions. In other words, WCC deals with clustering in each partition equally, by which cluster reliability information cannot be incorporated and discarded in WCC. The next section shows that this difference between INA and WCC leads to the performance advantage of INA over WCC.

4 Experiments

We conducted three experiments to evaluate the performance of the two proposed methods, NMFE and INA, comparing with six cutting-edge methods, CSPA, HGPA, MCLA, HBGF, NMFC and WCC. Note that HBGF is one of the highest performance graph-based method and NMFC and WCC

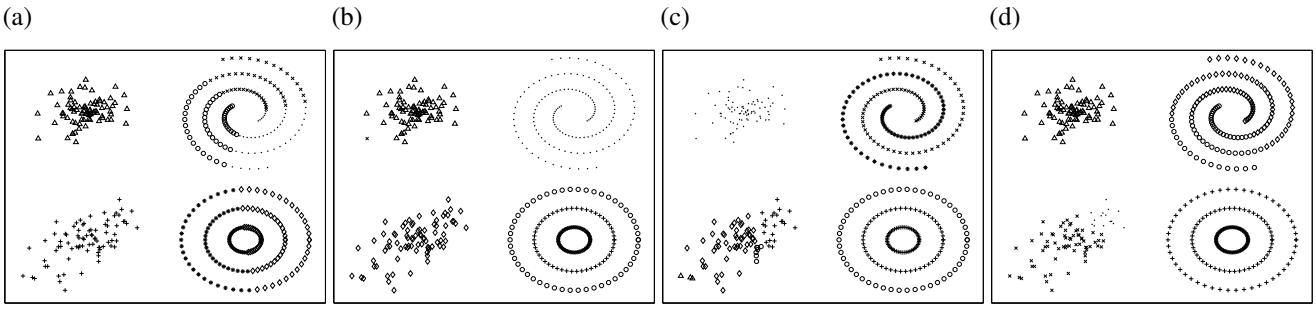

Figure 3: Results of Experiment 1 by component clustering methods: (a) k-means, (b) hierarchical clustering with single-linkage, (c) spectral clustering with the manifold distance and (d) that with the Euclidean distance.

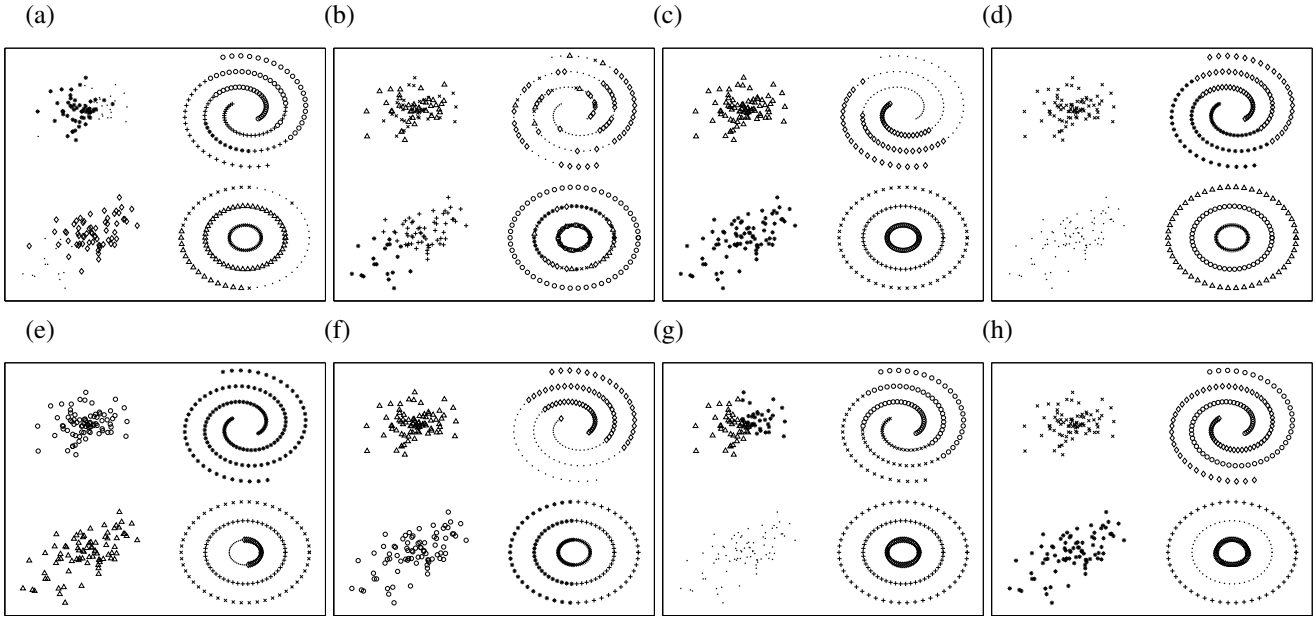

Figure 4: Ensemble clustering results of Experiment 1 by (a) CSPA, (b) HGPA, (c) MCLA, (d) HBGF, (e) NMFC, (f) WCC, (g) NMFE and (h) INA.

are the two latest NMF-based methods. Through all experiments, for a fair comparison, the true number of clusters was given as a priori parameter to all methods. The performance was evaluated by normalized mutual information (NMI), a well-used information theoretic measure for evaluating clustering methods [Fred and Jain, 2003], averaging over 100 trials for each method, with paired t-test to examine the significance on the performance difference between two methods.

4.1 Experiment 1: Benchmark Data in [Jain, 2010]

We first used a famous benchmark dataset with seven clusters, which are shaped by two Gaussians, two scrolls and three cycles [Jain, 2010]. The structures of these seven clusters are very diverse, which makes it very hard to separate these clusters each other. No existing clustering methods could completely separate these seven clusters [Jain, 2010].

We obtained twenty different partitions by running the following twenty different clustering algorithms: two k-means (the Euclidean distance and the city block distance), three hierarchical clustering (single-linkage, average-linkage and complete linkage) and 15 spectral clustering (eight with the manifold distance under different parameters, six with the Gaussian distance under different parameters and the last one with the Euclidean distance). Fig. 3 shows clustering results by typical clustering methods (points shown by the same shape are in the same cluster). We can easily see that all these clustering results were partially correct but failed to separate the seven clusters clearly.

Fig. 4 shows results obtained by running eight ensemble clustering methods, CSPA, HGPA, MCLA, HBGF, NMFC, WCC, NMFE and INA over the twenty partitions (points of the same shape are in the same resultant cluster). This figure shows that any clustering methods cannot separate seven true clusters clearly, except INA (Fig. 4h), which obtained the perfect clustering result. The second best method was HBGF (Fig. 4d), which gave a perfect result except two

Table 1: Data in Experiment 2. N is #instances. #clus is #clusters, N_LC is #instances in the largest cluster and N_SC is #instances in the smallest cluster.

Datasets	N	#clusters	N_LC	N_SC
Anneal	100	4	76	6
Breast	106	6	22	14
Ecoli	336	8	143	2
Glass	214	6	76	9
Protein	116	6	32	15
Statlog	946	4	240	226
Sponge	503	7	118	36
Zoo	101	7	41	4
Tr11	414	9	132	6
Tr12	313	8	93	9
Tr23	204	6	91	6
Tr31	927	7	352	2
Tr41	878	10	243	9
Tr45	690	10	160	14

scrolls which were wrongly clustered. Overall INA outperformed all other seven ensemble clustering methods on the very difficult benchmark dataset. HBGF was the best among the graph-based methods, and so we will show the results of HBGF only out of them in the subsequent experiments, due to space limitations. Thus hereafter we focused on the five best methods, i.e. HBGF, NMFC, WCC, NMFE and INA.

4.2 Experiment 2: UCI and CLUTO

Table 1 shows the summary of the datasets we used in this experiment: 1) eight from the UCI data repository [Newman *et al.*, 1998] under the criterion that each dataset has less than 1,000 instances and the number of labels (clusters) is between 4 to 10 to make all compared methods run in a practical amount of computation time. 2) six from the CLUTO data repository [Karypis, 2002], a standard document database often used for document clustering, extracted under the same criterion as that on the UCI data repository.

For each dataset, we generated input partitions from the ground truth with imposed local property, since controlled experiments are helpful for understanding the characteristic of competing methods. We first assigned true cluster labels to instances and then made instances move to different clusters, with a certain probability (which we call *perturbation rate*). For example, when the perturbation rate is 0.1, we move 10% of all instances to different clusters. However, in order to check the performance of capturing correct (or reliable) clusters, we kept one cluster per partition correct with a certain probability (which we call *selection probability*). The following is a simple example with 15 instances for 4 clusters (with the sizes of 5, 4, 3 and 3) where the first cluster (assigned by 1) is the correct cluster (with the selection probability of 1.0).

$$|11111|3212|432|143|$$

Table 2: NMI values of Experiment 2: The significantly best method for each dataset is highlighted with boldface. "Ave" means the simple average of NMI over all input partitions.

Dataset	Ave	HBGF	NMFC	WCC	NMFE	INA
Anneal	0.169	0.450	0.441	0.170	0.691	**0.850**
Breast	0.145	0.343	0.713	0.219	0.772	**0.921**
Ecoli	0.107	0.203	0.524	0.127	0.453	**0.527**
Glass	0.124	0.462	0.680	0.201	0.763	**0.871**
Protein	0.140	0.344	0.709	0.203	0.789	**0.934**
Statlog	0.136	0.979	0.985	0.219	0.985	**0.987**
Sponge	0.081	0.398	0.761	0.153	0.839	**0.866**
Zoo	0.175	0.289	0.541	0.190	0.613	**0.818**
Tr11	0.096	0.243	0.693	0.149	0.691	**0.759**
Tr12	0.098	0.226	0.726	0.249	0.746	**0.822**
Tr23	0.121	0.403	0.671	0.275	0.734	**0.859**
Tr31	0.072	0.511	**0.713**	0.122	0.696	0.704
Tr41	0.069	0.212	0.616	0.114	0.594	**0.641**
Tr45	0.072	0.182	0.622	0.143	0.623	**0.675**

We fixed the perturbation rate and the selection probability at 0.9 and 0.5, respectively, arbitrarily. For each dataset, 30 partitions were generated.

Table 2 is the results of all five methods, clearly showing that INA outperformed all other competing methods in all 14 datasets, being statistically significant (paired t-test not shown due to space limitations), except only one dataset. The second best was NMFE, which outperformed the other three competing methods (WCC, NMFC and HBGF) in all datasets except only five cases. We further stress that even under the selection probability of 0.5 (meaning that only one reliable cluster per two partitions on average), the NMI values by INA were very high, implying the high robustness of INA.

We further checked the change of reliability weights of INA during the optimization procedure by using Breast in Table 1. Fig. 5 (a) shows the changes of weights over 30 partitions, in which 15 partitions were generated in the above manner (one cluster in each partition can be correct) while the other 15 partitions were with random cluster assignment. Starting with random values, the weights for partitions with correct clusters became larger (more red) through the iterations, because the same set of examples share the same correct clusters two or three times. On the other hand, Fig. 5 (b) shows the weight change for only 5 partitions with correct clusters and 25 partitions with random assignment. In this case, the correct clusters could not be captured well, because of no examples, which share the same cluster more than once (since, for six true clusters, only five partitions with one different correct cluster). This result directly shows that the reliability weights can capture the cluster reliability in terms of how often clusters are shared by the same set of instances.

4.3 Experiment 3: Multi-views Dataset

We finally used real text datasets, called 3-Sources [2], which was retrieved from three online news sources: BBC, Reuters,

[2] http://mlg.ucd.ie/datasets/3sources.html

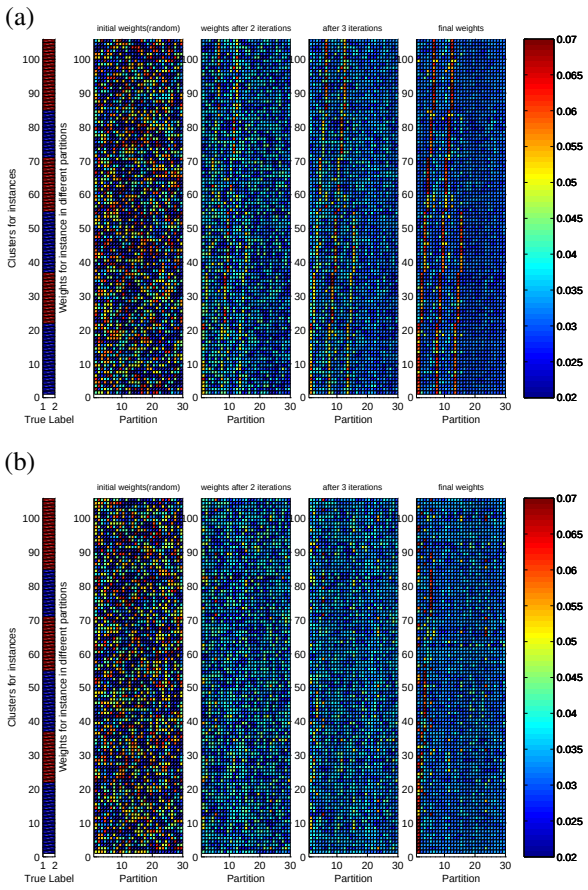

Figure 5: Weight values during iterations from initial (left) to final (right), and the most left column shows true clusters.

and The Guardian. From February to April 2009, 169 distinct news stories were reported by all three sources. By manual annotation, each story was labeled (and used as true cluster label) with one of six different topics: business, entertainment, health, politics, sport and technology. Table 3 shows the statistics of the 3-Sources dataset.

Each news source, which we call a *view*, has its own characteristic and should be separated. We generated features by processing words in original sentences using TFIDF [Manning *et al.*, 2008] and randomly selected P% of all features to make the data more diverse. We then generated 10 partitions from each view by using k-means.

Table 4 shows the results, indicating that INA outperformed other competing methods in all cases, being statistically significant (detailed numbers of paired t-test not shown due to space limitations). The second best method was NMFE, which was followed by HBGF, NMFC and WCC.

5 Concluding Remarks

We have proposed INA, a new NMF-based ensemble clustering formulation and an efficient algorithm for factorizing matrices and estimating cluster weights. A key feature of INA is the weights over clusters (per instance) which are trained from given partitions. The trained weights allow to capture locally reliable clusters, coherent instances and eventually key clusters, which are important to obtain the most consistent partition. We empirically showed that this unique feature makes INA the most powerful method in performance among the cutting-edge ensemble clustering methods.

Table 3: 3-Sources. Dim is # features.

Datasets	N	#clusters	Dim	N_LC	N_SC
BBC	169	6	3560	56	11
Reuters	169	6	3631	56	11
Guardian	169	6	3068	56	11

Table 4: NMI values of Experiment 3, changing P. The significantly best method for each P value is highlighted with boldface. "Ave" means the same as that in Experiment 2.

P(%)	Ave	HBGF	NMFC	WCC	NMFE	INA
50	0.610	0.752	0.733	0.644	0.772	**0.775**
60	0.625	0.768	0.741	0.661	0.778	**0.781**
70	0.638	0.769	0.745	0.669	0.781	**0.783**
80	0.645	0.766	0.744	0.675	0.783	**0.787**
90	0.654	0.770	0.751	0.690	0.790	**0.792**
100	0.660	0.779	0.747	0.709	0.790	**0.793**

Acknowledgments

This work has been partially supported by National Natural Science Foundation of China (61170097 and 61272480), KAKENHI (2430054) of MEXT, JSPS Invitation Fellowships for Research in Japan, and ICR-KU International Short-term Exchange Program for Young Researchers.

A Time Complexity

The time complexity of the optimization algorithm of INA is $O(K^2NMT)$ (T is the number of iterations), while that of NMFC is $O(KN^2T)$, implying that INA is much faster than NMFC (and WCC), since in reality $KM \ll N$.

B Convergence Proof

We denote the loss function, $\|\boldsymbol{L} \odot \boldsymbol{\Psi} - \boldsymbol{H}\boldsymbol{W}^T\|^2 + \lambda\|\boldsymbol{\Psi}\|^2$, by J. Let J_1^t be J before Step 1 at the t-th iteration of the optimization algorithm, and similarly let J_2^t be J before Step 2 at iteration t.

First $J_1^t \geq 0$. We then prove that J is reduced by each step of the algorithm, i.e. $J_1^t \geq J_2^t$ and $J_2^t \geq J_1^{t+1}$.

For Step 1, we can see $J_1^t \geq J_2^t$ by the standard derivation of NMF [Lee and Seung, 2001].

For Step 2, in Eq. (8), κ_{ij} is updated as follows:

$$\kappa_{ij} \quad = \quad \frac{b-\gamma}{a},$$

where $a = \boldsymbol{L}_{ij} + \frac{\lambda}{\boldsymbol{L}_{ij}}$, $b = \boldsymbol{L}^*_{ij}$ and $\gamma = \mu_i$.

Lemma 1 *Let a_1, a_2, b_1, b_2, c and γ be real values. We assume that the sum of two real values x and y is given by $x + y = \frac{b_1-\gamma}{a_1} + \frac{b_2-\gamma}{a_2}$.*

$f(x, y)(= a_1x^2 + b_1x + a_2y^2 + b_2y + c)$ is minimized when $x = x_0$ and $y = y_0$, where $x_0 = \frac{b_1-\gamma}{a_1}$ and $y_0 = \frac{b_2-\gamma}{a_2}$.

Proof Assume that $x = \frac{b_1-\gamma}{a_1} + \epsilon$ and $y = \frac{b_2-\gamma}{a_2} - \epsilon$, then $f(x, y) = a_1(x - \frac{b_1}{a_1})^2 + a_2(y - \frac{b_2}{a_2})^2 + C = a_1(\frac{\gamma}{a_1} - \epsilon)^2 + a_2(\frac{\gamma}{a_2} - \epsilon)^2 + C = (\frac{1}{a_1} + \frac{1}{a_2})\gamma^2 + (a_1 + a_2)\epsilon^2 + C > (\frac{1}{a_1} + \frac{1}{a_2})\gamma^2 + C = f(x_0, y_0)$. ($C$ is a constant.) □

Theorem 1 $J_2^t \geq J_1^{t+1}$.

Proof In Step 2, for each i, κ_{ij} is updated by Eq. (8), which is in the form of $\frac{b_{ij}-\gamma}{a_{ij}}$ (again $\gamma = \mu_i$). We prove that Eq. (8) is the only updating rule which can reduce J most at Step 2. We assume that the optimized κ_{ij} at Step 2 can have different γ for different j. However Lemma 1 says that the loss can be reduced more if we use the same γ for different j. This leads to a contradiction, meaning that optimal κ_{ij} must be obtained from the same γ for different j. This further means that Eq. (8) is the only updating rule of κ_{ij} that minimizes the loss at Step 2. This proves $J_2^t \geq J_1^{t+1}$. □

The proof is complete, since $J_1^t \geq 0$ and $J_1^t \geq J_2^t \geq J_1^{t+1}$.

References

[Azimi and Fern, 2009] Javad Azimi and Xiaoli Z. Fern. Adaptive cluster ensemble selection. In *IJCAI*, pages 993–997, 2009.

[Brucker, 1978] P. Brucker. On the complexity of clustering problems. *Optimization and Operations Research*, 157:45–54, 1978.

[Cohen, 1960] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[Ding *et al.*, 2006] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*, pages 126–135. ACM, 2006.

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

[Fern and Lin, 2008] Xiaoli Z. Fern and Wei Lin. Cluster ensemble selection. *Statistical Analysis and Data Mining*, 1(3):128–141, 2008.

[Fred and Jain, 2003] Ana L. N. Fred and Anil K. Jain. Robust data clustering. In *CVPR*, volume 2, pages 128–133. IEEE, 2003.

[Ghosh and Acharya, 2011] Joydeep Ghosh and Ayan Acharya. Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4):305–315, 2011.

[Gionis *et al.*, 2005] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. In *ICDE*, pages 341–352. IEEE, 2005.

[Hadjitodorov *et al.*, 2006] Stefan Todorov Hadjitodorov, Ludmila I. Kuncheva, and Ludmila P. Todorova. Moderate diversity for better better cluster ensembles. *Information Fusion Journal*, 7(3):264–275, 2006.

[Huang *et al.*, 2011] Xiaodi Huang, Xiaodong Zheng, Wei Yuan, Fei Wang, and Shanfeng Zhu. Enhanced clustering of biomedical documents using ensemble non-negative matrix factorization. *Information Sciences*, 181(11):2293–2302, 2011.

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

[Karypis, 2002] George Karypis. Cluto-a clustering toolkit. Technical report, Minnesota Univ Minneapolis Dept of Computer Science, 2002.

[Lee and Seung, 2001] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. *NIPS*, 13:556–562, 2001.

[Li and Ding, 2008] Tao Li and Chris Ding. Weighted consensus clustering. *SDM*, 1:798–809, 2008.

[Li *et al.*, 2007] Tao. Li, Chris Ding, and Michael I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM*, pages 577–582. IEEE, 2007.

[Li *et al.*, 2010] Tao Li, Mitsunori Ogihara, and Sheng Ma. On combining multiple clusterings: an overview and a new perspective. *Applied Intelligence*, 33(2):207–219, 2010.

[Manning *et al.*, 2008] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[Moret, 1997] Bernard M. Moret. *The theory of computation*. Addison-Wesley Longman Publishing Co., Inc., 1997.

[Newman *et al.*, 1998] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998.

[Nguyen and Caruana, 2007] Nam Nguyen and Rich Caruana. Consensus clusterings. In *ICDM*, pages 607–612, 2007.

[Strehl and Ghosh, 2003] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[Topchy *et al.*, 2005] 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.

[Wang *et al.*, 2011] Hongjun Wang, Hanhuai Shan, and Arindam Banerjee. Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, 4(1):54–70, 2011.