Ensemble clustering using factor graph

Dong Huang a,d, Jianhuang Lai a,*, Chang-Dong Wang b,c

a School of Information Science and Technology, Sun Yat-sen University, Guangzhou Higher Education Mega Center, Panyu District, Guangzhou, Guangdong 510006, China
b School of Mobile Information Engineering, Sun Yat-sen University, China
c SYSU-CMU Shunde International Joint Research Institute (JRI), China
d College of Mathematics and Informatics, South China Agricultural University, China

A R T I C L E   I N F O

Article history:
Received 24 January 2015
Received in revised form 6 July 2015
Accepted 17 August 2015

Keywords:
Ensemble clustering
Factor graph
Belief propagation
Super-object
Automatic cluster number estimate

A B S T R A C T

In this paper, we propose a new ensemble clustering approach termed ensemble clustering using factor graph (ECFG). Compared to the existing approaches, our approach has three main advantages: (1) the cluster number is obtained automatically and need not to be specified in advance; (2) the reliability of each base clustering can be estimated in an unsupervised manner and exploited in the consensus process; (3) our approach is efficient for processing ensembles with large data sizes and large ensemble sizes. In this paper, we introduce the concept of super-object, which serves as a compact and adaptive representation for the ensemble data and significantly facilitates the computation. Through the probabilistic formulation, we cast the ensemble clustering problem into a binary linear programming (BLP) problem. The BLP problem is NP-hard. To solve this optimization problem, we propose an efficient solver based on factor graph. The constrained objective function is represented as a factor graph and the max-product belief propagation is utilized to generate the solution insensitive to initialization and converged to the neighborhood maximum. Extensive experiments are conducted on multiple real-world datasets, which demonstrate the effectiveness and efficiency of our approach against the state-of-the-art approaches.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

The ensemble clustering technique has been receiving increasing attention in recent years due to its ability to combine multiple clusterings into a probably better and more robust clustering [1–10]. Many ensemble clustering approaches have been developed in the past few decades [11]. Despite the great success, there are still three limitations to most of the existing ensemble clustering methods. First, they generally take the cluster number of the final clustering as input and lack the ability to automatically estimate the cluster number. Second, most of them treat each base clustering equally and overlook the different reliability of the base clusterings. Third, many of the existing methods work at the object-level and does not scale well for large ensembles. In this paper, we refer to the ensemble clustering approaches with the object-level and does not scale well for large ensembles. In this paper, we refer to the ensemble clustering approaches with the automatic and the approaches without the ability of automatic cluster number estimate as the non-automatic approaches. Recently some efforts have been made to address one or two of the three limitations [12–14]. However, to the best of our knowledge, none of the existing ensemble clustering approaches is capable of dealing with all of the three limitations in a unified model.

To overcome the aforementioned three limitations, in this paper, we propose a novel ensemble clustering approach termed ensemble clustering using factor graph (ECFG). We introduce the concept of super-object as a compact and adaptive representation for ensemble data. Instead of using the original data objects, we use the super-objects as the primitive objects, which greatly reduces the problem size and facilitates the overall process. We formulate the ensemble clustering problem into a probabilistic framework. The clustering results are represented by a set of binary decisions. Each binary decision indicates whether the corresponding two super-objects are in the same cluster or not. We assume that each base clustering is associated with a probability of making the correct decisions, which can be viewed as the reliability of a base clustering. The consensus clustering and the reliability of each base clustering are estimated iteratively by solving an instance of binary linear programming (BLP) problem. However, the BLP problem is NP-hard. The high computational complexity is the most significant hurdle for it. In this work, we present an efficient solver for this instance of BLP problem based on the factor graph technique [15]. The factor graph is a powerful tool for solving optimization problems and has many successful applications.
in the field of pattern recognition and machine learning [16–19]. In our work, the constrained objective function is represented by a factor graph. Then the max-product belief propagation [15] is applied, which generates the solution insensitive to initialization and converged to the neighborhood maximum.

The main contributions of this paper are summarized as follows:

1. We introduce the concept of super-object, which serves as a compact and adaptive representation for ensemble data and significantly facilitates the computation of the consensus process.
2. We cast the ensemble clustering problem into a BLP problem and propose an efficient solver for the BLP problem based on factor graph.
3. We propose a novel ensemble clustering approach termed ECGF, which has three advantages: (i) it can automatically estimate the cluster number of the final clustering, (ii) the reliability of each base clustering can be estimated and exploited, (iii) it is efficient w.r.t. both large data sizes and large ensemble sizes.
4. Experimental results on multiple real-world datasets have shown that our approach significantly outperforms the state-of-the-art approaches in terms of both clustering accuracy and efficiency (see Section 4).

The remainder of this paper is organized as follows. We review the related work in Section 2. The proposed ensemble clustering approach termed ECGF is introduced in Section 3. The experimental results are reported in Section 4. We conclude this paper in Section 5.

2. Related work

The purpose of ensemble clustering is to combine multiple base clusterings into a more accurate and robust clustering. With regard to the difference of the input information of the ensemble clustering system, there are two formulations of the ensemble clustering problem. In the first formulation, the ensemble clustering system uses only the information of the multiple clusterings as input and has no access to the original data features [1,5,7,20,21]. In the other formulation, the ensemble system uses both the multiple clusterings and the original features as inputs [2,4]. In this paper, we comply with the former formulation of the ensemble clustering problem with no access to the original data features, which is also the common practice for most ensemble clustering approaches [1,5,7,20,21].

In the past decade, there is a large amount of literature on ensemble clustering [11]. The pair-wise co-occurrence based approaches [12,21,22] construct a co-association matrix by considering how many times two objects occur in the same cluster among the multiple base clusterings. Then the consensus clustering is obtained by partitioning the graph into a certain number of segments, each treated as a cluster of the final clustering [20,23]. Strehl and Ghosh [23] formulated the ensemble clustering problem into a graph partitioning problem and introduced three classic algorithms, namely, the cluster-based similarity partitioning algorithm (CSPA), the hypergraph-partitioning algorithm (HGP), and the meta-clustering algorithm (MCLA). The data objects are treated as graph nodes in CSPA and HGP, while the clusters are treated as graph nodes in MCLA. Different from the three algorithms in [23], Fern and Brodley [20] constructed a bipartite graph by treating both objects and clusters as nodes and obtained the final clustering by partitioning the bipartite graph into a number of disjoint sets of nodes.

The median partition based approaches are also one of the main categories of ensemble clustering [8,22,24]. In the median partition based methods, the ensemble clustering problem is cast into an optimization problem that aims to maximize the similarity between the consensus clustering (or median clustering) and the base clusterings. The median partition problem is NP-hard [25]. Cristofero and Simovici [22] utilized a genetic algorithm to find an approximate solution to the median partition problem. Singh et al. [24] used an agreement measure based on 2D string encoding and maximized the new agreement measure using 0–1 semidefinite programming (SDP) so as to find the consensus clustering. Franek and Jiang [8] cast the median partition problem into the Euclidean median problem by clustering embedding in vector spaces. The median vector is computed by the Weiszfeld algorithm [26], and then converted back into a clustering, which is taken as the consensus clustering, by an inverse transformation [8].

Although many successful ensemble clustering approaches have been developed, there are still three limitations to most of the existing approaches. First, they generally take the cluster number of the final clustering as input and cannot automatically estimate the cluster number. Second, many of them work at the object-level and are not capable of dealing with the clustering ensembles with very large data sizes and ensemble sizes. Third, they mostly treat each base clustering equally and lack the ability to weight the base clusterings with regard to their reliability. Recently some efforts have been made to address these limitations [9,12,14,27,28]. Mimaroglu and Erdil [27] constructed a similarity graph based on the evidence accumulated from multiple base clusterings and proposed a pivot-based algorithm termed COMULA to obtain the consensus clustering with the cluster number automatically found. Based on COMUSA, Mimaroglu and Erdil [14] further proposed two scalable ensemble clustering algorithms, termed COMUSACL and COMUSACL-DEW, respectively. Li and Ding [12] introduced a weighted ensemble clustering method based on non-negative matrix factorization. Yu et al. [9] proposed a weighting and selecting scheme for ensemble clustering based on the feature selection technique. Alush and Goldberger [28] addressed the problem of ensemble segmentation, which can be viewed as a special instance of ensemble clustering, and obtained the final segmentation with regard to the reliability of each base segmentation. The method in [28] is not applicable to the general ensemble clustering problem due to two major hurdles: (i) its efficiency heavily relies on the super-pixel map, which is constructed by exploiting the spatial constraints of image data and cannot be used for the general ensemble clustering problem; (ii) the computational complexity of integer linear programming in [28] is prohibitively high for large datasets and is not feasible even for a graph with over 200 nodes. On the whole, each of these methods [9,12,14,27,28] addressed one or two of the aforementioned three limitations. However, none of them is able to tackle all of the three limitations simultaneously. How to overcome all of the three limitations in a unified ensemble clustering framework remains an unaddressed and very challenging problem.

3. Ensemble clustering using factor graph

In this section, we introduce the proposed approach termed ensemble clustering using factor graph (ECGF). The formulation of the ensemble clustering problem is presented in Section 3.1. We
introduce the concept of super-object in Section 3.2 and describe the overall algorithm of ECFG in Section 3.3. To solve the BLP problem in Section 3.3, we introduce a novel optimization method based on factor graph in Section 3.4.

3.1. The ensemble clustering problem

Let $\mathcal{O} = \{o_1, \ldots, o_N\}$ be a dataset of $N$ objects, where $o_i$ is the $i$-th object in $\mathcal{O}$. Let $\Pi = \{\pi^1, \ldots, \pi^M\}$ be an ensemble of $M$ base clusterings for $\mathcal{O}$, where $\pi^m$ is the $m$-th base clustering in $\Pi$. The number of the objects, i.e., $N$, is referred to as the data size of $\Pi$. The number of the base clusterings, i.e., $M$, is referred to as the ensemble size of $\Pi$. Each base clustering contains a certain number of disjoint sets of objects, each set referred to as a cluster. Formally, the $m$-th base clustering is denoted as $\pi^m = \{C^m_1, \ldots, C^m_{\Pi^m}\}$, where $C^m_i$ is the $i$-th cluster and $\Pi^m$ is the number of clusters in $\pi^m$. Note that the clusters in the same clustering do not intersect with each other, i.e., given $\pi^m$, for any $i \neq j$, it holds that $C^m_i \cap C^m_j = \emptyset$. Each object belongs to one and only one cluster given a base clustering. Let $C_{\pi^m}(o)$ denote the cluster in $\pi^m$ that contains object $o_i$. Therefore, if $o_i \in C^m_i$, then $C_{\pi^m}(o_i) = C^m_i$. The goal of ensemble clustering is to build a probably compact and adaptive representation for the ensemble $\Pi$.

3.2. Super-object representation

To solve the ensemble clustering problem, the data objects are often used as the basic operating units [11]. Many successful algorithms have been developed by solving the ensemble clustering problem at the object-level [1,5,7,8,12,16]. However, these methods are generally infeasible for dealing with very large ensembles, since their computational burden may become very high when the data size $N$ and the ensemble size $M$ become very large. To improve the scalability, some researchers proposed to use the clusters as the operating units [13,14]. Typically, the clusters are used as graph nodes and then the graph is partitioned into a certain number of meta-clusters [13,14]. The final clustering is obtained by assigning each data object to a meta-cluster via majority voting [13,14]. By treating clusters as the operating units, the meta-cluster based approaches scale significantly better than the approaches which work at the object-level, especially when the ensemble size is small. However, as the ensemble size $M$ becomes larger, the total number of clusters increases, which leads to an increasing computational burden. According to our empirical study, the approaches which work at the cluster-level generally scale well with regard to the data size, but they often scale poorly with regard to the ensemble size and may become very time-consuming when the ensemble size $M$ is large (see Section 4.7 and Fig. 9). Instead of treating the objects or the clusters as the operating units, in this paper, we introduce the concept of the super-object, which can be viewed as an intermediate level of granularity between the object-level and the cluster-level. Unlike the original clusters in the base clusterings, the super-object is a compact and adaptive representation for the ensemble data and is able to significantly enhance the scalability of the ensemble clustering process with regard to both the data size and the ensemble size. In the following, we will describe the construction of the super-objects in detail.

Definition 1. Let $\mathcal{R}$ be a set of data objects. The set $\mathcal{R}$ is a raw super-object (RSO) if and only if (i) $\forall o_i \in \mathcal{R}$, $\forall \pi^m \in \Pi$, $C_{\pi^m}(o_i) = C_{\pi^m}(o_i)$, and (ii) $\forall o_i \in \mathcal{R}$ and $o_j \notin \mathcal{R}$, $3 \pi^m \in \Pi$, $C_{\pi^m}(o_i) \neq C_{\pi^m}(o_j)$.

Given the clustering ensemble $\Pi$, we first produce a set of $\mathcal{N}$ non-overlapping raw super-objects (RSOs), denoted as

\[ \mathcal{N} = \{N_1, \ldots, N_{\Pi}\}, \]

where $\Pi_i$ is the $i$-th RSO and $\mathcal{N}$ is the number of the RSOs for the ensemble $\Pi$. The formal definition of the RSO is given in Definition 1. Two objects are in the same RSO if and only if the two objects are in the same cluster in all of the $M$ base clusterings. Intuitively, the RSOs are generated by intersecting the $M$ base clusterings in the clustering ensemble $\Pi$. In Table 1 and Fig. 1, we provide an example of three base clusterings, namely, $\pi^1$, $\pi^2$, and $\pi^3$. There are 16 data objects in this example, which are grouped into three clusters, three clusters, and four clusters, respectively, by the three base clusterings (as shown in Fig. 1(a), (b), and (c), respectively). The intersection of the cluster boundaries of the three base clusterings is shown in Fig. 1(d), which leads to eight RSOs as shown in Fig. 1(e).

A straightforward strategy is to use the RSOs as the primitive objects instead of the original objects to reduce the problem size, which, however, suffers from two drawbacks:

1. The number of the RSOs increases dramatically as the number of base clusterings increases, which leads to an increasing computational burden.
2. There may exist many very tiny fragments in the generated RSOs for real-world datasets, especially when the datasets are complex and noisy.

In Fig. 2, we show the number of RSOs with varying ensemble sizes $M$ for the MNIST dataset, which consists of 5000 data objects. The base clusterings in the ensemble are randomly generated by $k$-means. Note that $\#\text{RSO}(\text{all})$ represents the total number of the RSOs, $\#\text{RSO}(\text{size} < 3)$ represents the number of the RSOs that contains less than three data objects, and $\#\text{RSO}(\text{size} > \varepsilon)$ represents the number of the RSOs that contains no less than $\varepsilon$ data objects. Obviously, it holds that $\#\text{RSO}(\text{all}) = \#\text{RSO}(\text{size} < 3) + \#\text{RSO}(\text{size} > \varepsilon)$. As shown in Fig. 2, with the ensemble size $M = 10$, the total number of the RSOs is 829. With the ensemble size $M = 20$, the total number of the RSOs gets much greater than $M = 10$ and becomes 1411. With the ensemble size $M = 50$, the total number of the RSOs goes beyond 2000, which is nearly a half of the total number of objects in the MNIST dataset. When the ensemble size $M$ grows from 10 to 100, $\#\text{RSO}(\text{all})$ and $\#\text{RSO}(\text{size} < 3)$ both increase quickly, i.e., the total number of the RSOs and the number of the tiny RSOs increase quickly, while $\#\text{RSO}(\text{size} > 3)$ is consistently about 200 (Fig. 2). Therefore, it stands to reason that, as the ensemble size increases, the dramatic growth of the total number of the RSOs mainly results from the growth of the number of the very tiny RSOs, while the number of the larger RSOs are much more stable than that of the very tiny ones. Based on these observations, to tackle the aforementioned two drawbacks of the RSOs,

<table>
<thead>
<tr>
<th>Data objects</th>
<th>Cluster labels</th>
<th>RSO labels</th>
<th>Super-object labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>$C_1^1$</td>
<td>$R_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>$C_1^2$</td>
<td>$R_1$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$o_3$</td>
<td>$C_1^2$</td>
<td>$R_2$</td>
<td>$S_2$</td>
</tr>
<tr>
<td>$o_4$</td>
<td>$C_1^2$</td>
<td>$R_2$</td>
<td>$S_2$</td>
</tr>
<tr>
<td>$o_5$</td>
<td>$C_1^2$</td>
<td>$R_3$</td>
<td>$S_3$</td>
</tr>
<tr>
<td>$o_6$</td>
<td>$C_1^2$</td>
<td>$R_3$</td>
<td>$S_3$</td>
</tr>
<tr>
<td>$o_7$</td>
<td>$C_2^3$</td>
<td>$R_3$</td>
<td>$S_3$</td>
</tr>
<tr>
<td>$o_8$</td>
<td>$C_2^3$</td>
<td>$R_4$</td>
<td>$S_4$</td>
</tr>
<tr>
<td>$o_9$</td>
<td>$C_2^3$</td>
<td>$R_4$</td>
<td>$S_4$</td>
</tr>
<tr>
<td>$o_{10}$</td>
<td>$C_2^3$</td>
<td>$R_4$</td>
<td>$S_4$</td>
</tr>
<tr>
<td>$o_{11}$</td>
<td>$C_3^4$</td>
<td>$R_5$</td>
<td>$S_5$</td>
</tr>
<tr>
<td>$o_{12}$</td>
<td>$C_3^4$</td>
<td>$R_5$</td>
<td>$S_5$</td>
</tr>
<tr>
<td>$o_{13}$</td>
<td>$C_3^4$</td>
<td>$R_5$</td>
<td>$S_5$</td>
</tr>
<tr>
<td>$o_{14}$</td>
<td>$C_3^4$</td>
<td>$R_6$</td>
<td>$S_6$</td>
</tr>
<tr>
<td>$o_{15}$</td>
<td>$C_3^4$</td>
<td>$R_6$</td>
<td>$S_6$</td>
</tr>
<tr>
<td>$o_{16}$</td>
<td>$C_3^4$</td>
<td>$R_6$</td>
<td>$S_6$</td>
</tr>
</tbody>
</table>

Table 1
An example of the super-object representation.
we propose a defragmentation strategy to construct the super-objects from the RSOs by iteratively rearranging the tiny fragments (or tiny RSOs). We first define the tiny fragment in Definition 2 by introducing a size threshold $\tau$.

**Definition 2.** A tiny fragment is a set of $n_0$ objects such that $n_0 \leq \tau$, where $\tau \geq 0$ is the tiny fragment threshold.

**Definition 3.** The similarity between two sets of objects w.r.t. the clustering ensemble $\Pi$ is measured as

$$Sim(R_i, R_j) = \frac{\sum_{A_{ij}} \sum_{o_k \in R_i \cap R_j} A_{ij} \cdot n_{ij}}{|R_i| \cdot |R_j|}.$$  

(2)

where

$$A_{ij} = \frac{n_{ij}}{M}$$

(3)

and $n_{ij}$ is the number of times that objects $o_k$ and $o_l$ occur in the same cluster in the $M$ base clusterings.

We collect the tiny fragments among the RSOs and merge each of the tiny fragments into its nearest neighbor according to the similarity measure in Eq. (2). The nearest neighbor of a tiny fragment may also be a tiny fragment or not. When one tiny fragment is merged into another tiny fragment, it is possible that the new and larger fragment is still a tiny fragment. Therefore, we perform the fragment merging process iteratively. After each iteration, if there are still tiny fragments, we proceed to another iteration to merge each of the new tiny fragments into its nearest neighbor. When there are no more tiny fragments, the iteration stops and we have a set of new and compact fragments, each referred to as a super-object. Obviously, the number of data objects in each super-object is greater than the tiny fragment threshold $\tau$.

The process of iteratively merging the tiny fragments into their nearest neighbors is referred to as the defragmentation process in this paper. By the defragmentation process, the tiny fragments are rearranged and the super-objects are constructed from the RSOs. We denote the set of super-objects as $S = \{S_1, ..., S_N\}$, where $S_i$ is the $i$-th super-object and $N$ is the number of the super-objects. The tiny fragment threshold $\tau$ exerts an influence on the number of the generated super-objects. A large $\tau$ leads to a small $N$, and vice versa. If $\tau = 0$, then no RSOs are regarded as tiny fragments and thus the set of super-objects is equivalent to the set of RSOs. For the example in Table 1 and Fig. 1, the tiny fragment threshold $\tau$ is set to 1 and six super-objects are generated based on the RSOs (as shown in Fig. 1(e) and (f)).

Then we need to map the ensemble of base clusterings from original objects to super-objects. Due to the defragmentation process, we map the ensemble of base clusterings from the original objects to super-objects.
process, the objects in a super-object may belong to more than one cluster. We adopt the majority voting strategy to define which cluster a super-object belongs to. Let $C^m(S_i) \subseteq \mathcal{X}^m$ be the cluster in $\mathcal{X}^m$ that $S_i$ belongs to. Then we define

$$C^m(S_i) = \arg\max_{S_j \subseteq \mathcal{X}^m} \sum \{1 | \alpha_k \in C^m_j \},$$

where

$$1(\text{statement}) = \begin{cases} 1 & \text{if statement is true.} \\ 0 & \text{if statement is false.} \end{cases}$$

By the majority voting strategy, given a base clustering $\mathcal{X}^m$, a super-object belongs to the cluster in $\mathcal{X}^m$ that gains the most votes (see Eq. (5)). If two or more clusters in $\mathcal{X}^m$ simultaneously gain the most votes (though it rarely happens in real applications), then $C^m(S_i)$ is randomly selected among these winning clusters. As it is defined, given $\mathcal{X}^m$, if $C^m(S_i) = C^m_j$, then the size of the intersection of $S_i$ and $C^m_j$ should be no less than the size of the intersection of $S_i$ and $C^m_k$ for any $k \neq j$. In this paper, we use the super-objects as the primitive objects for ensemble clustering. We write it as $S_i \in C^m_j$ if $C^m(S_i) = C^m_j$. If $S_i \in C^m_j$ and $S_k \in C^m_k$ such that $i \neq k$, then we say $S_j$ and $S_k$ occur in the same cluster in the base clustering $\mathcal{X}^m$.

### 3.3 Overall algorithm

In this section, we introduce the overall algorithm of the proposed approach termed ensemble clustering using factor graph (ECFG). We use the super-objects as the primitive objects and represent each clustering result as a set of binary decisions. The ensemble clustering problem is formulated into a probabilistic framework $[28]$. The consensus clustering and the reliability of each base clustering are estimated iteratively by the EM algorithm. In the E-step, we solve a binary linear programming (BLP) problem based on the factor graph technique and obtain the estimation of the true clustering. In the M-step, the reliability of each base clustering is estimated by fixing the estimated true clustering.

Given the $M$ base clusterings, we treat each clustering result as a set of binary decisions rather than cluster labels. Let $F = \{f_i\}_{i=1}^M$ be the set of binary decisions by the $m$-th base clustering $\pi^m$, where $f_i \in \{0, 1\}$ is the binary decision in $\pi^m$ for the super-objects $S_i$ and $S_j$. Specifically, $f_i = 1$ indicates the super-objects $S_i$ and $S_j$ are in the same cluster in $\pi^m$, whereas $f_i = 0$ indicates the super-objects $S_i$ and $S_j$ are in different clusters in $\pi^m$. Let $X = \{x_{ij}\}_{i,j=1}^N$ be the set of binary decisions of the true clustering, where $x_{ij} \in \{0, 1\}$ denotes the true relationship between $S_i$ and $S_j$. If $x_{ij} = 1$, then $S_i$ and $S_j$ are in the same cluster in the true clustering; otherwise, $S_i$ and $S_j$ are in different clusters in the true clustering. Thus, the purpose of ensemble clustering is to estimate the true decisions $X$ given the binary decisions of the $M$ base clusterings $E = \{e_i\}_{i=1}^M$.

The base clusterings may be unreliable, which means a binary decision $f_i$ can be either correct or wrong. We assume that each base clustering is associated with an unknown probability of making the correct decision. Let $p^m \in \{0, 1\}$ be the probability of the $m$-th base clustering making a correct decision and $\mathcal{P} = \{p^m, \ldots, p^M\}$ is the set of probabilities of the $M$ base clusterings. Then we have

$$p(f_i = b | x_{ij} = b; \mathcal{P}) = p^m, \quad b \in \{0, 1\}. \quad (7)$$

Similarly,

$$p(f_i = 1 - b | x_{ij} = b; \mathcal{P}) = 1 - p^m, \quad b \in \{0, 1\}. \quad (8)$$

Let $l_j = \{l_{ij} \}_{i=1}^M$ be the set of decisions of the $M$ base clusterings on the super-objects $S_i$ and $S_j$. We assume that the decisions of different base clusterings are independent. Then we have the conditional probability associated with $S_i$ and $S_j$ as follows:

$$p(l_j | x_{ij}; \mathcal{P}) = \prod_{m=1}^M p(l_{ij}^m | x_{ij} = b; \mathcal{P}) = p^m \prod_{b \neq b} (1 - p^m), \quad b \in \{0, 1\}. \quad (9)$$

The likelihood of $l_j$ is

$$p(l_j; \mathcal{P}) = p(l_{ij}^1 | x_{ij} = 1; \mathcal{P})p(x_{ij} = 1) + p(l_{ij}^1 | x_{ij} = 0; \mathcal{P})p(x_{ij} = 0). \quad (10)$$

The likelihood of all decisions in $\mathcal{P}$ is

$$p(L; \mathcal{P}) = \prod_{i \neq j} p(l_{ij}; \mathcal{P}). \quad (11)$$

Then we want to estimate $\mathcal{P}$ by maximum likelihood estimation (MLE) $[29]$, that is

$$\hat{\mathcal{P}} = \arg\max_{\mathcal{P}} p(L; \mathcal{P}). \quad (12)$$

In this model, $\mathcal{P} = \{p^1, \ldots, p^M\}$ are the unknown parameters and $X = \{x_{ij}\}_{i,j=1}^N$ are the hidden variables. We use the EM algorithm $[29]$ to estimate the maximum likelihood parameters.

In the E-step, we fix the parameters $p^1, \ldots, p^M$ and obtain the most likely clustering

$$\hat{X} = \arg\max_X p(X | L; \mathcal{P}). \quad (13)$$

In the M-step, we fix $\hat{X}$ and estimate the most likely parameters, that is

$$\hat{p}^m = \frac{2}{N(N-1)} \sum_{i < j} (\hat{x}_{ij}^m + (1 - \hat{x}_{ij}^m)(1 - l_{ij}^m)). \quad (14)$$

Note that there are totally $N(N-1)/2$ decisions for $N$ super-objects. It is straightforward to solve the M-step. However, solving the optimization problem associated with the E-step is a very hard problem.

As there is no prior knowledge about $X$, for all $i, j$, the prior probability $p(x_{ij} = 1) = p(x_{ij} = 0) = 0.5$. Thus, according to the Bayes’ rule, we have

$$\arg\max_X p(X | L; \mathcal{P}) = \arg\max_X p(L | X; \mathcal{P}). \quad (15)$$

We can write $\log p(l_j | x_{ij}; \mathcal{P})$ as follows:

$$\log p(l_j | x_{ij}; \mathcal{P}) = \begin{cases} \log p(l_{ij}^1 | x_{ij} = 1; \mathcal{P}) & x_{ij} = 1, \\ \log p(l_{ij}^1 | x_{ij} = 0; \mathcal{P}) & x_{ij} = 0, \end{cases} \quad (16)$$

which is equivalent to

$$\log p(l_j | x_{ij}; \mathcal{P}) = \log p(l_{ij}^1 | x_{ij} = 1; \mathcal{P})x_{ij} + \log p(l_{ij}^1 | x_{ij} = 0; \mathcal{P})(1 - x_{ij}) \quad (17)$$

Then we can write $\log p(L | X; \mathcal{P})$ as follows:

$$\log p(L | X; \mathcal{P}) = \sum_{i < j} \log p(l_{ij} | x_{ij}; \mathcal{P}) = \sum_{i < j} w_{ij}x_{ij} + C, \quad (18)$$

where

$$w_{ij} = \frac{\log p(l_{ij}^1 | x_{ij} = 1; \mathcal{P})}{p(l_{ij}^1 | x_{ij} = 0; \mathcal{P})} \quad (19)$$

and

$$C = \sum_{i < j} \log p(l_{ij} | x_{ij} = 0; \mathcal{P}) \quad (20)$$

are both independent on $X$ and can be computed directly given $\mathcal{P}$ (see Eq. (9)). Then we have

$$\arg\max_X p(L | X; \mathcal{P}) = \arg\max_X \sum_{i < j} w_{ij}x_{ij}. \quad (21)$$
Therefore, we can estimate $X$ by solving the following optimization problem:

$$\max_X \sum_{i < j} w_{ij} x_{ij}$$

subject to $x_{ij} + x_{ik} - x_{jk} \leq 1 \quad \forall i, j, k$

$$x_{ij} \in \{0, 1\},$$

(22)

where $x_{ij} + x_{ik} - x_{jk} \leq 1$ is the transitive constraint. The optimization problem in Eq. (22) is a binary linear programming (BLP) problem, which is NP-hard [30].

In this paper, we solve the optimization problem in Eq. (22) by an efficient solver based on factor graphs [see Section 3.4] and obtain the estimation of $X$ and the reliability of each base clustering iteratively and simultaneously. For clarity, we describe the overall algorithm of the proposed ECFG approach in Algorithm 1.

### Algorithm 1. Ensemble clustering using factor graph (ECFG).

**Input:** $\Pi = (\pi^1, ..., \pi^m)$, $\tau$.

1. Produce $N$ RSOs by intersecting base clusterings.
2. Construct $N$ super-objects from the RSOs with the tiny fragment threshold $\tau$.
3. Obtain the binary decisions $L$ for the $N$ super-objects according to the $M$ base clusterings.
4. Estimate the true clustering $X$ and the base clustering reliability measure $\mathcal{P}$ iteratively by the EM algorithm.

The E-step:

Compute $w_{ij}$ for all $i, j$ by Eq. (19). Estimate $X$ by solving the problem in Eq. (22).

The M-step:

Estimate the reliability measure $\mathcal{P}$ by Eq. (14).

5. Map the super-objects back to the data objects and obtain the consensus cluster $\pi^*$ using $X$.

**Output:** the consensus clustering $\pi^*$.

### 3.4. Optimization by factor graph

In this section, we introduce a novel optimization algorithm based on factor graph for the BLP problem in Eq. (22). The objective function (Eq. (22)) is equivalent to

$$\max_X \sum_{i < j} w_{ij} x_{ij}$$

subject to $x_{ij} + x_{ik} - x_{jk} \leq 1$

$$\forall i, j, k$$

$$x_{ij} \in \{0, 1\}.$$

(23)

Let $f_y(x_{ij}) = w_{ij} x_{ij}$ and

$$\delta_{ijk}(x_{ij}, x_{ik}, x_{jk}) = \begin{cases} 0 & x_{ij} + x_{ik} - x_{jk} \leq 1 \\
& x_{ij} + x_{ik} - x_{jk} \leq 1 \\
& x_{ij} + x_{ik} - x_{jk} \leq 1 \\
-\infty & \text{Otherwise} \end{cases}$$

(24)

The objective function (Eq. (23)) can be written as

$$\max_X \left[ \sum_{i < j} f_y(x_{ij}) + \sum_{i < j < k} \delta_{ijk}(x_{ij}, x_{ik}, x_{jk}) \right]$$

subject to $x_{ij} \in \{0, 1\}.$

(25)

The maximization term can be represented by factor graph [15] as shown in Fig. 3. In the factor graph representation, square nodes represent functions, circle nodes represent variables, and edges represent the function-variable relation [15]. There is one real-valued message associated with each edge, which will be updated iteratively.

In the max-sum algorithm, the message updating involves either a message from a variable to each adjacent function or that from a function to each adjacent variable. The message from a variable to a function sums together the messages from all adjacent functions except the one receiving the message [15]:

$$\mu_{v \rightarrow f}(x) = \sum_{h \in \text{ne}(f)} \mu_{h \rightarrow v}(x)$$

(26)

where $\text{ne}(x)$ is the set of adjacent functions of variable $x$. The message from a function to a variable maximizes a combination over all arguments of the function except the variable receiving the message [15]:

$$\mu_{f \rightarrow v}(x) = \max_{X \setminus \{x\}} \left[ f(X) + \sum_{y \in X \setminus \{x\}} \mu_{f \rightarrow y}(v) \right]$$

(27)

where $X = \text{ne}(f)$ is the set of arguments of function $f$.

According to Eqs. (26) and (27), the belief propagation messages passing in the factor graph are as follows (shown in Fig. 4), with $v \in \{0, 1\}:

$$\theta_y(v) = \mu_{x_1 \rightarrow x_y}(v) = f_y(v)$$

(28)

$$\alpha^1_y(v) = \mu_{x_1 \rightarrow \delta_{y_1x_2}x_{y_2x_1}}(v) = \theta_y(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v)$$

(29)

$$\alpha^2_y(v) = \mu_{x_2 \rightarrow \delta_{y_3x_4}x_{y_4x_3}}(v) = \theta_y(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v)$$

(30)

$$\alpha^3_y(v) = \mu_{x_3 \rightarrow \delta_{y_5x_6}x_{y_6x_5}}(v) = \theta_y(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v) + \sum_{v, m < i < j} \beta^1_{ij}(v)$$

(31)

$$\beta^1_{ij}(v) = \mu_{\delta_{y_1x_2}x_{y_2x_1} \rightarrow x_1}(v) = \max(\delta_{y_1}(v, x_1, x_2) + \alpha^2_1(y_1) + \alpha^2_1(y_2))$$

(32)

$$\beta^2_{ij}(v) = \mu_{\delta_{y_3x_4}x_{y_4x_3} \rightarrow x_2}(v) = \max(\delta_{y_1}(v, x_2, x_1) + \alpha^1_1(y_3) + \alpha^1_1(y_4))$$

(33)

$$\beta^3_{ij}(v) = \mu_{\delta_{y_5x_6}x_{y_6x_5} \rightarrow x_3}(v) = \max(\delta_{y_1}(v, x_3, x_2) + \alpha^1_1(y_5) + \alpha^1_1(y_6))$$

(34)

Fig. 3. The factor graph representation of the optimization problem.
4. Experiments

In this section, we conduct experiments on multiple real-world datasets to evaluate the proposed ECFG approach. We compare the proposed approach against seven ensemble clustering approaches, namely, COMUSA [27], DICLENS [13], COMUSACL [14], COMUSACL-DEW [14], EAC [1], WEAC [10], and WCT [3]. Among these baseline approaches, COMUSA, DICLENS, COMUSACL, and COMUSACL-DEW are automatic ensemble clustering approaches, while the other three are non-automatic approaches.

4.1. Datasets and evaluation metric

In the experiments, we use eight real-world datasets, namely, Iris, Vehicle Silhouettes (VS), Vowel, Semeion, MNIST, ISOLET, Pen Digits (PD), and Letters. The MNIST dataset is from [31], while the other seven datasets are from the UCI machine learning repository [32]. The details of the benchmark datasets are given in Table 2.

In our experiments, the normalized mutual information (NMI) [23] is utilized for evaluating the quality of clusterings. The NMI score measures the similarity between two clusterings based on the shared information between them. Given a dataset \(O\) of size \(N\), the clustering labels \(\pi\) of \(c\) clusters and actual class labels \(\zeta\) of \(c\) classes, a confusion matrix is formed first. Let \(n_{i}\) be the number of objects in cluster \(i\), \(n_{ij}\) be the number of objects in class \(j\), and \(n_{ij}\) be the number of objects in both cluster \(i\) and class \(j\). Then NMI can be computed from the confusion matrix as follows [23]:

\[
\text{NMI} = \frac{2 \sum_{j=1}^{c} \sum_{i=1}^{c} n_{ij} \log \frac{n_{ij} N}{\sum_{j=1}^{c} n_{ij} \sum_{i=1}^{c} n_{ij}}} \quad H(\pi) + H(\zeta),
\]

where \(H(\pi) = -\sum_{j=1}^{c} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N}\) and \(H(\zeta) = -\sum_{j=1}^{c} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N}\) are Shannon entropy of cluster labels \(\pi\) and class labels \(\zeta\), respectively. A higher NMI indicates that the test clustering matches the ground-truth clustering better. See [23] for further details.

In our experiments, we use \(k\)-means to produce the ensemble of base clusterings. Unless specially mentioned, the ensemble size \(M=20\) is used in this paper. Each of the \(M\) base clusterings is produced by \(k\)-means with the cluster number randomly chosen in the interval of \([2, \sqrt{3}M]\). To rule out the factor of getting lucky occasionally, we run the proposed approach and the baseline approaches on randomly generated ensembles repeatedly and compare these methods by their average performances over a large number of runs.

4.2. The tiny fragment threshold \(\tau\)

The parameter \(\tau\) is the size threshold of the tiny fragments and exerts an influence on the number of the generated super-objects (see Section 3.2). A larger \(\tau\) generally leads to less super-objects and thus less computational expenses for the follow-up ensemble process. In this section, we test the influence of \(\tau\) on the number of objects.

<table>
<thead>
<tr>
<th>Table 2 Description of datasets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>Iris</td>
</tr>
<tr>
<td>VS</td>
</tr>
<tr>
<td>Vowel</td>
</tr>
<tr>
<td>Semeion</td>
</tr>
<tr>
<td>MNIST</td>
</tr>
<tr>
<td>ISOLET</td>
</tr>
<tr>
<td>PD</td>
</tr>
<tr>
<td>Letters</td>
</tr>
</tbody>
</table>

Please cite this article as: D. Huang et al., Ensemble clustering using factor graph, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.08.015
super-objects, the NMI scores of consensus results, and the time expenses (as shown in Table 3, Figs. 5 and 6, respectively).

For the generation of super-objects, the RSOs with sizes no greater than \( \tau \) will be merged to their nearest neighbors. As shown in Table 3, even when setting \( \tau = 1 \), the number of the generated super-objects is still significantly less than that of the RSOs. By comparing the number of RSOs and the number of super-objects, it can be found that there are a large number of very tiny fragments in the set of RSOs. Especially, for the Semeion, MNIST, ISOLET, PD, and Letters datasets, the number of super-objects with \( \tau = 1 \) is less than a half of the number of RSOs, which means that more than a half of the RSOs only contain a single data object. With the ability of collecting the tiny fragments, the super-object is a more adaptive and compact representation than the RSO. By using different values of \( \tau \), the number of the generated super-objects can be adjusted (as shown in Table 3).

The performance of the proposed ECFG approach w.r.t. varying \( \tau \) is illustrated in Fig. 5. The execution time of ECFG w.r.t. varying \( \tau \) is illustrated in Fig. 6. Note that the X-axis in Figs. 5 and 6 corresponds to \( \log_2 \tau \). On the benchmark datasets, the proposed approach achieves consistent performance with varying parameter \( \tau \). In one aspect, setting \( \tau \) to smaller values leads to slightly better consensus results on the Vowel, Semeion, and Letters datasets (see Fig. 5). In another aspect, setting \( \tau \) to smaller values also results in higher execution time (see Fig. 6) for that the number of super-objects gets greater with smaller \( \tau \). Empirically, it is suggested that the parameter \( \tau \) be set in the interval of \( [1, \sqrt{N}/2] \). In this paper, we use \( \tau = \sqrt{N}/2 \) in all of the experiments on the benchmark datasets.

4.3. Comparison against base clusterings

The purpose of ensemble clustering is to combine multiple base clusterings into a probably better and more robust clustering. In this section, we compare the ECFG approach against the base clusterings on the benchmark datasets. We run the ECFG approach 100 times on each dataset. For each run, an ensemble of \( M \) base clusterings is randomly generated (see Section 4.1) and the NMI scores of the base clusterings and the consensus clustering by ECFG are computed, respectively. The average NMI scores over 100 runs of ECFG and the base clusterings are illustrated in Fig. 7. For the Iris, Vowel, Semeion, MNIST, ISOLET, and PD datasets, the ECFG approach produces significantly better clustering results than the base clusterings in terms of NMI. For the VS and Letters datasets, the average NMI scores of the base clusterings are 0.165 and 0.230, respectively, which indicates the low quality of the base clusterings. Based on the low-quality clustering ensembles for the VS and Letters datasets, the proposed approach is still capable of producing better consensus clusterings with average NMI scores of 0.201 and 0.285, respectively. As shown in Fig. 7, the proposed approach produces overall better and more robust clusterings than the base clusterings on the benchmark datasets.

4.4. Comparison against non-automatic ensemble clustering approaches

In this section, we compare the proposed ECFG approach against three non-automatic ensemble clustering approaches, that is, EAC [1], WCT [3], and WEAC [10]. Note that the proposed ECFG approach is an automatic approach. These non-automatic approaches lack the ability to estimate the cluster number automatically and need to take the cluster number of the final clustering as input. In our experiments, for comparison, the cluster number of these non-automatic approaches is set to be the same as the automatically estimated cluster number by ECFG.
The average NMI scores over 100 runs of the proposed approach and the three non-automatic approaches are shown in Table 4. For the VS and MNIST datasets, the proposed ECFG approach achieves nearly the best performance with the NMI scores slightly lower than WCT and EAC, respectively. For the other six datasets, the proposed approach achieves the best clustering results among the test approaches. Especially, for Iris, Semeion, ISOLET, and Letters datasets, the proposed approach achieves significantly higher scores than the baseline approaches. As an automatic ensemble clustering approach, the ECFG approach achieves overall comparable or better performance than the baseline non-automatic approaches on the benchmark datasets.

4.5. Comparison against automatic ensemble clustering approaches

In this section, we compare the proposed approach against four automatic ensemble clustering approaches, namely, COMUSA [27], DICLENS [13], COMUSACL [14], and COMUSACL-DEW [14]. Although the proposed approach and these four baseline approaches are all automatic approaches, which are capable of estimating the cluster number of the final clustering automatically, there are mainly three differences between our approach and the four baseline automatic approaches. First, in our approach, each base clustering is associated with a reliability which is estimated by the EM algorithm and further used for estimating the consensus clustering, whereas in the four baseline approaches all base clusterings are treated equally. Second, our approach uses the super-object representation to facilitate the computation, whereas the four baseline approaches use the data objects or the clusters as the operating units, which may suffer from the growth of the data size and the ensemble size. The third difference between our approach and the four baseline approaches is that the four baseline approaches obtain the consensus clustering heuristically, whereas our approach models the
ensemble clustering problem into a probabilistic framework and obtains the consensus clustering by solving an optimization problem via factor graph.

To achieve a fair comparison, for each benchmark dataset, we run the proposed approach and the four baseline automatic approaches 100 times, respectively. For each run, the ensemble of base clusterings is randomly generated as introduced in Section 4.1. The average performance over 100 runs in terms of NMI is shown in Table 5. For all benchmark datasets, the ECFG approach yields the best average performance among the test approaches (see Table 5). Especially, for the Iris, Vowel, Semeion, PD, and Letters datasets, the ECFG approach significantly outperforms the baseline automatic approaches in terms of the average NMI scores over 100 runs. As shown in Table 5, the proposed approach achieves overall more accurate clustering results on the benchmark datasets when compared against the other automatic ensemble clustering approaches.

4.6. Robustness to ensemble size \( M \)

In this section, we further test the robustness of our approach w.r.t. varying ensemble size \( M \). For each ensemble size \( M \), we run ECFG, DICLENS, COMUSACL, and COMUSACL-DEW 100 times, respectively, and report their average performance w.r.t. varying \( M \) in Fig. 8. The ECFG approach yields consistent performance when the ensemble size \( M \) varies from 10 to 100. For the Iris, Vowel, Semeion, and PD datasets, the ECFG approach significantly outperforms the other three approaches. For the other four datasets, the ECFG approach also achieves the best or nearly the best performance in terms of NMI among the test approaches. As shown in Fig. 8, the proposed approach yields overall better and more robust performance than the baseline approaches with varying ensemble sizes on the benchmark datasets.

4.7. Execution time

In this section, we test the execution time of ECFG and the baseline approaches. We first use a fixed ensemble size \( M = 20 \) to evaluate the time expenses of different approaches on the benchmark datasets. Then we compare the efficiency of these ensemble clustering approaches with varying ensemble sizes. All experiments are conducted in Matlab R2014a 64-bit on a workstation (Windows Server 2008 R2 64-bit, 8 Intel 2.40 GHz processors, 96GB of RAM).

As shown in Table 6, for the two smallest datasets, namely, Iris and VS, the COMUSACL and COMUSACL-DEW approaches are the two fastest approaches, whereas the proposed approach is slightly slower than COMUSACL and COMUSACL-DEW. For all of the benchmark datasets except Iris and VS, the proposed approach is faster than all baseline approaches. Especially, for the large datasets such as PD and Letters, our approach exhibits a significant advantage in execution time over the baseline approaches. To process the clustering ensemble for the Letters dataset, the time costs (in seconds) of the seven baseline approaches, namely, EAC, WCT, WEAC, COMUSA, DICLENS, COMUSACL, and COMUSACL-DEW, are 369.30, 558.24, 369.62, 5140.94, 199.83, 42.26, and 43.09, respectively, whereas the time cost (in seconds) of ECFG is only 12.99. The significant difference in efficiency among the test approaches is mainly due to the different levels of granularity that they work at. Specifically, the EAC, WCT, WEAC, and COMUSA approaches work at the object-level, the DICLENS, COMUSACL, and COMUSACL-DEW approaches work at the cluster-level, and the proposed ECFG approach works at the super-object-level. On the whole, the ECFG approach shows overall the best efficiency on the benchmark datasets. Although slower than ECFG when dealing with large datasets, the DICLENS, COMUSACL, and COMUSACL-DEW approaches still show better efficiency than the other four baseline approaches which work at the object-level.

We further compare the execution time of ECFG against DICLENS, COMUSACL, and COMUSACL-DEW with varying ensemble sizes on the benchmark datasets. As illustrated in Fig. 9, with the ensemble size \( M = 10 \), the ECFG approach shows slightly better or comparable efficiency than the DICLENS, COMUSACL, and COMUSACL-DEW approaches on the benchmark datasets. However, when the ensemble size goes beyond 30, the ECFG approach significantly outperforms the other approaches in terms of time efficiency. For the LR dataset, with the ensemble size \( M = 100 \), the time costs (in seconds) of DICLENS, COMUSACL, and COMUSACL-DEW are 4505.70, 967.76, and 934.28, respectively, whereas the time cost (in seconds) of ECFG is only 72.70. As shown in Table 6 and Fig. 9, the DICLENS, COMUSACL, and COMUSACL-DEW approaches scale well with regard to the data size \( N \), but they scale poorly with regard to the ensemble size \( M \). Compared to the baseline approaches, our approach shows a significant advantage in the scalability with regard to both the data size and the ensemble size.

5. Conclusions

In this paper, we propose a novel ensemble clustering approach termed ECFG. Compared to the existing approaches, our approach is distinguished in three aspects: (i) the cluster number is automatically obtained; (ii) the reliability of each base clustering can be estimated and exploited in the ensemble clustering process; (iii) our approach is efficient in terms of both large data sizes and large ensemble sizes. In this paper, the super-object representation is introduced to facilitate the computation of the overall ensemble process. We probabilistically formulate the ensemble clustering problem into a binary linear programming problem. To solve this optimization problem, we propose an efficient solver based on the factor graph technique. We represent the objective function as a factor graph and utilize the max-product belief propagation to generate the approximate solution. In our approach, the consensus clustering and the reliability of each base clustering are estimated.
Table 6

Execution time (in seconds) of different approaches with ensemble size $M = 20$. The shortest execution time in each column is highlighted in bold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Iris</th>
<th>VS</th>
<th>Vowel</th>
<th>Semeion</th>
<th>MNIST</th>
<th>ISOLET</th>
<th>PD</th>
<th>Letters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECFG</td>
<td>0.07</td>
<td>0.14</td>
<td>0.18</td>
<td>0.29</td>
<td>1.13</td>
<td>0.96</td>
<td>0.99</td>
<td>12.99</td>
</tr>
<tr>
<td>EAC</td>
<td>0.05</td>
<td>0.60</td>
<td>0.75</td>
<td>1.77</td>
<td>17.25</td>
<td>45.38</td>
<td>109.11</td>
<td>369.30</td>
</tr>
<tr>
<td>WCT</td>
<td>0.06</td>
<td>0.85</td>
<td>1.10</td>
<td>2.95</td>
<td>19.72</td>
<td>76.30</td>
<td>162.66</td>
<td>558.24</td>
</tr>
<tr>
<td>WEAC</td>
<td>0.16</td>
<td>0.75</td>
<td>0.91</td>
<td>2.01</td>
<td>17.85</td>
<td>46.32</td>
<td>110.59</td>
<td>369.62</td>
</tr>
<tr>
<td>COMUSA</td>
<td>0.07</td>
<td>1.29</td>
<td>2.09</td>
<td>5.91</td>
<td>365.23</td>
<td>346.27</td>
<td>998.60</td>
<td>5140.94</td>
</tr>
<tr>
<td>DICLENS</td>
<td>0.09</td>
<td>0.41</td>
<td>1.03</td>
<td>1.49</td>
<td>14.20</td>
<td>9.71</td>
<td>5.51</td>
<td>43.09</td>
</tr>
<tr>
<td>COMUSACL</td>
<td>0.03</td>
<td>0.12</td>
<td>0.58</td>
<td>0.99</td>
<td>6.19</td>
<td>4.21</td>
<td>3.11</td>
<td>42.26</td>
</tr>
<tr>
<td>COMUSACL-DEW</td>
<td>0.03</td>
<td>0.13</td>
<td>0.62</td>
<td>0.90</td>
<td>6.36</td>
<td>4.17</td>
<td>3.03</td>
<td>43.09</td>
</tr>
</tbody>
</table>

Fig. 8. The average performance over 100 runs in terms of NMI by different methods with varying ensemble size $M$.

Fig. 9. The execution time (in seconds) of different methods with varying ensemble size $M$. 

Please cite this article as: D. Huang, et al., Ensemble clustering using factor graph, Pattern Recognition (2015), http://dx.doi.org/10.1016/j.patcog.2015.08.015
ilitively and simultaneously. Extensive experiments have been conducted on multiple real-world datasets. The experimental results have shown that our approach significantly outperforms the state-of-the-art approaches in terms of both clustering accuracy and efficiency.

Conflict of interest
None declared.

Acknowledgments
This work was supported by National Science & Technology Pillar Program (No. 2012BAK16B06), NSFC (61173084 & 61502543), the GuangZhou Program (Grant no. 201508010032), the CCF-Tencent Open Research Fund, and the PhD Start-up Fund of Natural Science Foundation of Guangdong Province, China (No. 2014A030310180).

References

Dong Huang received his B.S. degree in computer science in 2009 from South China University of Technology, China. He received his M.Sc. degree in computer science in 2011 and his Ph.D. degree in computer science in 2015, both from Sun Yat-sen University, China. He joined South China Agricultural University in 2015 as an assistant professor with College of Mathematics and Informatics. His research interests include data mining and pattern recognition.

Jianhuang Lai received his M.Sc. degree in applied mathematics in 1989 and his Ph.D. in mathematics in 1999 from Sun Yat-sen University, China. He joined Sun Yat-sen University in 1989 as an assistant professor, where currently he is a professor with the Department of Automation of School of Information Science and Technology and dean of School of Information Science and Technology. His current research interests are in the areas of digital image processing, pattern recognition, multimedia communication, wavelet and its applications. He has published over 100 scientific papers in the international journals and conferences on image processing and pattern recognition, e.g. IEEE TPAMI, IEEE TKDE, IEEE TNN, IEEE TIP, IEEE TSMC (Part B), Pattern Recognition, ICCV, CVPR and ICDM. Prof. Lai serves as a standing member of the Image and Graphics Association of China and also serves as a standing director of the Image and Graphics Association of Guangdong.

Chang-Dong Wang received the B.S. degree in applied mathematics in 2008, M.Sc. degree in computer science in 2010 and Ph.D. degree in computer science in 2013, all from Sun Yat-sen University, China. He is a visiting student at University of Illinois at Chicago from January 2012 to November 2012. He joined Sun Yat-sen University in 2013 as an assistant professor with School of Mobile Information Engineering. His current research interests include machine learning and data mining. He has published over 20 scientific papers in international journals and conferences such as IEEE TPAMI, IEEE TKDE, IEEE TSMC-C, Pattern Recognition, KAIR, Neurocomputing, ICDM and SDM. His ICDM 2010 paper won the Honorable Mention for Best Research Paper Awards.