Advanced Review

Cluster ensembles

Joydeep Ghosh* and Ayan Acharya

Cluster ensembles combine multiple clusterings of a set of objects into a single consolidated clustering, often referred to as the *consensus* solution. Consensus clustering can be used to generate more robust and stable clustering results compared to a single clustering approach, perform distributed computing under privacy or sharing constraints, or reuse existing knowledge. This paper describes a variety of algorithms that have been proposed to address the cluster ensemble problem, organizing them in conceptual categories that bring out the common threads and lessons learnt while simultaneously highlighting unique features of individual approaches. *WIREs Data Mining Knowl Discov* 2011 00 1–11 DOI: 10.1002/widm.32

INTRODUCTION

Cluster ensembles address the problem of combining multiple 'base clusterings' of the same set of objects into a single consolidated clustering. Each base clustering refers to a *grouping* of the same set of objects or its transformed (or perturbed) version using a suitable clustering algorithm. The consolidated clustering is often referred to as the *consensus* solution. At first glance, this problem sounds similar to the widely prevalent use of combining multiple classifiers to solve difficult classification problems, using techniques such as bagging, boosting, and output combining.[1–3] However, combining multiple clusterings poses additional challenges. First, the number of clusters produced may differ across the different *base* solutions.[4] The appropriate number of clusters in the consensus is also not known in advance and may depend on the scale at which the data is inspected. Moreover, cluster labels are symbolic and thus aligning cluster labels across different solutions requires solving a potentially difficult correspondence problem. Also, in the typical formulation,[a] the original data used to yield the base solutions are not available to the consensus mechanism, which has only access to the sets of cluster labels. In some schemes, one does have control on how the base clusterings are produced,[6] whereas in others even this is not granted in order to allow applications involving knowledge reuse,[7] as described later. There are many reasons for using a cluster ensemble. In fact, the potential motivations and benefits are much broader than those for using classification or regression ensembles, where one is primarily interested in improving predictive accuracy. These reasons include:

1. *Improved quality of solution*. Just as ensemble learning has been proved to be more useful compared to single-model solutions for classification and regression problems, one may expect that cluster ensembles will improve the quality of results as compared to a single clustering solution. It has been shown that using cluster ensembles leads to more accurate results on average as the ensemble approach takes into account the biases of individual solutions.[8,9]
2. *Robust clustering*. It is well known that the popular clustering algorithms often fail spectacularly for certain datasets that do not match well with the modeling assumptions.[10] A cluster ensemble approach can provide a 'meta' clustering model that is much more robust in the sense of being able to provide good results across a very wide range of datasets. As an example, by using an ensemble that includes approaches such as *k*-means, SOM, and DBSCAN that are typically better suited to low-dimensional metric spaces, as well as base clusterers designed for high-dimensional sparse spaces (spherical *k*-means, Jaccard-based graph clustering, etc.), one can perform well across a wide range of data dimensionality.[7] Authors in Ref 11 present several empirical results on the robustness of the results in document clustering by using feature diversity and consensus clustering.

*Correspondence to: ghosh@ece.utexas.edu

Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA

DOI: 10.1002/widm.32

3. *Model selection*. Cluster ensembles provide a novel approach to the model selection problem by considering the match across the base solutions to determine the final number of clusters to be obtained.[12]
4. *Knowledge reuse*. In certain applications, domain knowledge in the form of a variety of clusterings of the objects under consideration may already exist due to past projects. A consensus solution can integrate such information to get a more consolidated clustering. Several examples are provided in Ref 7, where such scenarios formed the main motivation for developing a consensus clustering methodology. As another example, a categorization of web pages based on text analysis can be enhanced by using the knowledge of topical document hierarchies available from Yahoo! or DMOZ.
5. *Multiview clustering*. Often the objects to be clustered have multiple aspects or 'views', and base clusterings may be built on distinct views that involve nonidentical sets of features or subsets of data points. In marketing applications, for example, customers may be segmented based on their needs, psychographic or demographic profiles, attitudes, etc. Different views can also be obtained by considering qualitatively different distance measures, an aspect that was exploited in clustering multifaceted proteins to multiple functional groups in Ref 13. Consensus clustering can be effectively used to combine all such clusterings into a single consolidated partition. Strehl and Ghosh[7] illustrated empirically the utility of cluster ensembles in two orthogonal scenarios:
   (a) Feature distributed clustering (FDC): Different base clusterings are built by selecting different subsets of the features but utilizing all the data points.
   (b) Object distributed clustering (ODC): Base clusterings are constructed by selecting different subsets of the data points but utilizing all the features.

   Fern and Brodley[14] also showed that clustering in high dimension is much more effective compared to clustering with principal component analysis (PCA) when the data points are randomly projected onto a subspace, clustered in that subspace and consensus clustering is performed with this ensemble.
6. *Distributed computing*. In certain situations, data is inherently distributed and it is not possible to first collect the entire data at a central site due to privacy/ownership issues or computational, bandwidth and storage costs.[15] An ensemble can be used in situations where each clusterer has access only to a subset of the features of each object, as well as where each clusterer has access only to a subset of the objects.[7,12]

The problem of combining multiple clusterings can be viewed as a special case of the more general problem of comparison and consensus of data 'classifications', studied in the pattern recognition and related application communities in the 1970s and 1980s. In this literature, 'classification' was used in a broad sense to include clusterings, unrooted trees, graphs, etc., and problem-specific formulations were made (see Ref 16 for a broad, more conceptual coverage). For example, in the building of phylogenetic trees, it is important to get a strict consensus solution, wherein two objects occur in the same consensus partition if and only if they occur together in all individual clusterings,[17] typically resulting in a consensus solution at a much coarser resolution than the individual solutions. A quick overview with pointers to such literature is given by Ayad and Kamel.[4] A reasonable coverage of this broader class of problems is not feasible here, instead this paper focuses on the cluster ensemble formulations and associated algorithms that have been proposed in the past decade. Section *The Cluster Ensemble Problem* formally defines the cluster ensemble problem within this context. Section *Cluster Ensemble Algorithms* presents a variety of approaches to designing cluster ensembles.

THE CLUSTER ENSEMBLE PROBLEM

We denote a vector by a bold-faced letter and a scalar variable or a set in normal font. We start by considering r base clusterings of a data set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ with the qth clustering containing $k^{(q)}$ clusters. The most straightforward representation of the qth clustering is $\lambda^{(q)} = \{\mathcal{C}_\ell | \ell = 1, 2, \ldots, k^{(q)}$ and $\mathcal{C}_\ell \subseteq \mathcal{X}\}$. Here, each clustering is denoted by a collection of subsets (not necessarily disjoint) of the original dataset. For hard partitional clustering (clustering where each object is assigned to a single cluster only), the qth clustering can alternatively be represented by a label vector $\lambda^{(q)} \in \mathbb{Z}_+^n$. In this representation, each object is assigned some cluster label and 0 is used if the corresponding object was not available to that clusterer. The third possible way of representation of an

TABLE 1 | Contingency Table Explaining the Three Measures

	$\mathcal{C}_1{}^{(b)}$	$\mathcal{C}_2{}^{(b)}$	...	$\mathcal{C}_{k^{(b)}}{}^{(b)}$	Sum
$\mathcal{C}_1{}^{(a)}$	n_{11}	n_{12}	...	$n_{1k^{(b)}}$	$n_1{}^{(a)}$
$\mathcal{C}_2{}^{(a)}$	n_{21}	n_{22}	...	$n_{2k^{(b)}}$	$n_2{}^{(a)}$
...	...	...	...	...	...
...	...	...	...	...	...
$\mathcal{C}_{k^{(a)}}{}^{(a)}$	$n_{k^{(a)}1}$	$n_{k^{(a)}2}$	...	$n_{k^{(a)}k^{(b)}}$	$n_{k^{(a)}}{}^{(a)}$
Sum	$n_1{}^{(b)}$	$n_2{}^{(b)}$	...	$n_{k^{(b)}}{}^{(b)}$	n

individual clustering is by the binary membership indicator matrix $\mathbf{H}^q \in \{0,1\}^{1\times k^{(q)}}$, which is defined as $\mathbf{H}^q = \{h^q_{i\ell} | h^q_{i\ell} \in \{0,1\}\ \forall \mathbf{x}_i, \mathcal{C}_\ell, \lambda^{(q)}\}$. For partitional clustering, we additionally have $\sum_{\ell=1}^{k^{(q)}} h^q_{i\ell} = 1 \forall \mathbf{x}_i \in \mathcal{X}$.

A *consensus function* Γ is defined as a function $\mathbb{Z}_+^{n\times r} \to \mathbb{Z}_{++}^n$ mapping a set of clusterings to an integrated clustering $\Gamma : \lambda^{(q)} | q \in \{1, 2, \ldots\ldots, r\} \to \hat{\lambda}$. For conciseness, we shall denote the set of clusterings $\{\lambda^{(q)}\}_{q=1}^r$ that is available to the consensus mechanism by Λ. Moreover, the results of any hard clustering[b] of n objects can be represented as a binary, symmetric $n \times n$ *coassociation matrix*, with an entry being 1 if the corresponding objects are in the same cluster and 0 otherwise. For the qth base clustering, this matrix is denoted by $S^{(q)}$ and is given by

$$S_{ij}^{(q)} = \begin{cases} 1 & (i,j) \in C_\ell(\lambda^{(q)}) \quad \text{for some } \ell \in \{1,2,\ldots,k^{(q)}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Broadly speaking, there are two main approaches to obtaining a consensus solution and determining its quality. One can postulate a probability model that determines the labelings of the individual solutions, given the true consensus labels, and then solve a maximum likelihood formulation to return the consensus.[18,19] Alternately, one can directly seek a consensus clustering that agrees the most with the original clusterings. The second approach requires a way of measuring the similarity between two clusterings, for example to evaluate how close the consensus solution is to each base solution. Popular measures for such a purpose include: (1) adjusted Rand index (ARI), (2) normalized mutual information (NMI), and (3) variation of information (VI). We now briefly introduce these measures as they are used in several algorithms covered in the section *Cluster Ensemble Algorithms*.

Adjusted Rand Index

Suppose we have two candidate clusterings $\lambda^{(a)} = \{\mathcal{C}_h^{(a)} | h = 1, 2, \ldots, k^{(a)}\}$ and $\lambda^{(b)} = \{\mathcal{C}_\ell{}^{(b)} | \ell = 1, 2, \ldots, k^{(b)}\}$. Let $n_h^{(a)}$ be the number of objects in cluster $C_h^{(a)}$ and $n_l^{(b)}$ be the number of objects in cluster $C_l^{(b)}$. Table 1 is a contingency table that shows the overlap between different clusters of these clusterings, where $n_{h\ell} = |\mathcal{C}_h^{(a)} \cap C_\ell^{(b)}|$. The ARI, proposed by Hubert and Arabie,[20] is defined as

$$\phi^{(\mathrm{ARI})}(\lambda^{(a)}, \lambda^{(b)}) = \frac{\sum_{h\ell}\binom{n_{h\ell}}{2} - S_a S_b / \binom{n}{2}}{\frac{1}{2}(S_a + S_b) - S_a S_b / \binom{n}{2}}, \tag{2}$$

where $S_a = \sum_h \binom{n_h^{(a)}}{2}$ and $S_b = \sum_\ell \binom{n_\ell^{(b)}}{2}$. The second term in both numerator and denominator adjusts for the expected number of overlaps that will occur 'by chance', i.e., if the cluster labels are totally uncorrelated.

Normalized Mutual Information

Strehl and Ghosh[7] proposed NMI to measure the similarity between two candidate clusterings. The entropy associated with clustering $\lambda^{(a)}$ is $H(\lambda^{(a)}) = -\sum_h \frac{n_h^{(a)}}{n} \log(\frac{n_h^{(a)}}{n})$ and that with clustering $\lambda^{(b)}$ is $H(\lambda^{(b)}) = -\sum_\ell \frac{n_\ell^{(b)}}{n} \log(\frac{n_\ell^{(b)}}{n})$. Similarly, the joint entropy of $\lambda^{(a)}$ and $\lambda^{(b)}$ is defined as, $H(\lambda^{(a)}, \lambda^{(b)}) = -\sum_{h,\ell} \frac{n_{h\ell}}{n} \log(\frac{n_{h\ell}}{n})$. Now, the NMI between $\lambda^{(a)}$and $\lambda^{(b)}$ is defined as

$$\begin{aligned} \phi^{(\mathrm{NMI})}(\lambda^{(a)}, \lambda^{(b)}) &= \frac{H(\lambda^{(a)}) + H(\lambda^{(b)}) - H(\lambda^{(a)}, \lambda^{(b)})}{\sqrt{H(\lambda^{(a)}) H(\lambda^{(b)})}} \\ &= \frac{I(\lambda^{(a)}, \lambda^{(b)})}{\sqrt{H(\lambda^{(a)}) H(\lambda^{(b)})}}. \end{aligned} \tag{3}$$

Here, $I(\lambda^{(a)}, \lambda^{(b)})$ is the mutual information between two clusterings $\lambda^{(a)}$ and $\lambda^{(b)}$ which is normalized by the geometric mean of $H(\lambda^{(a)})$ and $H(\lambda^{(b)})$ to compute the NMI. It should be noted that $I(\lambda^{(a)}, \lambda^{(b)})$ is non-negative and has no upper bound. $\phi^{(NMI)}(\lambda^{(a)}, \lambda^{(b)})$, on the other hand, lies between 0 and 1 and is suitable for easier interpretation and comparisons.

Variation of Information

VI is another information theoretic measure proposed for cluster validation,[21] and defined as

$$\phi^{(VI)}(\lambda^{(a)}, \lambda^{(b)}) = H(\lambda^{(a)}) + H(\lambda^{(b)}) - 2I(\lambda^{(a)}, \lambda^{(b)}) \tag{4}$$

It turns out that VI is a metric. But its original definition is not consistent if data sets of different sizes and clusterings with different number of clusters are considered. Wu et al.[22] proposed a normalized version of VI (NVI) which is empirically shown to alleviate this inconsistency. NVI is defined as follows:

$$\begin{aligned}\phi^{(NVI)}(\lambda^{(a)}, \lambda^{(b)}) &= \frac{\phi^{(VI)}(\lambda^{(a)}, \lambda^{(b)})}{H(\lambda^{(a)}) + H(\lambda^{(b)})} \\ &= 1 - \frac{2I(\lambda^{(a)}, \lambda^{(b)})}{H(\lambda^{(a)}) + H(\lambda^{(b)})}.\end{aligned} \tag{5}$$

It can be seen that NVI and NMI are closely related to each other. Also, all of the above three measures lie in the range [0, 1] with a unity value signifying maximum agreement between two clusterings and a zero value implying no agreement.

Given any of the three pairwise measures of similarity between two clusterings given above, one can express the average normalized similarity measure between a set of r labelings, Λ, and a single consensus labeling, $\hat{\lambda}$, by

$$\phi(\Lambda, \hat{\lambda}) = \frac{1}{r}\sum_{q=1}^{r} \phi(\lambda^{(q)}, \hat{\lambda}), \tag{6}$$

where ϕ refers to any of $\phi^{(\mathrm{ARI})}$, $\phi^{(\mathrm{NMI})}$, and $\phi^{(\mathrm{NVI})}$. This serves as the objective function in certain cluster ensemble formulations, where the goal is to find the combined clustering $\hat{\lambda}$ with $\hat{k}$ clusters such that $\phi(\Lambda, \hat{\lambda})$ is maximized. It turns out though that this objective is in general intractable, so heuristic approaches have to be resorted to.

Topchy et al.[23] offered a different perspective on the problem of consensus clustering and answered a fundamental question related to the asymptotic accuracy of the ensemble solution. Given a dataset $\mathcal{X}$ of n number of points, we can determine the number of all possible clusterings of the dataset into k nonempty clusters. This number, often denoted by $\mathcal{S}_n^{(k)}$, is in fact called Stirling number of second kind.[24] Let the set corresponding to this Stirling number be denoted by $\Lambda_n^{(k)} = \{\lambda^{(q)}\}_{q=1}^{\mathcal{S}_n^{(k)}}$. With a probability measure μ imposed on $\Lambda_n^{(k)}$, the optimal (or the true) clustering is given by

$$\lambda^0 = \arg\min_{\lambda \in \Lambda_n^{(k)}} \sum_{q=1}^{\mathcal{S}_n^{(k)}} \mu(\lambda^{(q)}) \phi(\lambda^{(q)}, \lambda). \tag{7}$$

A cluster ensemble Λ of the data points $\mathcal{X}$ can now be built by choosing $r \leq \mathcal{S}_n^{(k)}$ clusterings from $\Lambda_n^{(k)}$. The optimal solution to this ensemble is given by

$$\hat{\lambda} = \arg\min_{\lambda \in \Lambda_n^{(k)}} \sum_{q=1}^{r} \mu(\lambda^{(q)}) \phi(\lambda^{(q)}, \lambda) \text{ such that } \lambda^{(q)} \in \Lambda. \tag{8}$$

This notion of consensus solution is termed as *consensus as the mean clustering* in Ref 23. The authors showed that $P(\hat{\lambda} \neq \lambda^0) \leq \exp(-r\epsilon)$ with ϵ being some positive number. The result implies that the probability of incorrect consensus according to Eq. (8) decreases exponentially with number of clusterings in the ensemble. When r approaches infinity, $P(\hat{\lambda} \neq \lambda^0)$ approaches zero. In Ref 25, the authors, however, viewed cluster ensemble as a *median clustering* problem and used generalized entropy to modify the NMI measure explained earlier.

CLUSTER ENSEMBLE ALGORITHMS

Cluster ensemble methods are now presented under three categories: (1) probabilistic approaches, (2) approaches based on coassociation, and (3) direct and other heuristic methods.

Probabilistic Approaches to Cluster Ensembles

The two basic probabilistic models for solving cluster ensembles are described in this subsection.

A Mixture Model for Cluster Ensembles

In a typical mixture model[26] approach to clustering, such as fitting the data using a mixture of Gaussians, there are $\hat{k}$ mixture components, one for each cluster. A component-specific parametric distribution is used to model the distribution of data attributed to a specific component. Such an approach can be applied to form the consensus decision if the number of consensus clusters is specified. This immediately yields the pioneering approach taken in Ref 18. We describe it in a bit more detail as this work is essential to build an understanding of later works.[19,27]

In the basic mixture model of cluster ensembles,[18] each object $\mathbf{x}_i$ is represented by $\mathbf{y}_i = \Lambda(\mathbf{x}_i)$, i.e., the labels provided by the base clusterings.

We assume that there are $\hat{k}$ consensus clusters each of which is indexed by $\hat{\ell}$. Corresponding to each consensus cluster $\hat{\ell}$ and each base clustering q, we have a multinomial distribution $\boldsymbol{\beta}_{\hat{\ell}}^{(q)}$ of dimension $k^{(q)}$. Therefore, a single draw from this distribution is a cluster label corresponding to the qth base clustering. The underlying generative process is assumed as follows:

For ith data point $\mathbf{x}_i$,

1. Choose $\mathbf{z}_i = \mathbf{I}_{\hat{\ell}}$ such that $\hat{\ell} \sim$ multinomial($\boldsymbol{\theta}$). Here $\mathbf{I}_{\hat{\ell}}$ is a vector of dimension $k^{(q)}$ with only the $\hat{\ell}$th component being 1 and rest of the components being 0. $\boldsymbol{\theta}$ is a multinomial distribution of dimension $\hat{k}$.
2. For the qth base clustering of ith data point, choose the base clustering result $y_{iq} = \ell \sim$ multinomial($\boldsymbol{\beta}_{\hat{\ell}}^{(q)}$).

These probabilistic assumptions give rise to a simple maximum log-likelihood problem that can be solved using the expectation maximization algorithm. This model also takes care of the missing labels in a natural way.

Bayesian Cluster Ensembles

A Bayesian version of the multinomial mixture model described above was subsequently proposed by Wang et al.[19] As in the simple mixture model, we assume $\hat{k}$ consensus clusters with $\boldsymbol{\beta}_{\hat{\ell}}^{(q)}$ being the multinomial distribution corresponding to each consensus cluster $\hat{\ell}$ and each base clustering q. The complete generative process for this model is as follows:

For ith data point $\mathbf{x}_i$,

1. Choose $\boldsymbol{\theta}_i \sim$ Dirichlet($\boldsymbol{\alpha}$) where $\boldsymbol{\theta}_i$ is a multinomial distribution with dimension $\hat{k}$.
2. For the qth base clustering:
   (a) Choose $\mathbf{z}_{iq} = \mathbf{I}_{\hat{\ell}}$ such that $\hat{\ell} \sim$ multinomial($\boldsymbol{\theta}_i$). $\mathbf{I}_{\hat{\ell}}$ is a vector of dimension $k^{(q)}$ with only $\hat{\ell}$th component being 1 and rest of the components being 0.
   (b) Choose the base clustering result $y_{iq} = \ell \sim$ multinomial($\boldsymbol{\beta}_{\hat{\ell}}^{(q)}$).

So, given the model parameters ($\boldsymbol{\alpha}, \beta = \{\boldsymbol{\beta}_{\hat{\ell}}^{(q)}\}$), the joint distribution of latent and observed variables $\{\mathbf{y}_i, \mathbf{z}_i, \boldsymbol{\theta}_i\}$ is given by

$$p(\mathbf{y}_i, \mathbf{z}_i, \boldsymbol{\theta}_i | \boldsymbol{\alpha}, \beta) = p(\boldsymbol{\theta}_i|\boldsymbol{\alpha}) \prod_{q=1}^{r} \left\{ p(\mathbf{z}_{iq} = \mathbf{I}_{\hat{\ell}}|\boldsymbol{\theta}_i) p\left(y_{iq}|\boldsymbol{\beta}_{\hat{\ell}}^{(q)}\right) \right\}^{\mathbf{1}_{y_{iq}}}, \quad (9)$$

where $\mathbf{1}_{y_{iq}}$ is 1 if there exists a qth base clustering result for $\mathbf{y}_i$ and 0 otherwise. The marginals $p(\mathbf{y}_i|\boldsymbol{\alpha}, \beta)$ can further be calculated by integrating over the hidden variables $\{\mathbf{z}_i, \boldsymbol{\theta}_i\}$. The authors used variational expectation maximization and Gibbs' sampling for inference and parameter estimation. The graphical model corresponding to this Bayesian version is given in Figure 1(a). To highlight the difference between Bayesian cluster ensembles and the mixture model for cluster ensembles, the graphical model corresponding to the latter is also shown alongside in Figure 1(b). Very recently, a nonparametric version of Bayesian cluster ensemble has been proposed in Ref. 27 which facilitates the number of consensus clusters to adapt with data. It should be noted that although both of the generative models presented above were used only with hard partitional clustering, they could be used for overlapping clustering as well.

Pairwise Similarity-Based Approaches

In pairwise similarity-based approaches, one takes the weighted average of all r coassociation matrices to form an *ensemble coassociation matrix* S which is given as follows:

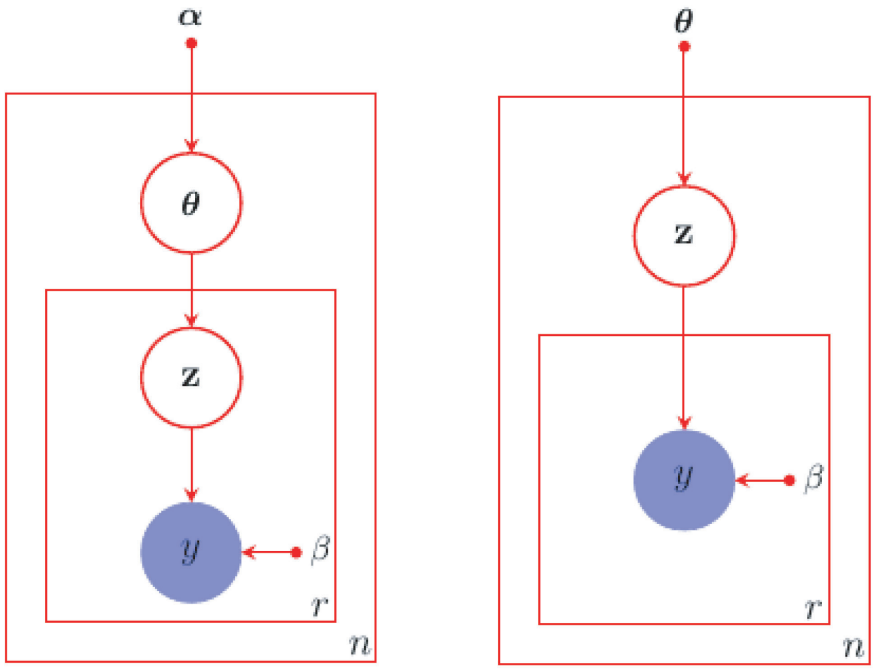
$$S = \frac{1}{r} \sum_{q=1}^{r} w_q S^{(q)}. \quad (10)$$

Here w_q specifies the weight assigned to the qth base clustering. This ensemble coassociation matrix captures the fraction of times a pair of data points is placed in the same cluster across the r base clusterings. The matrix can now be viewed as a similarity matrix (with a corresponding similarity graph) to be used by the consensus mechanism for creating the consensus clusters. This matrix is different from the similarity matrix $\hat{S}$ that we obtain from the consensus solution $\hat{\lambda}$. We will explain the difference in detail in the section *Methods based on Ensemble Coassociation Matrix*.

Note that the coassociation matrix size is itself quadratic in n, which thus forms a lower bound on computational complexity as well as memory requirements, inherently handicapping such a technique for applications to very large datasets. However, it is independent of the dimensionality of the data.

Methods Based on Ensemble Coassociation Matrix

The cluster-based similarity partitioning algorithm (CSPA)[7] used METIS[28] to partition the induced consensus similarity graph. METIS was chosen for its scalability and because it tries to enforce comparable



(a) Bayesian cluster ensembles (b) Mixture model of cluster ensembles

FIGURE 1 | Graphical models for probabilistic approaches to cluster ensembles.

sized clusters. This added constraint is desirable in several application domains[29]; however, if the data is actually labeled with imbalanced classes, then it can lower the match between cluster and class labels. Assuming quasilinear graph clustering, the worst case complexity for this algorithm is $\mathcal{O}(n^2kr)$. Punera and Ghosh[30] later proposed a soft version of CSPA, i.e., one that works on soft base clusterings. Al-Razgan and Domeniconi[31] proposed an alternative way of obtaining nonbinary coassociation matrices when given access to the raw data.

The evidence accumulation approach[6] obtains individual coassociation matrices by random initializations of the k-means algorithm, causing some variation in the base cluster solutions. This algorithm is used with a much higher value of k than the range finally desired. The ensemble coassociation matrix is then formed, each entry of which signifies the relative co-occurrence of two data points in the same cluster. A minimum spanning tree (MST) algorithm (also called the single-link or nearest neighbor hierarchical clustering algorithm) is then applied on the ensemble coassociation matrix. This allows one to obtain nonconvex shaped clusters. Essentially, this approach assumes the designer has access to the raw data, and the consensus mechanism is used to get a more robust solution than what can be achieved by directly applying MST to the raw data.

A related approach was taken by Monti et al.,[32] where the perturbations in the base clustering were achieved by resampling. Any of bootstrapping, data subsampling or feature subsampling can be used as a resampling scheme. If either of the first two options are selected, then it is possible that certain objects will be missing in a given base clustering. Hence when collating the r base coassociation matrices, the (i, j)th entry needs to be divided by the number of solutions that included both objects rather than by a fixed r. This work also incorporated a model selection procedure as follows: The consensus coassociation matrix is formed multiple times. The number of clusters is kept at k_i for each base clustering during the ith experiment, but this number is changed from one experiment to another. A measurement termed as *consensus distribution* describes how the elements of a consensus matrix are distributed within the 0–1 range. The extent to which the consensus matrix is skewed toward a binary matrix denotes how good the base clusterings match one another. This enables one to choose the most appropriate number of consensus clusters $\hat{k}$. Once $\hat{k}$ is chosen, the corresponding ensemble coassociation matrix is fed to a hierarchical

clustering algorithm with average linkage. Agglomeration of clusters is stopped when $\hat{k}$ branches are left.

The iterative pairwise consensus (IPC) algorithm[33] essentially applies model-based k-means[34] to the ensemble coassociation matrix S. The consensus clustering solution $\hat{\lambda} = \{\mathcal{C}_\ell\}_{\ell=1}^{\hat{k}}$ is initialized to some solution, after which a reassignment of points is carried out based on the current configuration of $\hat{\lambda}$. The point $\mathbf{x}_i$ gets assigned to cluster $\mathcal{C}_\ell$, if $\mathbf{x}_i$ has maximum average similarity with the points belonging to cluster $\mathcal{C}_\ell$. Then the consensus solution is updated, and the cycle starts again.

However, both Mirkin[16] and Li et al.[35] showed that the problem of consensus clustering can be framed in a different way than what has been discussed so far. In these works, the distance $d(\lambda^{(q_1)}, \lambda^{(q_2)})$ between two clusterings $\lambda^{(q_1)}$ and $\lambda^{(q_2)}$ is defined as the number of pairs of objects that are placed in the same cluster in one of $\lambda^{(q_1)}$ or $\lambda^{(q_2)}$ and in different cluster in the other, essentially considering the (unadjusted) Rand index. Using this definition, the consensus clustering problem is formulated as

$$\begin{aligned}\arg\min_{\hat{\lambda}} J &= \arg\min_{\hat{\lambda}} \frac{1}{r}\sum_{q=1}^{r} d(\lambda^{(q)}, \hat{\lambda}) \\ &= \arg\min_{\hat{S}} \frac{1}{r}\sum_{q=1}^{r} w_q \sum_{i<j} \left[S_{ij}^{(q)} - \hat{S}_{ij}\right]^2. \quad (11)\end{aligned}$$

Mirkin (Ref 16, Section 5.3.4, p. 260) further proved that the consensus clustering according to criterion (11) is equivalent to clustering over the ensemble coassociation matrix by subtracting a 'soft' and 'uniform' threshold from each of the different consensus clusters. This soft threshold, in fact, serves as a tool to balance cluster sizes in the final clustering. The subtracted threshold has also been used in Ref 36 for consensus clustering of gene-expression data.

In Ref 37, consensus clustering result is obtained by minimizing a weighted sum of the Bregman divergence[38] between the consensus partition and the input partitions with respect to their coassociation matrices. In addition, the authors also show how to generalize their framework in order to incorporate must-link and cannot-link constraints between objects.

Note that the optimization problem in Eq. (11) is over the domain of $\hat{S}$. The difference between the matrices S and $\hat{S}$ lies in the way the optimization problem is posed. If optimization is performed with cluster labels only (as illustrated in the section *Direct Approaches using Cluster Labels*), there is no guarantee of achieving the optimum value $\hat{S} = S$. However, if we are optimizing over the domain of the coassociation matrix we can achieve this optimum value in theory.

Relating Consensus Clustering to Other Optimization Formulations

The coassociation representation of clustering has been used to relate consensus clustering with two other well-known problems.

Consensus Clustering as Nonnegative Matrix Factorization

Li et al.,[35,39] using the same objective function as mentioned in Eq. (11), showed that the problem of consensus clustering can be reduced to a consensus clustering as nonnegative matrix factorization (NNMF) problem. Assuming $U_{ij} = \hat{S}_{ij}$ to be a solution to this optimization problem. we can rewrite Eq. (11) as

$$\arg\min_{U} \sum_{i,j=1}^{n} (S_{ij} - U_{ij})^2 = \arg\min_{U} \|S - U\|_F^2, \quad (12)$$

where the matrix norm is the Frobenius norm. This problem formulation is similar to the NNMF formulation[40] and can be solved using an iterative update procedure. In Ref 41, the cost function J used in Eq. (11) was further modified via normalization to make it consistent with data sets with different number of data points (n) and different number of base clusterings (r).

Consensus Clustering as Correlation Clustering

Gionis et al.[42] showed that a certain formulation of consensus clustering is a special case of correlation clustering. Suppose we have a data set $\mathcal{X}$ and some kind of dissimilarity measurement (distance) between every pair of points in $\mathcal{X}$. This dissimilarity measure is denoted by $d_{ij} \in [0, 1]\, \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$. The objective of correlation clustering[43] is to find a partition $\hat{\lambda}$ such that

$$\begin{aligned}\hat{\lambda} &= \arg\min_{\lambda} d(\lambda) \\ &= \arg\min_{\lambda} \left[\sum_{(i,j):\lambda(\mathbf{x}_i)=\lambda(\mathbf{x}_j)} d_{ij} + \sum_{(i,j):\lambda(\mathbf{x}_i)\neq\lambda(\mathbf{x}_j)} (1 - d_{ij})\right]. \quad (13)\end{aligned}$$

In the above equation, $\lambda(\mathbf{x}_i)$ is the cluster label imposed by λ on $\mathbf{x}_i$. The coassociation view of the cluster ensemble problem reduces to correlation clustering if the distance d_{ij} is defined as $d_{ij} = \frac{1}{r}|\{\lambda^{(q)} : \lambda^{(q)}(\mathbf{x}_i) \neq \lambda^{(q)}(\mathbf{x}_j)\}|\ \forall\ i, j$.

Direct Approaches Using Cluster Labels

Several consensus mechanisms take only the cluster labels provided by the base clusterings as input, and try to optimize an objective function such as Eq. (6), without computing the coassociation matrix.

Graph Partitioning

In addition to CSPA, Strehl and Ghosh[7] proposed two direct approaches to cluster ensembles: hyper graph partitioning algorithm (HGPA) which clusters the objects based on their cluster memberships, and meta clustering algorithm (MCLA), which groups the clusters based on which objects are contained in them. HGPA considers a graph with each object being a vertex. A cluster in any base clustering is represented by a hyperedge connecting the member vertices. The hypergraph clustering package HMETIS (Karypis et al.[44]) was used as it gives quality clusterings and is very scalable. As with CSPA, employing a graph clustering algorithm adds a constraint that favors clusterings of comparable size. Though HGPA is fast with a worst case complexity of $\mathcal{O}(nkr)$, it suffers from an additional problem: if all members of a base cluster are not assigned the same cluster in the consensus solution, the corresponding hyperedge is broken and incurs a constant penalty; however, it cannot distinguish between a situation where only one object was clustered differently and one where several objects were allocated to other groups. Due to this issue, HGPA is often not competitive in terms of cluster quality.

MCLA first forms a meta-graph with a vertex for each base cluster. The edge weights of this graph are proportional to the similarity between vertices, computed using the binary Jaccard measure (number of elements in common divided by the total number of distinct elements). Because the base clusterings are partitional, this results in an r-partite graph. The meta-graph is then partitioned into k balanced meta-clusters. Each meta-cluster, therefore, contains approximately r vertices. Finally, each object is assigned to its most closely associated meta-cluster. Ties are broken randomly. The worst case complexity is $\mathcal{O}(nk^2r^2)$.

Noting that CSPA and MCLA consider either the similarity of objects or similarity of clusters only, a hybrid bipartite graph formulation (HBGF) was proposed in Ref 45. A bipartite graph models both data points and clusters as vertices, wherein an edge exists only between a cluster vertex and a object vertex if the latter is a member of the former. Either METIS or other multiway spectral clustering methods are used to partition this bipartite graph. The corresponding soft versions of CSPA, MCLA, and HBGF have also been developed by Punera and Ghosh.[30] It should be noted that all of CSPA, MCLA, and HGPA were compared with one other using the NMI measure in Ref 7.

Cumulative Voting

The concept of cumulative voting was first introduced in Ref 46 where the authors used bagging to improve the accuracy of clustering procedure. Once clustering is done on a bootstrapped sample, the cluster correspondence problem is solved using iterative relabeling via Hungarian algorithm. Clustering on each bootstrapped sample gives some votes corresponding to each data point and cluster label pair which, in aggregate, decides the final cluster assignment.

A similar approach was adopted in Ref 4. Each base clustering in this contribution is thought of as providing a soft or probabilistic vote on which clusters in the consensus solution its data points should belong to. These votes are then gathered across the base solutions and thresholded to determine the membership of each object to the consensus clusters. Again, this requires a mapping function from the base clusterings to a stochastic one. An information-theoretic criterion based on the information bottleneck principle was used in Ref 4 for this purpose. The mean of all the stochastic clusterings then yields the consensus partition. This approach is able to cater to a range of 'k' in the base clusterings, is fast as it avoids the quadratic time/space complexity of forming a coassociation matrix, and has shown good empirical results as well. Noting that the information bottleneck solutions can be obtained as a special case of Bregman clustering,[38] it should be possible to recast this approach as a probabilistic one.

A variety of heuristic search procedures have also been suggested to hunt for a suitable consensus solution. These include a genetic algorithm formulation[47] and one using a multi-ant colony.[48] These approaches tend to be computationally expensive and the lack of extensive comparisons with the methods covered in this paper currently make it difficult to assess their quality.

CONCLUDING REMARKS

This paper first showed that cluster ensembles are beneficial in a wide variety of scenarios. It then provided a framework for understanding many of the approaches taken so far to design such ensembles.

Even though there seems to be many different algorithms for this problem, we showed that there are several commonalities among these approaches. Apart from the applications already mentioned in introduction, cluster ensembles have been utilized in detecting overlapping clusters,[49] clustering with categorical data,[16,50] for automatic malware categorization,[51] for clustering gene expression time series[52] and many more. The design domain, however, is still quite rich leaving space for more efficient heuristics as well as formulations that place additional domain constraints to yield consensus solutions that are useful and actionable in diverse applications.

NOTES

[a]In this paper, we shall not consider approaches where the feature values of the original data or of the cluster representatives are available to the consensus mechanism, e.g., Ref 5.
[b]This definition is also valid for overlapping clustering.

ACKNOWLEDGMENTS

We are thankful to Dr. Eduardo Raul Hruschka and the anonymous reviewers for their insightful comments and suggestions. This work was supported in part by NSF-IIS 1016614.

REFERENCES

1. Sharkey A. *Combining Artificial Neural Nets*. Secaucus, NJ, USA: Springer-Verlag, New York, Inc.; 1999.
2. Tumer K, Ghosh J. Robust order statistics based ensembles for distributed data mining. In: Hillol Kargupta H, Chan P, eds, *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press; 2000, 85–110.
3. Kuncheva LI. *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: John Wiley & Sons; 2004.
4. Ayad HG, Kamel MS. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Trans Pattern Anal Mach Intell* 2008, 30:160–173.
5. Hore P, Hall LO, Goldgof DB. A scalable framework for cluster ensembles. *Pattern Recognit* 2009, 42:676–688.
6. Fred A, Jain AK. Combining multiple clusterings using evidence accumulation. *IEEE Trans Pattern Anal Mach Intell* 2005, 27:835–850.
7. Strehl A, Ghosh J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J Mach Learn Res* 2002, 3:583–617.
8. Kuncheva LI, Hadjitodorov ST. Using diversity in cluster ensemble. *IEEE Int Conf Syst Man Cybern* 2004, 2:1214–1219.
9. Hu X, Yoo I. Cluster ensemble and its applications in gene expression analysis. In: *APBC '04: Proceedings of the second conference on Asia-Pacific bioinformatics*, Darlinghurst, Australia: Australian Computer Society, Inc; 2004.
10. Karypis G, Han E-H, Kumar V. Chameleon: hierarchical clustering using dynamic modeling. *IEEE Comput* 1999, 32:68–75.
11. Sevillano X, Cobo G, Alías F, Socoró JC. Feature diversity in cluster ensembles for robust document clustering. In: *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York: ACM; 2006, 697–698.
12. Ghosh J, Strehl A, Merugu S. A consensus framework for integrating distributed clusterings under limited knowledge sharing. In: *Proceedings of NSF Workshop on Next Generation Data Mining, Baltimore*; 2002, 99–108.
13. Asur S, Parthasarathy S, Ucar D. An ensemble framework for clustering protein-protein interaction networks. In: *Proceedings of 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB)*; 2007, 29–40.
14. Brodley CE. Random projection for high dimensional data clustering: a cluster ensemble approach. In: *Proceedings of 20th International Conference on Machine Learning (ICML'03)*, Washington, DC; 2003.
15. Merugu S, Ghosh J. A distributed learning framework for heterogeneous data sources. In: *Proc. KDD*; 2005, 208–217.

16. Mirkin B. *Mathematical Classification and Clustering*. Dordrecht: Kluwer; 1996.
17. Day WHE. Foreword: comparison and consensus of classifications. *J. Classi* 1986, 3:183–185.
18. Topchy A, Jain A, Punch W. A mixture model for clustering ensembles. In: *Proceedings of SIAM International Conference on Data Mining*; 2004, 379–390.
19. Wang H, Shan H, Banerjee A. Bayesian cluster ensembles. In: *Proceedings of the Ninth SIAM International Conference on Data Mining*; 2009, 211–222.
20. Hubert L, Arabie P. Comparing partitions. *J Classif* 1985, 2:193–218.
21. Meila M. Comparing clusterings by the variation of information. In: *Proceedings of Conference on Learning Theory*; 2003, 173–187.
22. Wu J, Chen J, Xiong H, Xie M. External validation measures for *k*-means clustering: a data distribution perspective. *Expert Syst Appl* 2009, 36:6050–6061.
23. Topchy AP, Law MHC, Jain AK, Fred AL. Analysis of consensus partition in cluster ensemble. In: *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, Washington, DC: IEEE Computer Society; 2004, 225–232.
24. Hilton P, Pedersen J, Stigter J. On partitions, surjections and stirling numbers. In: *Bull Belgian Math Soc* 1994, 1:713–725, 1994.
25. Topchy A, Jain AK, Punch W. Combining multiple weak clusterings. In: *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*; page 331, Washington, DC: IEEE Computer Society; 2003, 331
26. Bishop CM. *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer, New York, Inc.; 2006.
27. Wang P, Domeniconi C, Laskey K. Nonparametric bayesian clustering ensembles. In: *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science, Vol. 6323, Ch. 28. Berlin/Heidelberg: Springer; 2010.
28. Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J Sci Comput* 1998; 20:359–392.
29. Strehl A, Ghosh J. A scalable approach to balanced, high-dimensional clustering of market-baskets. In: *Proc. HiPC 2000, Bangalore*, LNCS, Vol. 1970. Springer; 2000.
30. Punera K, Ghosh J. Consensus based ensembles of soft clusterings. In: *Proc. MLMTA'07 – International Conference on Machine Learning: Models, Technologies & Applications*; 2007.
31. Al Razgan M, Domeniconi C. Weighted cluster ensemble. In: *Proceedings of SIAM International Conference on Data Mining*; 2006, 258–269.
32. Monti S, Tamayo P, Mesirov J, Golub T. Consensus clustering-a resampling-based method for class discovery and visualization of gene expression microarray data. *J Mach Learn* 2003, 52:91–118.
33. Nguyen N, Caruana R. Consensus clusterings. In: *Proceedings of International Conference on Data Mining*; 2007, 607–612.
34. Zhong S, Ghosh J. A unified framework for model-based clustering. *J Mach Learn Res* 2003, 4:1001–1037.
35. Li T, Ding C, Jordan M. Solving consensus and semi-supervised clustering problems using non-negative matrix factorization. In: *Proceedings of Eighth IEEE International Conference on Data Mining*; 2007, 577–582.
36. Swift S, Tucker A, Vinciotti V, Martin M, Orengo C, Liu X, Kellam P. Consensus clustering and functional interpretation of gene-expression data. *Genome Biol* 2004, 5:R94.
37. Wang F, Wang X, Li T. Generalized cluster aggregation. In: *IJCAI'09: Proceedings of the 21st International Jont Conference on Artifical Intelligence*, San Francisco, CA: Morgan Kaufmann Publishers Inc.; 2009, 1279–1284.
38. Banerjee A, Merugu S, Dhillon I, Ghosh J. Clustering with Bregman divergences. *J. Mach Learn Res* 2005, 6:1705–1749.
39. Li T, Ding C. Weighted consensus clustering. In: *Proceedings of Eighth SIAM International Conference on Data Mining*; 2008, 798–809.
40. Lee DD, Seung HS. Algorithms for non-negative matrix factorization. In: *NIPS*. Denver, CO, USA: MIT Press; 2000.
41. Goder A, Filkov V. Consensus clustering algorithms: Comparison and refinement. In: *Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments*; 2008, 109–117.
42. Gionis A, Mannila H, Tsaparas P. Clustering aggregation. *ACM Trans Knowl Discov Data* 2007, 1:109–117.
43. Bansal N, Blum AL, Chawla S. Correlation clustering. In: *Proceedings of Foundations of Computer Science*; 2002, 238–247.
44. Karypis G, Aggarwal R, Kumar V, Shekhar S. Multilevel hypergraph partitioning: applications in VLSI domain. In: *Proceedings of the Design and Automation Conference*; 1997, 526–529.
45. Fern X, Brodley C. Solving cluster ensemble problems by bipartite graph partitioning. In: *Proceedings of International Conference on Machine Learning*; 2004, 281–288.
46. Dudoit S, Fridlyand J. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* 2003, 19:1090–1099.
47. Yoon SY Ahn, SH Lee, SB Cho, JH Kim. Heterogeneous clustering ensemble method for combining

different cluster results. In: *Proceedings of BioDM 2006*, Lecture Notes in Computer Science, Vol. 3916; 2006, 82–92.

48. Yang Y, Kamel MS. An aggregated clustering approach using multi-ant colonies algorithms. *Pattern Recognit* 2006, 39:1278–1289.
49. Deodhar M, Ghosh J. Consensus clustering for detection of overlapping clusters in microarray data. In: *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining Workshops*, Washington, DC: IEEE Computer Society; 2006, 104–108.
50. He Z, Xu X, Deng S. A cluster ensemble method for clustering categorical data. *Inform Fusion* 2005, 6:143–151.
51. Ye Y, Li T, Chen Y, Jiang Q. Automatic malware categorization using cluster ensemble. In: *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York: ACM; 2010, 95–104.
52. Chiu T-Y, Hsu T-C, Wang J-S. Ap-based consensus clustering for gene expression time series. In: *International Conference on Pattern Recognition*; 2010, 2512–2515.

