Cluster Ensembles

Joydeep Ghosh[1], Ayan Acharya[2]
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin Texas 78712
✉[1]ghosh@ece.utexas.edu, [2]masterayan@gmail.com

Abstract

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 article 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.

1 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 [37, 44, 25]. However, combining multiple clusterings poses additional challenges. First, the number of clusters produced may differ across the different *base* solutions [3]. 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,[1] 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 [13], while in others even this is not granted in order to allow applications involving knowledge reuse [38], 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 [26, 20].

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 [22]. 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,

[1] 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. [19]

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 [38]. Authors in [36] present several empirical results on the robustness of the results in document clustering by using feature diversity and consensus clustering.

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 [14].

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 [38], 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. **Multi-view Clustering**

   Often the objects to be clustered have multiple aspects or "views", and base clusterings may be built on distinct views that involve non-identical 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 [2]. Consensus clustering can be effectively used to combine all such clusterings into a single consolidated partition. Strehl & Ghosh [38] illustrated empirically the utility of cluster ensembles in two orthogonal scenarios,

   - Feature Distributed Clustering (FDC): different base clusterings are built by selecting different subsets of the features but utilizing all the data points.
   - Object Distributed Clustering (ODC): base clusterings are constructed by selecting different subsets of the data points but utilizing all the features.

   Fern & Brodley [12] also showed that clustering in high dimension is much more effective compared to clustering with 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 [31]. An ensemble can be used in situations where each clusterer has access to only a subset of the features of each object, as well as where each clusterer has access to only a subset of the objects [14], [38].

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 70's and 80's. In this literature, "classification" was used in a broad sense to include clusterings, unrooted trees, graphs, etc, and problem-specific formulations were made (see [32] 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 [8], 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 [3]. A reasonable coverage of this broader class of problems is not feasible here, instead this article focuses on the cluster ensemble formulations and associated algorithms that have been proposed in the past decade. Section 2 formally defines the cluster ensemble problem within this context. Section 3 presents a variety of approaches to designing cluster ensembles, organized in three different categories.

2 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 q^{th} clustering containing $k^{(q)}$ clusters. The most straightforward representation of the q^{th} clustering is $\lambda^{(q)} = \{\mathcal{C}_\ell | \ell = 1, 2, ..., k^{(q)} \text{ and } \mathcal{C}_\ell \subseteq \mathcal{X}\}$. 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 q^{th} 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 individual clustering is by the binary membership indicator matrix $\mathbf{H}^q \in \{0,1\}^{1 \times k^{(q)}}$ which is defined as $\mathbf{H}^q = \{h_{i\ell}^q | h_{i\ell}^q \in \{0,1\} \ \forall \mathbf{x}_i, \mathcal{C}_\ell, \lambda^{(q)}\}$. For partitional clustering, we additionally have $\sum_{\ell=1}^{k^{(q)}} h_{i\ell}^q = 1 \ \forall \mathbf{x}_i \in \mathcal{X}$.

A *consensus function* Γ is defined as a function $\mathbb{Z}_+^{n \times r} \rightarrow \mathbb{Z}_{++}^n$ mapping a set of clusterings to an integrated clustering $\Gamma : \lambda^{(q)} | q \in \{1, 2, ..., r\} \rightarrow \hat{\lambda}$. 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 [2] of n objects can be represented as a binary, symmetric $n \times n$ *co-association matrix*, with an entry being 1 if the corresponding objects are in the same cluster and 0 otherwise. For the q^{th} base clustering, this matrix is denoted by $S^{(q)}$ and is given by

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

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 labellings of the individual solutions, given the true consensus labels, and then solve a maximum likelihood formulation to return the consensus [41, 46]. 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: i) *Adjusted Rand Index*, ii) *Normalized Mutual Information* and iii) *Variation of Information*. We now briefly introduce these measures as they are used in several algorithms covered in Section 3.

1. **Adjusted Rand Index (ARI)**

   Suppose we have two candidate clusterings $\lambda^{(a)} = \{\mathcal{C}_h^{(a)} | h = 1, 2, ..., k^{(a)}\}$ and $\lambda^{(b)} = \{\mathcal{C}_\ell{}^{(b)} | \ell = 1, 2, ..., k^{(b)}\}$. 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 & Arabie [21], is defined as,

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

   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.

2. **Normalized Mutual Information (NMI)**

   Strehl & Ghosh [38] 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

[2] This definition is also valid for overlapping clustering.

Table 1: Contingency Table Explaining Adjusted Rand Index

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

$H(\lambda^{(b)}) = -\sum_\ell \frac{n_\ell^{(b)}}{n} log(\frac{n_\ell^{(b)}}{n})$. 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:

$$\phi^{(NMI)}(\lambda^{(a)}, \lambda^{(b)}) = \frac{H(\lambda^{(a)}) + H(\lambda^{(b)}) - H(\lambda^{(a)}, \lambda^{(b)})}{\sqrt{H(\lambda^{(a)})H(\lambda^{(b)})}} = \frac{I(\lambda^{(a)}, \lambda^{(b)})}{\sqrt{H(\lambda^{(a)})H(\lambda^{(b)})}} \tag{3}$$

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.

3. **Variation of Information (VI)**

   VI is another information theoretic measure proposed for cluster validation [30], 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* [48] proposed a normalized version of VI (NVI) which is empirically shown to alleviate this inconsistency. NVI is defined as follows:

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

   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 $\phi^{(ARI)}$, $\phi^{(NMI)}$ and $\phi^{(NVI)}$. 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 intractable, so heuristic approaches have to be resorted to.

Topchy *et al* [43] 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 non-empty clusters. This number, often denoted by $\mathcal{S}_n^{(k)}$, is in fact called Stirling number of second kind [18]. 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 [43]. 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 equation (8) decreases exponentially with number of clusterings in the ensemble. When r approaches infinity, $P(\hat{\lambda} \neq \lambda^0)$ approaches zero. In [42], the authors, however, viewed cluster ensemble as a *median clustering* problem and used generalized entropy to modify the NMI measure explained earlier.

3 Cluster Ensemble Algorithms

Cluster ensemble methods are now presented under three categories: i) probabilistic approaches, ii) approaches based on co-association and iii) direct and other heuristic methods.

3.1 Probabilistic Approaches to Cluster Ensembles

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

3.1.1 A Mixture Model for Cluster Ensembles

In a typical mixture model [6] 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 [41]. We describe it in a bit more detail as this work is essential to build an understanding of later works [46, 47].

In the basic mixture model of cluster ensembles [41], 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 sample from this distribution is a cluster label corresponding to the q^{th} base clustering. The underlying generative process is assumed as follows:

For i^{th} data point $\mathbf{x}_i$,

1. Choose $\mathbf{z}_i = \mathbf{I}_{\hat{\ell}}$ such that $\hat{\ell} \sim \text{multinomial}(\boldsymbol{\theta})$. Here $\mathbf{I}_{\hat{\ell}}$ is a probability vector of dimension $k^{(q)}$ with only the $\hat{\ell}^{\text{th}}$ component being 1, and $\boldsymbol{\theta}$ is a multinomial distribution of dimension $\hat{k}$.

2. For the q^{th} base clustering of i^{th} data point, choose the base clustering result $y_{iq} = \ell \sim \text{multinomial}(\boldsymbol{\beta}_{\hat{\ell}}^{(q)})$.

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.

3.1.2 Bayesian Cluster Ensembles

A Bayesian version of the multinomial mixture model described above was subsequently proposed by Wang *et al* [46]. 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:

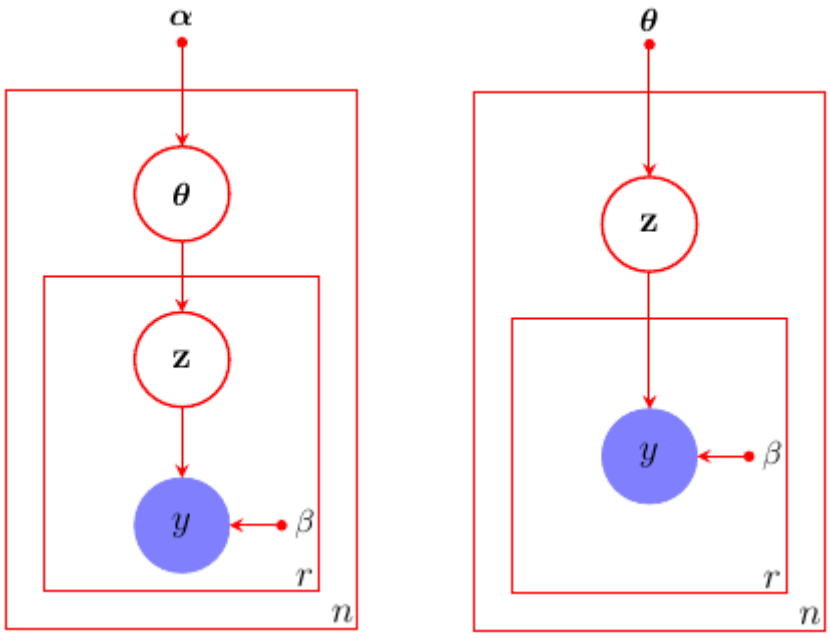
For i^{th} data point $\mathbf{x}_i$,

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

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,

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

where $\exists y_{iq}$ implies that there exists a q^{th} base clustering result for $\mathbf{y}_i$. 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 EM and Gibb's 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 non-parametric version of Bayesian cluster ensemble has been proposed in [47] 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.



(a) Bayesian Cluster Ensembles (b) Mixture Model of Cluster Ensembles

Figure 1: Graphical Models for Probabilistic Approaches to Cluster Ensembles

3.2 Pairwise Similarity Based Approaches

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

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

Here w_q specifies the weight assigned to the q^{th} base clustering. This ensemble co-association 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 section 3.2.1.

Note that the co-association 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.

3.2.1 Methods based on Ensemble Co-association Matrix

The Cluster-based Similarity Partitioning Algorithm (CSPA) [38] used METIS [23] to partition the induced consensus similarity graph. METIS was chosen for its scalability and because it tries to enforce comparable sized clusters. This added constraint is desirable in several application domains [39]; however, if the data is actually labeled with imbalanced classes, then it can lower the match between cluster and class labels. Assuming quasi-linear graph clustering, the worst case complexity for this algorithm is $\mathcal{O}(n^2kr)$. Punera & Ghosh [35] later proposed a soft version of CSPA, i.e. one that works on soft base clusterings. Al-Razgan & Domeniconi [1] proposed an alternative way of obtaining non-binary co-association matrices when given access to the raw data.

The Evidence Accumulation approach [13] obtains individual co-association 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 co-association 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 co-association matrix. This allows one to obtain non-convex 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* [33], where the perturbations in the base clustering were achieved by re-sampling. Any of bootstrapping, data sub-sampling or feature sub-sampling can be used as a re-sampling 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 co-association matrices, the $(i,j)^{\text{th}}$ 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 co-association matrix is formed multiple times. The number of clusters is kept at k_i for each base clustering during the i^{th} 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 towards 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 co-association 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 [34] essentially applies model-based k-means [52] to the ensemble co-association matrix S. The consensus clustering solution $\hat{\lambda} = \{\mathcal{C}_\ell\}_{\ell=1}^{\hat{k}}$ is initialized to some solution, after which a re-assignment 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 [32] and Li *et al* [29] 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,

$$\arg\min_{\hat{\lambda}} J = \arg\min_{\hat{\lambda}} \frac{1}{r}\sum_{q=1}^{r} d(\lambda^{(q)}, \hat{\lambda}) = \arg\min_{\hat{S}} \frac{1}{r}\sum_{q=1}^{r} w_q \sum_{i<j} [S_{ij}^{(q)} - \hat{S}_{ij}]^2 \tag{11}$$

Mirkin ([32], section 5.3.4, p. 260) further proved that the consensus clustering according to criterion (11) is equivalent to clustering over the ensemble co-association 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 [40] for consensus clustering of gene-expression data.

In [45], consensus clustering result is obtained by minimizing a weighted sum of the Bregman divergence [4] between the consensus partition and the input partitions wrt their co-association 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 (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 section 3.3), there is no guarantee of achieving the optimum value $\hat{S} = S$. However, if we are optimizing over the domain of the co-association matrix we can achieve this optimum value in theory.

3.2.2 Relating Consensus Clustering to other Optimization Formulations

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

1. **Consensus Clustering as Non-Negative Matrix Factorization (NNMF)**

   Li *et al* ([29], [28]), using the same objective function as mentioned in (11), showed that the problem of consensus clustering can be reduced to an NNMF problem. Assuming $U_{ij} = \hat{S}_{ij}$ to be a solution to this optimization problem. we can rewrite (11) as,

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

   where the matrix norm is the Frobenius norm. This problem formulation is similar to the NNMF formulation [27] and can be solved using an iterative update procedure. In [16], the cost function J used in equation (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).

2. **Consensus Clustering as Correlation Clustering**

   Gionis *et al* [15] 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 [5] is to find a partition $\hat{\lambda}$ such that

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

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

3.3 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 (6), without computing the co-association matrix.

3.3.1 Graph Partitioning

In addition to CSPA, Strehl & Ghosh [38] 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* [24]) 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). Since 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 [11]. 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 multi-way 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 & Ghosh [35]. It should be noted that all of CSPA, MCLA and HGPA were compared with one other using the NMI measure in [38].

3.3.2 Cumulative Voting

The concept of cumulative voting was first introduced in [10] 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 re-labeling 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 [3]. 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 [3] 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 co-association 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 [4], 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 [51] and one using a multi-ant colony [49]. These approaches tend to be computationally expensive and the lack of extensive comparisons with the methods covered in this article currently make it difficult to assess their quality.

4 Concluding Remarks

This article 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 [9], clustering with categorical data [17], for automatic malware categorization [50], for clustering gene expression time series [7] 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.

Acknowledgement

This work has been supported by NHARP, NSF Grants (IIS-0713142 and IIS-1016614) and by the Brazilian Research Agencies FAPESP and CNPq. We are thankful to Dr. Eduardo Raul Hruschka and the anonymous reviewers for their insightful comments and suggestions.

References

[1] M. Al-Razgan and C. Domeniconi. Weighted cluster ensemble. In *Proceedings of SIAM International Conference on Data Mining*, pages 258–269, 2006.

[2] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An ensemble framework for clustering protein-protein interaction networks. In *In Proc. 15th Annual Intl Conference on Intelligent Systems for Molecular Biology (ISMB*, page 2007, 2007.

[3] Hanan G. Ayad and Mohamed S. Kamel. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):160–173, 2008.

[4] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Jl. Machine Learning Research (JMLR)*, 6:1705–1749, October 2005.

[5] N. Bansal, A.L. Blum, and S. Chawla. Correlation clustering. In *Proceedings of Foundations of Computer Science*, page 238247, 2002.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Tai-Yu Chiu, Ting-Chieh Hsu, and Jia-Shung Wang. Ap-based consensus clustering for gene expression time series. *Pattern Recognition, International Conference on*, 0:2512–2515, 2010.

[8] W.H.E. Day. Foreword: Comparison and consensus of classifications. *J. Classification*, 3:183–185, 1986.

[9] Meghana Deodhar and Joydeep Ghosh. Consensus clustering for detection of overlapping clusters in microarray data. In *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 104–108, Washington, DC, USA, 2006. IEEE Computer Society.

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

[11] X. Fern and C. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of International Conference on Machine Learning*, pages 281–288, 2004.

[12] Fern, Xiaoli Z. and Brodley, Carla E. Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. In *Proc. 20th International Conference on Machine Learning (ICML'03)*, Washington, August 2003.

[13] A. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

[14] J. Ghosh, A. Strehl, and S. Merugu. A consensus framework for integrating distributed clusterings under limited knowledge sharing. In *Proc. NSF Workshop on Next Generation Data Mining, Baltimore*, pages 99–108, Nov 2002.

[15] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data*, 1(4):109–117, March 2007.

[16] A. Goder and V. Filkov. Consensus clustering algorithms: Comparison and refinement. In *Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments*, pages 109–117, 2008.

[17] Zengyou He, Xiaofei Xu, and Shengchun Deng. A cluster ensemble method for clustering categorical data. *Information Fusion*, 6(2):143 – 151, 2005.

[18] Peter Hilton, Jean Pedersen, and Jurgen Stigter. On partitions, surjections and stirling numbers. In *Bulletin of the Belgian Mathematical Society 1*, pages 713 – 725, 1994.

[19] P. Hore, Lawrence O. Hall, and Dmitry B. Goldgof. A scalable framework for cluster ensembles. *Pattern Recogn.*, 42(5):676–688, 2009.

[20] Xiaohua Hu and Illhoi Yoo. Cluster ensemble and its applications in gene expression analysis. In *APBC '04: Proceedings of the second conference on Asia-Pacific bioinformatics*, pages 297–302, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[21] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[22] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32(8):68–75, August 1999.

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

[24] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. Multilevel hypergraph partitioning: Applications in VLSI domain. In *Proceedings of the Design and Automation Conference*, pages 526–529, 1997.

[25] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, Hoboken, NJ, 2004.

[26] L. I. Kuncheva and S. T. Hadjitodorov. Using diversity in cluster ensemble. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1214–1219, 2004.

[27] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *In NIPS*, pages 556–562. MIT Press, 2000.

[28] T. Li and C. Ding. Weighted consensus clustering. In *Proceedings of Eighth SIAM International Conference on Data Mining*, pages 798–809, 2008.

[29] T. Li, C. Ding, and M. Jordan. Solving consensus and semi-supervised clustering problems using non-negative matrix factorization. In *Proceedings of Eighth IEEE International Conference on Data Mining*, pages 577–582, 2007.

[30] M. Meila. Comparing clusterings by the variation of information. In *Proceedings of Conference on Learning Theory*, pages 173–187, 2003.

[31] S. Merugu and J. Ghosh. A distributed learning framework for heterogeneous data sources. In *Proc. KDD*, pages 208–217, 2005.

[32] B. Mirkin. *Mathematical Classification and Clustering*. Kluwer, 1996.

[33] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering-a resampling-based method for class discovery and visualization of gene expression microarray data. In *Journal of Machine Learning*, pages 52: 91–118, 2003.

[34] N. Nguyen and R. Caruana. Consensus clusterings. In *Proceedings of International Conference on Data Mining*, pages 607–612, 2007.

[35] K. Punera and J. Ghosh. Consensus based ensembles of soft clusterings. In *Proc. MLMTA'07 - Int'l Conf. on Machine Learning: Models, Technologies & Applications*, 2007.

[36] Xavier Sevillano, Germán Cobo, Francesc Alías, and Joan Claudi Socoró. 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*, pages 697–698, New York, NY, USA, 2006. ACM.

[37] A. Sharkey. *Combining Artificial Neural Nets*. Springer-Verlag, 1999.

[38] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Jl. Machine Learning Research (JMLR)*, 3 (Dec):583–617, 2002.

[39] Alexander Strehl and Joydeep Ghosh. A scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proc. HiPC 2000, Bangalore*, volume 1970 of *LNCS*, pages 525–536. Springer, December 2000.

[40] Stephen Swift, Allan Tucker, Veronica Vinciotti, Nigel Martin, Christine Orengo, Xiaohui Liu, and Paul Kellam. Consensus clustering and functional interpretation of gene-expression data. In *Genome Biology;5(11):R94*, 2004.

[41] A. Topchy, A. Jain, and W. Punch. A mixture model for clustering ensembles. In *Proceedings of SIAM International Conference on Data Mining*, pages 379–390, 2004.

[42] Alexander Topchy, Anil K. Jain, and William Punch. Combining multiple weak clusterings. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 331, Washington, DC, USA, 2003. IEEE Computer Society.

[43] Alexander P. Topchy, Martin H. C. Law, Anil K. Jain, and Ana L. Fred. Analysis of consensus partition in cluster ensemble. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 225–232, Washington, DC, USA, 2004. IEEE Computer Society.

[44] K. Tumer and J. Ghosh. Robust order statistics based ensembles for distributed data mining. In Hillol Kargupta and Philip Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*, pages 85–110. AAAI Press, 2000.

[45] Fei Wang, Xin Wang, and Tao Li. Generalized cluster aggregation. In *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1279–1284, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[46] H. Wang, H. Shan, and A. Banerjee. Bayesian cluster ensembles. In *Proceedings of the Ninth SIAM International Conference on Data Mining*, pages 211–222, 2009.

[47] Pu Wang, Carlotta Domeniconi, and Kathryn Laskey. Nonparametric bayesian clustering ensembles. In *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, chapter 28, pages 435–450. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2010.

[48] Junjie Wu, Jian Chen, Hui Xiong, and Ming Xie. External validation measures for k-means clustering: A data distribution perspective. *Expert Syst. Appl.*, 36(3):6050–6061, 2009.

[49] Y. Yang and M.S. Kamel. An aggregated clustering approach using multi-ant colonies algorithms. 39:109–117, July 2006.

[50] Yanfang Ye, Tao Li, Yong Chen, and Qingshan Jiang. Automatic malware categorization using cluster ensemble. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2010. ACM.

[51] H.S. Yoon, S.Y. Ahn, S.H. Lee, S.B. Cho, and J.H. Kim. Heterogeneous clustering ensemble method for combining different cluster results. In *Proceedings of BioDM 2006, Lecture Notes in Computer Science*, volume 3916, pages 82–92, 2006.

[52] S. Zhong and J. Ghosh. A unified framework for model-based clustering. *Jl. Machine Learning Research (JMLR)*, 4:1001–1037, 2003.