To appear in ICDM 2004

Analysis of Consensus Partition in Cluster Ensemble

Alexander P. Topchy Martin H. C. Law Anil K. Jain
Dept. of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824, USA
{topchyal, lawhiu, jain}@cse.msu.edu

Ana L. Fred
Dept. of Electr. and Comp. Engineering
Instituto Superior Tecnico
1049-001, Lisbon, Portugal
afred@lx.it.pt

Abstract

In combination of multiple partitions, one is usually interested in deriving a consensus solution with a quality better than that of given partitions. Several recent studies have empirically demonstrated improved accuracy of clustering ensembles on a number of artificial and real-world data sets. Unlike certain multiple supervised classifier systems, convergence properties of unsupervised clustering ensembles remain unknown for conventional combination schemes. In this paper we present formal arguments on the effectiveness of cluster ensemble from two perspectives. The first is based on a stochastic partition generation model related to re-labeling and consensus function with plurality voting. The second is to study the property of the "mean" partition of an ensemble with respect to a metric on the space of all possible partitions. In both the cases, the consensus solution can be shown to converge to a true underlying clustering solution as the number of partitions in the ensemble increases. This paper provides a rigorous justification for the use of cluster ensemble.

1 Introduction

Recent research on data clustering is increasingly focusing on combining multiple data partitions as a way to improve the robustness of clustering solutions. It has been shown that a meaningful consensus of multiple clusterings is possible by using a consensus function that maps a given ensemble (a collection of different partitions of a data set) to a combined clustering result. Several efficient consensus functions have been derived from statistical, graph-based and information-theoretic principles. A variety of known consensus functions are based on co-association matrix [5, 6, 7], hypergraph cuts [13, 9], mutual information [15], mixture models [14] and voting [3, 4]. Extensive experiments with these functions indicate that a combination of clusterings is capable of detecting novel cluster structures. Empirical evidence also supports the idea that requirements for individual clustering algorithms can be significantly relaxed in favor of weaker and inexpensive partition generation.

With all this progress, we are still lacking a rigorous understanding of why clustering ensembles can converge to a better consensus solution as compared with individual components of an ensemble. Basic properties of the consensus solutions have not been rigorously analyzed for existing consensus functions. The challenge of explaining the cluster ensembles is two-fold. First, a consensus of unsupervised classifications does not conform to the rules established for multiple classifier systems due to the invariance of clustering to class label permutations. All the partitions which differ only in cluster labeling are identical. Second, analysis of consensus solutions must include both consensus functions as well as assumptions about the ensemble generation mechanism. Indeed, clustering ensembles merely represent a more sophisticated class of clustering algorithms utilizing a two-step process – ensemble generation and search for consensus.

Several difficulties must be resolved in providing the "proof of consensus". Primarily, we need a good probabilistic model of individual unsupervised classifications as the components of ensemble. Many studies use the k-means algorithm and its randomness in choosing the initial cluster centers to generate diverse components. Unfortunately, at present we are unable to formally characterize the distribution of the k-means partitions of a data set produced from random initializations. In general, it is very difficult to make any analytic statement about a partition generated by a clustering algorithm. Moreover, some consensus functions, which act on the samples from such distributions of partitions, are either heuristic in nature or have no explicit objective function. For instance, the consensus results of hypergraph partitioning or agglomerative clustering algorithms for co-association matrices are difficult to predict.

Research supported by ONR grant no. N00014-04-1-0183.

To circumvent these difficulties, we have made several simplifying assumptions. As in the case of analyzing classifier combinations in supervised learning, we model the output of a clustering algorithm without referring to any property of the algorithm. Rather, the partition generated by an algorithm is interpreted as a noisy version of the ground-truth partition of the data set. Two approaches of analysis are considered. The first is based on the assumption that the unsupervised classifications of data are produced in two steps: each partition is a noisy version of the true partition, where all the cluster labels undergo a random permutation. The goal of consensus function is to discover the true underlying partition. As we explain below, voting with re-labeling can detect the true partition with probability approaching to 1 as the size of ensemble increases. Our second approach is based on the fact that a consensus function can be regarded as finding a "mean" partition of different partitions in a cluster ensemble, with respect to a metric defined on a space of partitions. By using results of large deviation theory, we can show that the chance of failing to discover the true partition drops exponentially with increasing number of partitions in the ensemble.

The notation used in this paper is as follows. A data set which we want to cluster into k clusters is represented by $\mathcal{X} = \{x_1, \ldots, x_n\}$, with $|\mathcal{X}| = n$. The set of all possible partitions of $\mathcal{X}$ into k clusters is denoted by $\mathbb{P}$, with $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$, where each P_i represents a partition of $\mathcal{X}$ into k clusters. The cardinality of $\mathbb{P}$, also known as the Stirling number of the second kind, is denoted by m. The (unknown) "ground-truth" partition of $\mathcal{X}$ is denoted by C. The cluster ensemble of N partitions is denoted by $D_N = (C_1, \ldots, C_N)$, where C_i represents a random partition of $\mathcal{X}$ that follows the probability measure μ, i.e., $P(C_i = P_j) = \mu(P_j)$, and $\sum_j \mu(P_j) = 1$.

2 Consensus based on voting

In this section, we analyze consensus solution obtained by plurality voting. We first define the probabilistic model for generating partitions that is based on mis-labeling and label permutation in section 2.1. Section 2.2 discusses the process of re-labeling, and section 2.3 shows that plurality voting used in the consensus function can indeed recover the ground truth partition, even with an imperfect collection of partitions.

2.1 Stochastic partition generation model

The true partition C of the data set $\mathcal{X}$ can be written as $C = \{C(x_1), C(x_2), \ldots, C(x_n)\}$, where $C(x_j) = l$, if the object x_j belongs to the l-th cluster, $l \in \{1, \ldots, k\}$. The cluster labels by themselves are irrelevant and simply used to specify the partition. Let the first n_1 objects belong to the cluster 1, next n_2 objects belong to the cluster 2, etc., such that $n_1 + n_2 + \cdots + n_k = n$. Each observed partition C_i in the ensemble D_s is generated by two transformations of the true partition C: noise $C' = F(C)$ and label permutation $C^* = T(C')$.

First, a random noise with probability $(1-p)$ is applied to a cluster label $C(x_j)$ of each object x_j, $j = 1, 2, \ldots, n$. The value $C(x_j)$ is replaced by a new random label l from $\{1, \ldots, k\}$ with equal probability q, for all values $l \neq C(x_j)$. Hence, an object keeps a correct label $C(x_j)$ with probability p, and acquires an incorrect label with probability $(1-p)$. We assume that all the incorrect labels are equally probable:

$$q = \frac{1-p}{k-1}.$$

We say that the first step generates a noisy version $C'(X)$ of the true partition C:

$$C' = \{C'(x_1), C'(x_2), \ldots, C'(x_n)\}.$$

The second step performs a random permutation of the labels in a noisy partition C'. The label permutation $T = \{\sigma(1), \sigma(2), \ldots, \sigma(k)\}$ is drawn from a set of all possible permutations of k labels with uniform probability. The partition $C^*(X) = T(C(X))$ becomes a member of an ensemble:

$$C^* = \{C^*(x_1), C^*(x_2), \ldots, C^*(x_n)\},$$

and C_1, the first partition in the cluster ensemble, takes this value of C^*. The above process is repeated with different realizations of $F(.)$ and $T(.)$ to generate other partitions C_i in the ensemble. The observed ensemble D_N is just the collection of N random partitions:

$$D_N = \{C_1, C_2, \ldots, C_N\}.$$

The label permutation, which is absent in supervised classifier combination, is a major difficulty in deriving a consensus solution from multiple clusterings. One can note that this ensemble generation procedure can be also described as a sampling of object's labels from a finite mixture of multivariate multinomial components. A mixture model admits maximum likelihood solution for consensus clustering, but is difficult to analyze in terms of convergence. That is why we proceed with the voting-type consensus function, where explicit parallels with the supervised case of multiple experts are possible. Figure 1 illustrates ensemble generation with 4 partitions of 7 objects.

2.2 Voting consensus function

A consensus function maps a given set of partitions in the ensemble to a single consensus partition. Voting procedure can be used to find a target partition if (i) all the

true
1
1
2
2
2
3
3

	Noisy partitions			
X1	1	1	1	2
X2	1	1	2	1
X3	3	2	2	2
X4	2	3	1	2
X5	2	2	2	1
X6	3	3	3	3
X7	3	2	2	1

	relabeled partitions			
X1	2	3	1	3
X2	2	3	3	2
X3	1	2	3	3
X4	3	1	1	3
X5	3	2	3	2
X6	1	1	2	1
X7	1	2	3	2

Figure 1. An example of ensemble generation.

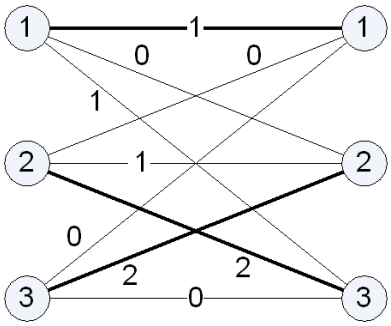

Figure 2. The bipartite graph for two partitions C_r and C_t.

ensemble's partitions use exactly the same set of cluster labels and (ii) clusters in different partitions are *consistently* labeled. A notion of consistent labeling can be made precise using our assumption about partition generation process: the best possible labeling of clusters in the ensemble should minimize the number of incorrectly labeled objects in comparison with the true partition. In order to achieve the most consistent labeling of clusters in a partition, we must solve an assignment problem equivalent to maximum weight bipartite matching problem. Equivalent matching problem is constructed from a contingency table between two partitions. A contingency table contains a number of cluster label co-occurrences counted for two partitions of the same set of objects. For example, for two partitions $C_r = \{1, 1, 2, 2, 2, 3, 3\}$ and $C_t = \{3, 1, 3, 3, 2, 2, 2\}$, we find the following contingency table

		C_t		
		1	2	3
C_r	1	1	0	1
	2	0	1	2
	3	0	2	0

and the equivalent weighted bipartite graph (figure 2). The minimum number of misassigned objects in partition C_t with respect to C_r is achieved by the re-labeling: $1 \leftrightarrow 1$, $2 \leftrightarrow 3$, $3 \leftrightarrow 2$ (shown by bold edges in the graph). Hence, the most consistent re-labeling would return $C_t = \{2, 1, 2, 2, 3, 3, 3\}$. In general, minimization of clustering error with respect to the true partition is equivalent to maximization of the weight of complete bipartite matching:

$$\max_{\{y_{ij}\}} \sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} y_{ij}$$
$$\text{Subject to } \sum_{j=1}^{k} y_{ij} = \sum_{i=1}^{k} y_{ij} = 1, y_{ij} \in \{0, 1\}, \quad (1)$$

where $\{w_{ij}\}$ are the values in the contingency table, and $\{y_{ij}\}$ are indicator variables which determine the correspondence between the clusters in the two partitions. An optimal solution of the problem (1) can be found by Hungarian algorithm [12] with the complexity $O(k^3)$.

A consistent re-labeling of all the partitions in the ensemble can be obtained by using a single common reference partition C_r. Ideally, the true partition is the best choice for a reference partition C_r, which, of course, is unavailable to us. In practice, any partition from the ensemble can be chosen as a reference partition. Then, all the remaining components of the ensemble can be relabeled by solving the problem in Equation (1) for every pair of partitions $(C_r, C_i), i = 1, ..., N, i \neq r$. Once all the partitions are re-labeled, plurality voting can be used to determine a consensus label for each object. *Plurality* voting decides in favor of a label, which is most frequently selected by individual experts in the ensemble for the given object. Unlike majority voting, more than half of the votes are not required for plurality consensus. Clearly, the accuracy of the consensus solution depends on the accuracy of: (i) decisions made by individual experts (clusterings), and (ii) their correct relabeling through a solution of the matching problem. Both these procedures are analyzed in detail now.

2.3 Supervised plurality consensus

Suppose that a re-labeling of an ensemble has been done. We make this assumption before considering the accuracy of such a re-labeling in the next section. Given this, one has to deal with the ensemble of experts (partitions in the ensemble) where each expert is accurate with probability p^*. Note that the probability p^* that an object has a correct true label is generally different than the probability p, that the object label was not changed by noise $F(.)$, because of additional label permutation $T(.)$. Our first goal is to demonstrate that the accuracy of consensus solution, p_c, improves with the increasing ensemble size if each expert performs better than random. Specifically, we expect that ensemble accuracy p_c for k-class problem using the plurality voting

combination rule satisfies:

$$\lim_{q\to\infty} p_c(p^*, q) = 1, \text{ for } p^* > \frac{1}{k}\,. \tag{2}$$

Note that p_c is a function of p^* and q because the performance of the ensemble, in general, depends on the noise level. The correctness of Equation (2) is commonly assumed in the literature for multiple classifier systems, yet its proof was given only recently for $k = 3$ and for general k with "best" non-independent classifiers [2]. Here, we would like to provide a simple proof of this convergence property for independent classifiers. Further, it will be utilized for unsupervised case as well.

We will assume that all the individual experts are independent. We shall focus on the label of a particular object x_l. Each decision on the object's class (label) is correct with probability h. All the incorrect decisions are equally probable for any given object. Let Z_1 be the number of votes in favor of the correct class and Z_i be the number of votes for an incorrect class i, $2 \le i \le k$. Given N independent decisions, the joint probability of random variables $Z_1, Z_2, \ldots, Z_k$ is a multinomial:

$$\begin{aligned} &P(Z_1 = N_1, Z_2 = N_2, \ldots, Z_K = N_k) \\ &= \frac{N!}{N_1!N_2!\ldots N_k!} h^{N_1} g^{N_2} \ldots g^{N_k} \end{aligned} \tag{3}$$

Here, $g = (1-h)/(k-1)$ is probability of each possible incorrect class, and $N = N_1+N_2+\ldots+N_k$. The probability, p_c, of correct classification by N experts using plurality voting is:

$$\begin{aligned} p_C &= P(Z_1 > Z_2, Z_1 > Z_3, \ldots, Z_1 > Z_k) \\ &= 1 - P(Z_1 \le Z_2 \text{ or } Z_1 \le Z_3 \text{ or } \ldots \text{ or } Z_1 \le Z_k) \\ &\ge 1 - \sum_{i=2}^{k} P(Z_1 \le Z_i) \end{aligned} \tag{4}$$

Consider any term $P(Z_1 \le Z_i)$ in the sum in Equation (4). We intend to show that for all $i > 1$, as $P(Z_1 \le Z_i) \to 0$ as $N \to \infty$, that would also imply $p_c \to 1$ as stated in Equation (2). Probability $P(Z_1 \le Z_i)$ can be rewritten as $P(Y_i \le 0)$, with $Y_i = (Z_1 - Z_i)/N$. Expected value and variance of Y_i can be obtained from multinomial distribution in Equation (3):

$$\begin{aligned} E[Y_i] &= \frac{Nh - Ng}{N} = h - g, \\ Var[Y_i] &= \frac{1}{N^2}\begin{bmatrix}1 & -1\end{bmatrix} Cov(\begin{bmatrix}X_1 \\ X_i\end{bmatrix}) \begin{bmatrix}1 \\ -1\end{bmatrix}, \\ \text{where } Cov(\begin{bmatrix}X_1 \\ X_i\end{bmatrix}) &= N \begin{bmatrix}h(1-h) & -hg \\ -hg & g(1-g)\end{bmatrix} \end{aligned} \tag{5}$$

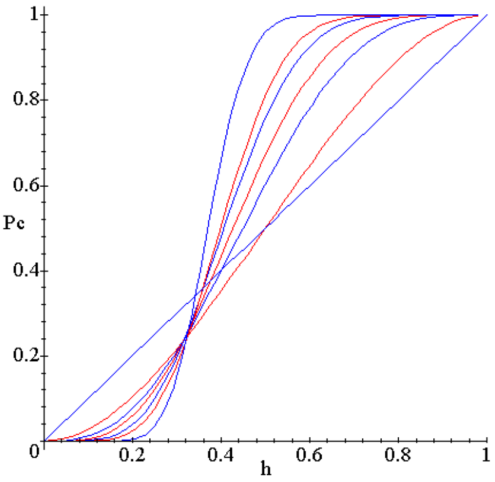

Figure 3. Accuracy of plurality voting

Since each expert is better than random, $h \ge 1/k$, we find that $E[Y_i] = h - g > 0$. Furthermore, Equation (5) implies $\lim_{N\to\infty} Var[Y_i] = 0$, which leads to $\lim_{N\to\infty} P(Y_i \le 0) = 0$. As a result, perfect accuracy is asymptotically achieved:

$$\lim_{N\to\infty} p_c = \lim_{N\to\infty} P(Z_1 > Z_2, Z_1 > Z_3, \ldots, Z_1 > Z_K) = 1 \tag{6}$$

Figure 3 illustrates the dependence of plurality voting accuracy p_c for $N = 1, 4, 7, 10, 15, 20, 50$ when $k = 3$. The value of ensemble accuracy in this case is given as:

$$\begin{aligned} p_c = &\sum_{i=\lceil\frac{N+1}{2}\rceil}^{N} \frac{N!}{i!(N-i)!} h^N (1-h)^{N-i} \\ &+ \sum_{i=\lceil\frac{N+1}{3}\rceil}^{\lceil\frac{N+1}{2}\rceil - 1} \sum_{j=N-2i+1}^{N-1} \frac{N!}{i!j!(N-i-j)!} h^i \Big(\frac{1-h}{2}\Big)^{N-i} \end{aligned} \tag{7}$$

For $k > 3$, the complexity of similar expressions quickly becomes prohibitive for closed form analysis. Certain Monte-Carlo estimations of the values of p_c were obtained in [10]. Note that for small values of N, combination accuracy p_c can be smaller than the accuracy h of each member of an ensemble if $1/k < h < 1/2$.

2.4 Probability of label permutation

The stochastic partition generation process in section 2.1 includes a label permutation step $T(.)$ because the Hungarian algorithm that matches the cluster labels of C_r with those of C_i can make an error. Consequently, the probability that an expert (an aligned partition in the ensemble) assigns the correct label to an object, p^* (defined in section

2.3), can be less than p. Strictly speaking, an expert can give the correct label to an object even when the Hungarian algorithm has made a mistake, because the expert may as well have assigned a wrong label to the object, and these two types mistakes can to cancel each other. This does not affect our analysis below, however, because we are only interested in an upper bound on the probability of error. Also, these "double mistakes" are rare and ignoring it does not have any practical consequences.

It is, however, difficult to conduct a probabilistic analysis of the result of the cluster label matching process, because Hungarian algorithm is defined only algorithmically. It is hard to obtain the distribution of its output based on a distribution of the input. Instead, we try to find bounds on the probabilities of the output. Without loss of generality, suppose the correct matching between the partitions C_i and C_r is to match the j-th cluster in C_i to the j-th cluster in C_r. Hungarian algorithm is guaranteed to find this matching if the (j,j)-th entry in the j-th row of the contingency table is the largest for all j. Since the conversion of the cluster label $C(x_l)$ to $C'(x_l)$ due to noise is assumed to be independent for different l, each row of the contingency table can be considered separately. It is easy to see that the entries in each row of the contingency table follow a multinomial distribution. Let γ_j be the probability that the (j,j)-th entry is not the largest in the j-th row in the contingency table. We have

$$\begin{aligned}\gamma_j &\le P(\ (j,j)\text{-th entry } \le n_j/2) \\ &= \sum_{i=0}^{\lceil \frac{n_j}{2} \rceil} \frac{n_j!}{(n_j-i)!i!} p^i (1-p)^{n-i},\end{aligned}$$

where n_j is the size of the j-th cluster. If n_j is large, by large deviation principle [1], the summand above can be approximated using $I(x)$, the rate function of Bernoulli trial, defined as

$$I(x) = x \log \frac{x}{p} + (1-x) \log \frac{1-x}{1-p}. \tag{8}$$

Assuming $p > 0.5$, we obtain $\gamma_j \le e^{-n_j I(0.5)} = (4p(1-p))^{n_j/2}$. The probability that the Hungarian algorithm makes a mistake can be upper bounded by $ke^{-n_j I(0.5)} = k(4p(1-p))^{n_j/2}$. So, a lower bound for p^* in section 2.3 is

$$p^* \ge p(1 - k(4p(1-p))^{n_j/2}), \tag{9}$$

under the approximation that the dependence between the error of the Hungarian algorithm and the error of labeling an individual object by the expert is negligible. Since the argument in section 2.3 requires only $p^* > 1/k$, a lower bound of p^* is sufficient to derive the convergence of the consensus of a cluster ensemble to the true partition.

3 Consensus as the mean partition

In this section, we present an alternative proof of the effectiveness of cluster ensemble by considering the properties of the mean partition with respect to a metric on a space of partitions.

3.1 Definitions

Let $d(P_i, P_j)$ denote a metric for two elements in $\mathbb{P}$. Many metrics have been proposed in the literature to compare two partitions of a data set, such as Rand index, Jaccard coefficient, Fowlkes and Mallows index, and Hubert's Γ, all of which are discussed in [8]. Although these are similarity measures, they can be easily converted to metrics because their values are upper bounded by one and the upper bound is attained if and only if two partitions are identical. Recently, normalized mutual information [7] and variation of information [11] have been used to compare two partitions. Here, we only require $d(.,.)$ to be symmetric, non-negative and $d(P_i, P_j) = 0$ only when P_i and P_j are the same. In particular, we do not require $d(.,.)$ to satisfy the triangle equality. One interpretation of consensus function is that it attempts to find the "mean" of the partitions in the ensemble. Formally, a consensus function applied to ensemble D_s of size s should return partition $\hat{C}$ such that

$$\hat{C} = \arg\min_{P_j \in \mathbb{P}} \sum_{i=1}^{s} d(P_j, C_i). \tag{10}$$

One such example is the consensus function based on voting, which uses the Hamming distance between P_j and C_i, after the matching of cluster labels by the Hungarian algorithm, as the metric. The consensus function based on mutual information in [15] can be regarded as another example, though, in this case, the partition that maximizes the sum of a similarity measure is returned instead.

The second tool for our analysis is a probability measure μ on $\mathbb{P}$, which characterizes the noise process that distorts the ground-truth partition. Each partition C_i in the ensemble D_s is a random variable sampled according to this measure, i.e., $P(C_i = P_j) = \mu(P_j)$. In order for C to be interpreted as the ground-truth, we require C to be the "mean" according to μ, i.e.,

$$C = \arg\min_{P \in \mathbb{P}} \sum_{i=1}^{m} \mu(P_i) d(P, P_i). \tag{11}$$

Intuitively, the distance of C from all the partitions in the ensemble should be the smallest. This is consistent with the definition of mean in the usual sense. There are two ways to specify probability measure μ. Usually, a stochastic partition generation process can be specified (as in section 2.1) that automatically induces a probability measure

on $\mathbb{P}$, given C. Alternatively, μ can be defined using distance $d(.,)$:

$$\mu(P_i) \propto \exp(-\lambda d(P_i, C)) \tag{12}$$

where C is a location-type parameter and λ is a scale-type parameter. The proportionality constant ommited in Equation (12), in general, depends on C.

3.2 Analysis of a simple case

The final goal of our analysis is to answer the question: what is the probability that the consensus partition $\hat{C}$ in Equation (10) is equal to the true partition C? Will this probability approach to 1 when the size of the ensemble increases indefinitely? The answer is affirmative. For simplicity, we shall assume Equation (11) has only one solution, i.e., P is unique.

We begin by studying a simpler version of the problem. Consider two arbitrary partitions, α and β, such that α is more "appropriate" mean partition than β, in the sense that

$$u \equiv \sum_{i=1}^{m} \mu(P_i) d(\alpha, P_i) - \sum_{i=1}^{m} \mu(P_i) d(\beta, P_i) < 0. \tag{13}$$

Intuitively, the average distance (with respect to the distribution μ) from α to all the partitions is smaller than that from β. Consider a random ensemble $D_s = (Y_1, \ldots, Y_s)$ of s partitions, such that $Y_i \in \mathbb{P}$ are i.i.d. random variables drawn from the distribution μ. If, for a particular realization of D_k, $\sum_{i=1}^{s} d(\alpha, Y_i) \geq \sum_{i=1}^{s} d(\beta, Y_i)$ is satisfied, then β is more appropriate than α as the mean partition. Hence ensemble D_s leads to a wrong consensus solution. We want to estimate the probability for this error to occur. Let $Z_i = d(\alpha, Y_i) - d(\beta, Y_i)$ be a real-valued random variable. Assuming that μ gives positive probabilities to both partitions α and β, we can see that $v \equiv \mathrm{Var}[Z_i]$, the variance of Z_i, is positive. Error (incorrect consensus) occurs if $\sum_{i=1}^{s} Z_i \geq 0$. Define

$$\Gamma(\lambda) \equiv \log E[e^{\lambda Z_i}] = \log \sum_{i=1}^{m} \mu(P_i) e^{\lambda Z_i}, \tag{14}$$

$$\Gamma^*(x) \equiv \sup_{\lambda \in \mathbb{R}} \big(\lambda x - \Gamma(\lambda)\big), \tag{15}$$

where $\Gamma(\lambda)$ ($\lambda \in \mathbb{R}$) is the logarithm of the moment generating function of Z_i, also known as the cumulant generating function. $\Gamma^*(x)$ ($x \in \mathbb{R}$) is the Fenchel-Legendre transform of $\Gamma(\lambda)$ [1]. Since for all λ, $\Gamma(\lambda)$ is now finite, continuous, and the variance of Z_i is non-zero (implying the second derivative of $\Gamma(\lambda)$ evaluated at $\lambda = 0$ is positive), $\Gamma^*(x)$ is a continuous convex function with minimum value 0. This minimum value is attained only at $x = E[z_i] = u$. By Cramer's theorem [1] and the continuity of $\Gamma^*(x)$, we have

$$\lim_{s \to \infty} \frac{1}{s} \log P\big(\big(\frac{1}{s} \sum_{i=1}^{s} Z_i\big) \in [0, \infty]\big) = - \inf_{x \in [0,\infty]} \Gamma^*(x) \tag{16}$$

Since $\Gamma^*(x)$ is the smallest when $x = u < 0$, $\inf_{x \in [0,\infty]} \Gamma^*(x)$ is simply $\Gamma^*(0)$ and we write $e \equiv \inf_{\lambda \in \mathbb{R}} \Gamma(\lambda) = \Gamma^*(0) > 0$. Since e is a positive constant, substituting α by the true mean partition C leads to the following theorem.

Theorem 3.1. *If $s \to \infty$, the probability of obtaining any partition β other than C as the mean partition of an ensemble D_s decreases exponentially with respect to s: $O(\exp(-se))$, where e is a positive constant.*

The exact value of constant e depends on the minimum value of the moment generation function. If $d(.,.)$ and μ were given, we could, in principle, find the minimum of $\Gamma(\lambda)$ by setting its derivative to zero. However, such optimization can be very complicated and requires complete knowledge of μ. A reasonable approximation of e only from the values of $u = E[Z_i]$ and $v = VAR[Z_i]$ is possible. By the Central Limit Theorem, $\frac{1}{s}\sum_{i=1}^{s} Z_i$ is approximately normal with mean u and variance v/s. Using the rate function of the normal distribution, e can then be approximated as $u^2/(2v)$.

Consider one intuitive interpretation of this result. Suppose α and β are far apart and lie in regions of $\mathbb{P}$ with high and low probabilities, respectively. In this case β should be a much better estimate of the mean partition than α. The values of $d(\alpha, P_i) - d(\beta, P_i)$ are negative for most partitions $\{P_i\}$ with high probability, meaning that u takes a large negative value. So, e is large too, and the probability of error decreases very rapidly with s. This agrees with our expectation.

In practice, both u and v are hard to compute unless the support of μ contains only few elements. We can approximate u and v by either summation over P_i such that $\mu(P_i)$ are significant, or by adopting Monte Carlo simulation. The later is particularly appropriate when μ is defined only implicitly in an algorithmic form as a process to corrupt the true partition, such as the generation model described in section 2.1. The sampling according to distribution μ can be accomplished by following the noise model.

3.3 Analysis of the general problem

We now consider the general problem: what is the probability that the estimated consensus $\hat{C}$ (defined in Equation 10) based on D_s is different from C? $\hat{C}$ is different if there exists a partition $\beta \in \mathbb{P}$ such that β is more "central" than

C. In other words,

$$\begin{aligned}
&\text{Prob}(\hat{C} \neq C) = \\
&\quad \text{Prob}\Big(\bigcup_{\beta \in \mathbb{P}, \beta \neq C} \{ \sum_{i=1}^{s} \big(d(C, C_i) - d(\beta, C_i) \big) \geq 0 \} \Big) \\
&\leq \sum_{j=1, P_j \neq c}^{m} \text{Prob}\Big(\sum_{i=1}^{s} \big(d(C, Y_i) - d(P_j, C_i) \big) \geq 0 \Big) \\
&\qquad \approx \sum_{j=1, P_j \neq C}^{m} e^{-se(j)} \quad \text{when } s \to \infty
\end{aligned}$$

Here, we denote the dependence of e on β by writing $e(j)$, with $\beta = P_j$, and m is the total number of possible partitions. In the limiting case as s goes to infinity, the above summand will be dominated by the term with the smallest $e(j)$, i.e.,

$$\lim_{s \to \infty} \text{Prob}(\hat{C} \neq C) \leq e^{-s \min_j e(j)},$$

assuming there is a unique minimum of $e(j)$. The probability of a incorrect consensus decreases exponentially with increasing s, with the rate determined by the smallest $e(j)$.

Another possibility to obtain the decreasing rate of the error is to invoke the multi-dimensional version of the Cramer's theorem. Let $C = P_j$. Let Z_j be a random vector in $\mathbb{R}^{m-1}$ with $Z_i \equiv \big(d(C_i, C) - d(C_i, P_1), \ldots, d(C_i, C) - d(C_i, P_{j-1}), d(C_i, C) - d(C_i, P_{j+1}), \ldots, d(C_i, C) - d(C_i, P_m) \big)$. The event that the consensus partition is correct corresponds to the event that all the components of $\sum_{i=1}^{s} Z_i$ are non-negative. Let A denote the subset of $\mathbb{R}^{m-1}$ such that all the coordinates are non-negative. Let λ and x denote a $(m-1)$-dimensional vector of real numbers, and let $< x, y >$ denote the inner product between x and y. Define

$$\Gamma(\lambda) \equiv \log E[e^{<\lambda, Z_i>}] = \log \sum_{i=1}^{m} \mu(P_i) e^{<\lambda, Z_i>}, \quad (17)$$

$$\Gamma^*(x) \equiv \sup_{\lambda \in \mathbb{R}^{m-1}} \big(< \lambda, x > - \Gamma(\lambda) \big), \quad (18)$$

The Cramer's theorem states that

$$\lim_{s \to \infty} P(\frac{1}{s} \sum_{i=1}^{s} Z_i \in A) = - \inf_{x \in A} \Gamma^*(x), \quad (19)$$

and $\inf_{x \in A} \Gamma^*(x)$ is positive, assuming the variance of Z_i is non-zero. Once again we have the exponentially decreasing probability of error when s goes to infinity. This result is more of theoretical interest, however, because typical values of the number of partitions is exponential large with respect to the number of objects and estimating $\inf_{x \in A} \Gamma^*(x)$ is very difficult.

4 Conclusion

In this paper we have presented two approaches to prove the utility of the consensus partition of a cluster ensemble. The first approach is based on a plurality voting argument, while the second is based on a metric and a probability measure on the space of partitions. In both cases, we have shown that the consensus partition indeed converges to the true partition when the ensemble consists of a large number of partitions. Convergence of voting consensus solutions is guaranteed as long as each expert (partition) gives a better than random clustering result. In the second approach, we give an estimate of the rate of convergence. The current paper complements the existing empirical literature on cluster ensembles and provides rigorous proof of the utility of consensus partition.

References

[1] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Jones and Bartlett Publishers, 1993.

[2] M. Demirekler and H. Altincay. Plurality voting based multiple classifier systems: Statistically independent with respect to dependent classifier sets. *Pattern Recognition*, 35:2365–2379, 2002.

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

[4] B. Fischer and J. M. Buhmann. Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1411–1415, 2003.

[5] A. Fred. Finding consistent clusters in data partitions. In *Multiple Classifier Systems, volume LNCS 2096*, pages 309–318. Springer, 2001.

[6] A. Fred and A. K. Jain. Data clustering using evidence accumulation. In *Proc. of Sixteenth International Conference on Pattern Recognition*, pages IV:276–280, 2002.

[7] A. Fred and A. K. Jain. Robust data clustering. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II–128–133, 2003.

[8] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

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

[10] X. Lin, S. Yacoub, J. Burns, and S. Simske. Performance analysis of pattern classifier combination by plurality voting. *Pattern Recognition Letters*, 24(12):1959–1969, Aug. 2003.

[11] M. Meila. Comparing clusterings by the variation of information. In *Proc. of Computational Learning Theory*, pages 173–187, 2003.

[12] J. Munkres. Algorithms for the assignment and transportation problems. *J. SIAM*, 5:32–38, Mar. 1957.

[13] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, December 2002.

[14] A. Topchy, A. Jain, and W. Punch. A mixture model for clustering ensembles. In *Proc. SIAM Data Mining*, pages 379–390, 2004.

[15] A. Topchy, A. K. Jain, and W. Punch. Combining multiple weak clusterings. In *Proc. IEEE International Conference on Data Mining*, pages 331–338, 2003.