Sawtooth Software

RESEARCH PAPER SERIES

Achieving Consensus in Cluster Ensemble Analysis

Joseph Retzer, Sharon Alberg,
and Jianping Yuan

© Copyright 2009, Sawtooth Software, Inc.
530 W. Fir St.
Sequim, WA 98382
(360) 681-2300
www.sawtoothsoftware.com

ACHIEVING CONSENSUS IN CLUSTER ENSEMBLE ANALYSIS[1]

JOSEPH RETZER
SHARON ALBERG
JIANPING YUAN
MARITZ RESEARCH

CLUSTER ENSEMBLES: AN OVERVIEW

Cluster ensemble, or consensus clustering, analysis is a relatively new advance in unsupervised learning. It has been suggested as a generic approach for improving the accuracy and stability of "base" clustering algorithm results, e.g., k-means.

Cluster ensemble analysis may be described as follows:

Consider p1, p2, ... , pM to be a set of partitions of data set Z.
(together these partitions form an ensemble).

Goal: find a partition P based on p1, p2, ... , pM which best represents the structure of Z.(P is the combined decision called a "consensus")

Given the above, we may say, tongue in cheek, that there are only two concerns to address:

How do we generate diverse yet accurate partitions 1 through M ? and in addition;

How do we combine those partitions?

It turns out that both (1) and (2) may be accomplished in numerous ways. While the literature outlines various approaches to both (1) and (2), little work has been found by the authors that compares those methods. This paper will focus on comparative performance of competing methods for combining partitions aka "achieving consensus." We begin with a brief outline of ways to generate the set of partitions, known as the ensemble, and then describe consensus methods to be used for comparison.

In addition to comparing consensus methods, we also develop and describe a graphical depiction of ensemble diversity, useful in evaluating the performance of our ensemble-generating mechanism.

GENERATING THE ENSEMBLE

A brief list of ways in which ensemble member partitions may be generated is given below. It is important to note that this list is not exhaustive and that any one or combination of these techniques may be employed for this purpose.

Ensemble generation techniques:

Random selection of basis variable subsets / segment features:
Simply put, subsets of the variables are chosen and each is used to generate one or more partitions.

[1] Originally published in 2009 Sawtooth Software Conference Proceedings.

Random initializations:
E.g., multiple selection of starting centroids in k-means.

Sub-sampling / re-sampling:
This approach uses, for example, bootstrap samples of the data for generating partitions.

Use different types of clustering algorithms:
This particularly effective approach generates multiple partitions using differing clustering algorithms, e.g. k-means, latent class, hierarchical, etc.

Randomly choose number of clusters for each clusterer:
Another particularly effective approach which specifies a varying number of cluster solutions within a given algorithm.

A brief overview of various methods for combining ensemble partitions taken from the literature is given below. All but the last approach (hyper-graph) will be described in more detail in the next section.

Consensus Methods:

Direct Approach:
Re-label ensemble solutions to find single solution which best matches individual ones.

Feature Based:
Treat partitions as *M* categorical features and build a clusterer thereupon.

Pair-wise:
Average over similarity matrix depiction of each of the *M* ensemble partitions.

Hyper-graph:
Create hyper-graph representing total clusterers output and cut redundant edges.

Consensus Methods Compared

The following section will provide a brief overview of each consensus method used in our empirical comparisons. Consensus performance will be based on its ability to recover known cluster partitions from synthetic data sets.

Direct Approach

The first, and intuitively the most straightforward, technique is appropriately referred to as the "direct approach." The direct approach re-labels individual ensemble partitions and creates a consensus solution which is, on average, as close as possible to the individual partitions. It may be described as follows:

1. Generate M partitions on data with sample size N.
2. Specify a membership matrix for each of M partitions, where k is the number of groups in each partition:

$$U_{N\times k}^{(m)} \quad m = 1, \ldots, M$$

3. Define a dissimilarity measure between the true classification of individual i, (pi) and that produced by partition m, (ui(m)) as:

$$\|u_i^{(m)} - p_i\|^2.$$

4. Averaging over all cases, for partition *m*, gives the dissimilarity between $U^{(m)}$ and *P* as:

$$h\left(U^{(m)}, P\right) = \frac{1}{N}\sum_{i=1}^{N} \|u_i^{(m)} - p_i\|^2$$

5. The previous equation assumes cluster labels are "fixed." In actuality we need to consider all permutations of $U^{(m)} \rightarrow \prod_m(U^{(m)})$ when arriving at an optimal *P*. So our minimization problem becomes:

$$h\left(U^{(m)}, P\right) = \min_{p_1,\ldots,p_N} \min_{\prod_1,\ldots,\prod_M} \left(\frac{1}{M}\sum_{l=1}^{M} \frac{1}{N}\sum_{i=1}^{N} \|u_i^{(m)} - p_i\|^2\right)$$

Feature Based Approach

This method treats individual clusterer outputs as *M* categorical features and builds a cluster consensus thereupon. The steps necessary to carry out a feature based consensus analysis are given as:

1. Consider each cluster solution as representative of a "feature" of the data.
2. Replace the raw data (cluster basis variables) with k-tuple cluster labels.
3. Assume the data arises, in varying proportions, from a mixture of probability distributions, each representing a different cluster.
4. The goal is then to partition data into groups associated with component distributions (clusters).
5. The analysis necessary to accomplish this is referred to as Finite Mixture Modeling.

Pair-wise Approach

The pair-wise approach depicts each ensemble member with a similarity matrix, averages across all member similarity matrices and uses that average to generate a consensus solution. This approach may best be described with an illustration.

Assume our first ensemble partition contains 6 respondents assigned to 2 clusters as shown below. A similarity matrix $S^{(1)}$ may be constructed as follows:

Resp.	Cluster
Bill	(1)
Amy	(1)
Jeff	(2)
Pam	(2)
Mike	(1)
Kate	(2)

$\Longrightarrow$

	Bill	Amy	Jeff	Pam	Mike	Kate
Bill	1	1	0	0	1	0
Amy	1	1	0	0	1	0
Jeff	0	0	1	1	0	1
Pam	0	0	1	1	0	1
Mike	1	1	0	0	1	0
Kate	0	0	1	1	0	1

Next, a similarity matrix depiction of each ensemble partition is generated and labeled as:

$$S^{(1)}, \ldots, S^{(M)}$$

The similarity matrices are then averaged to get a "consensus" similarity matrix "*S"* as:

$$S = \frac{1}{M}\left(S^{(1)} + S^{(2)} + \cdots + S^{(M)}\right)$$

Finally, we may apply any clustering algorithm which accepts a similarity matrix as its input (e.g. ,*"single linkage," PAM*, etc.) to *S* in order to produce a consensus solution.

Sawtooth Software Approach

The Sawtooth Software (hereafter, "Sawtooth") approach is a modification of the meta-clustering algorithm discussed in Strehl and Gosh (2002). The first step is to dummy code ensemble members as shown in the tables below.

Three ensemble members, e.g., for the first four cases:

Resp.	Partition 1	Partition 2	Partition 3
1	1	4	2
2	2	2	1
3	2	3	1
4	1	4	2

Dummy code above to create basis variables:

Resp.	Partition 1		Partition 2				Partition 3	
1	1	0	0	0	0	1	0	1
2	0	1	0	1	0	0	1	0
3	0	1	0	0	1	0	1	0
4	1	0	0	0	0	1	0	1

The second step varies from the Strehl and Gosh approach of repeatedly clustering using a graph partitioning approach with relabeling of clusterers. A secondary cluster analysis is performed on the dummy coded values (8 variables above) using Sawtooth Software CCA's (Convergent Cluster Analysis) standard approach. This involves running multiple replicates and selecting the most reproducible solution. If several solutions are created from the second step, a third step involves clustering on cluster solutions of cluster solutions (CCC). Additional CCA's can be performed indefinitely (CCC…C). Sawtooth has found that the process converges very quickly.

Evaluating the Consensus with the Adjusted Rand Index

Standard cluster analysis quality measures may be used to evaluate cluster solutions when the true underlying partition is unknown. These include measures such as:

Hubert's Gamma: Correlation between distances and a 0-1-vector where 0 means same cluster, 1 means different clusters.

Dunn Index: Minimum separation / maximum diameter.

Silhouette Index

Calinski & Harabasz C(g)

This investigation, however, employs known "true" groupings and hence the focus of partition evaluation shifts away from "cluster quality" to "cluster recovery" where the partition to be recovered is the known solution.

While various cluster recovery measures were considered, support in the literature along with other aspects such as intuitive appeal, led the authors to choose the "Adjusted Rand Index" (ARI) for this purpose. Since the ARI is critical to our comparisons and in addition leads to an innovative depiction of ensemble diversity, its derivation and underpinnings are next presented in some detail.

The adjusted rand index (ARI) is based on Rand Index (Rand 1971). Hubert and Arabie (1985) adjusts the Rand Index to correct for chance levels of agreement, thereby avoiding spuriously large obtained values. Anecdotal evidence of its support in the literature is found in a 1988 article by Collins & Dent where the authors note *" ... based on current evidence it seems that the Hubert and Arabie ARI is the cluster recovery index of choice."* We begin by providing an intuitive description of the Rand Index and next show how it may be extended to the "Adjusted" Rand index.

The Rand Index measures the correspondence between two cluster partitions by focusing on pairs of objects. Specifically, it classifies pairs of objects in disjoint cluster solutions (partitions) in one of two ways:

together (same cluster) or

apart (different clusters).

The Rand Index is then a measure of the degree to which each pair of objects is classified the same by the two cluster solutions being compared.

Consider the following cross tabulation table:

	Solution V		
Solution U	Pair in same cluster	Pair in different clusters	
Pair in same cluster	a	b	$a+b$
Pair in different clusters	c	d	$c+d$
	$a+c$	$b+d$	N

where e.g., a = frequency of two objects in same cluster in both U and V
b = frequency two objects same in U, apart in V, etc.

It is clear that given n objects,

$$\frac{n(n-1)}{2} = N = \text{ number of pairs } = a+b+c+d.$$

Using the table above, we may define the Rand Index as:

$$\frac{a+d}{N} = \frac{\text{pairs classified in agreement}}{\text{total number of pairs}}$$

A problem with the Rand Index is that as the number of segments in the compared partitions falls, spuriously higher values of the index may result. Hubert & Arabie set about to correct this by creating what is referred to as the Adjusted Rand Index (ARI). Simply put, the ARI is:

$$ARI = \frac{\text{observed RI - expected RI}}{\text{max RI - expected RI}} = \frac{\text{observed improvement over chance}}{\text{max possible improvement over chance}}$$

(Where clearly the max Rand index = 1).

In order to calculate the expected RI, we need only replace *a*, *b*, *c* and *d* with expected frequencies conditioned on the assumption of partition independence using rules of probability.

Specifically,

replace *a* with *(a+b)(a+c)* $\rightarrow$

$$E(\text{agreements}) = \frac{(a+b)(a+c)}{N}$$

replace *d* with *(c+d)(b+d)* $\rightarrow$

$$E(\text{disagreements}) = \frac{(c+d)(b+d)}{N}$$

The expected Rand Index is then:

$$\frac{E(a+d)}{N} = \frac{[(a+b)*(a+c)+(c+d)*(b+d)]}{N^2}$$

Substituting this back into our original ARI formula we find the ARI is equal to:

$$\frac{N(a+d)-[(a+b)(a+c)+(c+d)(b+d)]}{N^2-[(a+b)(a+c)+(c+d)(b+d)]}$$

Reflecting Ensemble Diversity

Another useful application of the ARI is to compare ensemble member partitions and hence portray overall ensemble diversity (a necessary and critical condition for arriving at useful consensus solutions). This can be accomplished in the following way:

1. Calculate all pairwise partition ARI values.
2. Subtract each ARI value from previous step from 1 and create an $M \times M$ diversity matrix.
3. Graphically depict the diversity matrix from step (2) with a heat map.

Step (3) above produces a novel graphical depiction of overall ensemble diversity that provides an easily comprehendible synopsis of a single ensemble's diversity as well as an effective means for comparison of diversity across multiple ensembles.

THE DATA

We employ 10 synthetic data sets with known underlying clusters provided by Bryan Orme of Sawtooth Software for comparison of consensus algorithms. The data sets were deliberately designed to reflect fairly different sorts of underlying groups that may be found in market research. A brief overview describing each is given in the table below.

Data Set Descriptions

Data Set	Group Type	Basis Variables	σ	Seg 1	Seg 2	Seg 3	Seg 4	Seg 5	Seg 6
1	Extreme group sizes	No overlap on the means	1.5	100	300	600			
2	Moderately different sizes	No overlap on the means	2	200	300	500			
3	Equal sizes	No overlap on the means	2	333	333	334			
4	Extreme group sizes	Group 3 overlaps with 1 & 2	1.5	100	300	600			
5	Extreme group sizes	Group 3 overlaps with 1 & 2	1.5	600	300	100			
6	Extreme group sizes	Respondent data pattern based		50	100	150	200		
7	Random sizes	Means generated randomly	1	300	50	100	200	150	200
8	Random sizes	Means generated randomly	2	300	50	100	200	150	200
9	Random sizes	Means generated randomly	3	300	50	100	200	150	200
10	Random sizes	Means generated randomly	4	300	50	100	200	150	200

METHODOLOGY

First and foremost, the focus of this paper is on comparing consensus performance in terms of its ability to recover known underlying clusters. To that end, the following steps were taken:

1. Run each method on each data set
2. Calculate the ARI comparing the consensus solution to the known underlying groups for all runs
3. Compare the performance of each consensus algorithm, for each data set, using the ARI values from step (2)

The table below provides ARI measures comparing the consensus solution with the known underlying clusters for each algorithm on each data set.

Data Set	Group Type	Basis Variables	Feature Based	Direct	Sawtooth	Pair-wise
1	Extreme group sizes	No overlap on the means	0.61	**0.67**	0.66	0.60
2	Moderately different sizes	No overlap on the means	0.46	**0.47**	0.46	0.44
3	Equal sizes	No overlap on the means	**0.43**	0.36	**0.43**	0.40
4	Extreme group sizes	Group 3 overlaps with 1 & 2	0.35	**0.43**	0.34	0.30
5	Extreme group sizes	Group 3 overlaps with 1 & 2	0.64	**0.72**	0.71	0.56
6	Extreme group sizes	Respondent data pattern based	0.87	0.87	**0.89**	0.61
7	Random sizes	Means generated randomly	0.62	**0.66**	**0.66**	**0.66**
8	Random sizes	Means generated randomly	0.58	**0.65**	**0.65**	0.58
9	Random sizes	Means generated randomly	**0.53**	0.48	**0.53**	0.51
10	Random sizes	Means generated randomly	0.31	**0.36**	0.35	0.34
Means			0.54	0.57	0.57	0.50

It's clear from the results that, for these data, the Direct and Sawtooth approaches outperform all others. It is also apparent however that the Pair-wise and Feature Based approaches are not far behind in their ability to recover underlying true partitions.

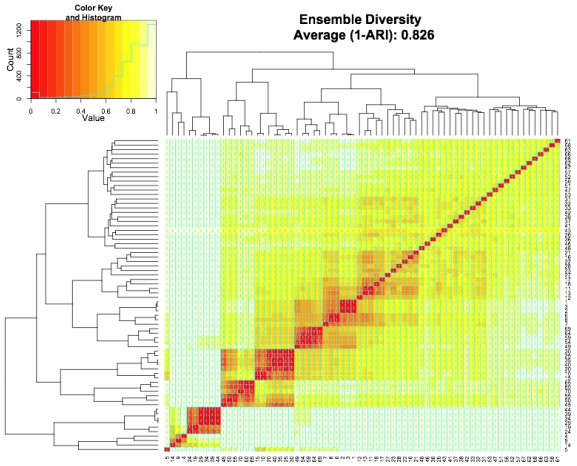
DEPICTING ENSEMBLE DIVERSITY

In addition to comparing consensus performance, the authors also employed the ARI to examine ensemble diversity. Diversity, as noted earlier, is an essential ensemble property which is necessary, but not sufficient, in arriving at useful consensus solutions. Ensemble diversity was depicted with a heat map graphic of the matrix composed of ARI pairwise comparisons of all ensemble members. For purposes of illustration, the authors chose 2 data sets and created heat map depictions of each ensemble. We then reduced the ensemble by removing specific partitions which added to the diversity of the set. The "diversity reduced" ensemble was also depicted graphically via heat maps and resultant ARI's associated with their consensus' ability to recover the known partition were estimated for each consensus algorithm.

It is important to note that the approach described above may not be used as a general measure of the effectiveness of a consensus algorithm's ability to deal with less diverse ensembles. The reason for this is that another critical property of the ensemble is being ignored, i.e., quality. If quality were controlled for, this approach may be used to add empirical support for or against a specific consensus algorithm's robustness to lack of diversity.

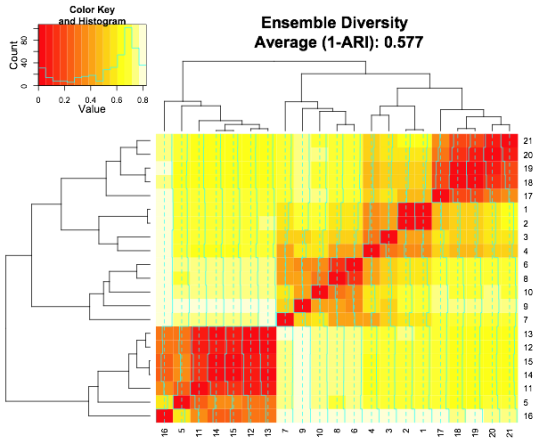
The following section presents heat map representations of ensemble diversity for both full (RHS) and reduced (LHS - less diverse) sets of partitions using data sets (1) and (2). Note that in both cases the drop in diversity is easily discerned by comparing adjacent heat maps. In addition, we see that a drop in performance (as measured by the ARI comparing consensus vs. true solution) is evident when using the less diverse ensemble. As noted however, the drop in ARI across ensembles may not be attributed solely to a drop in diversity since ensemble member quality is not being controlled.

Data Set 1



Full Ensemble

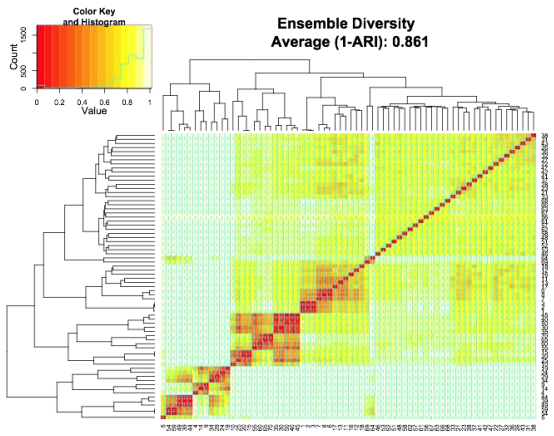
	F.B.	D.A.	S.S.	P.W.
ARI	.61	.67	.66	.60



Reduced Ensemble

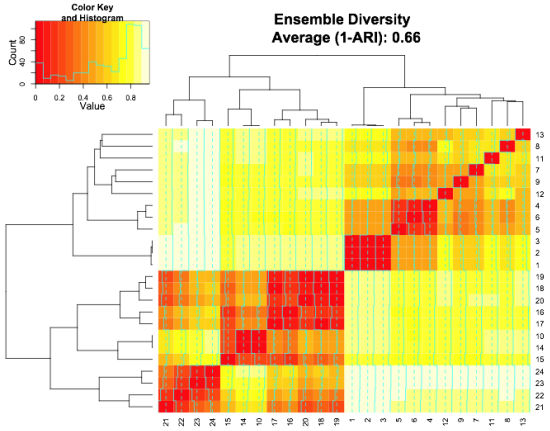
	F.B.	D.A.	S.S.	P.W.
ARI	.54	.60	.59	.59

Data Set 2



Full Ensemble

	F.B.	D.A.	S.S.	P.W.
ARI	.46	.47	.46	.44



Reduced Ensemble

	F.B.	D.A.	S.S.	P.W.
ARI	.30	.41	.45	.40

Conclusions & Future Work

Superior consensus performance provided by the Direct and Sawtooth approaches is the primary observation of this exercise. A secondary observation which, while not central to this paper is of no less importance, is the highly diverse ensemble produced by Sawtooth's program as evidenced in the heat maps. Details on Sawtooth's approach may be found in the appendix.

It is also apparent that cluster diversity may impact consensus solution performance directly. Specifically, as diversity decreases the risk of poorer consensus performance increases. This interpretation must be tempered by the realization that an important determinant of ensemble viability, individual partition quality, was not controlled for in these experiments.

We note that the literature recommends generating a large number of partitions and reducing down to a subset which is both diverse and of high quality. The authors feel this approach could be facilitated both numerically and graphically using pairwise ARI measurement to depict ensemble diversity. Specifically, partitions could be grouped based on similarity and selections from each group made using a measure of cluster quality. Overall diversity of the final ensemble could be depicted graphically via a diversity heat map.

Lastly, it is important to keep in mind the flexibility of the C.E. approach which adds value beyond its ability to create high quality solutions. For example, C.E.'s may also be used to construct partitions which profile well on marketing strategy variables by incorporating solutions from supervised learning analyses. Such a hybrid model is known as a Semi-Supervised Learning model and may be straightforwardly implemented in a cluster ensemble framework.

Appendix I: Generating the Ensemble

While this paper focuses on the comparison of consensus clustering techniques, it is important to be aware of the ensemble generation algorithm employed as well. The algorithm used is part of Sawtooth Software's Cluster Ensemble package (CCEA) which proved quite

capable of handling the difficult and critical task of generating a diverse ensemble. A detailed discussion of this process is given below.

CCEA software allows for creating an ensemble that can vary by the number of groups and cluster strategies. The default setting consists of 70 separate solutions, 2 to 30 groups, and the following five cluster strategies:

1. k-means (distance-based starting point)
2. k-means (density-based starting point)
3. k-means (hierarchical starting point)
4. hierarchical (average linkage criterion)
5. hierarchical (complete linkage criterion)

The ensemble is very diverse, but not assessed for quality, e.g., using CCA's reproducibility procedure. Reproducibility, however, is assessed for the ensemble consensus solution during the CCEA consensus stage.[2]

[2] In addition we may note that partitions generated outside CCEA may also be included in the ensemble for consensus creation using the software.

REFERENCES

Gordon, A. D. (1999), "Classification." Chapman & Hall/CRC.

Hornik, K. (2007), "A CLUE for CLUster Ensembles." R package version 0.3-18. URL http://cran.r-project.org/doc/vignettes/clue/clue.pdf.

Kaufman, L. & P. J. Rousseeuw (2005), "Finding Groups in Data, An Introduction to Cluster Analysis." Wiley-Interscience.

Kuncheva, L. I. & D. P. Vetrov (2006), "Evaluation of stability of k-Means cluster ensembles with respect to random initialization." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vo. 28, 11, November.

Orme, B. & R. Johnson (2008), "Improving K-Means Cluster Analysis: Ensemble Analysis Instead of Highest Reproducibility Replicates." Sawtooth Software.

Sawtooth Software (2008), "CCEA System," Sequim, WA.

Strehl, A. & J. Gosh (2002), "Cluster Ensembles - A knowledge reuse framework for combining multiple partitions." Journal of Machine Learning Research, 3, 583-617.

Topchy, A., A. Jain & W. Punch (2004), "A mixture model for clustering ensembles." Proc. of the SIAM conference on Data Mining, 379-390.

Vermunt and Magidson (2003), "LatentGOLD Version 3.0.1." Belmont Massachusetts: Statistical Innovations Inc.

Weingessel, A., E. Dimitriadou & K. Hornik (2003), "An ensemble method for clustering." DSC Working Papers.

Xiaoli, F., & C. Brodley (2003), "Random projection for high dimensional data clustering: a cluster ensemble approach." Proc. of the twentieth conference on machine learning, Washington D.C.