Cluster merging and splitting in hierarchical clustering algorithms

Chris Ding and Xiaofeng He
NERSC Division, Lawrence Berkeley National Laboratory
University of California, Berkeley, CA 94720. {chqding,xhe}@lbl.gov

Abstract

Hierarchical clustering constructs a hierarchy of clusters by either repeatedly merging two smaller clusters into a larger one or splitting a larger cluster into smaller ones. The crucial step is how to best select the next cluster(s) to split or merge. Here we provide a comprehensive analysis of selection methods and propose several new methods. We perform extensive clustering experiments to test 8 selection methods, and find that the average similarity is the best method in divisive clustering and the MinMax linkage is the best in agglomerative clustering. Cluster balance is a key factor to achieve good performance. We also introduce the concept of objective function saturation and clustering target distance to effectively assess the quality of clustering.

1 Introduction

Hierarchical clustering methods are among the first methods developed and analyzed for clustering problems[5]. There are two main approaches. (i) The agglomerative approach, which builds a larger cluster by merging two smaller clusters in a bottom-up fashion. The clusters so constructed form a binary tree; individual objects are the leaf nodes and the root node is the cluster that have all data objects. (ii) The divisive approach, which splits a cluster into two smaller ones in a top-down fashion. All clusters so constructed also form a binary tree.

The hierarchical cluster structure provides a comprehensive description of the data, which is quite useful for a number of applications. Given a dataset, however, the hierarchical cluster structure is not unique. It depends crucially on the criterion of choosing the clusters to merge or split. This is the main subject of this paper.

Besides hierarchical methods, there are many other clustering methods[5, 4]. A popular method is the K-means method, which is essentially a function minimization, where the objective function is the squared error. In most applications, one initializes K mean-vectors and directly optimize the objective function to obtain the optimal clusters. One can also follow a hierarchical divisive approach, and split a current cluster into two using the K-means method [7]. The gaussian mixture model using EM algorithm directly improves over the K-means method by using a probabilistic model of cluster membership of each object.

Both K-means method and Gaussian mixtures utilize directly the coordinates (attributes or variables) of the data points. From a general data clustering perspective, given all distances between data points, the cluster structure of the dataset is uniquely determined. Using the concept of *similarity*, we can equivalently say that given all pairwise similarities, the clustering is uniquely decided.

Recently, a MinMaxCut algorithm[3] is developed using similarity concepts. It is based on a min-max clustering principle: data should be grouped into clusters such that similarity between different clusters is minimized while the similarities within each clusters are maximized individually. MinMaxCut is extensively analyzed and experimented on two-cluster problems in [3], and is shown to be more effective than other current competitive methods such as the normalized cut [8] and PDDP [1]. Here we analyze the MinMaxCut on multi-cluster problems using hierarchical approaches.

2 Clustering objective functions

In this paper, we emphasize the view that clustering is an objective function optimization problem. This is a consistent and useful viewpoint. For example, we believe that after hierarchical clustering, one round of refinement of the leaf clusters based on an objective function will improve the clustering. The results of all 40 clustering experiments support this view.

2.1 K-means

The popular K-means algorithm is an error minimization algorithm where the objective function is the sum of error squared, sometimes called distorsion,

$$J_{\text{Kmeans}}(K) = \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \mathbf{c}_k)^2$$

where $\mathbf{c}_k = \sum_{i \in C_k} \mathbf{x}_i / n_k$ is the centroid of cluster C_k and $n_k = |C_k|$. When objects within each cluster are distributed according to a spherical Gaussian, with the same covariance for all clusters, J_{Kmeans} is a good measure.

2.2 Gaussian mixture model using EM

Both the K-means and MinMaxCut algorithms produce *hard* clustering, i.e., each data object belongs to exactly one cluster. For real life data, many points are situated near the boundaries between different clusters; it would be more natural to assign them fractionally to different clusters. A popular probabilistic model to allow this partial cluster membership is the Gaussian mixtures,

$$p(\mathbf{x}_i; K) = \pi_1 g_1(\mathbf{x}_i) + \cdots + \pi_K g_K(\mathbf{x}_i) \qquad (1)$$

where each component $g_k(\mathbf{x})$ is a Gaussian distribution, and π_k are prior distributions, satisfying $\sum_k \pi_k = 1$. The model parameters and cluster membership are determined by maximize the log-likelihood

$$\ell(K) = \sum_i \log(p(\mathbf{x}_i; K))$$

An efficient algorithm to achieve this is the Expectation-Maximization (EM) algorithm [2].

2.3 MinMax Cut

We briefly describe the MinMaxCut algorithm. Given n data objects and the pairwise similarity matrix $W = (w_{ij})$ where w_{ij} is the similarity between i, j, we wish to partition the data into two clusters C_1, C_2 using the min-max clustering principle. The similarity between C_1, C_2 is defined to be $s(C_1, C_2) \equiv \sum_{i\in C_1}\sum_{j\in C_2} w_{ij}$, which is also called the *overlap* between C_1, C_2. The similarity within a cluster C_1 is the sum of pairwise similarities within C_1: $s(C_1, C_1)$. We call $s(C_1, C_1)$ self-similarity of cluster C_1. The clustering principle requires minimizing $s(C_1, C_2)$ while maximizing $s(C_1, C_1)$ and $s(C_2, C_2)$ simultaneously. These requirements lead to the minimization of the MinMaxCut objective function,

$$J_{\text{MMC}} = \frac{s(C_1, C_2)}{s(C_1, C_1)} + \frac{s(C_1, C_2)}{s(C_2, C_2)}. \qquad (2)$$

Finding a partition to minimize J_{MMC} is NP-hard. However, an order $O(n^2)$ method exists that computes a globally near-optimal solution. A clustering is represented by an indicator vector $\mathbf{q}$, where $q(i) = a \quad \text{if} \quad i \in C_1; -b \quad \text{if} \quad i \in C_2$, and a,b are constants. One can show

$$\min_{\mathbf{q}} J_{\text{MMC}}(C_1, C_2) \Rightarrow \min_{\mathbf{q}} \frac{\mathbf{q}^T(D-W)\mathbf{q}}{\mathbf{q}^T D\mathbf{q}}, \qquad (3)$$

subject to $\mathbf{q}^T W\mathbf{e} = \mathbf{q}^T D\mathbf{e} = 0$, where $D = \text{diag}(d_i)$ is a diagonal matrix and $d_i = \sum_j w_{ij}$, and $\mathbf{e} = (1, \cdots, 1)^T$. We relax $q(i)$ from discrete indicators $\{a, -b\}$ to continuous values in $(-1, 1)$. The solution of $\mathbf{q}$ for minimizing the Rayleigh quotient of Eq.(3) is given by $(D-W)\mathbf{q} = \lambda D\mathbf{q}$. Since $\mathbf{q}_1 = \mathbf{e}^T$ is the eigenvector corresponding to the lowest eigenvalue, the second lowest eigenvector $\mathbf{q}_2$ is the desired solution (details and references are given in [3]). For $K \geq 3$, we define $J_{\text{MMC}}(C_1, \cdots, C_K) \equiv J_{\text{MMC}}(K)$,

$$J_{\text{MMC}}(K) = \sum_{p,q} J_{\text{MMC}}(C_p, C_q) = \sum_k \frac{s(C_k, \bar{C}_k)}{s(C_k, C_k)} \qquad (4)$$

where $\bar{C}_k = \sum_{p\neq k} C_p$ is the complement of C_k. In this paper, we use hierarchical approach to optimize this K-way clustering objective function.

3 Cluster merging and splitting

In hierarchical clustering, clusters are either merged into larger clusters or split to smaller clusters. It is instructive to see how clustering objective functions change with respect to the change of K, the number of clusters. Here we point out a fundamental difference between the graph-based MinMaxCut and the Euclidean distance based the K-means and Gaussian mixture.

Given the dataset and similarity measure (Euclidean distance in K-means and similarity graph weight in MinMaxCut), the global optimal value of the objective function is a function of K. An important property of these clustering objective functions is the monotonicity. We can prove that as K increases $K = 2, 3, \cdots$, the MinMaxCut objective increases monotonically, while the K-means objective decreases monotonically.

Theorem 1. Given the dataset and the similarity metric, as K increases, (i) the optimal value of the K-means objective function decreases monotonically:

$$J^{\text{opt}}_{\text{Kmeans}}(K) > J^{\text{opt}}_{\text{Kmeans}}(K+1)$$

(ii) the maximun log-likelihood of the Gaussian mixture increases monotonically:

$$\ell^{\text{opt}}(K) < \ell^{\text{opt}}(K+1)$$

(iii) the optimal value of the MinMax Cut objective function increases monotonically:

$$J^{\text{opt}}_{\text{MMC}}(K) < J^{\text{opt}}_{\text{MMC}}(K+1)$$

Proof. It is sometimes assumed that (i) is obvious, based on the arguments that for $K+1$ clusters, there are more parameters (than K cluster case) in the function optimization. This argument is incorrect, since it can be equally appled to (iii) and obtain wrong conclusion. We prove it for $K = 2$ by considering the 2-cluster problem. Assume we have found the optimal clustering for $K = 2$ and denote the clusters as A, B, with

$$J^{\text{opt}}_{\text{Kmeans}}(A, B) = \sum_{i\in A}(\mathbf{x}_i - \mathbf{c}_A)^2 + \sum_{i\in B}(\mathbf{x}_i - \mathbf{c}_B)^2.$$

Now we fix A and split B into B_1, B_2 in an optimal way using K-means:

$$J_{\rm Kmeans}^{\rm B-split}(A, B_1, B_2) = \sum_{i\in A}(\mathbf{x}_i - \mathbf{c}_A)^2 + \sum_{i\in B_1}(\mathbf{x}_i - \mathbf{c}_{B_1})^2 + \sum_{i\in B_2}(\mathbf{x}_i - \mathbf{c}_{B_2})^2.$$

It is easy to show that

$$\sum_{i\in B_1}(\mathbf{x}_i - \mathbf{c}_B)^2 \geq \sum_{i\in B_1}(\mathbf{x}_i - \mathbf{c}_{B_1})^2$$

$$\sum_{i\in B_2}(\mathbf{x}_i - \mathbf{c}_B)^2 \geq \sum_{i\in B_2}(\mathbf{x}_i - \mathbf{c}_{B_2})^2;$$

This is because for any $\mathbf{c}$, the quadratic function $f(\mathbf{c}) = \sum_{i\in B_1}(\mathbf{x}_i - \mathbf{c})^2$ achieves its minimum at $\mathbf{c} = \mathbf{c}_{B_1} = \sum_{i\in B_1}\mathbf{x}_i/n_{B_1}$, n_{B_1} is the size of B_1. Similar results hold for B_2. Note that the equality sign can not hold simultaneously for B_1, B_2. Therefore, we have

$$J_{\rm Kmeans}^{\rm B-split}(A, B_1, B_2) < J_{\rm Kmeans}^{\rm opt}(A, B) \tag{5}$$

Now the true global minimum $J_{\rm Kmeans}^{\rm opt}(K=3)$ must be lower than or equal to the particular instance of A, B_1, B_2. Thus we have

$$J_{\rm Kmeans}^{\rm opt}(K=3) \leq J_{\rm Kmeans}^{\rm B-split}(A, B_1, B_2) < J_{\rm Kmeans}^{\rm opt}(K=2)$$

This proves (i) for $K=2$. For $K=3$, we fix A, B and split C; the proof goes through similarly. By induction, this holds for arbitrary K.

We can follow the same steps to prove (ii); details are omitted here.

To prove (iii), we assume A, B_1, B_2 are the optimal clusters for $K=3$ for a given dataset. Now we merge B_1, B_2, and show that

$$J_{\rm MMC}^{\rm B-merge}(A, B) < J_{\rm MMC}^{\rm opt}(A, B_1, B_2). \tag{6}$$

This is because

$$\begin{aligned}
J_{\rm MMC}^{\rm opt}(A, B_1, B_2) - J_{\rm MMC}^{\rm B-merge}(A, B) &= -\frac{s(A,B)}{s(B,B)} \\
&\quad + \frac{s(B_1,A)+s(B_1,B_2)}{s(B_1,B_1)} + \frac{s(B_2,A)+s(B_2,B_1)}{s(B_2,B_2)} \\
&= \left[\frac{s(B_1,A)}{s(B_1,B_1)} - \frac{s(B_1,A)}{s(B,B)}\right] + \left[\frac{s(B_2,A)}{s(B_1,B_1)} - \frac{s(B_2,A)}{s(B,B)}\right] \\
&\quad + \left[\frac{s(B_2,B_1)}{s(B_1,B_1)} + \frac{s(B_2,B_1)}{s(B_2,B_2)}\right]
\end{aligned} \tag{7}$$

Since $s(B_1, B_1) < s(B, B)$ and $s(B_2, B_2) < s(B, B)$, every term is positive; thus we have Eq.(6). Now the true global minimum for $K=2$ must be lower than or equal to the particular instance of A, B. Thus we have

$$J_{\rm MMC}^{\rm opt}(K=2) \leq J_{\rm MMC}^{\rm B-merge}(A, B) < J_{\rm MMC}^{\rm opt}(K=3)$$

This proves (iii) for $K=2$, which can be generalized to any K. □

We note that (i) and (ii) in Theorem 1 are previously known, although we are not aware of a concrete proof. However, they are generally considered true based on the arguments of number of parameters involved. We make two contributions in Theorem 1: (1) prove the monotonicity of MinMaxCut; (2) the proof of (1) shows that the general arguments based on the number of parameters are incorrect, and we provide a concrete proof of (i) and (ii).

Theorem 1 shows the fundamental difference between graph-based MinMaxCut objective function and the K-means and Gaussian mixture. If we use the optimal value of the objective function to judge what is the optimal K, then K-means and Gaussian mixture favor large number of clusters while MinMaxCut favors small number of clusters. The monotonic increase or decrease indicate that one cannot determine optimal K from objective function alone. This is a well-known fact in data mining and Theorem 1 is a concise proof in support of this fact.

Another consequence of Theorem 1 is that in the top-down hierarchical divisive clustering, as clusters are split into more clusters, the K-means objective will steadily decrease while the MinMaxCut objective will steadily increase.

4 Objective function saturation

If a dataset has K reasonably distinguishable clusters, these natural clusters could have many different shapes and sizes. But in many datasets, clusters overlap substantially and natural clusters cannot be defined clearly. Therefore, in general, a single (even the "best" if exists) objective function J can not effectively model the vast different types of datasets. For many datasets, as J is optimized, the accuracy (quality) of clustering is usually improved. But this works only up to a point. Beyond that, further optimization of the objective will not improve the quality of clustering because the objective function does not necessarily model the data in fine details. We here formalize this characteristics of clustering objective function as the *saturation* of objective function.

Definition. For a given measure η of quality of clustering (i.g, accuracy), the saturation objective, J_{sat}, is defined to be the value when J is further optimized beyond $J_{\rm sat}$, η is no longer improved. We say η reaches its saturation value η_{sat}.

Saturation accuracy is a useful concept and also a useful measure. Given a dataset with known class labels, there is a unique saturation accuracy for a clustering method. Saturation accuracy gives a good sense on how well the clustering algorithm will do on the given dataset.

In general we have to use the clustering method to do extensive clustering experiments to compute saturation accuracy. Here we propose an effective method to compute an upper bound on saturation accuracy for a clustering method. The method is the following. (a) Initialize with the perfect clusters constructed from the known class labels. At this stage, the accuracy is 100%. (b) Run the refinement algorithm on this clustering until convergence. (c) Compute accuracy and other measures. These values are the upper bounds on saturation values.

5 Hierarchical divisive clustering

Divisive clustering starts from the top, treating the whole dataset as a cluster. It repeatedly partitions a current cluster (a leaf node in a binary tree) until the number of clusters reaches a predefined value K, or some other stopping criteria are met. The most important issue here is how to select the next candidate cluster to split. After discussing the size-priority selection, we introduce 4 new selection methods.

5.1 Size-priority cluster split

A common approach is to select the cluster with largest size to split [5]. This approach gives priority to produce size-balanced clusters. This can be written as choose p according to $p = \arg\min_k(1/n_k)$. This is a reasonable approach. However, natural clusters are not restricted to the situation where each cluster has the same size. Thus this approach is not necessarily the optimal approach.

5.2 Average similarity

Here we propose a new cluster choice. The idea is from the min-max clustering principle. The self-similarity of cluster C_k is $s(C_k, C_k) \equiv s_{kk}$, which is to be maximized during clustering. Define *average* self-similarity for each cluster, computed as $\bar{s}_{kk} = s_{kk}/n_k^2$. A cluster C_k with large $\bar{s}_{kk}$ implies that cluster members are more homogeneous; if we define similarity as the inverse of distance, $w_{ij} = 1/d_{ij}$, we may say that cluster members are more close to each other in Euclidean space, i.e., C_k is compact or tight. A cluster with small $\bar{s}_{kk}$ is less homogeneous or loose. A goal of min-max clustering principle is to produce clusters as compact and balanced as possible. Therefore, our priority in splitting clusters is to increase average similarity for all clusters. Our criterion is to choose the loosest (smallest average similarity) cluster p to split, $p = arg\min_k(s_{kk}/n_k^2)$. Our experiments (§7) show that this choice works well when natural clusters have either similar or different sizes.

5.3 Cluster cohesion

For a cluster with a given average similarity, there could be many different shapes. An elongated cluster (or a cluster consisting of two well-separated subclusters) could have the same average similarity as a highly spherical cluster. We introduce a new quantity to measure the difficulty for breaking the cluster into two:
Definition. Cluster cohesion is the smallest value of the MinMaxCut objective function when the cluster is split into two sub-clusters.
A quantity similar to the idea of cluster cohesion is implicitly used in [6].

A cluster k with small cohesion h_k implies it can be meaningfully split into two. Therefore a cohesion-based criterion is to choose the cluster p with the smallest cohesion among the current leaf clusters: $p = \arg\min_k h_k$.

The arguments above also suggest the combination of cohesion with average similarity could be a good cluster selection criterion. In this similarity-cohesion criterion, we select the cluster p according to

$$p = arg\min_k \ \left(s_{kk}/n_k^2\right)^\gamma h_k^{(1-\gamma)}. \tag{8}$$

by setting $\gamma = 1/2$. Note that setting $\gamma = 1$, we get similarity criterion; setting $\gamma = 0$, we get cohesion criterion.

5.4 Temporary objective

All above cluster choices are based on cluster characteristics and do not involve the clustering objective Eq.(4). Since the goal of clustering is to optimize the objective function, we can choose the cluster C_k such that the split of C_k leads to the smallest increase in the overall objective temporarily. This step-wise greedy approach is similar to the cohesion criterion, since the cohesion, the last term in Eq.(7), is the dominant contribution to the increase in J_{MMC}.

5.5 Stopping criteria

There are two stopping criterion for terminating the divisive procedure. (i) Terminate when the number of leaf nodes reaches the pre-defined K. (ii) Terminate when J_{MMC} computed based on current clusters on leaf nodes goes above a threshold value J_{stop}. Theorem 1 indicates that as the divisive process continues and the number of leaf clusters increase, J_{MMC} increases. Since J_{MMC} measures the overlop between different clusters (properly weighted against self-similarities), a large J_{MMC} (above J_{stop}) indicates that the current cluster is already highly homogeneous and it is better not to cut it further.

In applications where we do not know the correct K, we prefer to use (ii) as stopping criterion. In this paper, K is already known for the datasets, thus we use (i).

5.6 Refinement

In standard hierarchical clustering, clusters are taken as they are in the clustering tree. However, one can improve the clusters by refining them according to a proper clustering objective function. In this paper, we use MinMaxCut to split clusters, and thus we optimize the multi-K MinMaxCut objective function on the clusters produced by the divisive process.

Cluster refinement based on MinMax objective for $K = 2$ is discussed in [3]. Here we extend to $K > 2$. We use a greedy procedure. For each data point $\mathbf{x}_i$, we move it from its current cluster C_k to the new cluster C_p that gives the smallest objective function value. Note that often $p = k$, i.e., $\mathbf{x}_i$ should stay with C_k. Only those points near the boundaries are likely to be moved to other clusters.

6 Clustering quality measures

The results for clustering are specified by the $m \times n$ confusion matrix, $Z = (z_{pq})$, where z_{pq} is the number of data points in the discovered cluster C_q (column q) which are in fact belongs to the true cluster R_p (row p). Therefore, $n_k = \sum_{p=1}^{m} z_{pk}$ (sum of column k) is the size of discovered cluster C_k, while $m_k = \sum_{q=1}^{n} z_{kq}$ (sum of row k) is the size of true cluster R_k. Note that $\sum_{k=1}^{m} m_k = \sum_{k=1}^{n} n_k = N$, total number of data points.

The standard accuracy for k-th cluster is $q_k = z_{kk}/m_k$ (when $n = m$), the fraction of C_k that truly belongs to R_k. Since each true cluster contribute m_k to the total N data points, i.e, their contribution has a weight m_k/N, the usual global accuracy is the weighted sum of $Q = \sum_k (m_k/N) q_k = \sum_k z_{kk}/N$.

Accuracy measures only use the diagonal elements in Z. To account for the off-diagonal entries in Z, the mutual information (quotient) is often used,

$$I(Z) = \frac{1}{H(R)} \sum_{pq} \frac{z_{pq}}{N} \log\left(\frac{z_{pq} N}{m_p n_q}\right) = 1 - \frac{H(R|C)}{H(R)}, \quad (9)$$

where $H(R)$ is the entropy over the row distribution $(m_1, \cdots, m_K)$, and $H(R|C) = \sum_k (n_k/N) H(R|C_k)$, where $H(R|C_k)$ is the conditional entropy for column C_k. Note that we include $H(R)$ in the denominator, such that (i) for perfect clustering, $I(Z) = 1$; (ii) for random clustering, $I(Z) = 0$. Since $H(R)$ is fixed for a given dataset, $H(R|C)$ is sometimes used instead[10]. I is also useful when $n \neq m$.

Here we argue that there are certain desirable characteristics of clustering (and classification) that are not captured by mutual information. We further propose a new metric, *target distance*, which correctly captures these desirable features and is also more sensitive than accuracy. The first cluster feature is illustrated by two clustering results represented by contingency tables Z_A and Z_B,

$$Z_A = \begin{bmatrix} 20 & 20 \\ 20 & 20 \end{bmatrix}, \; Z_B = \begin{bmatrix} 39 & 1 \\ 39 & 1 \end{bmatrix}.$$

They both have same mutual information $I = 0$ and the same accuracy $Q = 50\%$. However, we believe that the results of Z_A is better, because the two clusters obtained are more balanced; the two clusters obtained in Z_B are highly unbalanced (one has 78 points and the other has 2 points). The desired (target) contingency table for perfect clustering is $T = \begin{bmatrix} 40 & \cdot \\ \cdot & 40 \end{bmatrix}$. We thus define the *target distance* as

$$d(Z) = ||Z - T||_F / ||T||_F = (\Sigma_{ij} (z_{ij} - t_{ij})^2 / \Sigma_{ij} t_{ij}^2)^{1/2},$$

where the Frobenius norm of T is $||T||_F^2 = \sum_{ij} t_{ij}^2$. One can show that (i) $d = 0$ for perfect clustering; (ii) $0 \le d \le 2$. For the contingency tables Z_A, Z_B, $d(Z_A) = 0.707$ is lower than $d(Z_B) = 0.975$, indicating Z_A is better. Clearly, $d(Z)$ captures the desired feature of balanced clustering. Also note that both $d(Z_A)$ and $d(Z_B)$ are far above 0, indicating poor clustering.

The second clustering feature is illustrated in the clustering of contingency tables Z_A and Z_B,

$$Z_A = \begin{bmatrix} 10 & 1 & 1 & 1 \\ \cdot & 10 & 1 & 1 \\ \cdot & \cdot & 10 & 1 \\ \cdot & \cdot & \cdot & 10 \end{bmatrix}, Z_B = \begin{bmatrix} 10 & \cdot & \cdot & 3 \\ \cdot & 10 & 2 & \cdot \\ \cdot & \cdot & 10 & 1 \\ \cdot & \cdot & \cdot & 10 \end{bmatrix}.$$

The accuracy are 0.93% for both. Mutual information $I(Z_A) = 0.678$ and $I(Z_B) = 0.748$, indicating Z_B is better. However, we believe Z_A is better because there are 6 small errors in Z_A (six 1's), whereas there are 3 larger errors in Z_B (3, 2 and 1). In other words, we prefer *many but small* random errors, instead of *fewer but large* errors. The target distance, $d(Z_A) = 0.194, d(Z_B) = 0.229$ correctly asserts that Z_A is better.

The fundamental reason is due to squares of off-diagonal entries in Z: a larger error get squared ($3^2 = 9$) is much larger than the sum of small errors squared ($1^2 + 1^2 + 1^2 = 3$). Thus target distance captures the desired clustering feature. However, mutual information (entropy) gives opposite results: given two distributions $p_A = \{4, 0, 0, 0, 4\}$ and $p_B = \{4, 1, 1, 1, 1\}$ (the corresponding columns in two different contingency tables for clustering same data). $H(p_A|C) = \log(2)$ whereas $H(p_B|C) = 2\log(2)$. Since $H(R)$ is same (see Eq.9), the mutual information criteria favor the clustering p_A with one large error, instead of the clustering p_B with 4 small errors.

7 Clustering Internet newsgroups

We apply the MinMaxCut with the divisive clustering algorithm to document clustering. We perform experiments on Internet newsgroup articles in 20 newsgroups.

Dataset	MB		MU	
method	t-dist	acc(%)	t-dist	acc(%)
Saturation	0.111(26)	92.5(2.0)	0.117(26)	91.7(1.6)
Size-P I	0.259(75)	82.8(3.4)	0.371(191)	77.1(10.8)
Size-P F	0.121(25)	91.8(1.7)	0.309(194)	81.7(9.9)
cohesion I	0.594(200)	66.1(10.6)	0.374(214)	75.6(13.8)
cohesion F	0.531(233)	73.0(10.8)	0.355(227)	78.8(13.2)
Tmp-obj I	0.300(167)	80.3(9.0)	0.481(28)	70.9(2.2)
Tmp-obj F	0.218(238)	87.0(11.6)	0.454(23)	75.0(1.3)
avg-sim I	0.246(47)	83.5(2.0)	0.168(29)	88.4(1.8)
avg-sim F	0.124(21)	91.7(1.1)	0.114(22)	91.7(1.3)
sim-coh I	0.246(47)	83.5(2.0)	0.168(29)	88.4(1.8)
sim-coh F	0.120(17)	91.8(1.2)	0.122(17)	91.0(1.0)

Table 1: Clustering results of the divisive MinMaxCut for the datasets MB and MU. Errors are in parenthesis.

Dataset	LB		LU	
method	t-dist	acc(%)	t-dist	acc(%)
Saturation	0.249(26)	81.4(2.1)	0.292(72)	79.0(4.4)
Size-P I	0.509(54)	67.2(2.9)	0.536(113)	62.9(6.7)
Size-P F	0.429(109)	71.8(4.8)	0.465(47)	68.4(1.9)
Cohesion I	0.891(140)	46.3(11.6)	0.722(209)	50.9(14.7)
Cohesion F	0.822(37)	49.6(5.3)	0.622(206)	58.1(13.8)
Tmp-obj I	0.652(54)	56.9(4.9)	0.597(96)	60.1(4.2)
Tmp-obj F	0.651(48)	58.7(5.6)	0.466(35)	68.8(2.8)
avg-sim I	0.469(27)	69.3(2.3)	0.345(47)	74.8(4.6)
avg-sim F	0.395(71)	72.4(4.1)	0.344(32)	74.1(2.5)
sim-coh I	0.601(134)	63.5(5.4)	0.428(38)	71.0(2.3)
sim-coh F	0.502(181)	67.1(8.0)	0.354(32)	72.6(2.3)

Table 2: Clustering results of divisive MinMaxCut for the datasets LB and LU.

We focus on two sets of 5-cluster cases. The choice of $K = 5$ is to have enough levels in the cluster tree; we avoid $K = 4, 8$ where the clustering results are less sensitive to cluster selection. The first dataset includes

```
NG2:  comp.graphics
NG9:  rec.motorcycles
NG10: rec.sport.baseball
NG15: sci.space
NG18: talk.politics.mideast
```

In this dataset, cluster overlap at medium level. The second dataset includes

```
NG2:  comp.graphics
NG3:  comp.os.ms-windows
NG8:  rec.autos
NG13: sci.electronics
NG18: talk.politics.mideast
```

Here the overlaps among different clusters are large.

The newsgroup dataset is from www.cs.cmu.edu/afs /cs/project/theo-11/www/naive-bayes.html. Word - document matrix is first constructed. 2000 words are selected according to the mutual information between words and documents. Standard `tf.idf` term weighting is used. Cosine similarities between all pair of document are calculated and stored in W. We perform the clustering as explained above.

From each set of newsgroups, we construct two datasets of different sizes: (A) randomly select 100 articles from each newsgroup. (B) randomly select 200, 140, 120, 100, 60 from the 5 newsgroups, respectively. Dataset (A) has clusters of equal sizes, which is presumably easier to cluster. Dataset (B) has clusters of significantly varying sizes, which is presumably difficult to cluster. Therefore, we have 4 newsgroup - cluster size combination categories

LB: large overlapping clusters of balanced sizes
LU: large overlapping clusters of unbalanced sizes
MB: medium overlapping clusters of balanced sizes
MU: medium overlapping clusters of unbalanced sizes

For each category, 5 different random datasets are generated and the divisive MinMaxCut clustering algorithm using the 4 cluster selection methods is applied to each of them. The final results are the average of these 5 random datasets in each categories. The results of clustering on the four datasets are listed in Tables 1,2. Two quality measures are computed: the target distance (t-dist) and accuracy (acc). The upper bounds of saturation values are computed as described in §4. Clustering results for each cluster selection method, size-priority (Size-P), average similarity (avg-sim), cohesion and similarity-cohesion (sim-coh) (see Eq.8) and temporary objective (tmp-obj) are given in 2 rows: "I" (initial) are the results immediately after divisive cluster; "F" (final) are the results after two rounds of greedy refinements.

A number of observations can be made from these extensive clustering experiments. (1) The best results are obtained by average similarity cluster selection. This is consistent for all 4 datasets. (2) The similarity-cohesion cluster selection gives very good results, statistically no different from average similarity selection method. (3) Cluster cohesion alone as the selection method gives consistently poorest results. The temporary objective choice performs slightly better than cohesion criterion, but still substantially below avg-sim and sim-coh choices. These results are somehow unexpected. We checked the details of several divisive processes. The temporary objective and cohesion often lead to unbalanced clusters because of the greedy nature and unboundedness of these choices[1]. (4) Size-priority selection method gives good results for datasets with balanced sizes, but not as

[1]A current cluster C_k is usually split into balanced clusters C_{k1}, C_{k2} by the MinMaxCut [3]. However, C_{k1} and C_{k2} may be quite smaller than other current clusters, because no mechanism exists in the divisive process to enforce cluster balances. After several divisive steps, they could become substantially out of balance. In contrast, avg-similarity and size-priority choices prevent large unbalance to occur.

good results for datasets with unbalanced cluster sizes. These are as expected.

(5) The refinement based on MinMaxCut objective improves the accuracy about 9%, and improve the target distances about 50% for dataset MB for avg-sim and sim-coh selection methods. For other datasets, the improvements are less profound, but can be clearly recognized for all cluster selection methods on all datasets. This indicates the importance of refinements in hierarchical clustering. (6) In all datasets and all methods, target distances are more sensitive than accuracies in showing the improvements due to refinements. This fact, together with the discussions in §6, that indicates target distance is a good metric for assessing the quality of clustering. (7) Both accuracies and target distances of the final clustering with avg-sim and sim-coh choices are very close to the saturation values, indicating the obtained clusters are as good as the MinMax cut objective function could provide.

Dataset MB has been studied in [9] using K-means methods. The standard K-means method achievers an accuracy of 66%, while two improved K-means methods achieve 76-80% accuracy. In comparison, the divisive MinMaxCut achieves 92% accuracy (Table 1).

8 Agglomerative clustering

In hierarchical agglomerative clustering, clusters are built from bottom up. During each recursive procedure, we merge two *current* clusters C_p and C_q that has the largest pairwise linkage between C_p and C_q:

$$\max_{\langle pq \rangle} \ \ell(C_p, C_q)$$

among all pairs of clusters. The key here is the choice of pairs, based on the *linkage* function. Using similarity measure $W = (w_{ij})$, we can translate standard linkage functions[5] using *distance* into those using *similarity*: (i) the single linkage, defined as the closest distance (largest similarity) among points in C_p, C_q,

$$\ell_{\text{single}}(C_p, C_q) = \max_{i \in C_p \ j \in C_q} w_{ij}.$$

(ii) the complete linkage, defined as the fartherest distance (smallest similarity) among points in C_p, C_q,

$$\ell_{\text{complete}}(C_p, C_q) = \min_{i \in C_p \ j \in C_q} w_{ij}$$

(iii) the average linkage, defined as the average of all distances (similarities) among points in C_p, C_q,

$$\ell_{\text{average}}(C_p, C_q) = \frac{s(C_p, C_q)}{|C_p||C_q|}.$$

(iv) the MinMax linkage, proposed and analyzed in [3], is defined as

$$\ell_{\text{MinMax}}(C_p, C_q) = \frac{s(C_p, C_q)}{s(C_p, C_p)s(C_q, C_q)}.$$

It is motivated by min-max principle : given the same overlap $s(C_p, C_q)$, we want to merge the pair of clusters that has small self-similarities (or loose clusters); in this way, clusters with larger similarities are retained and the final clusters have larger self-similarities.

At start, each data object is a cluster. The similarity or linkage between objects $\mathbf{x}_i, \mathbf{x}_j$ is w_{ij}. As cluster gets merged, we need a linkage between a single object and a cluster of several objects. All 4 linkages above have unique extension to this point-cluster linkage. For cluster with one point $\mathbf{x}_i$, $s(\mathbf{x}_i, \mathbf{x}_i) \equiv 1$, and $s(\mathbf{x}_i, C_k) \equiv \sum_{j \in C_k} w_{ij}$.

In our experiments, we stop the agglomerative process when the current number of cluster reaches K, the known number of clusters in the dataset. In applications when K is unknown, one may set a threshold J_{stop}. As the number of current clusters is reduced during the cluster merging process, J_{MMC} gradually decreases(Theorem 1). The process is terminated when J_{MMC} reaches J_{stop}.

Agglomerative clustering is uniquely determined for a given linkage, independent of any objective function. But, as in divisive clustering, we can refine the clusters according to MinMaxCut objective once the agglomerative process terminates. This improves the clustering results in all 4 linkages in all experiments we performed.

We perform agglomerative clustering using all 4 linkages on the datasets MB and LB in §7. The same 5 random samples for each datasets are used for clustering. The agglomerative process is terminated when the number of total current clusters reaches 5. The clustering results of this initial clustering and those after two rounds of greedy refinements are shown in Table 3.

Several observations from Table 3 are obtained. (1) Complete linkage performs worst consistently. and average linkage performs slightly better. (2) The MinMax linkage leads to the best initial and final clustering in all experiments. (3) Based on target distance, single linkage out-performs complete linkage and average linkage, but substantially less well compared to MinMax linkage. After examining several cases, we believe

Dataset	MB		LB	
method	t-dist	acc(%)	t-dist	acc(%)
Single I	0.791(73)	35.4(3.7)	0.763(70)	33.6(5.8)
Single F	0.283(113)	80.0(7.9)	0.485(69)	69.5(6.1)
Complet I	1.248(03)	20.6(0.3)	1.034(04)	32.4(0.2)
Complet F	0.952(202)	40.2(16.0)	0.809(140)	46.7(9.8)
Average I	1.145(175)	28.2(14.7)	0.877(161)	45.8(13.5)
Average F	1.086(197)	31.9(16.4)	0.733(178)	53.2(14.4)
MinMax I	0.315(72)	77.0(4.6)	0.490(72)	67.7(3.3)
MinMax F	0.124(32)	91.6(1.9)	0.453(54)	74.5(2.9)

Table 3: Clustering results of agglomarative MinMax-Cut with 4 different linkage functions.

the reason for these points. are due to cluster balance. For complete linkage and average linkage, large clusters tend to merge to form even larger clusters, while smaller clusters tend to be left alone. This snow-ball effect leads to high unbalanced clusters as observed in experiments. Single linkage can sometime link small clusters to larger ones; MinMax linkage weighs inversely with self-similarities; thus small clusters can produce large linkage, which cause them to merge with other clusters and lead to balanced clusters. Producing balanced clusters was one motivation for MinMaxCut algorithm[3].

(4) Comparing results in Table 3 with those in Tables 1 and 2, the agglomerative clustering appears to be substantially less effective than divisive clustering. Even in the best cases, the initial clustering obtained in agglomerative clustering is substantially poorer than the initial clustering obtained in divisive clustering.

Observation (4) is somehow unexpected. One would think that since in agglomerative clustering, each data point is individually compared to all existing clusters and is assigned to the best cluster, the final clustering should be better than those from divisive clustering, where no individual points are carefully considered. Observation (4) suggests that the collective optimizations (overlap vs self-similarities of clusters, etc) in divisive clustering is more important than the optimizations of individual points in a greedy fashion.

We also note that agglomerative clustering is much slower to run; its complexity is $O(n^3 \log(n))$, in contrast to $O(n^2)$ for divisive clustering. All these suggest that agglomerative clustering is not as competitive as the divisive clustering method.

9 Summary and discussions

In this paper, we provide a comprehensive analysis and experiments on divisive and agglomerative clustering. We use *similarities* instead of the traditional *distances*. We study the effects of merging and splitting clusters on clustering objective functions of MinMaxCut, K-means and gaussian mixture. The monotonicity property provides a general guidance about the changes of objective functions during divisive or agglomerative process, which is useful for stopping criteria.

For divisive clustering, we introduce 4 new cluster selection criteria, the average similarity, the cluster cohesion avg-cohesion, and temporary objective. Extensive experiments on internet newsgroups show that average similarity and similarity-cohesion selection perform well. For agglomerative clustering, we introduce the MinMax linkage and compared with single-linkage, complete linkage and average linkage. The MinMax linkage for merging clusters is found to be most effective. We found that maintaining cluster balance during divisive or agglomerative process is a key factor for good performance.

Besides algorithmic studies, we introduce two useful new concepts, the objective function saturation and cluster target distance. These concepts provide concrete and more effective means to assess the quality of clustering provided.

Although the cluster choices are specified in *similarity*, they can be converted to *distances*. For example, the average similarity in §5.2 is converted into

$$p = \arg\max_k \sum_{i,j \in C_k} \frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{n_k^2} = \arg\max_k \sum_{i \in C_k} \frac{(\mathbf{x}_i - \mathbf{c}_k)^2}{n_k}$$

and the MinMax linkage of §8 is converted into two

$$\ell_{\mathrm{MinMax}}(C_p, C_q) = \frac{d_{pq}}{d_{pp}^{1/2} d_{qq}^{1/2}},\ d_{pq} = \sum_{i \in C_p} \sum_{j \in C_q} \frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{n_p n_q}.$$

Acknowledgments. This work is supported by U.S. Department of Energy, Office of Science (MICS Office and LDRD) under contract DE-AC03-76SF00098.

References

[1] D. Boley. Principal direction divisive partitioning. *Data mining and knowledge discovery*, 2:325–344, 1998.

[2] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via em algorithm. *J. Royal Stat. Soc. B.*, pages 1–38, 1977.

[3] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. *Proc. IEEE Int'l Conf. Data Mining*, pages 107–114, 2001.

[4] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.

[5] A.K. Jain and R.C. Dubes. *Algorithms for clustering data.* Prentice Hall, 1988.

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

[7] S.M. Savaresi and D. Boley. On the performance of bisecting K-means and PDDP. *Proc. SIAM Data Mining Conf*, 2001.

[8] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[9] H. Zha, C. Ding, M. Gu, X. He, and H.D. Simon. Spectral relaxation for k-means clustering. *Proc. Neural Info. Processing Systems (NIPS 2001)*, 2001.

[10] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. *Univ. Minnesota, CS Dept. Tech Report*, 2001.