Cluster Aggregate Inequality and Multi-Level Hierarchical Clustering

Chris Ding and Xiaofeng He

Lawrence Berkeley National Laboratory
Berkeley, California 94720, USA.
{chqding,xhe}@lbl.gov

Abstract. We show that (1) in hierarchical clustering, many linkage functions satisfy a cluster aggregate inequality, which allows an exact $O(N^2)$ multi-level algorithm of the standard agglomerative hierarchical clustering algorithm rather than the traditional $O(N^3)$ algorithm; (2) a desirable close friends cohesion of clusters can be translated into kNN consistency which is guaranteed by the multi-level algorithm; (3) For similarity-based linkage functions, the multi-level algorithm is naturally implemented as graph contraction. The effectiveness of our algorithms is demonstrated on a number of real life applications.

1 Introduction

Agglomerative hierarchical clustering (AHC) is developed in 1960's and is widely used in practice. AHC produces a tree describing the hierarchical cluster structure. Such a comprehensive description of the data is quite useful for broad areas of applications. For example, in bioinformatics research, AHC is most commonly used for clustering genes in a DNA gene microarray expression data, because the resulting hierarchical cluster structure is readily recognizable by biologists. The phylogenetic tree (similar to binary clustering tree) of organisms is often built using the UPGMA (unweighted pair group method average) AHC algorithm. In social sciences, the hierarchical cluster structure often reveals gradual evolving social relationships that help explain complex social issues. Another application of AHC is in classification tasks on a large dataset using support vector machine [12]. The hierarchical cluster structure allows one to use most detailed local representation near the decision boundaries where support vectors lie; but as one moves away from the decision boundaries, the centroid representation of progressively larger clusters can be used.

Besides hierarchical clustering, many other clustering algorithms have been developed (see recent survey and text books [5, 1, 3]). K-means clustering is perhaps the most commonly used method and is well developed. 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 algorithms can be viewed as a global objective function optimization problem. A related set of graph clustering algorithms are developed that partition nodes into two sub-clusters based on well-motivated clustering objective functions [8]. They are typically applied in a top-down fashion (see also[7]), and thus complement the bottom-up AHC.

Standard AHC scales as $O(N^3)$. A detailed description of AHC and complete references can be found in [4, 9]. A number of efficient implementation based on approximations have been proposed [10, 6]. Several recent studies propose to integrate hierarchical clustering with additional information [2, 11] or summary statistics[13].

In this paper, we focus on making AHC scale to large data set. Over the last 40 years, the basic AHC algorithm remains unchanged. The the basic algorithm is an iterative procedure; at each iteration, among all pairs of current clusters, the pair with largest linkage function value (smallest distance) are selected and merged.

We start with a key observation. In the AHC algorithm, at each iteration, one may merge all mutual-nearest-neighbor (1mn) pairs (defined by the linkage function) simultaneously in the same iteration. As long as the linkage function satisfies a "cluster aggregate inequality", this modified algorithm of simultaneously merge all 1mn-pairs at each iteration produces identical clustering results as the standard AHC algorithm. The cluster aggregate inequality is satisfied by most common linkage functions (see §2.2). This modified algorithm provides a natural multi-level implementation of the AHC, where at each level we merge all 1mn pairs. This *multi-level hierarchical clustering* (MLHC) algorithm provides an order-N speedup over the standard AHC.

Next we propose "close friends" cohesion as a desirable feature for clustering, which requires that for every member in a cluster, its closest friend is also in the same cluster. We show that the MLHC guarantees the close-friends cohesion, a desirable feature for clustering. We further extend this cluster membership cohesion idea to (mutual) nearest neighbor consistency, and show that MLHC improves this KNN consistency compare to other clustering algorithms. (§3)

When the linkage function is expressed in similarity (in contrast to distance or dissimilarity), the new algorithm is identical to multi-level *graph contraction*. Graph contraction provides a natural framework for hierarchical clustering (§4).

The effectiveness of our algorithms is demonstrated on a number of real life applications: DNA gene expressions for lung cancer, global climate pattern, and internet newsgroups (§5).

2 Multi-level hierarchical clustering (MLHC)

2.1 Algorithm

The standard agglomerative hierarchical clustering is a bottom-up process. During each step, we merge two *current* clusters C_p and C_q which are closest, or *most similar* among all pairs of current clusters:

$$\min_{<pq>} d(C_p, C_q).$$

where $d(\cdot, \cdot)$ is the dissimilarity-based (distance) linkage function between C_p and C_q. Many researches have studied the effects of different choice of *linkage* functions [4].

At each step of AHC with p current clusters, $p-1$ new linkage functions need be computed, and we have total $O(p^2)$ pairwise linkage functions. It takes p^2 comparisons to search for the pair with the largest linkage. This is repeated $N-1$ times. The total computation is

$$N_{\text{search}}^{\text{AHC}} = N^2 + (N-1)^2 + \cdots + 2^2 + 1^2 = O(N^3/3).$$

The new MLHC algorithm is motivated by the following observation. In each iterative step in AHC, when all pairwise linkage functions are computed, we can form all mutual nearest neighbor pairs (1mn-pairs) of current clusters using the linkage as the distance metric. Two objects (i,j) are a 1mn-pair if j is the nearest neighbor of i and vice versa.

In this perspective, the standard AHC merges only the 1mn-pair with largest linkage value. It is then natural to ask if we may also merge all other 1mn-pairs simultaneously. Will this produces the same results? An analysis shows that if the linkage function satisfies a "cluster aggregate inequality", then the clustering results remain the same as the standard AHC.

This observation suggests a simultaneous merging algorithm which we call MLHC. At each level, all 1mn-pairs are identified and merged (not just the pair with largest linkage value). This is repeated until all objects are merged into one cluster. The total number of level is about $log_2 N$. Thus the required computation is approximately

$$N_{\text{search}}^{\text{MLHC}} = N^2 + (N/2)^2 + (N/4)^2 + 2^2 + 1^2 = O(4N^2/3).$$

The new algorithm speedups by a factor of order-N.

2.2 Cluster Aggregate Inequality

In this section, we show that the simultaneous merging of all 1mn-pairs in MLHC produces identical clustering results as the standard AHC, provided the linkage function satisfies a *Cluster Aggregate Inequality.*

Definition. *Cluster Aggregate Inequality* of the linkage function. Suppose we have three current clusters A, B, C. We try to merge B, C into a new cluster $B+C$. The cluster aggregate inequality is a property of the linkage function that the merged cluster $B+C$ is "no closer" to A than either one of its individual members B or C. More precisely, for distance (dissimilarity) based linkage $d(\cdot,\cdot)$, the cluster aggregate inequality is

$$d_{A,B+C} \geq \min(d_{A,C}, d_{A,B}) \tag{1}$$

for any triple (A, B, C).

What kind of linkage functions satisfy the cluster aggregate inequality? It is interesting to see that most commonly used linkage functions satisfy cluster aggregate inequality. Consider the four similarity-based linkage function. (i) the single linkage, defined as the closest distance among points in A, B, (ii) the complete linkage, defined as the fartherest distance among points in A, B, (iii) the average linkage, defined as the average of all distances among points in A, B, (iv) the minimum variance linkage.

$$d^{\text{sgl}}_{A,B} = \min_{i \in A \ j \in B} d_{ij} \tag{2}$$

$$d^{\text{cmp}}_{A,B} = \max_{i \in A \ j \in B} d_{ij} \tag{3}$$

$$d^{\text{avg}}_{A,B} = \frac{d(A,B)}{n_A\, n_B} \tag{4}$$

$$d^{\text{min-var}}_{A,B} = \frac{n_A\, n_B}{n_A + n_B} ||\mathbf{c}_A - \mathbf{c}_B||^2. \tag{5}$$

Theorem 1. The single link, the complete link and average link satisfy the strong cluster aggregate inequality.

Proof. For single link, one can easily see that

$$d^{\text{sgl}}_{A,B+C} = \min_{i \in A; j \in B+C} d_{ij} = \min(\min_{i \in A; j \in B} d_{ij}, \min_{i \in A; j \in C} d_{ij}) = \min(d^{\text{sgl}}_{A,B}, d^{\text{sgl}}_{A,C})$$

Thus the equality in Eq.(1) hold for single link. With same reasoning, one can see the inequality holds for complete linkage.

For average link, we have

$$\begin{aligned}
d^{\text{avg}}_{A,B+C} &= \sum_{i \in A; j \in B+C} \frac{d_{ij}}{|A||B+C|} \\
&= \sum_{i \in A; j \in B} \frac{d_{ij}}{|A||B+C|} + \sum_{i \in A; j \in C} \frac{d_{ij}}{|A||B+C|} \\
&= \frac{|B|}{|B+C|} d^{\text{avg}}_{A,B} + \frac{|C|}{|B+C|} d^{\text{avg}}_{A,C} \\
&\geq \frac{|B|}{|B+C|} \min(d^{\text{avg}}_{A,B}, d^{\text{avg}}_{A,C}) + \frac{|C|}{|B+C|} \min(d^{\text{avg}}_{A,B}, d^{\text{avg}}_{A,C}) \\
&= \min(d^{\text{avg}}_{A,B}, d^{\text{avg}}_{A,C})
\end{aligned}$$

$\square$

2.3 Equivalence of MLHC and AHC

Cluster aggregate inequality plays a fundamental role in hierarchical clustering. It is similar to the triangle inequality in metric space: for any three vectors $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ in the Hilbert space with the distance metric $d(\cdot, \cdot)$, the metric must satisfies the triangle inequality

$$d_{ik} \leq d_{ij} + d_{jk}.$$

Triangle inequality plays a fundamental role in deriving properties of the metric Space. Now we prove the main results of this paper.

Theorem 2. If the linkage function satisfies the cluster aggregate inequality, the clustering trees produced by MLHC is identical to that produced by standard AHC.

Proof. We view the linkage function as a distance metric and build all 1mn-pairs based on the linkage function. We show that the AHC is iteratively merging 1mn-pairs, which is same as MLHC. The details is broken into two features of AHC below. □

We first note a simple feature of AHC:

Feature 1. The closest pair must be a 1mn-pair.

Proof. Suppose this is not true, i.e., the closest pair is (a, b), but a is not the closest neighbor of b. There must exist another point c which is the closest neighbor of a. Then the pair (a, c) must be the closest pair, but this contradicts the fact that (a, b) is the closest pair. Thus a must be the closest neighbor of b. Similarly, b must be the closest neighbor of a. □

Next, we prove a key feature of AHC. This shows the essence of Theorem 2.

Feature 2. If the linkage function satisfies the cluster aggregate inequality, then any 1mn-pair must be preserved and will merge eventually in standard AHC.

Proof. Suppose at certain iteration, the current clusters are listed as $(C_{j_1}, C_{j_2}), (C_{j_3}, C_{j_4}), C_{j_5}, (C_{j_6}, C_{j_7}), \cdots$, where 1mn-pair is indicated by the parenthesis. In AHC, the 1mn-pair with largest linkage value , say (C_{j_6}, C_{j_7}), is merged. Due to the cluster aggregate inequality, the newly merged cluster $C_{(j_6,j_7)}$ will be "no closer" to any other current clusters. Thus the 1mn of $C_{(j_6,j_7)}$ can not be any member of the current remaining 1mn-pairs, say C_{j_1}.

This can seen as follows. By construction, neither C_{j_6} nor C_{j_7} is closer to C_{j_1} than C_{j_2} does. Due to the cluster aggregate inequality, the newly merged cluster $C_{(j_6,j_7)}$ will be "no closer" to C_{j_1} than either C_{j_6} or C_{j_7} does. Thus 1mn of $C_{(j_6,j_7)}$ can not be C_{j_1}.

This guarantees that the 1mn-pair (C_{j_1}, C_{j_2}) will never be broken by a merged cluster. Thus in the next iteration, either a current 1mn-pair, say (C_{j_3}, C_{j_4}), is merged, or the newly-merged $C_{(j_6,j_7)}$ is merged with a singleton cluster, say

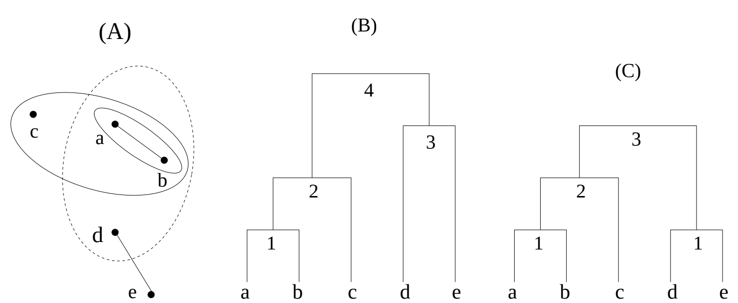

Fig. 1. (A) Dataset of 5 objects. 1mn-pairs are linked by a line. A merge is indicated by an elongated solid circle. (B) Dendrogram using AHC. Numbers indicate order of the merge. (C) Dendrogram using MLHC.

C_{j_5}. Therefore, the 1mn-pair (C_{j_1}, C_{j_2}) will preserve and eventually merge at some later iteration. $\square$

We give a simple example to illustrating some of the concepts. In Figure 1(b), we have 5 objects. They form two 1mn-pairs (a, b), (d, e) and one isolated object c. We do the standard AHC. Suppose (a, b) has the largest linkage value. So a,b are first merged into $(a + b)$. We assert that the 1mn-pair (d, e) must be preserved and will merge in later stages in AHC. This is done in two stages. First, we show that d cannot be the nearest neighbor of $(a + b)$, i.e.,

$$d(d, a + b) > d(d, e). \tag{6}$$

The fact that (d, e) is a 1mn-pair implies

$$d(d, a) > d(d, e), \tag{7}$$
$$d(d, b) > d(d, e), \tag{8}$$
$$d(d, c) > d(d, e). \tag{9}$$

From the cluster aggregate inequality, $d(d, a + b) \geq \min[d(d, a), d(d, b)]$. From this, and makeing use of Eqs.(7, 8), we obtain Eq.(6).

In the next round of AHC, either (i) the pair $(a+b, c)$ has the largest linkage value, or (ii) the pair (d, e) has the largest linkage value. If (ii) holds, our assertion is proved. Suppose (i) holds. Then $(a+b, c)$ are merged into $(a+b+c)$. Now we show that d cannot be the nearest neighbor of $(a + b + c)$, i.e.,

$$d(d, a + b + c) > d(d, e). \tag{10}$$

From the cluster aggregate inequality, $d(d, a + b + c) \geq \min[d(d, a + b), d(d, c)]$ From this, together with Eqs.(6, 9), we obtain Eq.(10). Therefore, (d, e) is the pair with the largest linkage value. Thus (d, e) are merged into $(d + e)$.

This example shows how 1mn pairs are preserved in AHC. Thus any cluster merge in MLHC (those 1mn-pairs) will also occur in AHC. There are total $N-1$ cluster merges in both MLHC and AHC. So any cluster merge in AHC also occur in MLHC.

Both AHC and MLHC algorithms lead to the same binary cluster tree. The difference is the sequential order they are merged. This is illustrated in Figure 1.(B,C). If we represent the tree hight by the standard linkage function value for each merge, the dendrograms of the two algorithm remains identical.

We emphasize that the equivalence of MLHC and AHC only requires 1mn-pair preservation during AHC, as shown in the Feature 2. The cluster aggregate inequality is a *sufficient* condition for 1mn-pair preservation.

This is a stronger requirement. It is possible for a given dataset, a particular linkage function maybe not satisfy the generic cluster aggregate inequality, but the 1mn-pair preservation holds during AHC and thus MLHC is equivalent to AHC. Thus in general MLGC is a fast and appealing alternative of AHC.

3 "Close friends" cohesion

One of the fundamental concept of data clustering is that members of the same cluster have high association with each other. One way to characterize the within cluster association is the cohesion of the cluster members via the preservation of "close friends". Suppose we divide 100 people into 10 teams. The cohesion of each team is greatly enhanced if for any member in a team, his/her close friends are also in the same team.

It is interesting to note that this close-friends cohesion is guaranteed by MLHC, if we interprete 1mn-pair relationship as close friends; by construction, this cohesion is guaranteed at all levels. We say the clustering results satisfy 1mn-consistency if for each object in a cluster, its 1mn is also in the same cluster.

By Theorem 2, clusters produced by the standard AHC also enjoy the same 1mn-consistency as in MLHC. We summarize this important result as

Theorem 3. In MLHC, 1mn-consistency is fully guaranteed. In agglomerative hierarchical clustering, if the linkage function satisfy the cluster aggregate inequality, 1mn-consistency is fully guaranteed.

3.1 Cluster membership kNN consistency

1mn describe the "most close" friend. Beyond the closest friend, It is also desirable that other less close friends are also in the same cluster. This will increase cohesiveness of the clustering.

We thus further extend the "close friends" into k-nearest-neighbor, the cohesiveness of a cluster becomes the following knn consistency:

Cluster Membership kNN Consistency: For any data object in a cluster, its k-nearest neighbors are also in the same cluster.

Note that the relationship of "close friend" is not symmetric. Although a's closest friend is b, b's closest friend could be c, not a. Thus the "mutual closest friend" implies the tightest friendship. Thus k-Mutual-Nearest-Neighbor Consistency is more desirable.

3.2 Enforcing kNN consistency

In general, clustering algorithms perform global optimizations, such as K-means algorithm, will gives final clusters with a smaller degree of cluster knn and kmn consistency than the bottom hierarchical clustering. This is because that in global optimizations, nearest-neighbor relations are not considered explicitly. In HC, a pair of clusters are merged if their linkage value is high. Thus clusters being merged are typically very similar to each other. Therefore the nearest-neighbor local information are utilized to some extent, leading to higher degree of knn consistency.

What about other knn/kmn consistency? First we note that cluster knn consistency defines a "transitive relationship". If x_1 is a member of cluster C_1, and x_2 is the 1nn of x_1, then by cluster 1nn consistency, x_2 should also be a member of C_1. This transitive relation implies that 100% 1nn consistency can be achieved only if entire connected component of the 1nn graph are in C_1. To generate clusters that guarantee 100% knn consistency, at each level of the MLHC, we must first generate knn-graph, identify all connected components, and for each CC, merge all current clusters into one cluster.

Because for any object, its 2nn set always include its 1nn set. Thus 2nn consistency guarantees 1nn consistency. Similarly, because any knn set include kmn set, knn consistency implies (k-1)-nn consistency.

4 Similarity-based hierarchical clustering: multi-level graph contraction

The above discussion uses the distance-based linkage. All results there can be easily translate into similarity-based linkage function.

For similarity-based linkage we select the pair with the largest linkage to merge: $\max_{<pq>} s(C_p, C_q)$, where $s(C_p, C_q)$ is the aggregate similarity between clusters C_p, C_q. Let the initial pairwise similarity are $W = (w_{ij})$. The aggregate similarity has a simple form, $s(C_p, C_q) = \sum_{i \in C_p} \sum_{j \in C_q} w_{ij}$.

Cluster aggregate inequality using similarity-based linkage can be written as

$$s(A, B + C) \leq \max[s(A, B), s(A, C)] \tag{11}$$

Consider the following similarity-based linkage functions. (i) the single linkage, defined as the largest similarity among points in A, B, (ii) the complete linkage, defined as the smallest similarity among points in A, B, (iii) the average linkage, defined as the average of all similarities among points in A, B,

$$s_{\text{single}}(A, B) = \max_{i \in A \; j \in B} s_{ij}, \; s_{\text{complete}}(A, B) = \min_{i \in A \; j \in B} s_{ij}, \; s_{\text{avg}}(A, B) = \frac{s(A, B)}{|A||B|}.$$

With similar analysis as in the case of distance-based clustering, we can proof **Theorem 4**. All three above similarity-based linkage functions satisfy the cluster aggregate inequality.

The similarity based linkage functions have an advantage that merging two cluster become graph node contraction. Defining the similarity between two objects as the weight on an edge between them, this forms a similarity graph. Thus the multi-level hierarchical clustering naturally become multi-level graph contraction of the similarity graph. Many well-known results in graph theory can be applied.

Merging two current clusters into a new cluster corresponds to contracting two nodes i, j into a new node k and with edge e_{ij} being eliminated. Weights of the graph are updated according to standard graph contraction procedure. Let $W^{(t)}, W^{(t+1)}$ be the weights of the similarity graph at steps $t, t+1$. The updated weights for contracting the edge e_{ij} and merging nodes i, j into k are

$$\begin{cases} w_{kk}^{(t+1)} = w_{ii}^{(t)} + w_{jj}^{(t)} + w_{ij}^{(t)} \\ w_{kp}^{(t+1)} = w_{ip}^{(t)} + w_{jp}^{(t)}, \quad \forall p \notin \{i, j, k\} \\ w_{pq}^{(t+1)} = w_{pq}^{(t)}, \qquad \forall p, q \notin \{i, j, k\} \end{cases}$$

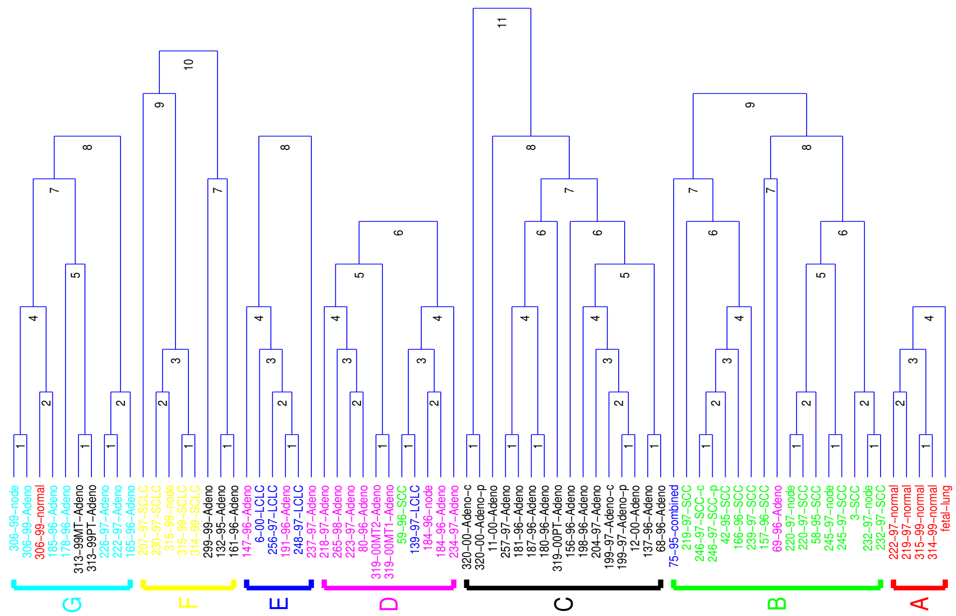

Fig. 2. MLHC clustering of lung cancer gene expressions.

Using the above graph contraction operation, AHC and MLHC can be described very succinctly. AHC can be viewed as contracting one edge (with heaviest weight) at each step. The steps are repeated until all nodes merge into one. This takes exactly $N-1$ steps. MLHC can be viewed as contracting all 1mn-pairs simultaneously at each step. It is repeated about $O(\log_2 N)$ times. Now the speedup of MLHC over AHC is clear. At each step, all-pair linkage function computation is necessary. But the number of steps required in AHC is $N-1$, and it is $O(\log_2 N)$ in MLHC.

5 Experiments

5.1 DNA Gene expression for Lung cancer

The DNA gene expressions of lung cancer patients (available online: http://genome-www.stanford.edu/lung_cancer/adeno/) contains 73 samples of 67 lung tumors from patients whose clinical course was followed for up to 5 years. The samples comprise of 916 DNA clones representing 835 unique genes. The samples are classified into 5 groups by visual examination (41 Adenocarcinomas (ACs), 16 squamous cell carcinomas (SCCs), 5 large cell lung cancers(LCLCs), 5 small cell lung cancer (SCLCs) and 5 normal tissue with one fetal lung tissue.). The largest group ACs is further classified into three smaller groups. The purpose is to see if we can recover this 7 groups using unsupervised learning method, i.e., the hierarchical clustering method. The Pearson correlations c_{ij} among tissue samples are computed first, and the similarity metric is defined as $s_{ij} = \exp(5c_{ij})$. We use MLHC and obtain the cluster structure as shown in Fig.2.

As the Figure shows, at 1st level, after an all-pair computation, 18 1mn-pairs are formed and merged. At 2nd level, 11 1mn-pairs are formed and merged. Total 11 levels of merges are required to obtain 7 clusters. In contrast, for standard AHC, We need 66 levels of merge steps to obtain 7 clusters. The clustering result is give in the confusion matrix $T = \begin{bmatrix} 5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 15 & \cdot & 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & 16 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 9 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 3 & 3 & \cdot & \cdot \\ \cdot & \cdot & 3 & \cdot & \cdot & 5 & \cdot \\ 1 & \cdot & 2 & \cdot & \cdot & \cdot & 7 \end{bmatrix}$ where $T = (t_{ij})$, t_{ij} is the number of data points which are observed to be in cluster i, but was computed via the clustering method to belong to cluster j. The accuracy is defined as $Q = \sum_k t_{kk}/N = 82\%$. indicating the effectiveness of the clustering algorithm.

5.2 Climate pattern

We tested MLHC on global precipitation data as shown in Fig.3. Regularly-spaced data points cover the surface of earth. Each data point is a 402-dimensional vector containing seasonal means over 100 years and geometric information: longitude and latitude. Similarity between two points are based two factors: (1) precipitation pattern similarity computed as Euclidean distance and (2) geometric closeness based on simple physical distance. The obtained stable regions (shown in different color and symbols) correlate well with continents, say, in Australia, south Americas.

5.3 Internet Newsgroups

We apply MLHC on Internet newsgroup articles. A 20-newsgroup dataset is from www.cs.cmu.edu /afs/cs/project/theo-11/www/naive-bayes.html. 1000 words are selected according to the mutual information between words and documents in unsupervised manner. Word - document matrix is first constructed using standard `tf.idf` term weighting. Cosine similarity between documents is used. We focus on two sets of 5-newsgroup combinations listed below:

```
      A5:                          B5:
NG2:  comp.graphics          NG2:  comp.graphics
NG9:  rec.motorcycles        NG3:  comp.os.ms-windows
NG10: rec.sport.baseball     NG8:  rec.autos
NG15: sci.space              NG13: sci.electronics
NG18: talk.politics.mideast  NG19: talk.politics.misc
```

In A5, clusters overlap at medium level. In B5, clusters overlap substantially. Table 1 contains the results of MLHC. To accumulate sufficient statistics, for each newsgroup combination, we generate 10 datasets, each of which is a random sample of documents from the newsgroups (with 100 documents per newsgroup).

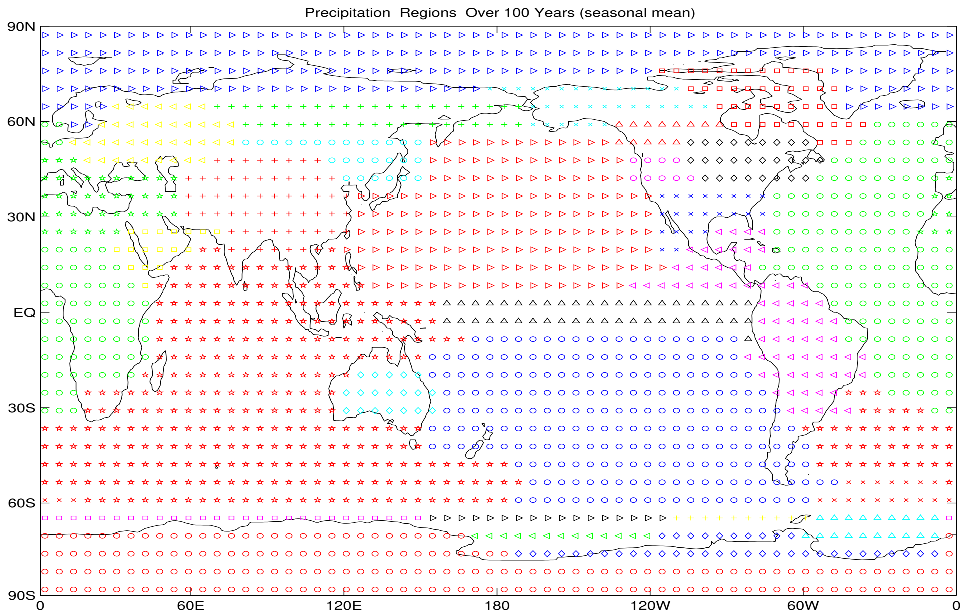

Fig. 3. Global precipitation pattern based on seasonal means over 100 years.

The results in the table are the average over these 10 random sampled datasets. For comparison purpose, we also run K-means clustering. For each dataset, we run 10 K-means clustering from random seeds for cluster centroids and selecte the best result as determined by the K-means objective function value. Results are given in Table 1. Note that because percentage consistency results are close to 1, we give inconsistency = 1 - consistency.

From Table 1, the MLHC results have better clustering accuracy (last column of Table 1) compared to K-means clustering. More important is MLHC always provides clustering with better kmn cluster membership consistency. For 1mn-consistency, MLHC is perfect since this is guaranteed by MLHC. With this, it is not surprising that MLHC has substantially better 1nn-consistency than K-means method, about half as smaller. In all categories, MLHC has better knn/kmn consistency than K-means .

6 Summary

In this paper, we propose a modification of the standard AHC algorithm that allow an order-N faster implementation. The modification is based on the recognition that all 1mn-pairs in each iteration of AHC can be merged if the linkage function satisfies the cluster aggregate inequality. This leads to the multi-level hierarchical clustering algorithm. Many commonly used linkage functions satisfy

	1nn	2nn	3nn	1mn	2mn	3mn	Accuracy
A5							
K-means	16.2	28.4	37.8	6.4	14.5	23.0	75.2%
MLHC	8.5	24.1	36.4	0	6.9	16.6	77.6%
B5							
K-means	23.1	39.4	50.6	8.5	21.6	32.8	56.3%
MLHC	10.2	28.9	45.0	0	9.3	21.3	60.7%

Table 1. Fractional knn and kmn inconsistency and clustering accuracy (last column) for newsgroup datasets A5 and B5. For dataset A5, 1nn inconsistency is 16.2% for K-means and 8.5% for MLHC.

this inequality and thus will benefit from this modification. We propose "close friends" cohesion as important feature of clustering and show that it is fully guarantees in the algorithm. This is further extended to cluster membership KNN consistency. Experiments on newsgroup show that kNN consistency is satisfied much better by MLHC than widely used algorithms such as K-means .

References

1. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, 2nd ed.* Wiley, 2000.
2. B. Fung, K. Wang, and M. Ester. Large hierarchical document clustering using frequent itemsets. *Proc. SIAM Data Mining Conf*, 2003.
3. T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning.* Springer Verlag, 2001.
4. A.K. Jain and R.C. Dubes. *Algorithms for clustering data.* Prentice Hall, 1988.
5. A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999.
6. S.Y. Jung and T.-S. Kim. An agglomerative hierarchical clustering using partial maximum array and incremental similarity computation method. *Proc. SIAM Conf. on Data Mining*, pages 265–272, 2001.
7. G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32:68–75, 1999.
8. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.
9. S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic Press, 1999.
10. E.M. Voorhees. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing and Management*, 22:465–476, 1986.
11. H Xiong, M. Steinbach, P-N. Tan, and V. Kumar. Hicap:hierarchial clustering with pattern preservation. *Proc. SIAM Data Mining Conf*, pages 279–290, 2004.
12. H. Yu, J. Yang, and J. Han. Classifying large data sets using svms with hierarchical clusters. In *Proc. ACM Int'l Conf Knowledge Disc. Data Mining (KDD)*, pages 306–315, 2003.
13. T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. *Proc. ACM Int'l Conf. Management of Data (SIGMOD)*, pages 103–114, 1996.