Optimal Implementations of UPGMA and Other Common Clustering Algorithms

Ilan Gronau Shlomo Moran

September 21, 2006

Abstract

We present an optimal $O(n^2)$-time algorithm, which uses only elementary data structures, for few common clustering algorithms including UPGMA. The correctness of our algorithm is proved by showing that the global-minimum selection rule in these algorithms can be replaced by a local-minimum selection rule.

Key Words: Hierachical clustering, UPGMA, design of algorithms, analysis of algorithms, computational complexity

1 Introduction

UPGMA (Unweighted Pair Grouping Method with Arithmetic-mean) is one of the simplest and most commonly used hierarchical clustering algorithms. It receives as input a set of elements and a dissimilarity matrix which contains pairwise distances[1] between all elements, and returns a *hierarchy* of clusters on this set (see [1], Chapter 3). It starts by initializing a singleton-cluster for each element in the set, and then follows the closest-pair hierarchical clustering scheme described in Table 1: two clusters of minimal distance from each other are selected and replaced by their union (a *joining event*); clustering then continues recursively on the reduced cluster-set. In each such iteration the distances from the new cluster to all other clusters are computed via the *reduction formula* in step 3. The reduction formula used by UPGMA defines the new distances as the arithmetic means of original distances (see Table 1). Notice that the scheme used by UPGMA is non-deterministic, since there could be more than one pair satisfying the selection criterion of step 2. Consequently, for certain inputs we may have more than one execution. Note also that the output of an execution is completely determined by the (unordered) set of joining events it performs (where two joining events are identical if they involve identical clusters).

Several other known algorithms, such as WPGMA [4] and the single linkage

[1] In the context of this paper, "distances" need not satisfy the triangle inequality.

Unweighted Pair Grouping Method with Arithmetic-mean:

Input: A dissimilarity matrix $D = [D(C_i, C_j)]$ over a set of clusters $\mathcal{C}$.
Output: A hierarchy over $\mathcal{C}$.

1. **Stopping condition:** If $|\mathcal{C}| = 1$ return the single cluster in $\mathcal{C}$.
2. **Cluster-pair selection:** Select a pair of distinct clusters $\{C_i, C_j\} \subseteq \mathcal{C}$ s.t. $D(C_i, C_j)$ is a minimal off-diagonal entry of D.
3. **Reduction:** Remove C_i, C_j from the cluster set $\mathcal{C}$ and replace them with $C_i \cup C_j$. For all $C_k \neq (C_i \cup C_j)$, set:
$$D(C_k, (C_i \cup C_j)) \leftarrow \frac{|C_i|}{|C_i|+|C_j|} D(C_k, C_i) + \frac{|C_j|}{|C_i|+|C_j|} D(C_k, C_j) .$$
– Recursively call UPGMA on the reduced cluster-set and dissimilarity matrix.
4. **Returning:** Add C_i and C_j to the returned hierarchy.

Table 1: The UPGMA algorithm

algorithm [1, 3], use this clustering scheme with different reduction formulae:

$$\textbf{WPGMA:} \quad D(C_k, (C_i \cup C_j)) \leftarrow \frac{1}{2}\left(D(C_k, C_i) + D(C_k, C_j)\right) \quad (1.1)$$

$$\textbf{Single-Linkage:} \quad D(C_k, (C_i \cup C_j)) \leftarrow \min\left\{D(C_k, C_i), D(C_k, C_j)\right\} \quad (1.2)$$

Other reduction formulae are possible as well. Those considered in this paper are assumed to be *convex*, meaning that for each cluster C_k, the value of $D(C_k, (C_i \cup C_j))$ lies between $D(C_k, C_i)$ and $D(C_k, C_j)$.

The naive implementation of UPGMA takes $O(n^3)$ time, and the most efficient implementation known today takes $O(n^2 log(n))$ time, using heaps or sorted arrays (see Section 2). This note presents a novel simple $O(n^2)$ implementation which uses only elementary data structure. This implementation is clearly asymptotically optimal, and is based on the technique we introduced in [2]. To the best of our knowledge, this is the first $O(n^2)$ implementation of UPGMA which is *faithful*, in the sense that it is guaranteed to produce clustering which corresponds to a correct execution of the UPGMA algorithm as described in Table 1.

Our implementation is based on a relaxation of the clustering scheme mentioned above: instead of selecting a *globally closest* cluster-pair C_i, C_j, s.t. $D(C_i, C_j)$ is a minimal off-diagonal entry in the entire matrix, we selcet a *locally closest* cluster pair, for which $D(C_i, C_j)$ is only required to be a minimal off-diagonal entry in the rows corresponding to C_i and C_j. We call this relaxed scheme the *locally closest pair* (LCP) scheme, as opposed to the original *globally*

closest pair (GCP) scheme. We show that in the case of the reduction formula used by UPGMA, the LCP scheme is equivalent to the GCP scheme, meaning that any implementation of LCP-UPGMA is a faithful implementation of GCP-UPGMA (as described in Table 1).

The rest of this note is organized as follows. In Section 2 we review various known implementations of UPGMA and present our $O(n^2)$ implementation of LCP-UPGMA. In Section 3 we prove the equivalence of the LCP and GCP schemes. Most of our analysis specifically considers UPGMA, but we state possible generalizations to other clustering algorithms as well.

2 An $O(n^2)$ Implementation of LCP-UPGMA

In this section we present an implementation of LCP-UPGMA whose time complexity is $O(n^2)$. The analysis in this section applies to all hierarchical clustering algorithms which follow the 'closest-pair' joining scheme, and which use a **convex reduction formula**, as will be detailed.

Given an input dissimilarity matrix D over a set $\mathcal{C}$ of n clusters, the UPGMA algorithm of Table 1 performs $n-1$ iterations (recursive calls). Each such iteration involves joining a cluster-pair and reducing the input matrix. It is easy to see that the reduction step can be implemented in linear time. Thus, the running time of the algorithm is dominated by the time required for selecting the cluster-pairs. A naive approach, which requires $\theta(n^2)$ time in **each iteration** (and a total time complexity of $\theta(n^3)$) scans the entire matrix D for a minimal off diagonal entry, corresponding to a globally closest cluster-pair.

Time complexity can be reduced to $O(n^2 log(n))$ as follows. Let $MIN_D(C_i) = \min_{C_k \neq C_i} D(C_i, C_k)$ denote the minimal off-diagonal value in the row corresponding to C_i in D. An **ordered** cluster-pair (C_i, C_j) is a *minimal pair* (for C_i and D) if $D(C_i, C_j) = MIN_D(C_i)$. Finding a minimal pair for each row in D can be done in $O(n^2)$ time. Once a minimal pair is kept for each row, a globally closest cluster-pair is found in linear time by scanning the set of minimal pairs and selecting a pair (C_i, C_j) for which $D(C_i, C_j)$ is minimized. Updating the set of minimal pairs after the reduction of D (in step 3 of the algorithm) can be done in $O(nlog(n))$ time by maintaining the entries in each row of D in a heap. This results in total time complexity of $\Theta(n^2 \log n)$.

As mentioned earlier, our $O(n^2)$ implementation is based on the LCP scheme, which joins at each stage a locally (rather than globally) closest cluster-pair.

Observation 2.1. *If (C_i, C_j) is a minimal pair and $MIN_D(C_i) = MIN_D(C_j)$, then $\{C_i, C_j\}$ is a locally closest cluster pair in D.*

Observation 2.1 implies that under the LCP scheme, cluster joining can be implemented by joining clusters C_i, C_j s.t. (C_i, C_j) is a minimal pair and $MIN_D(C_i) = MIN_D(C_j)$. To find such cluster pairs efficiently, we maintain a *complete descending path*. A sequence of distinct clusters $P = (C_{i_1}, C_{i_2}, \ldots, C_{i_l})$ is a *descending path* with respect to D if for $r = 1, .., l-1$, $(C_{i_r}, C_{i_{r+1}})$ are minimal pairs, implying also that $MIN_D(C_{i_r}) \geq MIN_D(C_{i_{r+1}})$. A descending

path is *complete* if D (and hence P) are of dimension 1, or if $MIN_D(C_{i_{l-1}}) = MIN_D(C_{i_l})$, that is: $(C_{i_{r-1}}, C_{i_r})$, the last clsuter pair in P, is a locally closest pair. Thus, constructing and maintaining a complete descending path throughout the execution of the algorithm in overall $O(n^2)$ time will imply the desired bound on the total time complexity.

Our method is based on the following *basic extension operation*, defined for a given descending path P w.r.t. a dissimiarity matrix D: if P is the empty path then insert to P an arbitrary cluster $C_{i_1} \in \mathcal{C}$. Else let $P = (C_{i_1}, \ldots, C_{i_l})$; compute $m = MIN_D(C_{i_l})$; if $l > 1$ and $m = D(C_{i_{l-1}}, C_{i_l})$ then terminate extension; otherwise (i.e. $l = 1$ or $m < D(C_{i_{l-1}}, C_{i_l})$), extend the path P by adding to it any cluster $C_{i_{l+1}}$, s.t. $D(C_{i_l}, C_{i_{l+1}}) = m$. Observe that a repeated application of this basic extension operation must end when the termination condition holds, and a **complete** descending path is achieved.

Given an input matrix D, the algorithm starts by constructing a complete descending path w.r.t. D as described above. Given a complete descending path $P = (C_{i_1}, \ldots, C_{i_l})$ of D, if $l = 1$ the algorithm stop (since D is of dimension 1). Else the algorithm performs a reduction step in which the cluster-pair $(C_{i_{l-1}}, C_{i_l})$ is joined. Let D' be the matrix obtained by this reduction. We observe that if the reduction is convex, then the path $\bar{P} = (C_{i_1}, \ldots, C_{i_{l-2}})$ is a (possibly empty) descending path of the reduced matrix D'. This holds since the convexity of the reduction guarantees that all consecutive pairs in $\bar{P}$ remain minimal pairs with respect to D'. Thus a complete descending path P' can be computed for D' by iteratively extending $\bar{P}$ using basic extension operations until the termination condition is met.

We now analyze the total time complexity of the process described above. This process consists of a series of basic extension operations, some of which lead to termination, whereas the rest lead to an extension of P by an additional vertex (cluster). Each operation requires the computation of $MIN_D(C_i)$ (for some cluster C_i), which can be done in linear time. Thus, the total time complexity of maintaining P is determined by the total number of basic extension operations invoked throughout the execution of the algorithm. $n-1$ such operations lead to termination of the path P (one in each iteration), whereas the rest result in an extension of P by a single vertex. Now, since in each iteration only two vertices are removed from P, and at the end of the execution this path is emptied (up to a single vertex), the total number of vertices added to P throughout the execution is no more than $2n - 2$. Thus the total number of basic extension operations is no more than $3n - 3$, leading to a total running time of $O(n^2)$.

3 Equivalence of the LCP and GCP schemes

The *complete descending paths* technique presented in the previous section yields an $O(n^2)$ implementation of LCP-UPGMA. In this section we prove that for each LCP-UPGMA execution there is a GCP-UPGMA execution on the same input which yields identical clustering (Theorem 3.3). This implies that the $O(n^2)$ algorithm presented in the previous section is a faithful (i.e., correct)

implementation of UPGMA. Although our discussion focuses on UPGMA, it can be easily generalized for several other hierarchical clustering algorithms which follow the 'closest-pair' joining scheme (such as WPGMA and the single-linkage algorithm). The analysis is based on the following two lemmas:

Lemma 3.1. *Let D be a dissimilarity matrix over a cluster set $\mathcal{C}$, s.t. $|\mathcal{C}| > 1$. Then during each execution of LCP-UPGMA on $(\mathcal{C}, D)$, a cluster-pair $\{C_i, C_j\} \subseteq \mathcal{C}$, which is a globally closest pair in D, is joined.*

Proof. The lemma is proved by induction on $|\mathcal{C}|$. It holds trivially when $|\mathcal{C}| = 2$. Assume that $|\mathcal{C}| > 2$, and let m denote the minimal off-diagonal value in D. Consider the first cluster-pair $\{C_k, C_l\}$ joined during the execution, and distinguish between the two complementary cases:
Case 1: $\min\{MIN_D(C_k), MIN_D(C_l)\} = m$. Then we must have that $D(C_k, C_l) = m$, and the lemma follows.
Case 2: $\min\{MIN_D(C_k), MIN_D(C_l)\} > m$, i.e. all the off-diagonal values in the rows corresponding to C_k, C_l are strictly greater than m. Let $\mathcal{C}' = \mathcal{C} \setminus \{C_k, C_l\} \cup \{(C_k \cup C_l)\}$ be the reduced cluster-set, and let D' denote the corresponding dissimilarity matrix obtained after the first iteration of the algorithm. Since the reduction is convex, we have that $MIN_{D'}(C_i \cup C_j) > m$, hence the minimum off-diagonal value in D' is also m. Moreover, A cluster pair $\{C_{i'}, C_{j'}\} \subseteq \mathcal{C}'$ is a globally closest pair in D' iff it is a globally closest pair in D. The induction hypothesis on $\mathcal{C}'$ implies that at some point in the execution, a globally closest cluster-pair $\{C_{i'}, C_{j'}\} \subseteq \mathcal{C}'$ must be joined. By the written above, $\{C_{i'}, C_{j'}\}$ is also a globally closest pair in D. □

Note that Lemma 3.1 applies for all hierarchical clustering algorithms which follow the 'locally-closest-pair' joining scheme and which use a convex reduction.

Lemma 3.2 (Swapping Lemma)**.** *Let D be a dissimilarity matrix over a cluster set $\mathcal{C}$ s.t. $|\mathcal{C}| \geq 4$, and assume $\{C_{i_1}, C_{j_1}, C_{i_2}, C_{j_2}\} \subseteq \mathcal{C}$. Let $\mathcal{C}' = \mathcal{C} \setminus \{C_{i_1}, C_{j_1}, C_{i_2}, C_{j_2}\} \cup \{(C_{i_1} \cup C_{j_1}), (C_{i_2} \cup C_{j_2})\}$ be the smaller cluster set obtained by joining cluster-pairs $\{C_{i_1}, C_{j_1}\}$ and $\{C_{i_2}, C_{j_2}\}$. Let further D'_1 be the dissimilarity matrix over $\mathcal{C}'$ resulting from first joining cluster-pair $\{C_{i_1}, C_{j_1}\}$ and then joining $\{C_{i_2}, C_{j_2}\}$, and let D'_2 be the matrix which results from first joining $\{C_{i_2}, C_{j_2}\}$ and then $\{C_{i_1}, C_{j_1}\}$, where all reductions use the UPGMA-formula. Then $D'_1 = D'_2$.*

Proof. Denote $\alpha_1 = \frac{|C_{i_1}|}{|C_{i_1}|+|C_{j_1}|}$ and $\alpha_2 = \frac{|C_{i_2}|}{|C_{i_2}|+|C_{j_2}|}$. By substitution in the reduction formula of UPGMA (see Step 3 in Table 1) we get for an arbitrary cluster-pair $A, B \in \mathcal{C}' \setminus \{(C_{i_1} \cup C_{j_1}), (C_{i_2} \cup C_{j_2})\}$:

$$\begin{aligned}
D'_1(A, B) = &\quad D(A, B) &&= D'_2(A, B) \\
D'_1(A, (C_{i_1} \cup C_{j_1})) = &\quad \alpha_1 D(A, C_{i_1}) + (1 - \alpha_1) D(A, C_{j_1}) &&= D'_2(A, (C_{i_1} \cup C_{j_1})) \\
D'_1(A, (C_{i_2} \cup C_{j_2})) = &\quad \alpha_2 D(A, C_{i_2}) + (1 - \alpha_2) D(A, C_{j_2}) &&= D'_2(A, (C_{i_2} \cup C_{j_2}))
\end{aligned}$$

By substitution in the same formula we get for $(C_{i_1} \cup C_{j_1}), (C_{i_2} \cup C_{j_2})$:

$$\begin{aligned}
&D'_1((C_{i_1} \cup C_{j_1}), (C_{i_2} \cup C_{j_2})) = \\
&\alpha_1\alpha_2 D(C_{i_1}, C_{i_2}) \quad + \quad (1-\alpha_1)\alpha_2 D(C_{j_1}, C_{i_2}) + \\
&\alpha_1(1-\alpha_2) D(C_{i_1}, C_{j_2}) \quad + \quad (1-\alpha_1)(1-\alpha_2) D(C_{j_1}, C_{j_2}) = \\
&D'_2((C_{i_1} \cup C_{j_1}), (C_{i_2} \cup C_{j_2}))
\end{aligned}$$

□

Lemmas 3.1 and 3.2 imply our main result:

Theorem 3.3. *Let D be an arbitrary dissimilarity matrix over a cluster-set $\mathcal{C}$. Then every LCP-UPGMA execution on $(\mathcal{C}, D)$ has an equivalent GCP-UPGMA execution (yielding the same clustering as output).*

Proof. By induction on $|\mathcal{C}|$. The claim holds for $|\mathcal{C}| \leq 3$, since in such a case, a locally-closest cluster-pair is globally-closest as well. Assume, therefore, that $|\mathcal{C}| > 3$, and let LE be an LCP-UPGMA execution on $(\mathcal{C}, D)$. Let $\{C_i, C_j\} \subseteq \mathcal{C}$ be a globally closest cluster-pair in D which is joined during LE, as guaranteed by Lemma 3.1. By a repeated application of the swapping lemma, we can move up the joining of this cluster-pair to the beginning of LE, without changing the output of the execution. Thus we can assume that the first cluster-pair joined by the execution LE is $\{C_i, C_j\}$. Observe LE', the suffix execution of LE defined on the reduced cluster set $\mathcal{C}' = \mathcal{C} \setminus \{C_i, C_j\} \cup \{(C_i \cup C_j)\}$ and the corresponding reduced matrix D'. The induction hypothesis implies that there is a GCP-UPGMA execution GE' on $(\mathcal{C}', D')$ equivalent to LE'. By appending GE' to the joining of $\{C_i, C_j\}$ we get a GCP-UPGMA execution GE equivalent to LE. □

We note that Theorem 3.3 applies to any 'closest-pair' hierarchical clustering algorithm, providing that the reduction formula it uses is **convex** and **satisfies the swapping lemma** (this in particular includes WPGMA and the single linkage algorithm). Hence, each such clustering algorithm has an optimal $O(n^2)$ faithful implementation using complete descending paths.

References

[1] J. Barthelemy and A. Guenoche. *Trees and proximities representations.* Wiley, 1991.

[2] I. Gronau and S. Moran. Neighbor joining algorithms for inferring phylogenies via LCA-distances. September 2006.

[3] M. Krivanek. The complexity of ultrametric partitions on graphs. *Inform. Process. Lett.*, 27:265–270, 1988.

[4] P. Sneath and R. Sokal. *Numerical Taxonomy : the principles and practice of numerical classification.* W. H. Freeman, San Francisco, 1973.