Approximation Bounds for Hierarchical Clustering: Average Linkage, Bisecting K-means, and Local Search

Ben Moseley (CMU), Joshua R. Wang (Stanford)

Hierarchical Clustering
- Paradigm: clusters form a hierarchy.
- Key Example: Biological Taxonomy.
- Two types of algorithms in common use.
- “Agglomerative” (bottom-up) algorithms begin with each point in its own cluster, and try to merge similar clusters.
- Example: Average linkage picks the two clusters with the biggest average (all-pairs) similarity.
- “Divisive” (top-down) algorithms begin with all points in one cluster, and try to recursively split into dissimilar clusters.
- Example: Bisecting K-means uses Lloyd’s algorithm to divide clusters.
- Question: Can theory provide guidance about which algorithm is “best” for a given application?

Dasgupta’s Objective Function [Dasgupta ’16]
- Input: weighted graph $G = (V,E)$ with weights $w_e$ (similarity).
- Minimize: $\text{cost}_G(T) = \sum_{i,j \in V} w_{ij}[\text{leaves}(T_{ij})]$. 
- Fact: Optimal tree is binary for any data set.
- Fact: When the input is a line graph, the optimal tree recursively splits the graph into equally sized line graphs (when viewed top-down).
- Fact: All binary trees score the same when the input is a clique (with equal weights on all edges).
- [Dasgupta ’16]: NP-hard to compute exactly.
- [Charikar-Chatsifrazitis ’17]: Divisive algorithm based on recursive sparse cut computations, $O(\sqrt{\log n})$-approximation.
- [Charikar-Chatsifrazitis ’17]: Hard to approximate within any constant factor assuming Small-Set Expansion Hypothesis.

Graph + Clustering + Sample Calculation

A New Approach to Dasgupta’s Program
- We study a closely related objective function.
- Maximize: $\text{rew}_G(T) = \sum_{i \in V} w_{ii}[\text{nonleaves}(T_{ii})]$.
- The sum of cost and reward is an invariant based on the graph, so our objective has the same optimal tree.

$\text{cost}_G(T) + \text{rew}_G(T) = \sum_{i,j \in V} w_{ij} \cdot n$

Proof of Theorem 1
- Focus: single iteration of average linkage; just merged clusters $A$ and $B$. The remainder of the graph is $C$.
- What is the contribution towards the final revenue?
  $\text{merge-rev}_{G}(A,B) = |C| \sum_{a \in A,c \in C} w_{ac}$
- What is the contribution towards the final cost?
  $\text{merge-cost}_{G}(A,B) = |B| \sum_{a \in A,c \in C} w_{ac} + |A| \sum_{b \in B,c \in C} w_{bc}$
- Insight: rewrite expressions in terms of average linkage.
  $\text{merge-rev}_{G}(A,B) = \frac{|A||B||C|}{|A||B|} \sum_{a \in A,c \in C} w_{ac}$
  $\text{merge-cost}_{G}(A,B) = \frac{|A||B||C|}{|A||B|} \sum_{a \in A,c \in C} w_{ac} + \frac{|A||B||C|}{|B||C|} \sum_{b \in B,c \in C} w_{bc}$
- Upshot: rewrite merges in terms of average linkage.
- Sum over all iterations:
  $\text{rew}_G(T) \geq 1/2 \cdot \text{cost}_G(T) $
  $\text{rew}_G(T) + \text{cost}_G(T) = n \sum_{i,j \in V} w_{ij}$
  $\text{rew}_G(T) \leq n \sum_{i,j \in V} w_{ij}$
- By construction, $A$ and $B$ are more similar than any other pair of clusters.

Proof of Theorem 2
- Starting Point: Star Graph.
- Merging interior node with leaves one-by-one:
  $(n-1) + (n-2) + \cdots + 1 = \frac{n(n+1)}{2} = 0.5n^2$ reward.
- Take #1: two star graphs, connect interior nodes with valuable edge.
- Average Linkage: merges valuable edge first, then left with a normal star graph. Scores roughly $0.5n^2$.
- Better Solution: separately cluster the two star graphs. Scores $2 \times \left( (n-2) + (n-3) + \cdots + n/2 \right)$ = $0.75n^2$.
- Take #2: $k$ star graphs, connect interior nodes with valuable clique.
- Average Linkage: Again, scores roughly $0.5n^2$.
- Better Solution: Approaches a score of $n^2$.

Main Results
- Theorem 1: Average Linkage is a $1/3$-apx for $\text{rew}_G(T)$.
- Theorem 2: Average Linkage is not a $(1/2 + \epsilon)$-apx for $\text{rew}_G(T)$.
- Theorem 3: Bisecting K-Means is a $1/\Omega(n)$-apx for $\text{rew}_G(T)$.
- Theorem 4: There are divisive algorithms that are $1/3$-apx for $\text{rew}_G(T)$.
- Upshot: Shows you when to use Average Linkage over Bisecting K-Means.

Average Linkage
- Data: Vertices $V$, weights $w : E \rightarrow \mathbb{R}^+$
- Initialize clusters $C = \cup_{\text{leaves}(V)}$:
- while $|C| \geq 2$ do
  Choose $A, B \in C$ to maximize $\text{rew}(A, B) = \sum_{a \in A, b \in B} \sum_{c \in C} w_{ac} \cdot \text{leaves}(T_{ac})$.
  Set $C \leftarrow C \cup \{A \cup B\} \setminus \{A, B\}$
- end

Counterexample Graph, Take #1
- $\frac{n}{2}$ nodes
- $1 + \delta$
- $\frac{n}{2}$ nodes

Counterexample Graph, Take #2
- $\frac{n}{k}$ nodes
- $k$-Clique with $1 + \delta$ edge weights
- $\frac{n}{k}$ nodes

Situation Immediately Before an Iteration of Average Linkage

$A \rightarrow B$

$C_k$

$C_2$

$C_1$