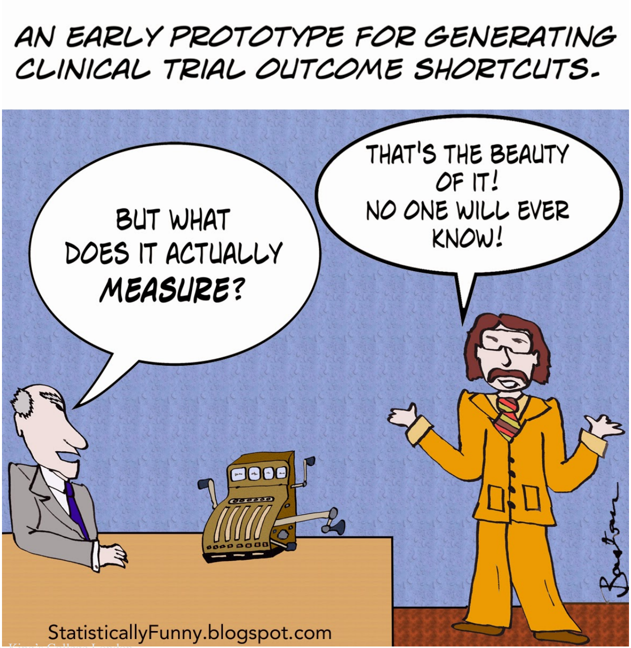
Hierarchical Clustering Objective

Frederik Mallmann-Trenn
6CCS3AIN

What are the hierarchical algorithms actually doing?



What quantity are these algorithms optimizing?

- For flat clustering, algorithms designed to optimize some objective function
  - Remember:
    for **flat clustering** the goal was to find k points $\mu_1, \ldots, \mu_k$ that minimize, e.g.

    1. k-median objective $\sum_{i=1}^{N} \left(\min_{j \in [k]} d(\mathbf{x}_i, \mu_j) \right)$
    2. k-means objective $\sum_{i=1}^{N} \left(\min_{j \in [k]} d(\mathbf{x}_i, \mu_j) \right)^2$
- For hierarchical clustering, algorithms have been studied procedurally
  - Thus, comparisons between hierarchical clustering algorithms are only qualitative!

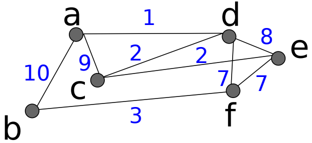
What quantity are these algorithms optimizing?

- For flat clustering, algorithms designed to optimize some objective function
  - Remember:
    for **flat clustering** the goal was to find k points $\mu_1, \ldots, \mu_k$ that minimize, e.g.
    1. k-median objective $\sum_{i=1}^{N} \left(\min_{j \in [k]} d(\mathbf{x}_i, \mu_j) \right)$
    2. k-means objective $\sum_{i=1}^{N} \left(\min_{j \in [k]} d(\mathbf{x}_i, \mu_j) \right)^2$
- For hierarchical clustering, algorithms have been studied procedurally
  - Thus, comparisons between hierarchical clustering algorithms are only qualitative!
- [Dasgupta '16]
  "The lack of an objective function has prevented a theoretical understanding"
- Dasgupta introduced an objective function to model the hierarchical clustering problem

Dasgupta's Cost Function

Input: a weighted similarity graph G

- Edge weights represent similarities



Output: T a tree with leaves labelled by nodes of G

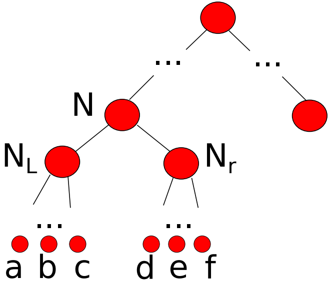
Cost of the output: Sum of the costs of the nodes of T

Cost of a node N of the tree:

$L = \{u \mid u \text{ is leaf of subtree rooted at } N_L\}$

$R = \{v \mid v \text{ is leaf of subtree rooted at } N_R\}$

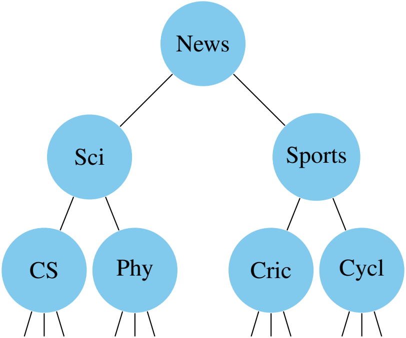
$$\boxed{\text{cost}(N) = (|L| + |R|) \cdot \sum_{\substack{u \in L \\ v \in R}} \text{similarity}(u, v)}$$



The total cost is the sum of costs of all subtrees.

Intuition: Better to cut a high similarity edge at a lower level

Results

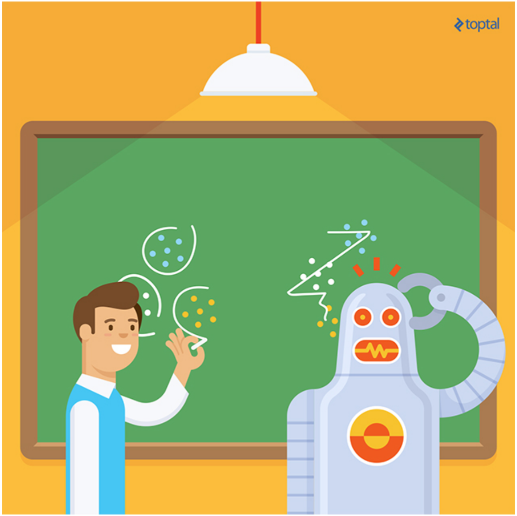


Hierarchical Clustering: Objective Functions and Algorithms
Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, Claire Mathieu
JACM 2019

Result 1: Is Dasgupta the only reasonable function?

- We characterize the set of ‘good’ objective functions based on axioms.
  - Disconnected components must be separated first
  - Symmetry
  - Only the ‘true’ hierarchical clustering should minimize the cost function (if there is one).
- In some sense Dasgupta is the most natural

Result 2: Algorithms



Hope: Recursive Sparsest Cut

- Dissimilarity graph:
  - We show Avg. Linkage has a $3/2$-approximation factor
  - We also show that other practical algorithms have a $\Omega(n^{1/4})$-approximation factor

Hope: Recursive Sparsest Cut

- Dissimilarity graph:
  - We show Avg. Linkage has a $3/2$-approximation factor
  - We also show that other practical algorithms have a $\Omega(n^{1/4})$-approximation factor
- Similarity graph:

Algorithm: Recursive Sparsest Cut

Input: Weighted graph $G = (V, E, w)$

$\{A, V \setminus A\} \leftarrow$ cut with sparsity $\leqslant \phi \cdot \min_{S \subseteq V} \dfrac{w(S, V \setminus S)}{|S| \cdot |V \setminus S|}$

Recurse on subgraphs $G[A]$, $G[V \setminus A]$ to obtain trees T_A, $T_{V \setminus A}$

Output: Return tree whose root has subtrees T_A, $T_{V \setminus A}$

Hope: Recursive Sparsest Cut

- Dissimilarity graph:
  - We show Avg. Linkage has a $3/2$-approximation factor
  - We also show that other practical algorithms have a $\Omega(n^{1/4})$-approximation factor
- Similarity graph:

Algorithm: Recursive Sparsest Cut

Input: Weighted graph $G = (V, E, w)$

$\{A, V \setminus A\} \leftarrow$ cut with sparsity $\leqslant \phi \cdot \min_{S \subseteq V} \frac{w(S, V \setminus S)}{|S| \cdot |V \setminus S|}$

Recurse on subgraphs $G[A], G[V \setminus A]$ to obtain trees $T_A, T_{V \setminus A}$

Output: Return tree whose root has subtrees $T_A, T_{V \setminus A}$

  - We show $O(\phi)$-approximation
  - Improves the $O(\log n \cdot \phi)$-approximation [Dasgupta '16]
  - Current best known value for ϕ is $O(\sqrt{\log n})$ [ARV '09]

- For worst case inputs, Recursive Sparsest Cut gives $O(\phi)$-approximation
- Assuming the "Small Set Expansion Hypothesis", no polytime $O(1)$-approx.

Real-world graphs are often not worst-case