Hierarchical Clustering: Objective Functions and Algorithms

Arne Nix
arne.nix@rwth-aachen.de

1 Introduction

1.1 Motivation

The goal of *hierarchical clustering* is to partition a set of datapoints into successively smaller clusters. Compared to *centroid-based clustering* like e.g. k-means clustering, hierarchical clustering has the key advantage that there is no need to specify the number of clusters in advance. A hierarchical clustering algorithm generates a cluster tree describing the partition steps at each level simultaneously and thus capturing the cluster structure at all levels of granularity, which is another benefit of hierarchical clustering.

There are several well established algorithms for hierarchical clustering, like e.g. single linkage, average linkage and complete linkage (Fionn and Pedro, 2011). However, those algorithms are specified procedurally without an underlying objective function. This means in particular that the output, i.e. the cluster tree, produced by such an algorithm is not defined in a precise way. An objective function has the further advantage that it enables a direct comparison of different algorithms in terms of complexity and efficency. Finally, an objective function is also very useful to incorporate constraints or prior information in the algorithm.

1.2 Problem Description

Following Dasgupta (Dasgupta, 2016), assume the input to a clustering problem to be a similarity graph[1] $G = (V, E, w)$ (see figure 1) with vertices V, undirected edges $E \subseteq \{\{u, v\} | u, v \in V\}$ and a weight function $w : E \to \mathbb{R}_+$. The weight function $w_{ij} = w(v_i, v_j)$ captures the degree of similarity between the nodes $v_i, v_j \in V$.[2] The output of a clustering algorithm is a clustering tree T (see figure 2) which has internal nodes $\mathcal{N}$ and exactly $|V|$ leaves $\mathcal{L}$, each corresponding to a distinct vertex $v \in V$. Such a cluster tree specifies a clustering at each level of horizontal cuts through the tree, where the leaves of each subtree resulting from this cut define one cluster.

The goal of an algorithm, which is defined according to an objective function Γ, is to minimize the cost $\Gamma(T)$ of the generated clustering tree.

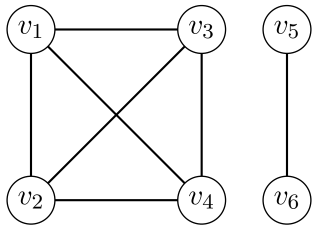

Figure 1: Example of a similarity graph

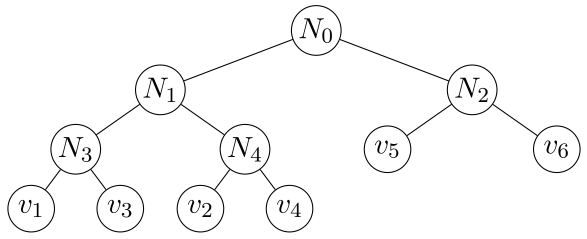

Figure 2: Example of a (generating) cluster tree corresponding to the graph in figure 1

1.3 Literature

Dasgupta (Dasgupta, 2016) frames similarity-based hierarchical clustering as a combinatorial optimization problem, where a "good" clustering minimizes a specified cost function. He introduces a simple and intuitive cost function and proposes the *Recursive ϕ-Sparsest-Cut* clustering algorithm for which he shows that it optimizes the cost function.

The work by Cohen-Addad et al. (Cohen-Addad et al., 2018) continues Dasgupta's efforts to formalize hierarchical clustering. They develop a general framework for cost functions in hierarchical clustering and formally define the idea of a "good" cost function, by introducing the term of an admissible cost function. The authors further propose several clustering algorithms and investigate the behavior of those, as well as some existing algorithms, based on the defined framework.

[1] Most of the following statements have a symmetric result for dissimilarity graphs as input.

[2] $\{v_i, v_j\} \notin E$ is equivalent to $w(v_i, v_j) = 0$.

1.4 Outline

This work aims to give a short overview of the topics discussed in (Cohen-Addad et al., 2018) and (Dasgupta, 2016). Section 2 gives an introduction to the framework of objective functions for hierarchical clustering introduced by Cohen-Addad and explains the concept of an admissible cost function in section 2.1. In section 3 the *recursive ϕ-sparsest-cut* and the *fast and simple* clustering algorithms are explained and proven in detail.

2 A Cost Function for Cluster Trees

The objective function defined in the following, which will finally be optimized by applying one of the clustering algorithms in section 3 is supposed to give a low cost for "good" clusterings. Recall the assumption from section 1.2 that a clustering is performed on the vertices of a similarity graph G. Therefore, each clustering step corresponds to cutting a set of edges in G. As stated by Dasgupta (Dasgupta, 2016), a "good" clustering should intuitively separate, i.e. be cutting the corresponding edges of, similar vertices as far down in the cluster tree as possible. Cosidering this requirement, Cohen-Adda et al. (Cohen-Addad et al., 2018) give a defintion for a general cost function:

Definition 1 (Cost Function) *For cluster tree T:*

$$\Gamma(T) = \sum_{N \in \mathcal{N}} \gamma(N)$$

For an individual internal node $N \in \mathcal{N}$:

$$\gamma(N) = \left(\sum_{\substack{u \in L(N_l),\\ v \in L(N_r)}} w(u,v) \right) \cdot g(|L(N_l)|, |L(N_r)|)$$

where N_l and N_r are the left and right children of N and $L(N)$ gives the leaves of the subtree induced by N.

By choosing for example $g(a,b) := a + b$, the cost of an internal node N is the number of nodes following below N times the sum of weights between the vertices in the right and left sub-cluster partitioned by N. This can be rewritten in the following way, to obtain the cost function proposed by Dasgupta:

$$\begin{aligned}\Gamma(T) &= \sum_{N \in \mathcal{N}} \sum_{\substack{u \in L(N_\text{l}),\\ v \in L(N_\text{r})}} w(u,v) \cdot (|L(N_\text{l})| + |L(N_\text{r})|) \\ &= \sum_{\{u,v\} \in E} w(u,v) \cdot |L(\text{LCA}_T(u,v))| \\ &= \sum_{\{u,v\} \in E} \gamma_E(u,v)\end{aligned}$$

Where $\text{LCA}_T(N_i, N_j)$ is the lowest common ancestor of $N_i, N_j \in \mathcal{N} \cup \mathcal{L}$ in T and $\gamma_E(u,v)$ defines an edge-wise charge for each $\{u,v\} \in E$.

2.1 Admissible Cost Function

Cohen-Addad et al. (Cohen-Addad et al., 2018) further characterize "good" objective functions by utilizing the properties of Ultrametrics:

Definition 2 (Ultrametric) *An ultrametric is a metric space (X,d) with*

$$d(x,y) \le \max\{d(x,z), d(y,z)\} \; \forall x,y,z \in X$$

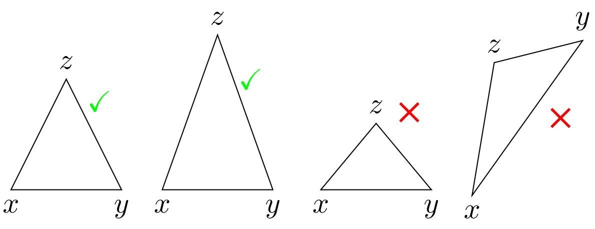

Figure 3: The triangles 3 and 4 violate the condition $d(x,y) \le \max\{d(x,z), d(y,z)\}$

Based on this definition, they state that G is generated from an ultrametric (X,d) if $V \subseteq X$ and every edge $e = \{x,y\}$ has $w(e) = f(d(x,y))$ for some non-decreasing function $f : \mathbb{R}_+ \to \mathbb{R}_+$. Such a graph G is called a *ground-truth input*. The cluster tree T corresponding to such a similarity graph G would be a *generating tree*, defined as follows:

Definition 3 (Generating Tree) *Given $G = (V,E,w)$ generated by minimal generating ultrametric (V,d), a* generating tree *is a rooted binary tree T with $|V|$ leaves $\mathcal{L}$, $|V|-1$ internal nodes $\mathcal{N}$ and a bijection $\sigma : \mathcal{L} \to V$ between the leaves of T and the nodes of V fulfilling the following condition:*

There exists a weight function $W : \mathcal{N} \to \mathbb{R}_+$ s.t. $W(N_i) \le W(N_j)$ for $N_i, N_j \in \mathcal{N}$ if N_i appears on the path from N_j to the root and

$w(\{v_i, v_j\}) = W(LCA_T(\sigma^{-1}(v_i), \sigma^{-1}(v_j)))$ *for every* $v_i, v_j \in V$.

The main advantage of defining a generating tree in this way is the fact that points in the same cluster are at least as similar to each other as they are to points outside this clusters. This fact is guaranteed by the definition of the weight function W and the ultrametric properties of the similarity graph G. This is obvioulsy a desirable property for a cluster tree. Therefore, Cohen-Addad et al. declare a cost function to be "good" (admissible) if it perfers such trees.

Definition 4 (Admissible Cost Function) *A cost function Γ is admissible if a cluster tree T for G achieves minimum cost if and only if it is a generating tree for G.*

A simplified characterization of admissible cost functions is also presented by Cohen-Addad et al. in form of a therom, proven in the long version of their paper.

Theorem 1 *A cost function Γ is admissible if and only if it satifies the follwoing conditions:*

1. *Let G be a clique, i.e. $w(u, v) = 1 \forall u, v \in V$, then $\Gamma(T) = \Gamma(T')$ for all cluster trees T, T' of G*
2. *Symmetry: $g(a, b) = g(b, a)\ \forall a, b \in \mathbb{N}$*
3. *Monotonicity: $g(a+1, b) > g(a, b) \forall a, b \in \mathbb{N}$*

3 Hierarchical Clustering Algorithms

3.1 Clustering Arbitrary Input Graphs

As mentioned above, the clustering of a similarity graph input can be seen as successively cutting edges until all vertices are disconnected. The intuition of the following clustering algorithm, introduced by Dasgupta (Dasgupta, 2016), is to minimize the sparsity $sp(\{A, V\backslash A\}) := \frac{w(A, V\backslash A)}{|A||V\backslash A|}$ in each of these cuts. Since finding the sparsest cut is known to be NP-hard, it is necessary to use an approximation algorithm for this problem. This is possible in $\mathcal{O}(\sqrt{\log(|V|)})$, as shown by Arora et al. in 2009 (Arora et al., 2009). The cluster tree resulting from algorithm 1 describes a ϕ-approximation of the sparsest cut in each cluster split, i.e. at each of the internal nodes $N \in \mathcal{N}$. Cohen-Addad et al. (Cohen-Addad et al., 2018) show the following theorem and subsequent proof for the optimality of the resulting cluster tree, considering the cost function defined by Dasgupta.

Algorithm 1 Recursive ϕ-Sparsest-Cut Clustering

```
1: Input: $G = (V, E, w)$
2: Find $\{A, V\backslash A\}$ with
       $sp(\{A, V\backslash A\}) \leq \phi \cdot \min_{S \subset V} sp\{S, V\backslash S\})$
3: Repeat recursively on $G[A]$ and $G[V\backslash A]$ to
   obtain trees $T_A$ and $T_{V\backslash A}$
4: Return: union of $T_A$ and $T_{V\backslash A}$
```

Theorem 2 *For any graph $G = (V, E, w)$ with $w : E \to \mathbb{R}_+$, the ϕ-sparsest-cut algorithm outputs a solution T of cost $\Gamma(T) \leq \frac{27}{4}\phi\Gamma(T^*)$ for optimal clustering T^**

The goal is to compare the cost of the cluster tree T resulting from algorithm 1 with an arbitrary tree T^* (notably including the optimal cluster tree). First, define the balanced cut $(L(N_{\text{BC}}), V\backslash L(N_{\text{BC}}))$ of T^* as the cut induced by N_{BC}. Where N_{BC} is the first node with $L(N_{\text{BC}}) < \frac{2n}{3}$ that is found in T^* by descending from the root and always taking the side with more leaves.

Based on the balanced cut, pick $A, B, C, D \subseteq V$ s.t. $(A \cup B, C \cup D) := (L(N_{\text{BC}}), V\backslash L(N_{\text{BC}}))$ is the balanced cut in T^* and $(A \cup C, B \cup D) := (L(N_0), V\backslash L(N_0))$ the cut induced by the root of T (compare figures 4 and 5).

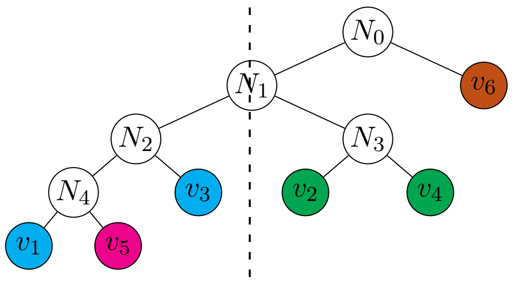

Figure 4: Cut of T^* ($N_1 = N_{\text{BC}}$)

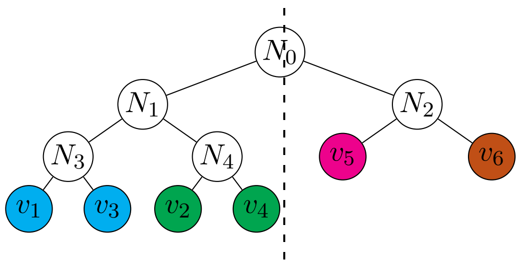

Figure 5: Cut of T

Since T is the output of Algorithm 1, the cut $(A \cup C, B \cup D)$ which is induced by the root of T

is a ϕ-approximate sparsest cut:

$$\frac{w(A \cup C, B \cup D)}{|A \cup C| \cdot |B \cup D|} \leq \phi \frac{w(A \cup B, C \cup D)}{|A \cup B| \cdot |C \cup D|}$$

The weight between $A \cup B$ and $C \cup D$ is at most as large as the summed weights between all the subsets, which leads to:

$$\begin{aligned} \Rightarrow & w(A \cup C, B \cup D) \\ \leq & \phi \frac{|A \cup C| \cdot |B \cup D|}{|A \cup B| \cdot |C \cup D|} \\ & \cdot (w(A, C) + w(A, D) + w(B, C) + w(B, D)) \end{aligned}$$

By definition of the balanced cut, the corresponding subsets must be of size $\frac{n}{3} \leq |A \cup B| \leq \frac{2n}{3}$ and $\frac{n}{3} \leq |C \cup D| \leq \frac{2n}{3}$. Since $|A \cup B| + |C \cup D| = n$, it is possible to compute the lower-bound $|A \cup B| \cdot |C \cup D| \geq \frac{2n^2}{9}$. Using this in the denominator of the inequality above, achieves:

$$\begin{aligned} \leq & \phi \frac{9}{2n^2} |A \cup C| \cdot |B \cup D| \\ & \cdot (w(A, C) + w(A, D) + w(B, C) + w(B, D)) \end{aligned}$$

Exploiting the fact that $\frac{|X|}{n} \leq 1$ for any $X \subseteq V$:

$$\begin{aligned} \leq & \phi \frac{9}{2} \left(\frac{|B \cup D|}{n} w(A, C) + w(A, D) \right. \\ & \left. + w(B, C) + \frac{|A \cup C|}{n} w(B, D) \right) \end{aligned}$$

Thus, the cost induced by the root N_0 satisfies:

$$\begin{aligned} \gamma(N_0) = & n w(A \cup C, B \cup D) \\ \leq & \frac{9}{2} \phi |A \cup C| w(B, D) \\ & + \frac{9}{2} \phi |B \cup D| w(A, C) \\ & + \frac{9}{2} n \phi [w(A, D) + w(B, C)] \end{aligned}$$

Therefore, to account for $\gamma(N_0)$, one must charge:

- $\frac{9}{2} \phi |B \cup D| w(e)$ to each edge e of (A, C)
- $\frac{9}{2} \phi |A \cup C| w(e)$ to each edge e of (B, D)
- $n \cdot \frac{9}{2} \phi w(e)$ to each edge e of (A, D) or (B, C)

By induction on the number of nodes n of the graph, Cohen-Addad et al. show the following Lemma:

Lemma 1 *Considering the above charging scheme for T and T^*, an edge $(u, v) \in E$ gets charged at most:*

$$\gamma_E(u, v) \leq \frac{9}{2} \phi \min \left\{ \frac{3}{2} |V(LCA_{T^*}(u, v))|, n \right\} w(u, v)$$

From this Lemma, it is possible to imply the theorem by computing the total cost as a sum over all edge charges:

$$\begin{aligned} \Gamma(T) &= \sum_{\{u,v\} \in E} \gamma_E(u, v) \\ &\leq \frac{9}{2} \phi \sum_{\{u,v\} \in E} \frac{3}{2} |V(\mathrm{LCA}_{T^*}(u, v))| w(u, v) \end{aligned}$$

Inserting the definition of $\Gamma(T^*)$ by Dasgupta,

$$= \frac{27}{4} \phi \Gamma(T^*)$$

finally proves the theorem. □

Adapting the definition of the balanced cut and rescaling γ_E by a factor of Γ_n implies the following, more general, result for any admissible cost function:

Remark 1 *Algorithm 1 achieves an $\mathcal{O}(\Gamma_n \phi)$-approximation for any admissible cost function Γ, where $\Gamma_n = \max_n \frac{\Gamma(n)}{\Gamma(\lceil n/3 \rceil)}$.*

3.2 Clustering Perfect Inputs

Algorithm 2 Fast and Simple Clustering

```
1: Input: $G = (V, E, w)$
2: $p \leftarrow$ random vertex of $V$
3: $w_1 > \ldots > w_k$ edge weights of edges $\{\cdot, p\}$
4: Let $B_i = \{v | w(p, v) = w_i\}$ for $1 \leq i \leq k$
5: Recurse on each $G[B_i]$ and obtain $T_1, \ldots, T_k$
6: $T_0^* \leftarrow$ tree with $p$ as single vertex
7: $T_i^* \leftarrow$ union of $T_{i-1}^*$ and $T_i$
8: Return: $T_k^*$
```

Cohen-Addad et al. (Cohen-Addad et al., 2018) propose a near-linear time algorithm for clustering *strict*[3] ground-truth inputs, i.e. graphs generated by an ultrametric. The algorithm works by selecting a pivot element p and partitioning the remaining vertices into sets based on the distance to p. This is repeated recursively for the parition sets B_i representing subclusters in which each vertex has the same distance to p, i.e. for which the corresponding leaves should have same LCA with p in T.

[3]Generating tree with $W(N_i) < W(N_j)$ in definition 3.

Theorem 3 *For any admissible objective function, the fast and simple clustering algorithm computes a tree of optimal cost in $\mathcal{O}(n \log^2 n)$ with high probability if the input is a strict ground-truth input or in $\mathcal{O}(n^2)$ if the input is a (not necessarily strict) ground-truth input.*

To prove that algorithm 2 computes a tree of optimal cost, let $p \in V$ be the pivot element chosen by the algorithm and $u \in B_i, v \in B_j$ elements from two partitions based on p with $j > i$. Let T be a generating tree and define $N_u := \text{LCA}_T(p, u)$ and $N_v := \text{LCA}_T(p, v)$, as visualized in figure 6. By construction of B_i, B_j, it is guaranteed that $w(p, u) > w(p, v)$. This implies that N_v is an ancestor of N_u in T, since T is a generating tree. Furthermore, it is obvious that $\text{LCA}_T(u, v) = N_v$ and therefore $w(u, v) = W(N_v) = w(p, v)$. Applying this inductively to all $G[B_i]$ proves that the output of algorithm 2 is a generating tree for G and therefore of optimal cost.

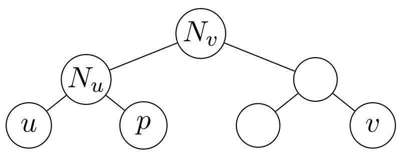

Figure 6: Visualizing the proof of theorem 3

Now prove the runtime complexity claim of theorem 3. Let input G be strongly generated by some T, consider a single recursive call on $G[B_i]$ with $|B_i| = n_0$. Assuming a bucket size of $|B_i| \leq \frac{2n}{3}$ $\forall i \in [1, k]$, it possible to apply the Master theorem (Cormen et al., 2009).

$$\text{Time}(n) = \sum_{i=1}^{k} \text{Time}(\alpha_i n) + f(n)$$

Where $\text{Time}(n)$ is the time complexity for algorithm 2 with G as input. It is easy to recognize that the recursive call $f(n)$, before further recursion, takes $\mathcal{O}(n)$ time. Thus, $f(n) \in \Theta(n^\kappa)$ holds with $\kappa = 1$ for the recursive call. Additionally, $\alpha_i = |B_i|$ fulfills the constraints $\sum_{i=1}^{k}(\alpha_i^\kappa) = 1$ and $\alpha_i < 1$ if the bucket size is limited to $\frac{2n}{3}$ as specified above. Using the Master theorem it can be concluded that $\text{Time}(n) \in \mathcal{O}(n \log n)$.

The assumption that $|B_i| \leq \frac{2n}{3}$ always holds is not reasonable, but it can be shown that the "problematic" event that a partitioning results in a single bucket of size $|B_i| = n - 1$ occurs with low probability. To prove this, select a node N^* and $v \in L(N^*)$ by walking through T down from the root in the direction of the most leaves until $\frac{2n}{3} > L(N^*) \geq \frac{n}{3}$. Then, any $u \in V$ with $\text{LCA}_T(u, v)$ being an ancestor of N^* in T, has the property that for any $x \in L(N^*)$ $w(u, v) < w(x, v)$. Thus, u and x belong to different buckets B_i and therefore $|B_i| < \frac{2n}{3} \forall i \in [1, k]$ for any partition induced by $v \in L(N^*)$. This means that the leaves of the subtree induced by such a node N^* are the leaves introducing "non-problematic" paritions. Since picking any of the $|L(N^*)| \geq \frac{n}{3}$ leaves leads to the same behavior, the probability for this event is at least $\frac{1}{3}$ and consequently the inverse event with $Pr_{p \sim V}(p \notin L(N^*)) \in \mathcal{O}(\frac{2}{3})$. Each pivot element is picked independently, so the probability of $p \notin L(N^*)$ after $c \log n$ recursive calls is $\mathcal{O}((\frac{2}{3})^{c \log n}) = \mathcal{O}(\frac{1}{3^{\log_3(n^c)}}) = \mathcal{O}(\frac{1}{n^c})$. Then, the probability that there are not $\log n$ "non-problematic" partitions after $\log(n) \cdot (c \log n)$ recursive calls can be computed by applying the union bound.

$$\begin{aligned} &\underset{p_1,\ldots,p_{\log n} \sim V}{Pr} \left(\bigcup_{i=1}^{\log n} p_i \notin L(N^*) \right) \\ &\leq \sum_{i=1}^{\log n} Pr_{p_i \sim V} \left(p_i \notin L(N^*) \right) \\ &\in \mathcal{O}(\frac{\log n}{n^c}) = \mathcal{O}(\frac{1}{n^{c-1}}) \end{aligned}$$

This shows that the inverse, i.e. the assumption of $|B_i| \leq \frac{2n}{3}$, and therefore the runtime $\mathcal{O}(n \log^2 n)$ holds with high probability.

The property that the "problematic" events occur with low probability can only be guaranteed if the input is strictly generated. If this is not the case, it may happen that every pivot element choice results in a partition with a single bucket of size $n - 1$. In this case, the runtime $\mathcal{O}(n^2)$ follows directly from the definition of the algorithm. □

4 Conclusion

This work presents the findings of Cohen-Addad et al. by focusing on the *Recursive ϕ-Sparsest Cut* and *Fast and Simple* clustering algorithms. The former approximates an optimal output for arbitrary input graphs and the latter guarantees an optimal clustering in near-linear time for perfect inputs. Additionally, Cohen-Addad et al.'s framework for objective functions and the corresponding definition of admissible objective functions is described and set in relation to Dasgupta's cost function.

References

Sanjeev Arora, Satish Rao, and Umesh Vazirani. 2009. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, April.

Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu, 2018. *Hierarchical Clustering: Objective Functions and Algorithms*, pages 378–397.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.

Sanjoy Dasgupta. 2016. A cost function for similarity-based hierarchical clustering. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 118–127, New York, NY, USA. ACM.

Murtagh Fionn and Contreras Pedro. 2011. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.