Improved Analysis of Complete-Linkage Clustering[*]

Anna Großwendt and Heiko Röglin

Department of Computer Science
University of Bonn, Germany
grosswen@cs.uni-bonn.de, roeglin@cs.uni-bonn.de

Abstract. Complete-linkage clustering is a very popular method for computing hierarchical clusterings in practice, which is not fully understood theoretically. Given a finite set $P \subseteq \mathbb{R}^d$ of points, the complete-linkage method starts with each point from P in a cluster of its own and then iteratively merges two clusters from the current clustering that have the smallest diameter when merged into a single cluster.

We study the problem of partitioning P into k clusters such that the largest diameter of the clusters is minimized and we prove that the complete-linkage method computes an $O(1)$-approximation for this problem for any metric that is induced by a norm, assuming that the dimension d is a constant. This improves the best previously known bound of $O(\log k)$ due to Ackermann et al. (Algorithmica, 2014). Our improved bound also carries over to the k-center and the discrete k-center problem.

1 Introduction

In a typical clustering problem, the goal is to partition a given set of objects into clusters such that similar objects belong to the same cluster while dissimilar objects belong to different clusters. Clustering is ubiquitous in computer science with applications ranging from biology to information retrieval and data compression. In applications where the number of clusters is not known a priori, *hierarchical clusterings* are of particular appeal. A hierarchical clustering of a set P of n objects is a sequence $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_n$, where $\mathcal{C}_i$ is a clustering of P into i non-empty clusters and $\mathcal{C}_{i+1}$ is a refinement of $\mathcal{C}_i$. Besides the advantage that the number of clusters does not have to be specified in advance, hierarchical clusterings are also appealing because they help to understand the hereditary properties of the data and they provide information at different levels of granularity.

In practice, *agglomerative methods* are very popular for computing hierarchical clusterings. An agglomerative clustering method starts with the clustering $\mathcal{C}_n$, in which every object belongs to its own cluster. Then it iteratively merges the two clusters from the current clustering $\mathcal{C}_{i+1}$ with the smallest distance to obtain the next clustering $\mathcal{C}_i$. Depending on how the distance between two clusters is defined, different agglomerative methods can be obtained. A common variant

[*] This research was supported by ERC Starting Grant 306465 (BeyondWorstCase).

is the *complete-linkage method* in which the distance between two clusters A and B is defined as the diameter or the (discrete) radius of $A \cup B$, assuming some distance measure on the objects from P is given.

The complete-linkage method is very popular and successful in a wide variety of applications. To name just a few of many recent examples, Rieck et al. [7] have used it for automatic malware detection, Ghaemmaghami et al. [6] have used it to design a speaker attribution system, and Cole et al. [2] use it as part of the Ribosomal Database Project. Yet the complete-linkage method is not fully understood in theory and there is still a considerable gap between the known upper and lower bounds for its approximation guarantee.

1.1 Problem Definitions and Algorithms

Let $P \subseteq \mathbb{R}^d$ denote a finite set of points and let $\mathrm{dist}\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ denote some metric on $\mathbb{R}^d$. A *k-clustering $\mathcal{C}$ of P* is a partition of P into k non-empty sets $C_1, \ldots, C_k$. We consider three different ways to measure the quality of the k-clustering $\mathcal{C}$, which lead to three different optimization problems.

- **diameter k-clustering problem**: Find a k-clustering $\mathcal{C}$ with minimum *diameter*. The diameter $\mathrm{diam}(\mathcal{C})$ of $\mathcal{C}$ is given by the maximal diameter $\max_i \mathrm{diam}(C_i)$ of one of its clusters, where the diameter of a set $C \subseteq P$ is defined as $\mathrm{diam}(C) := \max_{x,y \in C} \mathrm{dist}(x,y)$.
- **k-center problem**: Find a k-clustering $\mathcal{C}$ with minimum *radius*. The radius $\mathrm{rad}(\mathcal{C})$ of $\mathcal{C}$ is given by the maximal radius $\max_i \mathrm{rad}(C_i)$ of one of its clusters, where the radius of a set $C \subseteq P$ is defined as $\mathrm{rad}(C) := \min_{y \in \mathbb{R}^d} \max_{x \in C} \mathrm{dist}(x,y)$.
- **discrete k-center problem**: Find a k-clustering $\mathcal{C}$ with minimum *discrete radius*. The discrete radius $\mathrm{drad}(\mathcal{C})$ of $\mathcal{C}$ is given by the maximal discrete radius $\max_i \mathrm{drad}(C_i)$ of one of its clusters, where the discrete radius of a set $C \subseteq P$ is defined as $\mathrm{drad}(C) := \min_{y \in C} \max_{x \in C} \mathrm{dist}(x,y)$.

The *complete-linkage method* **CL** starts with the $|P|$-clustering $\mathcal{C}_{|P|}$ in which every point from P is in its own cluster. Then, for $i = |P|-1, |P|-2, \ldots, 1$, it merges two clusters from $\mathcal{C}_{i+1}$ to obtain $\mathcal{C}_i$. Regardless of the choice of which clusters are merged, this yields a hierarchical clustering $\mathcal{C}_1, \ldots, \mathcal{C}_{|P|}$. Which clusters are merged in an iteration depends on the optimization problem we consider. For the diameter k-clustering problem, the complete-linkage method chooses two clusters A and B from $\mathcal{C}_{i+1}$ such that $\mathrm{diam}(A \cup B)$ is minimized. Similarly, for the k-center problem and the discrete k-center problem it chooses two clusters A and B from $\mathcal{C}_{i+1}$ such that $\mathrm{rad}(A \cup B)$ or $\mathrm{drad}(A \cup B)$ is minimized, respectively. Hence, every objective function gives rise to a different variant of the complete-linkage method. When it is not clear from the context which variant is meant, we will use the notation $\mathbf{CL}^{\mathrm{drad}}$, $\mathbf{CL}^{\mathrm{rad}}$, and $\mathbf{CL}^{\mathrm{diam}}$ to make the variant clear.

1.2 Related Work

Let $P \subseteq \mathbb{R}^d$ and a metric dist on P be given and let $\mathcal{O}_k^{\mathrm{drad}}$, $\mathcal{O}_k^{\mathrm{rad}}$, and $\mathcal{O}_k^{\mathrm{diam}}$ be optimal k-clusterings of P for the discrete k-center problem, the k-center problem, and the diameter k-clustering problem, respectively. For each of these three problems, it is easy to find examples where no hierarchical clustering $\mathcal{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_{|P|})$ exists such that $\mathcal{C}_k$ is an optimal k-clustering for every k. We say that a hierarchical clustering $\mathcal{C}$ is an *α-approximate hierarchical clustering* for the diameter k-clustering problem if $\mathrm{diam}(\mathcal{C}_k) \leq \alpha \cdot \mathrm{diam}(\mathcal{O}_k^{\mathrm{diam}})$ holds for every k. In general, we allow α to be a function of k and d. We define α-approximate hierarchical clusterings analogously for the (discrete) k-center problem.

Dasgupta and Long [3] gave an efficient algorithm that computes 8-approximate hierarchical clusterings for the diameter k-clustering and the k-center problem, thereby giving a constructive proof of the existence of such hierarchical clusterings. Their result holds true for arbitrary metrics on $\mathbb{R}^d$ and it can even be improved to an expected approximation factor of $2e \approx 5.44$ by a randomized algorithm. They also studied the performance of the complete-linkage method and presented an artificial metric on $\mathbb{R}^2$ for which its approximation factor is only $\Omega(\log k)$ for the diameter k-clustering and the k-center problem. Ackermann et al. [1] showed for the diameter k-clustering and the discrete k-center problem a lower bound of $\Omega(\sqrt[p]{\log k})$ for the ℓ_p-metric for every $p \in \mathbb{N}$, assuming $d = \Omega(k)$.

Ackermann et al. [1] also showed that the complete-linkage method yields an $O(\log k)$-approximation for any metric that is induced by a norm, assuming that d is a constant. Here the constant in the big O notation depends on the dimension d. For the discrete k-center problem the dependence on d is only linear and additive. For the k-center problem the dependence is multiplicative and exponential in d, while for the diameter k-clustering problem it is even multiplicative and doubly exponential in d. The analysis of Ackermann et al. proceeds in two phases. The first phase ends when $2k$ clusters are left and the second phase consists of the last k merge operations. In the first phase a factor depending only on d but not on k is incurred. To make this precise, let $\mathcal{C}_{2k}^{\mathrm{drad}}$, $\mathcal{C}_{2k}^{\mathrm{rad}}$, and $\mathcal{C}_{2k}^{\mathrm{diam}}$ denote the $2k$-clusterings computed by the corresponding variants of **CL**. Ackermann et al. prove that for each objective $X \in \{\mathrm{drad}, \mathrm{rad}, \mathrm{diam}\}$ there exists a function κ_X such that

$$X(\mathcal{C}_{2k}^X) \leq \kappa_X(d) \cdot X(\mathcal{O}_k^X). \tag{1}$$

The function κ_{drad} is linear in d, the function κ_{rad} is exponential in d, and the function κ_{diam} is doubly exponential in d. The factor $O(\log k)$ is only incurred in the last k merge operations. Let $\mathcal{C}_k^{\mathrm{drad}}$, $\mathcal{C}_k^{\mathrm{rad}}$, and $\mathcal{C}_k^{\mathrm{diam}}$ denote the k-clusterings computed by the corresponding variants of **CL**. Ackermann et al. show that for each objective $X \in \{\mathrm{drad}, \mathrm{rad}, \mathrm{diam}\}$, it holds

$$X(\mathcal{C}_k^X) \leq O(\log k) \cdot X(\mathcal{C}_{2k}^X),$$

where the constant in the big O notation depends again on the dimension d. Additionally, Ackermann et al. [1] studied the case $d = 1$ separately and proved that the complete-linkage method computes 3-approximate hierarchical clusterings for the diameter k-clustering problem and the k-center problem for $d = 1$.

The approximability of non-hierarchical clustering problems is well understood. Feder and Greene [5] proved that for the Euclidean metric the k-center problem and the diameter k-clustering problem cannot be approximated better than a factor of 1.822 and 1.969, respectively. For the ℓ_1 and the ℓ_∞-metric they prove a lower bound of 2 for the approximability of both problems. On the positive side, they also provide a 2-approximation algorithm for any ℓ_p-metric.

A naive implementation of the complete-linkage method has a running time of $O(|P|^3)$. Defays gave an implementation with running time $O(|P|^2)$ [4].

1.3 Our Results

Our main result is a proof that the complete-linkage method yields an $O(1)$-approximation for the (discrete) k-center problem and the diameter k-clustering problem for any metric on $\mathbb{R}^d$ that is induced by a norm, assuming that d is a constant. This does not contradict the lower bound of Ackermann et al. because this lower bound assumes that the dimension depends linearly on k. In light of our result, the dependence of this lower bound on k is somewhat misleading and it could also be expressed as $\Omega(\sqrt[p]{\log d})$.

In order to obtain our result, we improve the second phase of the analysis of Ackermann et al. [1] and we prove that for each objective $X \in \{\mathrm{drad}, \mathrm{rad}, \mathrm{diam}\}$,

$$X(\mathcal{C}_k^X) \leq O(1) \cdot X(\mathcal{C}_{2k}^X).$$

The constant in the big O notation depends neither on d nor on k. It is 37, 19, and 17 for the discrete k-center problem, the k-center problem, and the diameter k-clustering problem, respectively. Together with (1) this yields the desired bound for the approximation factor.

In our analysis we introduce the concept of *clustering intersection graphs*. Given an ℓ-clustering $C_1, \dots, C_\ell$ computed by the complete-linkage method and an optimal k-clustering $O_1, \dots, O_k$, the clustering intersection graph contains a node for each cluster C_j and a hyperedge for every optimal cluster O_i. The hyperedge corresponding to O_i contains all clusters C_j with $O_i \cap C_j \neq \emptyset$. We then observe that merge operations of the complete-linkage method correspond to the contraction of two nodes in the clustering intersection graph. We obtain our results by carefully exploiting the structural properties of clustering intersection graphs.

In Section 2 we introduce formally the concept of clustering intersection graphs and prove some elementary properties. In Section 3 we combine our analysis with the result of Ackermann et al. about the first phase to prove that the complete-linkage method yields an $O(1)$-approximation. Due to space constraints, proofs of statements marked by ($\star$) are deferred to the full version of this paper.

2 Clustering Intersection Graphs

Our analysis is based on studying the clustering intersection graph induced by **CL** at certain points of time. Before we introduce the concept of clustering intersection graphs formally, we will define these points of time. Let $P \subseteq \mathbb{R}^d$

be arbitrary and let $\mathcal{O}_k$ denote some arbitrary optimal k-clustering of P (w.r.t. the chosen objective function diameter or (discrete) radius). By scaling our point set we may assume that the objective value of $\mathcal{O}_k$ equals 1. We define $\mathrm{t}_{\leq x}$ to be the last step before some cluster of size larger than x (w.r.t. the chosen objective function) is obtained and denote the clustering of **CL** at time $\mathrm{t}_{\leq x}$ by $\mathcal{A}_x$. The following lemma is crucial for our analysis.

Lemma 1 ($\star$). *Let $x > 0$. In $\mathcal{A}_x$ there do not exist two clusters a_1 and a_2 such that*

$$\begin{aligned} \mathrm{diam}(a_1) + \mathrm{dist}(a_1, a_2) + \mathrm{diam}(a_2) &\leq x, \text{ for } \boldsymbol{CL}^{\mathrm{diam}}, \\ \mathrm{rad}(a_1) + \mathrm{dist}(a_1, a_2) + 2\,\mathrm{rad}(a_2) &\leq x, \text{ for } \boldsymbol{CL}^{\mathrm{rad}}, \\ \mathrm{drad}(a_1) + \mathrm{dist}(a_1, a_2) + 2\,\mathrm{drad}(a_2) &\leq x, \text{ for } \boldsymbol{CL}^{\mathrm{drad}}, \end{aligned}$$

where $\mathrm{dist}(a_1, a_2)$ is defined as the minimum distance between two points $p_1 \in a_1$ and $p_2 \in a_2$.

This implies that if we have at $\mathrm{t}_{\leq x}$ two clusters $a_1, a_2 \in \mathcal{A}_x$ and some cluster $o \in \mathcal{O}_k$ with $a_1 \cap o \neq \emptyset$ and $a_2 \cap o \neq \emptyset$, then depending on the objective function at $\mathrm{t}_{\leq 2x+1}$ or $\mathrm{t}_{\leq 3x+1}$ either a_1 or a_2 or both were merged.

2.1 Definition and Fundamental Properties

The fact that we can guarantee for certain pairs of clusters that one of it is merged at a certain point of time motivates us to define a clustering intersection graph (which is in general a hypergraph) with the clusters from $\mathcal{A}_x$ as vertices, where two vertices are neighbored if and only if there exists a cluster $o \in \mathcal{O}_k$ with which both have a non-empty intersection.

Definition 2. *Let $\mathcal{O}_k$ be an optimal k-clustering of some finite point set $P \subseteq \mathbb{R}^d$. Let $\mathcal{A}_x$ be the clustering of P computed by **CL** at time $\mathrm{t}_{\leq x}$. We define the* clustering intersection graph (CI-graph) *$G_x(\mathcal{A}_x, \mathcal{O}_k)$ at point of time $\mathrm{t}_{\leq x}$ as a graph with vertex set $\mathcal{A}_x$. A set of vertices $N = \{v_1, \dots, v_\ell\}$ forms a hyperedge if there exists some cluster $o \in \mathcal{O}_k$ such that for each cluster v_i we have that $v_i \cap o \neq \emptyset$ and furthermore there does not exist a cluster $v \notin N$ with $v \cap o \neq \emptyset$.*

In general, the CI-graph is a hypergraph with exactly k edges and $|\mathcal{A}_x|$ vertices. If a statement holds for arbitrary points of time or the point of time is clear from context we omit the index x and just write G. Note that for each cluster $a \in \mathcal{A}_x$ each point $p \in a$ in the cluster is contained in some optimal cluster o. Thus, the CI-graph does not contain isolated vertices where isolated means that the vertex has no incident edge. We call a vertex ℓ a *leaf* if ℓ is incident to exactly one edge e and moreover ℓ is not the only vertex incident to e. Moreover an edge e is called a *loop* if e is only incident to one vertex. We define the *degree* of a vertex v to be number of non-loop edges that contain v plus twice the number of loops that consist of v. The CI-graph has the crucial property that merging two clusters in $\mathcal{A}_x$ corresponds to contracting the corresponding vertices in the CI-graph.

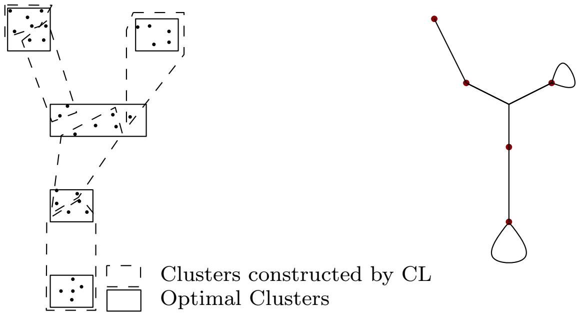

Fig. 1. Example of a clustering instance with an optimal clustering and a clustering computed by **CL** (left side) and the corresponding CI-graph (right side). Note that the figure is only schematic and does not depict the actual clustering computed by **CL** on the given instance.

Lemma 3. *There is a homomorphism between pairs of clusterings* $(\mathcal{O}, \mathcal{A})$ *where* $\mathcal{O}$ *and* $\mathcal{A}$ *are both clusterings of a finite point set* $P \subseteq \mathbb{R}^d$ *and the set of CI-graphs with respect to the operations* merging two clusters in $\mathcal{A}$ *and* contracting two vertices in the corresponding CI-graph*.*

Assume that two clusters a_1 and a_2 are merged in a step of **CL**. Then all clusters $o \in \mathcal{O}$ that have a nonempty intersection with a_1 or a_2 clearly have a nonempty intersection with $a_1 \cup a_2$. Let G and G' denote the CI-graph before and after this merge operation, respectively. Then it is easy to see that G' is obtained from G by contracting the two vertices v_1 and v_2 corresponding to a_1 and a_2. The vertex that results from this contraction is incident to each edge that was incident to v_1 or v_2 before.

To prove that the approximation factor of **CL** is at most x, it is sufficient to show that at time $\mathrm{t}_{\leq x}$ the CI-graph G_x contains at least as many edges as vertices. Clearly this is equivalent to $|\mathcal{A}_x| \leq k$, which means that **CL** has terminated.

2.2 The One-Dimensional Case

One can prove that **CL** yields a constant approximation factor for all finite point sets $P \subseteq \mathbb{R}$, all metrics dist: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ and all $k \in \mathbb{N}$ analyzing the structure of the CI-graph after certain time periods showing that at $\mathrm{t}_{\leq 3}$ (or $\mathrm{t}_{\leq 5}$) the number of vertices is smaller or equal to the number of edges. The result is known for the diameter k-clustering problem and the k-center problem [1]. Our result also holds for the discrete k-center problem. For a detailed proof see the full version of our paper.

Theorem 4 **($\star$).** *For* $d = 1$ *and arbitrary* k*,*
CL$^{\mathrm{diam}}$ *computes a* 3*-approximation for the diameter* k*-clustering problem,*
CL$^{\mathrm{rad}}$ *computes a* 5*-approximation for the* k*-center problem,*
CL$^{\mathrm{drad}}$ *computes a* 5*-approximation for the discrete* k*-center problem.*

2.3 Completion of the CI-Graph

In the one-dimensional case one has the crucial property that all vertices of a CI-graph can be arranged in increasing order on a line such that only neighbored vertices on the line may be contracted. Additionally, it follows from Lemma 1 that at least one vertex of every neighbored pair must be contracted until a certain time step. This implies that each edge is incident to at most 3 vertices at $\mathrm{t}_{\leq 1}$, which is essential in the proof of Theorem 4. This property is not true anymore in higher dimensions.

Given a CI-graph G, we construct a weighted multi-graph $\Gamma(G)$, which we call the *completion* of G. The graph $\Gamma(G)$ has the same vertex set as the CI-graph G. For every hyperedge $\{v_1, \dots, v_\ell\}$ in G, we introduce a clique with edge weights 1 in $\Gamma(G)$. For each pair of vertices v and w from the same connected component that are not adjacent we add an edge (v, w) to $\Gamma(G)$. If p denotes the length of the shortest v-w-path in G then the weight of the edge (v, w) in $\Gamma(G)$ is set to $p + (p-1)x$ for the objective function diam and $p + (p-1)2x$ for the objective functions rad and drad. This construction ensures the following important property: the weight of every edge (v, w) in $\Gamma(G)$ is an upper bound for the distance of the corresponding clusters (remember that the distance of two clusters is defined as the smallest distance between any pair of points from these clusters).

Lemma 5 ($\star$). *Assume that the shortest v-w-path in a CI-graph G has length p. Then the smallest distance between two points in v and w is at most $p+(p-1)x$ for the objective function* diam *and $p + (p-1)2x$ for the objective functions* rad *and* drad*.*

For the analysis of **CL** we choose a subgraph H of $\Gamma(G)$. Unfortunately, Lemma 3 cannot be applied to H since H is no CI-graph but we state a weaker version, which is still strong enough for our analysis.

Lemma 6 ($\star$). *Let $G_x = G_x(\mathcal{A}_x, \mathcal{O}_k)$ be a CI-graph of a clustering $(\mathcal{A}_x, \mathcal{O}_k)$ at point of time $\mathrm{t}_{\leq x}$. Let H_x be a subgraph of $\Gamma(G_x)$ with $V(H_x) = V(G_x)$. Now consider $G_{x'} = G_{x'}(\mathcal{A}_{x'}, \mathcal{O}_k)$ for some point of time $\mathrm{t}_{\leq x'}$ with $x' > x$. Let $H_{x'}$ be the graph that arises from H_x by performing the same contractions that are made between G_x and $G_{x'}$. Then $V(G_{x'}) = V(H_{x'})$ and moreover the weight of any edge (v, w) in $H_{x'}$ is an upper bound for the distance of the clusters corresponding to v and w.*

2.4 Analysis of H at Different Time Steps

The analysis of **CL** proceeds as follows. Let G_x be the CI-graph for a fixed point of time $\mathrm{t}_{\leq x}$. Assume that there exists a special subgraph H_x of $\Gamma(G_x)$ satisfying the properties

i) $V(H_x) = V(G_x)$,
ii) $|E(H_x)| \leq k$,

iii) and no vertex in H_x is isolated (i.e., every vertex in H_x has at least one incident edge).

We will prove that at a certain point of time t_c, depending on the maximum edge weight in H_x, we have that $|V(H_c)| \leq |E(H_c)|$. Because of property i) and Lemma 6 we conclude $V(H_c) = V(G_c)$. Together with property ii) we obtain $|V(G_c)| = |V(H_c)| \leq |E(H_c)| = |E(H_x)| \leq k$ and thus **CL** terminated. In the following we denote $H_{x'}$ by H if the point of time is clear from context or if a statement holds for all $H_{x'}$ with $x' \geq x$.

First note that H is a multi-graph. Multi-graphs have the crucial property that a connected component has at least as many edges as vertices if and only if a cycle exists (where a loop is considered as a special case of a cycle).

Definition 7. *We call a connected component of* H active *if the component is a tree. Otherwise we call it* inactive*.*

Observation 8. *If $H_{x'}$ has no active connected component, then **CL** has terminated at* $\mathrm{t}_{\leq x'}$*.*

Leaves of H and their neighbors have a key role in the analysis of the algorithm. We will show that between certain time steps either a leaf or its unique neighbor is merged. Define d_n as an upper bound for the distance between the clusters corresponding to any pair of adjacent vertices v_1 and v_2 in H_x. Because of Lemma 6 we have that d_n is smaller or equal to the maximum edge weight in H at any point of time. We use that fact later when choosing the subgraph H_x. We analyze time steps $\mathrm{t}_{\leq x+i(d_n+x)}$ for the diameter k-clustering problem and $\mathrm{t}_{\leq x+i(d_n+2x)}$ for the k-center and discrete k-center problem according to Lemma 1 and denote them by t_i. In accordance to that, we define $x_i = x + i(d_n + x)$ for $\mathbf{CL}^{\mathrm{diam}}$ and $x_i = x + i(d_n + 2x)$ for $\mathbf{CL}^{\mathrm{rad}}$ and $\mathbf{CL}^{\mathrm{drad}}$, respectively.

Definition 9. *We call a vertex $p \in H$ in an active connected component of H a* leaf-parent *if p is the neighbor of some leaf and has at least degree* 2*.*

At the beginning of our analysis at $\mathrm{t}_{\leq x}$ there does not necessarily exist a leaf-parent in each active component. This follows because the smallest possible active component consists of two connected vertices and is the only possibility of an active component without a leaf-parent (remember that in H there exist no isolated vertices by property iii); any connected component that consists of a single vertex must contain a loop and is hence inactive). Analogous to dimension one we show that at point of time t_1 for each active connected component by **CL** either one vertex was merged with a vertex from another component but thereby some vertex with degree 2 is built or two vertices from one component were merged. The latter means that a cycle was built and the component is no longer active. The following lemma ensures that at a certain point of time there exists a leaf-parent in each active component.

Lemma 10 ($\star$). *Each active component C of H containing a vertex v of degree 2 contains at least one leaf-parent p. In particular H_{x_1} contains at least one leaf-parent in each active component.*

The proof of Lemma 10 gives a hint that we have in general at least two leaf-parents in each component while components with exactly one leaf-parent are of a special form. We will use this structure later on to prove that if each active component contains at least 2 leaf-parents the algorithm terminates. Therefore we need some statement counting the number of remaining contractions depending on the number of leaf-parents. First, we need some statement how often contraction steps are performed in each component.

Lemma 11 ($\star$). *Let ℓ be some leaf in H_{x_i} at an arbitrary point of time t_i with $i \geq 0$. Then the leaf ℓ is also contained in H_{x_0} and it is not contracted between t_0 and t_i. Moreover between two steps of time t_i and t_{i+1} where $i \in \mathbb{N}_0$ we have that for each leaf ℓ either the leaf ℓ or its corresponding leaf-parent p_ℓ is contracted.*

We denote the number of leaf-parents of H at time t_i for a connected component C by $n_{\ell p}(C)$. Since in each active component the number of leaf-parents is at most the number of leaves, we may conclude that the algorithm performs at least $n_{\ell p}/2$ contractions between t_i and t_{i+1} where $n_{\ell p} = \sum\limits_{i=1}^{r} n_{\ell p}(C_i)$ is the sum over the number of leaf-parents in the active connected components. Now we count the number of leaf-parents contained in one active connected component. The idea is that if each active component contains at least two leaf-parents then we have at least as many contractions as active components and can conclude that the algorithm will terminate. Therefore we show that at a certain point of time every active component must contain at least two leaf-parents. First we will show that if the number of leaf-parents in an active component is at least two, then after contraction the number of leaf-parents does not decrease below two.

Lemma 12 ($\star$). *Assume that two vertices v_1 and v_2 from two different components C_1 and C_2 that contain each at least one leaf-parent are contracted in H. If the resulting component $C = C_1 \cup C_2$ is active then C has at least as many leaf-parents as the maximum of C_1 and C_2, i.e., $n_{\ell p}(C) \geq \max\{n_{\ell p}(C_1), n_{\ell p}(C_2)\}$.*

We may conclude that the only possibility to obtain an active component containing just one leaf-parent is that we contract vertices from two different components which contain only one leaf-parent. In particular for two such components C_1 and C_2 we have to contract the leaf-parents p_1 and p_2. If another vertex and therefore a leaf of C_1 is contracted another component $C_1 \cup C_2$ with at least two leaf-parents is built.

Lemma 13 ($\star$). *For $\boldsymbol{CL}^{\mathrm{diam}}$ each active component contains at least 2 leaf-parents at point of time t_3. For $\boldsymbol{CL}^{\mathrm{rad}}$ each active component contains at least 2 leaf-parents at t_2. For $\boldsymbol{CL}^{\mathrm{drad}}$ each active component contains at least 2 leaf-parents at t_6.*

It remains to prove that **CL** terminates if each component contains at least two leaf-parents.

Lemma 14 ($\star$). *If at t_i each active component of H_{x_i} contains at least two leaf-parents then **CL** has terminated at t_{i+1}.*

3 Approximation Factor of CL in the Case $d \geq 2$

In this section we combine our analysis with the result of Ackermann et al. [1] for the first phase of **CL** (i.e., the steps until $2k$ clusters are left) in order to prove the main theorem. From the analysis of Ackermann et al. it follows that there is a function κ such that for $x = \kappa(d)$ the CI-graph G_x contains at most $2k$ vertices. We will analyze the last k steps of **CL** more carefully. We consider the completion $\Gamma(G_x)$ of G_x and assume that it is connected. This is not necessarily the case but we will see later that this assumption is without loss of generality because our analysis can be applied to each connected component separately. In fact, the result of Ackermann et al. implies that for each connected component of G_x the number of vertices is at most twice the number of edges.

3.1 CI-Graphs with at most $2k$ Vertices

Let H_x be a subgraph of $\Gamma(G_x)$ with k edges and at most $2k$ vertices such that H_x fulfills the properties i)-iii). The goal is to find such a subgraph H_x whose maximum edge weight is small. Note that properties i), ii), and iii) imply $|V(G_x)| = |V(H_x)| \leq 2|E(H_x)| \leq 2k = 2|E(G_x)|$, which means $|V(G_x)| \leq 2|E(G_x)|$ is a necessary property of G_x to find a subgraph H_x.

We will prove that we can always find a subgraph H_x of G_x that satisfies properties i)-iii) and has the following additional property iv): for each edge $e' = (v, w) \in E(H_x)$ the vertices v and w have distance at most 2 in G_x, i.e., either there is an edge $e \in E(G_x)$ with $\{v, w\} \subseteq e$ or there are two edges $e_v \in E(G_x)$ and $e_w \in E(G_x)$ with $v \in e_v$, $w \in e_w$, and $v \cap w \neq \emptyset$.

Using this we will prove that **CL** terminates at time $t_{\leq O(x)}$ if for each connected component C of the CI-graph G_x we have that $|V(C)| \leq 2|E(C)|$.

In order to find a subgraph H_x of $\Gamma(G_x)$ that satisfies properties i)-iv) we let T be a spanning tree of $\Gamma(G_x)$ that uses only edges of weight 1. Such a spanning tree is guaranteed to exist because we assumed G_x to be connected. Such a spanning tree satisfies all properties except for ii) because the number of edges in T is $|V(G_x)| - 1$ and $|V(G_x)|$ can only be bounded by $2k$.

However, any perfect matching in the spanning tree T is a subgraph H that satisfies the properties i)-iv). If T does not contain a perfect matching, we show how to find a perfect 2-matching (according to the following definition).

Definition 15. *An α-matching in a graph G is a matching M in the complete graph $K_{|V(G)|}$ with $|V(G)|$ vertices such that for each matching edge $(v, w) \in M$ the distance of v and w in G is at most α. Moreover we call an α-matching perfect if M is a perfect matching in $K_{|V(G)|}$.*

Lemma 16 ($\star$). *Each tree T with an even number $|V(T)|$ of vertices has a perfect 2-matching.*

Construction of $\boldsymbol{H_x}$ We construct a graph H_x that satisfies the properties i), ii), iii), and iv) as follows. First we compute an arbitrary spanning tree T of $\Gamma(G_x)$ that uses only edges of weight 1. If $|V(G_x)| = |V(H_x)|$ is even, then the graph H_x is chosen as a perfect 2-matching of T. Then the properties i), iii), and iv) are satisfied by construction and property ii) is satisfied because of $|E(H_x)| = |V(H_x)|/2 \leq k$. If $|V(G_x)|$ is odd, we choose some leaf v from the spanning tree T. Then we find a perfect 2-matching M in $T \setminus \{v\}$. Since $|V(G_x)| \leq 2|E(G_x)|$ we have that the matching contains at most $|E(G_x)| - 1$ edges. Thus we set H_x to M and may add the edge from T that is incident to v to H_x such that property iii) becomes true.

Now we have a graph H_x fulfilling properties i), ii), iii), and iv). Property iv) and Lemma 6 imply that $d_n \leq 2 + x$ for the objective function diam and $d_n \leq 2 + 2x$ for the objective functions rad and drad. We conclude with the following theorem.

Theorem 17 ($\star$). *Assume that the CI-graph G_x is connected and contains k edges and at most $2k$ vertices at some point of time $\mathrm{t}_{\leq x}$. Then $\boldsymbol{CL}^{\mathrm{diam}}$ computes a $9x + 8$ approximation for the diameter k-clustering problem. Moreover $\boldsymbol{CL}^{\mathrm{rad}}$ computes a $13x+6$ approximation for the k-center problem and $\boldsymbol{CL}^{\mathrm{drad}}$ computes a $25x + 12$ approximation for the discrete k-center problem.*

3.2 Approximation Factor of CL

Now for each version of the algorithm $\mathbf{CL}^{\mathrm{diam}}$, $\mathbf{CL}^{\mathrm{rad}}$, and $\mathbf{CL}^{\mathrm{drad}}$ we combine our analysis with the special result of [1] corresponding to each of the methods. We state the following lemma from [1] deriving an upper bound for a point of time x where $|V(G_x)| \leq 2k$.

Lemma 18 ([1]). *Let $P \subseteq \mathbb{R}^d$ be finite. Then, for all $k \in \mathbb{N}$ with $2k \leq |P|$, the partition $\mathcal{A}$ of P into $2k$ clusters computed by $\boldsymbol{CL}^{\mathrm{drad}}$ satisfies*

$$\max_{a\in\mathcal{A}} \mathrm{drad}(a) < 20d \cdot \mathrm{drad}(\mathcal{O}_k^{\mathrm{drad}}).$$

Combining this result with Theorem 17 yields the following theorem.

Theorem 19 ($\star$). *For $d \in \mathbb{N}$ and a finite point set $P \subseteq \mathbb{R}^d$ the algorithm $\boldsymbol{CL}^{\mathrm{drad}}$ computes an $O(d)$-approximation for the discrete k-center problem.*

Lemma 20 ([1]). *Let $P \subseteq \mathbb{R}^d$ be finite. Then, for all $k \in \mathbb{N}$ with $2k \leq |P|$, the partition $\mathcal{A}$ of P into $2k$ clusters computed by $\boldsymbol{CL}^{\mathrm{rad}}$ satisfies*

$$\max_{a\in\mathcal{A}} \mathrm{rad}(a) < 24d \cdot e^{24d} \cdot \mathrm{rad}(\mathcal{O}_k^{\mathrm{rad}}).$$

Combining this result with Theorem 17 yields the following theorem.

Theorem 21. *For $d \in \mathbb{N}$ and a finite point set $P \subseteq \mathbb{R}^d$ the algorithm $\boldsymbol{CL}^{\mathrm{rad}}$ computes an $e^{O(d)}$-approximation for the k-center problem.*

Lemma 22 ([1]). *Let $P \subseteq \mathbb{R}^d$ be finite. Then, for all $k \in \mathbb{N}$ with $2k \leq |P|$, the partition $\mathcal{A}$ of P into $2k$ clusters computed by $\boldsymbol{CL}^{\mathrm{diam}}$ satisfies*

$$\max_{a \in \mathcal{A}} \mathrm{diam}(a) < 2^{3(42d)^d}(28d+6) \cdot \mathrm{diam}(\mathcal{O}_k^{\mathrm{diam}}).$$

Analogously to $\mathbf{CL}^{\mathrm{drad}}$ and $\mathbf{CL}^{\mathrm{rad}}$ we can conclude the following theorem.

Theorem 23. *For $d \in \mathbb{N}$ and a finite point set $P \subseteq \mathbb{R}^d$ the algorithm $\boldsymbol{CL}^{\mathrm{diam}}$ computes a $2^{O(d)^d}$-approximation for the diameter k-clustering problem.*

4 Conclusions

We have shown that the popular complete-linkage method computes $O(1)$-approximate hierarchical clusterings for the diameter k-clustering problem and the (discrete) k-center problem, assuming that d is a constant. For this it was sufficient to improve the second phase of the analysis by Ackermann et al. [1] (i.e., the last k merge operations). We used their results about the first phase to obtain our results. It is a very interesting question if the dependence on the dimension can be improved in the first phase. If we express the known lower bound of Ackermann et al. [1] in terms of d then it becomes $\Omega(\sqrt[p]{\log d})$. Hence, in terms of d, there is still a huge gap between the known upper and lower bounds. Another interesting question is whether the upper bound of $O(\log k)$ holds also for metrics that are not induced by norms.

References

1. Marcel R. Ackermann, Johannes Blömer, Daniel Kuntze, and Christian Sohler. Analysis of agglomerative clustering. *Algorithmica*, 69(1):184–215, 2014.
2. James R. Cole, Qiong Wang, Jordan A. Fish, Benli Chai, Donna M. McGarrell, Yanni Sun, C. Titus Brown, Andrea Porras-Alfaro, Cheryl R. Kuske, and James M. Tiedje. Ribosomal database project: data and tools for high throughput rrna analysis. *Nucleic Acids Research*, 2013.
3. Sanjoy Dasgupta and Philip M. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.
4. D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.
5. Tomás Feder and Daniel H. Greene. Optimal algorithms for approximate clustering. In *Proc. of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 434–444, 1988.
6. Houman Ghaemmaghami, David Dean, Robbie Vogt, and Sridha Sridharan. Speaker attribution of multiple telephone conversations using a complete-linkage clustering approach. In *Proc. of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4185–4188, 2012.
7. Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz. Automatic analysis of malware behavior using machine learning. *Journal of Computer Security*, 19(4):639–668, 2011.