Analysis of Agglomerative Clustering*

Marcel R. Ackermann[1], Johannes Blömer[1], Daniel Kuntze[1], and Christian Sohler[2]

1 **Department of Computer Science**
University of Paderborn
{mra,bloemer,kuntze}@upb.de
2 **Department of Computer Science**
TU Dortmund
christian.sohler@tu-dortmund.de

Abstract

The diameter k-clustering problem is the problem of partitioning a finite subset of $\mathbb{R}^d$ into k subsets called clusters such that the maximum diameter of the clusters is minimized. One early clustering algorithm that computes a hierarchy of approximate solutions to this problem for all values of k is the agglomerative clustering algorithm with the complete linkage strategy. For decades this algorithm has been widely used by practitioners. However, it is not well studied theoretically. In this paper we analyze the agglomerative complete linkage clustering algorithm. Assuming that the dimension d is a constant, we show that for any k the solution computed by this algorithm is an $O(\log k)$-approximation to the diameter k-clustering problem. Moreover, our analysis does not only hold for the Euclidean distance but for any metric that is based on a norm.

1998 ACM Subject Classification F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—Geometrical problems and computations; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Clustering; I.5.3 [Pattern Recognition]: Clustering—Algorithms, Similarity measures

Keywords and phrases agglomerative clustering, hierarchical clustering, complete linkage, approximation guarantees

Digital Object Identifier 10.4230/LIPIcs.STACS.2011.308

1 Introduction

Clustering is the process of partitioning a set of objects into subsets (called clusters) such that each subset contains similar objects and objects in different subsets are dissimilar. It has many applications including data compression [13], analysis of gene expression data [6], anomaly detection [10], and structuring results of search engines [3]. For every application a proper objective function is used to measure the quality of a clustering. One particular objective function is the largest diameter of the clusters. If the desired number of clusters k is given we call the problem of minimizing this objective function the *diameter k-clustering problem*.

One of the earliest and most widely used clustering strategies is agglomerative clustering. The history of agglomerative clustering goes back at least to the 1950s (see for example

* For all four authors this research was supported by the German Research Foundation (DFG), grants BL 314/6-2 and SO 514/4-2.

28th Symposium on Theoretical Aspects of Computer Science (STACS'11).
Editors: Thomas Schwentick, Christoph Dürr; pp. 308–319

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


[8, 11]). Later, biological taxonomy became one of the driving forces of cluster analysis. In [14] the authors, who where the first biologists using computers to classify organisms, discuss several agglomerative clustering methods.

Agglomerative clustering is a bottom-up clustering process. At the beginning, every input object forms its own cluster. In each subsequent step, the two 'closest' clusters will be merged until only one cluster remains. This clustering process creates a hierarchy of clusters, such that for any two different clusters A and B from possibly different levels of the hierarchy we either have $A \cap B = \emptyset$, $A \subset B$, or $B \subset A$. Such a hierarchy is useful in many applications, for example, when one is interested in hereditary properties of the clusters (as in some bioinformatics applications) or if the exact number of clusters is a priori unknown.

In order to define the agglomerative strategy properly, we have to specify a distance measure between clusters. Given a distance function between data objects, the following distance measures between clusters are frequently used. In the *single linkage strategy*, the distance between two clusters is defined as the distance between their closest pair of data objects. It is not hard to see that this strategy is equivalent to computing a minimum spanning tree of the graph induced by the distance function using Kruskal's algorithm. In case of the *complete linkage strategy*, the distance between two clusters is defined as the distance between their furthest pair of data objects. In the *average linkage strategy* the distance is defined as the average distance between data objects from the two clusters.

1.1 Related Work

In this paper we study the agglomerative clustering algorithm using the complete linkage strategy to find a hierarchical clustering of n points from $\mathbb{R}^d$. The running time is obviously polynomial in the description length of the input. Therefore, our only goal in this paper is to give an approximation guarantee for the diameter k-clustering problem. The approximation guarantee is given by a factor α such that the cost of the k-clustering computed by the algorithm is at most α times the cost of an optimal k-clustering. Although the agglomerative complete linkage clustering algorithm is widely used, only few theoretical results considering the quality of the clustering computed by this algorithm are known. It is known that there exists a certain metric distance function such that this algorithm computes a k-clustering with an approximation factor of $\Omega(\log k)$ [5]. However, prior to the analysis we present in this paper, no non-trivial upper bound for the approximation guarantee of the classical complete linkage agglomerative clustering algorithm was known, and deriving such a bound has been discussed as one of the open problems in [5].

The diameter k-clustering problem is closely related to the *k-center problem*. In this problem, we are searching for k centers and the objective is to minimize the maximum distance of any input point to the nearest center. When the centers are restricted to come from the set of the input points, the problem is called the *discrete k-center problem*. It is known that for metric distance functions the costs of optimal solutions to all three problems are within a factor of 2 from each other.

For the Euclidean case we know that the diameter k-clustering problem and the k-center problem are $\mathcal{NP}$-hard. In fact, it is already $\mathcal{NP}$-hard to approximate both problems with an approximation factor below 1.96 and 1.82 respectively [7].

For fixed k, i.e. when we are not interested in a hierarchy of clusterings, there exist provably good approxiamtion algorithms. For the discrete k-center problem, a simple 2-approximation algorithm is known for metric spaces [9], which immediately yields a 4-approximation algorithm for the diameter k-clustering problem. For the k-center prob-

lem, a variety of results is known. For example, for the Euclidean metric in [2] a $(1+\epsilon)$-approximation algorithm with running time $2^{O(k \log k/\epsilon^2)}dn$ is shown. This implies a $(2+\epsilon)$-approximation algorithm with the same running time for the diameter k-clustering problem.

Also, for metric spaces a hierarchical clustering strategy with an approximation guarantee of 8 for the discrete k-center problem is known [5]. This implies an algorithm with an approximation guarantee of 16 for the diameter k-clustering problem.

This paper as well as all of the above mentioned work is about static clustering, i.e. in the problem definition we are given the whole set of input points at once. An alternative model of the input data is to consider sequences of points that are given one after another. In [4] the authors discuss clustering in a so-called *incremental clustering* model. They give an algorithm with constant approximation factor that maintains a hierarchical clustering while new points are added to the input set. Furthermore, they show a lower bound of $\Omega(\log k)$ for the agglomerative complete linkage algorithm and the diameter k-clustering problem. However, since their model differs from ours, this result has no bearing on our lower bounds.

1.2 Our contribution

In this paper, we study the agglomerative complete linkage clustering algorithm for input sets $X \subset \mathbb{R}^d$, where d is constant. To measure the distance between data points, we use a metric that is based on a norm, e.g., the Euclidean metric. We prove that in this case the agglomerative clustering algorithm is an $O(\log k)$-approximation algorithm. Here, the O-notation hides a constant that is doubly exponential in d. This approximation guarantee holds for every level of the hierarchy computed by the algorithm. That is, we compare each computed k-clustering with an optimal solution for the particular value of k. These optimal k-clusterings do not necessarily form a hierarchy. In fact, there are simple examples where optimal solutions have no hierarchical structure.

Our analysis also yields that if we allow $2k$ instead of k clusters and compare the cost of the computed $2k$-clustering to an optimal solution with k clusters, the approximation factor is independent of k and depends only on d. Moreover, the techniques of our analysis can be applied to prove stronger results for the k-center problem and the discrete k-center problem. For the k-center problem we derive an approximation guarantee that is logarithmic in k and only single exponential in d. For the discrete k-center problem we derive an approximation guarantee that is logarithmic in k and the dependence on d is only linear and additive.

Furthermore, we give almost matching upper and lower bounds for the one-dimensional case. These bounds are independent of k. For $d \geq 2$ and the metric based on the ℓ_∞-norm we provide a lower bound that exceeds the upper bound for $d = 1$. For $d \geq 3$ we give a lower bound for the Euclidean case which is above the lower bound for $d = 1$. Finally, we construct instances providing lower bounds for any metric based on an ℓ_p-norm with $1 \leq p \leq \infty$. However, for these instances the lower bounds and the dimension d depend on k.

2 Preliminaries and problem definition

Throughout this paper, we consider input sets that are finite subsets of $\mathbb{R}^d$. Our results hold for arbitrary metrics that are based on a norm, i.e., the distance $\|x-y\|$ between two points $x, y \in \mathbb{R}^d$ is measured using an arbitrary norm $\|\cdot\|$. Readers who are not familiar with arbitrary metrics or are only interested in the Euclidean case, may assume that $\|\cdot\|_2$ is used, i.e. $\|x-y\| = \sqrt{\sum_{i=1}^{d}(x_i-y_i)^2}$. For $r \in \mathbb{R}$ and $y \in \mathbb{R}^d$ we denote the closed

d-dimensional ball of radius r centered at y by $\mathrm{B}_r^d(y) := \{x \mid \|x - y\| \leq r\}$.

Given $k \in \mathbb{N}$ and a finite set $X \subset \mathbb{R}^d$ with $k \leq |X|$ we say that $\mathcal{C}_k = \{C_1, \ldots, C_k\}$ is a k-clustering of X if the sets $C_1, \ldots, C_k$ (called clusters) form a partition of X into k non-empty subsets. We call a collection of k-clusterings of the same finite set X but for different values of k hierarchical, if it fulfills the following two properties. First, for any $1 \leq k \leq |X|$ the collection contains at most one k-clustering. Second, for any two of its clusterings $\mathcal{C}_i, \mathcal{C}_j$ with $|\mathcal{C}_i| = i < j = |\mathcal{C}_j|$ every cluster in $\mathcal{C}_i$ is the union of one or more clusters from $\mathcal{C}_j$. A hierarchical collection of clusterings is called a hierarchical clustering.

For a finite and non-empty set $C \subset \mathbb{R}^d$ we define the diameter of C to be $\mathrm{diam}(C) := \max_{x,y \in C} \|x - y\|$. Finally, we define the cost of a k-clustering $\mathcal{C}_k$ as its largest diameter, i.e. $\mathrm{cost}(\mathcal{C}_k) := \max_{C \in \mathcal{C}_k} \mathrm{diam}(C)$.

▶ **Problem 1** (diameter k-clustering)**.** Given $k \in \mathbb{N}$ and a finite set $X \subset \mathbb{R}^d$ with $|X| \geq k$ find a k-clustering $\mathcal{C}_k$ of X with minimal cost.

For our analysis of agglomerative clustering we repeatedly use the volume argument stated in Lemma 3. This argument provides an upper bound on the minimum distance between two points from a finite set of points lying inside the union of finitely many balls. For the application of this argument the following definition is crucial.

▶ **Definition 2.** Let $k \in \mathbb{N}$ and $r \in \mathbb{R}$. A set $X \subset \mathbb{R}^d$ is called (k, r)-*coverable* if there exist $y_1, \ldots, y_k \in \mathbb{R}^d$ with $X \subseteq \bigcup_{i=1}^k \mathrm{B}_r^d(y_i)$.

▶ **Lemma 3.** *Let $k \in \mathbb{N}$, $r \in \mathbb{R}$ and $P \subset \mathbb{R}^d$ be finite and (k, r)-coverable with $|P| > k$. Then there exist distinct $p, q \in P$ such that $\|p - q\| \leq 4r \sqrt[d]{\frac{k}{|P|}}$.*

The proof of Lemma 3 can be found in the full version of this paper [1].

3 Analysis

In this section we analyze the agglomerative algorithm for Problem 1 stated as Algorithm 1. Given a finite set $X \subset \mathbb{R}^d$ of input points, the algorithm computes hierarchical k-clusterings for all values of k between 1 and $|X|$. As mentioned before, the algorithm takes a bottom-up approach. It starts with the $|X|$-clustering that contains one cluster for each input point and then successively merges two of the remaining clusters that minimize the diameter of the resulting cluster.

▶ Observation 4. The greedy strategy guarantees that the following holds for all computed clusterings. First, the cost of the clustering is equal to the diameter of the cluster created last. Second, the diameter of the union of any two clusters is always an upper bound for the cost of the clustering to be computed next.

Note that our results hold for any particular tie-breaking strategy. However, to keep the analysis simple, we assume that there are no ties. Thus, for any input set X the clusterings computed by Algorithm 1 are uniquely determined.

Our main result is the following theorem.

▶ **Theorem 5.** *Let $X \subset \mathbb{R}^d$ be a finite set of points. Then for all $k \in \mathbb{N}$ with $k \leq |X|$ the partition $\mathcal{C}_k$ of X into k clusters as computed by Algorithm 1 satisfies*

$$\mathrm{cost}(\mathcal{C}_k) = O(\log k) \cdot \mathrm{opt}_k,$$

where opt_k denotes the cost of an optimal solution to Problem 1, and the constant hidden in the O-notation is doubly exponential in the dimension d.

AgglomerativeCompleteLinkage(X):
X finite set of input points from $\mathbb{R}^d$

1: $\mathcal{C}_{|X|} := \{\{x\} \mid x \in X\}$
2: **for** $i = |X| - 1, \ldots, 1$ **do**
3: find distinct clusters $A, B \in \mathcal{C}_{i+1}$ minimizing $\operatorname{diam}(A \cup B)$
4: $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$
5: **end for**
6: **return** $\mathcal{C}_1, \ldots, \mathcal{C}_{|X|}$

Algorithm 1 The agglomerative complete linkage clustering algorithm.

We prove Theorem 5 in two steps. First, Proposition 6 in Section 3.1 provides an upper bound to the cost of the intermediate $2k$-clustering. This upper bound is independent of k and $|X|$ and may be of independent interest. Second, in the remainder of Section 3, we analyze the k merge steps of Algorithm 1 down to the computation of the k-clustering.

In the following, let $X \subset \mathbb{R}^d$ be the finite set of input points for Algorithm 1 and $k \in \mathbb{N}$ be a fixed number of clusters with $k \leq |X|$. Furthermore, to simplify notation let $r := \operatorname{opt}_k$, where opt_k is the maximum diameter of an optimal solution to Problem 1. Since any cluster C is contained in a ball of radius $\operatorname{diam}(C)$, the set X is (k, r)-coverable, a fact that will be used frequently in our analysis. By $\mathcal{C}_1, \ldots, \mathcal{C}_{|X|}$ we denote the clusterings computed by Algorithm 1 on input X.

3.1 Analysis of the $2k$-clustering

▶ **Proposition 6.** *Let $X \subset \mathbb{R}^d$ be finite. Then for all $k \in \mathbb{N}$ with $2k \leq |X|$ the partition $\mathcal{C}_{2k}$ of X into $2k$ clusters as computed by Algorithm 1 satisfies*

$$\operatorname{cost}(\mathcal{C}_{2k}) < 2^{3\sigma}\,(28d + 6) \cdot \operatorname{opt}_k,$$

where $\sigma = (42d)^d$ and opt_k denotes the cost of an optimal solution to Problem 1.

To prove Proposition 6 we divide the merge steps of Algorithm 1 into two stages. The first stage consists of the merge steps down to a $2^{2^{O(d \log d)}}k$-clustering. The analysis of the first stage is based on the following notion of similarity. Two clusters are called similar if one cluster can be translated such that every point of the translated cluster is near a point of the second cluster. Then, by merging similar clusters, the diameter essentially increases by the length of the translation vector. During the whole first stage we guarantee that there is a sufficiently large number of similar clusters left. The cost of the intermediate $2^{2^{O(d \log d)}}k$-clustering can be upper bounded by $O(d) \cdot \operatorname{opt}_k$.

The second stage consists of the merge steps reducing the number of remaining clusters from $2^{2^{O(d \log d)}}k$ to only $2k$. In this stage we are no longer able to guarantee that a sufficiently large number of similar clusters exists. Therefore, we analyze the merge steps of the second stage using a weaker argument. The underlying reasoning of what we do for the second stage is the following. If there are more than $2k$ clusters left, we are able to find sufficiently many pairs of clusters that intersect with the same cluster of an optimal k-clustering. As long as one of these pairs is left, the cost of merging this pair gives an upper bound on the cost of the next merge step. Therefore, we can bound the diameter of the created cluster by the sum of the diameters of the two clusters plus the diameter of the optimal cluster. We find that the cost of the intermediate $2k$-clustering is upper bounded by $2^{2^{O(d \log d)}} \cdot \operatorname{opt}_k$. Let us remark that we do not obtain our main result if we already use this argument for the first stage.

3.2 Stage one

In our analysis the first stage is subdivided into phases, such that in each phase the number of remaining clusters is reduced by one fourth. The following lemma will be used to bound the increase of the cost during a single phase.

▶ **Lemma 7.** *Let* $\lambda \in \mathbb{R}$ *with* $0 < \lambda < 1$ *and* $\rho := \left\lceil \left(\frac{3}{\lambda}\right)^d \right\rceil$. *Furthermore let* $m \in \mathbb{N}$ *with* $2^{\rho+1}k < m \leq |X|$. *Then*

$$\operatorname{cost}(\mathcal{C}_{\lfloor \frac{3m}{4} \rfloor}) < (1+2\lambda) \cdot \operatorname{cost}(\mathcal{C}_m) + 4r \sqrt[d]{\frac{2^{\rho+1}k}{m}}. \tag{1}$$

Proof. Let $t := \lfloor \frac{3m}{4} \rfloor$ and $\mathcal{S} := \mathcal{C}_m \cap \mathcal{C}_{t+1}$ be the set of clusters from $\mathcal{C}_m$ that still exist $\lceil \frac{m}{4} \rceil - 1$ merge steps after the computation of $\mathcal{C}_m$. In each iteration of its loop, the algorithm can merge at most two clusters from $\mathcal{C}_m$. Thus $|\mathcal{S}| > \frac{m}{2}$.

From every cluster $C \in \mathcal{S}$ we fix an arbitrary point and denote it by p_C. Let $R := \operatorname{cost}(\mathcal{C}_m)$. Then the distance from p_C to any $q \in C$ is at most R and we get $C - p_C \subset \mathrm{B}_R^d(0)$.

A ball of radius R can be covered by ρ balls of radius λR (see [12]). Hence, there exist $y_1, \ldots, y_\rho \in \mathbb{R}^d$ with $\mathrm{B}_R^d(0) \subseteq \bigcup_{i=1}^{\rho} \mathrm{B}_{\lambda R}^d(y_i)$. For $C \in \mathcal{S}$ we call the set $\operatorname{Conf}(C) := \{y_i \mid 1 \leq i \leq \rho \text{ and } \mathrm{B}_{\lambda R}^d(y_i) \cap (C - p_C) \neq \varnothing\}$ the configuration of C. That is, we indentify each cluster $C \in \mathcal{S}$ with the subset of the balls $\mathrm{B}_{\lambda R}^d(y_1), \ldots, \mathrm{B}_{\lambda R}^d(y_\rho)$ that intersect with $C - p_C$. Note that no cluster from $C \in \mathcal{S}$ has an empty configuration. The number of possible configurations can be upper bounded by 2^ρ. With $|\mathcal{S}| > \frac{m}{2}$ it follows that there exist $j > \frac{m}{2^{\rho+1}}$ distinct clusters $C_1, \ldots, C_j \in \mathcal{S}$ with the same configuration. Using $m > 2^{\rho+1}k$ we deduce $j > k$.

Let $P := \{p_{C_1}, \ldots, p_{C_j}\}$. Since X is (k,r)-coverable, so is $P \subset X$. Therefore, by Lemma 3, there exist distinct $a, b \in \{1, \ldots, j\}$ such that $\|p_{C_a} - p_{C_b}\| \leq 4r \sqrt[d]{\frac{2^{\rho+1}k}{m}}$.

Next we want to bound the diameter of the union of the corresponding clusters C_a and C_b. The distance between any two points $u, v \in C_a$ or $u, v \in C_b$ is at most the cost of $\mathcal{C}_m$. Now let $u \in C_a$ and $v \in C_b$. Using the triangle inequality, for any $w \in \mathbb{R}^d$ we obtain $\|u - v\| \leq \|p_{C_a} - p_{C_b}\| + \|u + p_{C_b} - p_{C_a} - w\| + \|w - v\|$.

For $\|p_{C_a} - p_{C_b}\|$ we just derived an upper bound. To bound $\|u + p_{C_b} - p_{C_a} - w\|$, we let $y \in \operatorname{Conf}(C_a) = \operatorname{Conf}(C_b)$ such that $u - p_{C_a} \in \mathrm{B}_{\lambda R}^d(y)$. Furthermore, we fix $w \in C_b$ with $w - p_{C_b} \in \mathrm{B}_{\lambda R}^d(y)$. Hence, $\|u + p_{C_b} - p_{C_a} - w\| = \|u - p_{C_a} - (w - p_{C_b})\|$ can be upper bounded by $2\lambda R = 2\lambda \cdot \operatorname{cost}(\mathcal{C}_m)$. For $w \in C_b$ the distance $\|w - v\|$ is bounded by $\operatorname{diam}(C_b) \leq \operatorname{cost}(\mathcal{C}_m)$. We conclude that merging clusters C_a and C_b results in a cluster whose diameter can be upper bounded by

$$\operatorname{diam}(C_a \cup C_b) < (1+2\lambda) \cdot \operatorname{cost}(\mathcal{C}_m) + 4r \sqrt[d]{\frac{2^{\rho+1}k}{m}}.$$

Using Observation 4 and the fact that C_a and C_b are part of the clustering $\mathcal{C}_{t+1}$, we can upper bound the cost of $\mathcal{C}_t$ by $\operatorname{cost}(\mathcal{C}_t) \leq \operatorname{diam}(C_a \cup C_b)$. ◀

Note that the parameter λ from Lemma 7 establishes a trade-off between the two terms on the right-hand side of Inequality 1. To complete the analysis of the first stage, we have to carefully choose λ. In the proof of the following lemma we use $\lambda = {}^{\ln \frac{4}{3}}/_{4d}$ and apply Lemma 7 for $\left\lceil \log_{\frac{4}{3}} \frac{|X|}{2^{\sigma+1}k} \right\rceil$ consecutive phases, where $\sigma = (42d)^d$. Then, we are able to upper bound the total increase of the cost by a term that is linear in d and r and independent of $|X|$ and k. The number of remaining clusters is independent of the number of input points $|X|$ and depends only on the dimension d and the desired number of clusters k.

▶ **Lemma 8.** *Let $2^{\sigma+1}k < |X|$ for $\sigma = (42d)^d$. Then on input X Algorithm 1 computes a clustering $\mathcal{C}_{2^{\sigma+1}k}$ with $\operatorname{cost}(\mathcal{C}_{2^{\sigma+1}k}) < (28d+4)\cdot r$.*

Proof. Let $u := \left\lceil \log_{\frac{3}{4}} \frac{2^{\sigma+1}k}{|X|} \right\rceil$ and define $m_i := \left\lceil \left(\frac{3}{4}\right)^i |X| \right\rceil$ for all $i = 0, \ldots, u$. Furthermore let $\lambda = \ln\frac{4}{3}/4d$. This implies $\rho \le \sigma$ for the parameter ρ of Lemma 7. Then $m_u \le 2^{\sigma+1}k$ and $m_i > 2^{\sigma+1}k \ge 2^{\rho+1}k$ for all $i = 0, \ldots, u-1$. We apply Lemma 7 with $m = m_i$ for all $i = 0, \ldots, u-1$. Since $\left\lfloor \frac{3m_i}{4} \right\rfloor \le m_{i+1}$ and Algorithm 1 uses a greedy strategy we deduce $\operatorname{cost}(\mathcal{C}_{m_{i+1}}) \le \operatorname{cost}(\mathcal{C}_{\lfloor \frac{3m_i}{4} \rfloor})$ for all $i = 0, \ldots, u-1$. Using $\operatorname{cost}(\mathcal{C}_{2^{\sigma+1}k}) \le \operatorname{cost}(\mathcal{C}_{m_u})$ and $\operatorname{cost}(\mathcal{C}_{m_0}) = 0$ we get

$$\begin{aligned}\operatorname{cost}(\mathcal{C}_{2^{\sigma+1}k}) &< \sum_{i=0}^{u-1} \left((1+2\lambda)^i \cdot 4r \sqrt[d]{\frac{2^{\sigma+1}k}{\left(\frac{3}{4}\right)^{u-1-i}|X|}} \right) \\ &= 4r \sqrt[d]{\frac{2^{\sigma+1}k}{\left(\frac{3}{4}\right)^{u-1}|X|}} \cdot \sum_{i=0}^{u-1} \left((1+2\lambda)^i \cdot \sqrt[d]{\left(\frac{3}{4}\right)^i} \right).\end{aligned}$$

Using $u - 1 < \log_{\frac{3}{4}} \frac{2^{\sigma+1}k}{|X|}$ we get

$$\operatorname{cost}(\mathcal{C}_{2^{\sigma+1}k}) < 4r \sum_{i=0}^{u-1} \left(\frac{1+2\lambda}{\sqrt[d]{\frac{4}{3}}} \right)^i. \tag{2}$$

By taking only the first two terms of the series expansion of the exponential function we get $1 + 2\lambda = 1 + \frac{\ln\frac{4}{3}}{2d} < e^{\frac{\ln\frac{4}{3}}{2d}} = \sqrt[2d]{\frac{4}{3}}$. Substituting this bound into Inequality (2) and extending the sum gives

$$\operatorname{cost}(\mathcal{C}_{2^{\sigma+1}k}) < 4r \sum_{i=0}^{\infty} \left(\frac{1}{\sqrt[2d]{\frac{4}{3}}} \right)^i < 4r \sum_{i=0}^{\infty} \left(\frac{1}{1+2\lambda} \right)^i.$$

Solving the geometric series leads to

$$\operatorname{cost}(\mathcal{C}_{2^{\sigma+1}k}) < 4r \left(\frac{1}{2\lambda} + 1 \right) < (28d+4)\cdot r.$$

◀

3.3 Stage two

The second stage covers the remaining merge steps until Algorithm 1 computes the clustering $\mathcal{C}_{2k}$. The following lemma is the analogon of Lemma 8. Again, the proof subdivides the merge steps into phases of one fourth of the remaining steps. However, compared to stage one, the analysis of a single phase yields a weaker bound. The proof can be found in the full version of this paper [1].

▶ **Lemma 9.** *Let $n \in \mathbb{N}$ with $n \le 2^{\sigma+1}k$ and $2k < n \le |X|$ for $\sigma = (42d)^d$. Then on input X Algorithm 1 computes a clustering $\mathcal{C}_{2k}$ with*

$$\operatorname{cost}(\mathcal{C}_{2k}) < 2^{3\sigma} \left(\operatorname{cost}(\mathcal{C}_n) + 2r \right).$$

Proposition 6 follows immediately by combining Lemma 8 and Lemma 9.

3.4 Connected instances

For the analysis of the two stages in Section 3.1 we use arguments that are only applicable if there are enough clusters left (at least $2k$ in case of stage two). To analyze the remaining merge steps, we show that it is sufficient to analyze Algorithm 1 on a subset $Y \subseteq X$ satisfying a certain connectivity property. Using this property we are able to apply a combinatorial approach that relies on the number of merge steps left. This introduces the $O(\log k)$ term to the approximation factor of our main result.

We start by defining the connectivity property that will be used to relate clusters to an optimal k-clustering.

▶ **Definition 10.** Let $Z \subseteq \mathbb{R}^d$ and $r \in \mathbb{R}$. Two sets $A, B \subseteq \mathbb{R}^d$ are called (Z, r)-*connected* if there exists a $z \in Z$ with $\mathrm{B}_r^d(z) \cap A \neq \varnothing$ and $\mathrm{B}_r^d(z) \cap B \neq \varnothing$.

Note that for any two (Z, r)-connected clusters A, B we have

$$\operatorname{diam}(A \cup B) \leq \operatorname{diam}(A) + \operatorname{diam}(B) + 2r. \tag{3}$$

Next, we show that for any input set X we can bound the cost of the k-clustering computed by Algorithm 1 by the cost of the ℓ-clustering computed by the algorithm on a connected subset $Y \subseteq X$ for a proper $\ell \leq k$. Recall that by our convention from the beginning of Section 3, the clusterings computed by Algorithm 1 on a particular input set are uniquely determined.

▶ **Lemma 11.** *Let $X \subset \mathbb{R}^d$ be finite and $k \in \mathbb{N}$ with $k \leq |X|$. Then there exists a subset $Y \subseteq X$, a number $\ell \in \mathbb{N}$ with $\ell \leq \min(k, |Y|)$, and a set $Z \subset \mathbb{R}^d$ with $|Z| = \ell$ such that:*

1. *Y is (ℓ, r)-coverable;*
2. *$\operatorname{cost}(\mathcal{C}_k) \leq \operatorname{cost}(\mathcal{P}_\ell)$;*
3. *For all $n \in \mathbb{N}$ with $\ell + 1 \leq n \leq |Y|$, every cluster in $\mathcal{P}_n$ is (Z, r)-connected to another cluster in $\mathcal{P}_n$.*

Here, the collection $\mathcal{P}_1, \ldots, \mathcal{P}_{|Y|}$ denotes the hierarchical clustering computed by Algorithm 1 on input Y.

Proof. To define Y, Z, and ℓ we consider the $(k+1)$-clustering computed by Algorithm 1 on input X. We know that $X = \bigcup_{A \in \mathcal{C}_{k+1}} A$ is (k, r)-coverable. Let $E \subseteq \mathcal{C}_{k+1}$ be a minimal subset such that $\bigcup_{A \in E} A$ is $(|E| - 1, r)$-coverable, i.e., for all sets $F \subseteq \mathcal{C}_{k+1}$ with $|F| < |E|$ the union $\bigcup_{A \in F} A$ is not $(|F| - 1, r)$-coverable. Since a set F of size 1 cannot be $(|F| - 1, r)$-coverable, we get $|E| \geq 2$.

Let $Y := \bigcup_{A \in E} A$ and $\ell := |E| - 1$. Then $\ell \leq k$ and Y is (ℓ, r)-coverable. Thus, we can define $Z \subset \mathbb{R}^d$ with $|Z| = \ell$ and $Y \subset \bigcup_{z \in Z} \mathrm{B}_r^d(z)$. Furthermore, we let $\mathcal{P}_1, \ldots, \mathcal{P}_{|Y|}$ be the hierarchical clustering computed by Algorithm 1 on input Y.

Since Y is the union of the clusters from $E \subseteq \mathcal{C}_{k+1}$, each merge step between the computation of $\mathcal{C}_{|X|}$ and $\mathcal{C}_{k+1}$ merges either two clusters $A, B \subset Y$ or two clusters $A, B \subset X \setminus Y$. The merge steps inside $X \setminus Y$ have no influence on the clusters inside Y. Furthermore, the merge steps inside Y would be the same in the absence of the clusters inside $X \setminus Y$. Therefore, on input Y Algorithm 1 computes the $(\ell + 1)$-clustering $\mathcal{P}_{\ell+1} = E = \mathcal{C}_{k+1} \cap 2^Y$. Thus, $\mathcal{P}_{\ell+1} \subseteq \mathcal{C}_{k+1}$.

To compute $\mathcal{P}_\ell$, Algorithm 1 on input Y merges two clusters from $\mathcal{P}_{\ell+1}$ that minimize the diameter of the resulting cluster. Analogously, Algorithm 1 on input X merges two clusters from $\mathcal{C}_{k+1}$ to compute $\mathcal{C}_k$. Since $\mathcal{P}_{\ell+1} \subseteq \mathcal{C}_{k+1}$, Observation 4 implies $\operatorname{cost}(\mathcal{C}_k) \leq \operatorname{cost}(\mathcal{P}_\ell)$.

It remains to show that for all $n \in \mathbb{N}$ with $\ell + 1 \leq n \leq |Y|$ it holds that every cluster in $\mathcal{P}_n$ is (Z, r)-connected to another cluster in $\mathcal{P}_n$. We first show the property for $n = \ell + 1$.

For $\ell = 1$ this follows from the fact that $\mathrm{B}_r^d(z)$ with $Z = \{z\}$ has to contain both clusters from $\mathcal{P}_2$. For $\ell > 1$ we are otherwise able to remove one cluster from $\mathcal{P}_{\ell+1}$ and get ℓ clusters whose union is $(\ell-1, r)$-coverable. This contradicts the definition of $E = \mathcal{P}_{\ell+1}$ as a minimal subset with this property.

To prove 3. for general n, let $C_1 \in \mathcal{P}_n$ and $z \in Z$ with $\mathrm{B}_r^d(z) \cap C_1 \neq \varnothing$. There exists a unique cluster $\tilde{C}_1 \in \mathcal{P}_{\ell+1}$ with $C_1 \subseteq \tilde{C}_1$. Then we have $\mathrm{B}_r^d(z) \cap \tilde{C}_1 \neq \varnothing$. However, $\mathrm{B}_r^d(z)$ has to intersect with at least two clusters from $\mathcal{P}_{\ell+1}$. Thus, there exists another cluster $\tilde{C}_2 \in \mathcal{P}_{\ell+1}$ with $\mathrm{B}_r^d(z) \cap \tilde{C}_2 \neq \varnothing$. Since every cluster from $\mathcal{P}_{\ell+1}$ is a union of clusters from $\mathcal{P}_n$, there exists at least one cluster $C_2 \in \mathcal{P}_n$ with $C_2 \subseteq \tilde{C}_2$ and $\mathrm{B}_r^d(z) \cap C_2 \neq \varnothing$. ◀

3.5 Analysis of the remaining merge steps

Let Y, Z, ℓ, and $\mathcal{P}_1, \ldots, \mathcal{P}_{|Y|}$ be as given in Lemma 11. Then, Proposition 6 can be used to obtain an upper bound for the cost of $\mathcal{P}_{2\ell}$. In the following, we analyze the merge steps leading from $\mathcal{P}_{2\ell}$ to $\mathcal{P}_{\ell+1}$ and show how to obtain an upper bound for the cost of $\mathcal{P}_{\ell+1}$. As in Section 3.1, we analyze the merge steps in phases. The following lemma is used to bound the increase of the cost during a single phase.

▶ **Lemma 12.** *Let $m, n \in \mathbb{N}$ with $n \leq 2\ell$ and $\ell < m \leq n \leq |Y|$. If there are no two (Z, r)-connected clusters in $\mathcal{P}_m \cap \mathcal{P}_n$, it holds*

$$\mathrm{cost}(\mathcal{P}_{\lfloor \frac{m+\ell}{2} \rfloor}) \leq \mathrm{cost}(\mathcal{P}_m) + 2 \cdot (\mathrm{cost}(\mathcal{P}_n) + 2r)\,.$$

Proof. We show that there exist at least $m - \ell$ disjoint pairs of clusters from $\mathcal{P}_m$ such that the diameter of their union can be upper bounded by $\mathrm{cost}(\mathcal{P}_m) + 2 \cdot (\mathrm{cost}(\mathcal{P}_n) + 2r)$. By Observation 4, this upper bounds the cost of the computed clusterings as long as such a pair of clusters remains. Then the lemma follows from the fact that in each iteration of its loop the algorithm can destroy at most two of these pairs.

To bound the number of such pairs of clusters we start with a structural observation. Let $\mathcal{S} := \mathcal{P}_m \cap \mathcal{P}_n$ be the set of clusters from $\mathcal{P}_n$ that still exist in $\mathcal{P}_m$. By our definiton of Y, Z, and ℓ we find that any cluster $A \in \mathcal{S} \subseteq \mathcal{P}_m$ is (Z, r)-connected to another cluster $B \in \mathcal{P}_m$. If we assume that there are no two (Z, r)-connected clusters in $\mathcal{S}$, this implies $B \in \mathcal{P}_m \setminus \mathcal{S}$. Thus, using $A \in \mathcal{P}_n$, $B \in \mathcal{P}_m$, and Equation (3) the diameter of $A \cup B$ can be bounded by

$$\mathrm{diam}(A \cup B) \leq \mathrm{cost}(\mathcal{P}_m) + \mathrm{cost}(\mathcal{P}_n) + 2r. \tag{4}$$

Moreover, using similar argument, if two clusters $A_1, A_2 \in \mathcal{S} \subseteq \mathcal{P}_n$ are (Z, r)-connected to the same cluster $B \in \mathcal{P}_m \setminus \mathcal{S}$ we can bound the diameter of $A_1 \cup A_2$ by

$$\mathrm{diam}(A_1 \cup A_2) \leq \mathrm{cost}(\mathcal{P}_m) + 2 \cdot (\mathrm{cost}(\mathcal{P}_n) + 2r). \tag{5}$$

Next we show that there exist at least $\left\lceil \frac{|\mathcal{S}|}{2} \right\rceil$ disjoint pairs of clusters from $\mathcal{P}_m$ such that the diameter of their union can be bounded either by Inequality (4) or by Inequality (5). To do so, we first consider the pairs of clusters from $\mathcal{S}$ that are (Z, r)-connected to the same cluster from $\mathcal{P}_m \setminus \mathcal{S}$ until no candidates are left. For these pairs we can bound the diameter of their union by Inequality (5). Then, each cluster from $\mathcal{P}_m \setminus \mathcal{S}$ is (Z, r)-connected to at most one of the remaining clusters from $\mathcal{S}$. Thus, each remaining cluster $A \in \mathcal{S}$ can be paired with a different cluster $B \in \mathcal{P}_m \setminus \mathcal{S}$ such that A and B are (Z, r)-connected. For these pairs we can bound the diameter of their union by Inequality (4). Since for all pairs

either one or both of the clusters come from the set $\mathcal{S}$, we can lower bound the number of pairs by $\left\lceil \frac{|\mathcal{S}|}{2} \right\rceil$.

To complete the proof, we show that $m - \ell \leq \left\lceil \frac{|\mathcal{S}|}{2} \right\rceil$. In each iteration of its loop the algorithm can merge at most two clusters from $\mathcal{P}_n$. Therefore, to compute $\mathcal{P}_m$, at least $\left\lceil \frac{n-|\mathcal{S}|}{2} \right\rceil$ merge steps must have been done since the computation of $\mathcal{P}_n$. Hence, $m \leq n - \left\lceil \frac{n-|\mathcal{S}|}{2} \right\rceil \leq \frac{n}{2} + \frac{|\mathcal{S}|}{2}$. Using $n \leq 2\ell$ we get $m - \ell \leq \frac{|\mathcal{S}|}{2}$. ◀

▶ **Lemma 13.** *Let $n \in \mathbb{N}$ with $n \leq 2\ell$ and $\ell < n \leq |Y|$. Then*

$$\operatorname{cost}(\mathcal{P}_{\ell+1}) < 2(\log_2 \ell + 2) \cdot (\operatorname{cost}(\mathcal{P}_n) + 2r)\,.$$

Proof. For $n = \ell + 1$ there is nothing to show. Hence, assume $n > \ell + 1$. Then by definition of Z there exist two (Z, r)-connected clusters in $\mathcal{P}_n$. Now let $\tilde{n} \in \mathbb{N}$ with $\tilde{n} < n$ be maximal such that no two (Z, r)-connected clusters exist in $\mathcal{P}_{\tilde{n}} \cap \mathcal{P}_n$. The number $\tilde{n}$ is well-defined since at least the set $\mathcal{P}_1$ does not contain two clusters at all. It follows that the same holds for all $m \in \mathbb{N}$ with $m \leq \tilde{n}$. We conclude that Lemma 12 is applicable for all $m \leq \tilde{n}$ with $\ell < m$.

By the definition of $\tilde{n}$ there still exist at least two (Z, r)-connected clusters in $\mathcal{P}_{\tilde{n}+1} \cap \mathcal{P}_n$. Then, Observation 4 implies

$$\operatorname{cost}(\mathcal{P}_{\tilde{n}}) \leq 2 \cdot \operatorname{cost}(\mathcal{P}_n) + 2r. \tag{6}$$

If $\tilde{n} \leq \ell + 1$ then Inequality (6) proves the lemma. For $\tilde{n} > \ell + 1$ let $u := \lceil \log_2(\tilde{n} - \ell) \rceil$ and define $m_i := \left\lceil \left(\frac{1}{2}\right)^i (\tilde{n} - \ell) + \ell \right\rceil > \ell$ for all $i = 0, \ldots, u$. We apply Lemma 12 with $m = m_i$ for all $i = 0, \ldots, u-1$. Since $\left\lfloor \frac{m_i + \ell}{2} \right\rfloor \leq m_{i+1}$ and Algorithm 1 uses a greedy strategy we deduce $\operatorname{cost}(\mathcal{C}_{m_{i+1}}) \leq \operatorname{cost}(\mathcal{C}_{\left\lfloor \frac{m_i+\ell}{2} \right\rfloor})$ for all $i = 0, \ldots, u-1$. By summation over all i, we get

$$\operatorname{cost}(\mathcal{P}_{m_u}) < \operatorname{cost}(\mathcal{P}_{\tilde{n}}) + 2u \cdot (\operatorname{cost}(\mathcal{P}_n) + 2r) \overset{(6)}{<} 2(u+1) \cdot (\operatorname{cost}(\mathcal{P}_n) + 2r)\,.$$

Since $\tilde{n} < 2\ell$ we get $u < \log_2 \ell + 1$ and the lemma follows using $m_u = \ell + 1$. ◀

The following lemma finishes the analysis except for the last merge step.

▶ **Lemma 14.** *Let $Y \subset \mathbb{R}^d$ be finite and $\ell \leq |Y|$ such that Y is (ℓ, r)-coverable. Furthermore, let $Z \subset \mathbb{R}^d$ with $|Z| = \ell$ such that for all $n \in \mathbb{N}$ with $\ell + 1 \leq n \leq |Y|$ every cluster in $\mathcal{P}_n$ is (Z, r)-connected to another cluster in $\mathcal{P}_n$, where $\mathcal{P}_1, \ldots, \mathcal{P}_{|Y|}$ denotes the hierarchical clustering computed by Algorithm 1 on input Y. Then $\operatorname{cost}(\mathcal{P}_{\ell+1}) < 2(\log_2 \ell + 2) \cdot \left(2^{3\sigma}(28d + 6) + 2\right) \cdot r$ for $\sigma = (42d)^d$.*

Proof. Let $n := \min(|Y|, 2\ell)$. Then, using Proposition 6, we get $\operatorname{cost}(\mathcal{P}_n) < 2^{3\sigma}(28d + 6) \cdot r$. The lemma follows by using this bound in combination with Lemma 13. ◀

3.6 Proof of Theorem 5

Using Lemma 11 we know that there is a subset $Y \subseteq X$, a number $\ell \leq k$, and a hierarchical clustering $\mathcal{P}_1, \ldots, \mathcal{P}_{|Y|}$ of Y with $\operatorname{cost}(\mathcal{C}_k) \leq \operatorname{cost}(\mathcal{P}_\ell)$. Furthermore, there is a set $Z \subset \mathbb{R}^d$ such that every cluster from $\mathcal{P}_{\ell+1}$ is (Z, r)-connected to another cluster in $\mathcal{P}_{\ell+1}$. Thus, $\mathcal{P}_{\ell+1}$ contains two clusters A, B that intersect with the same ball of radius r. Hence $\operatorname{cost}(\mathcal{C}_k) \leq \operatorname{diam}(A \cup B) \leq 2 \cdot \operatorname{cost}(\mathcal{P}_{\ell+1}) + 2r$. The theorem follows using Lemma 14 and $\ell \leq k$. ◀

4 Further results and open problems

4.1 Lower bounds

For $d = 1$ we are able to show that Algorithm 1 computes an approximation to Problem 1 with an approximation factor between 2.5 and 3. We even know that for any input set $X \subset \mathbb{R}$ the approximation factor of the computed solution is strictly below 3. However, we do not show an approximation factor of $3 - \epsilon$ for some $\epsilon > 0$. The proof of the upper bound is very technical, makes extensive use of the total order of the real numbers, and is too long to be included in this extended abstract.

Furthermore, we know that the dimension d has an impact on the approximation factor of Algorithm 1. This follows from a 2-dimensional input set yielding a lower bound of 3 for the metric based on the ℓ_∞-norm. Note that this exceeds the upper bound from the one-dimensional case. Furthermore, for the Euclidean metric we know a 3-dimensional input set giving a lower bound of 2.56, thus exceeding our lower bound from the one-dimensional case.

Moreover, we can show that there exist input instances such that Algorithm 1 computes an approximation to Problem 1 with an approximation factor of $\Omega(\sqrt[p]{\log k})$ for metrics based on an ℓ_p-norm ($1 \leq p < \infty$) and $\Omega(\log k)$ for the metric based on the ℓ_∞-norm. In case of the ℓ_1- and the ℓ_∞-norm this matches the already known lower bound [5] that has been shown using a rather artificial metric. However, in our instances the dimension d is not fixed but depends on k.

All lower bounds mentioned above are proven in the full version of this paper [1].

4.2 Related clustering problems

As mentioned in the introduction, the cost of optimal solutions to the diameter k-clustering problem, the k-center problem, and the discrete k-center problem are within a factor of 2 from each other. That is, Algorithm 1 computes an $O(\log k)$-approximation for all three problems.

In case of the k-center problem and the discrete k-center problem, our techniques can be applied in a simplified way and yield stronger bounds. Here, we consider the agglomerative algorithm that minimizes the (discrete) k-center cost function in every merge step. In case of the k-center problem we are able to show an upper bound that is logarithmic in k and single exponential in d. More precisely, the dependency on d in the upper bound for the cost of the $2k$-clustering from doubly exponential to only single exponential. Mainly this is achieved because the analysis no longer requires configurations of clusters.

In case of the discrete k-center problem we are able to show an upper bound of $20d + 2\log_2 k + 4$ for the approximation factor. Here, the analysis benefits from the fact that we are able to bound the increase of the cost in each phase of the second stage by a term that is only additive.

The lower bound of $\Omega(\sqrt[p]{\log k})$ for any ℓ_p-norm and $\Omega(\log k)$ for the ℓ_∞-norm can be adopted to the discrete k-center problem (see full version of this paper [1]). In particular, in case of the ℓ_2-norm we obtain an instance in dimension $O(\log^3 k)$ for which we can show a lower bound of $\Omega(\sqrt{\log k})$. Applying the upper bound of $20d + 2\log_2 k + 4$ to this instance we see that Algorithm 1 computes a k-clustering whose cost is $O(\log^3 k)$ times the cost of an optimal solution. This implies that the approximation factor of Algorithm 1 cannot be simultaneously independent of d and $\log k$. More precisely, the approximation factor cannot be sublinear in $\sqrt[6]{d}$ and in $\sqrt{\log k}$.

4.3 Open problems

The main open problems our contribution raises are:

- Can the doubly exponential dependence on d in Theorem 5 be improved?
- Are the different dependencies on d in the approximation factors for the discrete k-center problem, the k-center problem, and the diameter k-clustering problem due to the limitations of our analysis or are they inherent to these problems?
- Can our results be extended to more general distance measures?
- Can the lower bounds for ℓ_p-metrics with $1 < p < \infty$ be improved to $\Omega(\log k)$, matching the lower bound from [5] for all ℓ_p-norms?

References

1 M. R. Ackermann, J. Blömer, D. Kuntze, and C. Sohler. Analysis of Agglomerative Clustering. *CoRR: Computing Research Repository*, 2010. `arXiv:1012.3697 [cs.DS]`.

2 M. Badoiu, S. Har-Peled, and P. Indyk. Approximate Clustering via Core-Sets. In *Proceedings of STOC '02*, pages 250–257, 2002.

3 A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic Clustering of the Web. In *Selected papers from the sixth intern. conf. on WWW*, pages 1157–1166, Essex, UK, 1997. Elsevier Science Publishers Ltd.

4 M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental Clustering and Dynamic Information Retrieval. *SIAM J. Comput.*, 33:1417–1440, June 2004.

5 S. Dasgupta and P. M. Long. Performance Guarantees for Hierarchical Clustering. *JCSS: Journal of Computer and System Sciences*, 70(4):555–569, 2005.

6 M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS: Proceedings of the National Academy of Sciences*, 95(25):14863–14868, December 1998.

7 T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *Proceedings of STOC '88*, pages 434–444, New York, NY, USA, 1988. ACM.

8 K. Florek, J. Lukaszewicz, J. Perkal, H. Steinhaus, and S. Zubrzycki. Sur la liaison et la division des points d'un ensemble fini. *Colloqium Math.*, 2:282–285, 1951.

9 T. F. Gonzalez. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science*, 38:293–306, 1985.

10 K. Lee, J. Kim, K. Kwon, Y. Han, and S. Kim. DDoS attack detection method using cluster analysis. *Expert Systems with Applications*, 34(3):1659–1665, 2008.

11 L. L. McQuitty. Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies. *Educational and Psychological Measurement*, 17:207–209, 1957.

12 M. Naszódi. Covering a Set with Homothets of a Convex Body. *Positivity*, 2009.

13 F. Pereira, N. Tishby, and L. Lee. Distributional Clustering of English Words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, 1993.

14 P. H. A. Sneath and R. R. Sokal. *Numerical taxonomy: the principles and practice of numerical classification*. W. H. Freeman, 1973.