Electronic Colloquium on Computational Complexity, Report No. 25 (2002)

Polynomial Time Approximation Schemes for Metric Min-Sum Clustering

W. Fernandez de la Vega [*] Marek Karpinski [†] Claire Kenyon[‡]
Yuval Rabani[§]

Abstract

We give polynomial time approximation schemes for the problem of partitioning an input set of n points into a fixed number k of clusters so as to minimize the sum over all clusters of the total pairwise distances in a cluster. Our algorithms work for arbitrary metric spaces as well as for points in $\mathbb{R}^d$ where the distance between two points x, y is measured by $\|x-y\|_2^2$ (notice that $(\mathbb{R}^d, \|\cdot\|_2^2)$ is not a metric space). Our algorithms can be modified to handle other objective functions, such as minimizing the sum over all clusters of the total distance to the best choice for cluster center.

[*] Email:lalo@lri.lri.fr LRI, CNRS UMR 8623, Université Paris-Sud, France.

[†] Email: marek@cs.uni-bonn.de, Dept. of Computer Science, University of Bonn

[‡] Email:kenyon@lri.fr. LRI, CNRS UMR 8623, Université Paris-Sud, France.

[§] Computer Science Department, Technion — IIT, Haifa 32000, Israel. Work at the Technion supported by Israel Science Foundation grant number 386/99, by US-Israel Binational Science Foundation grant number 99-00217, by the European Commission Fifth Framework Programme Thematic Networks contract number IST-2001-32007 (APPOL II), and by the Fund for the Promotion of Research at the Technion. Email: rabani@cs.technion.ac.il

ISSN 1433-8092

1 Introduction

Problem statement and motivation. The partition of a data set into a small number of *clusters*, each containing a set of seemingly related items, plays an increasingly crucial role in emerging applications such as web search and classification [12, 50], interpretation of experimental data in molecular biology and astrophysics [41, 57, 48], or market segmentation [45]. This task raises several fundamental questions about representing data, measuring affinity, estimating clustering quality, and designing efficient algorithms. For example, when searching or mining massive unstructured data sets, data items are often processed and represented as points in a high dimensional [1] space $\mathbb{R}^d$, where some standard distance function measures affinity (see, for example, [20, 58, 26, 12]).

This paper deals with the question of designing good algorithms for an attractive criterion for clustering quality in such a setting. More specifically, we consider a set V of n points endowed with a distance function $\delta : V \times V \to \mathbb{R}$. These points have to be partitioned into a fixed number k of subsets $C_1, C_2, \ldots, C_k$ so as to minimize the cost of the partition, which is defined to be the sum over all clusters of the total pairwise distances in a cluster. We refer to this problem as the Min-Sum All-Pairs k-Clustering problem. Our algorithms deal with the case that δ is an arbitrary metric (including, in particular, points in $\mathbb{R}^d$ with distances induced by some norm). We also handle the non-metric case of points in $\mathbb{R}^d$ where the distance between two points x, y is measured by $\delta(x,y) = \|x-y\|_2^2$. In the latter case, our algorithms can be modified to deal with other objective functions, including the problem of Min-Sum Median k-Clustering, where the cost of a clustering is the sum over all clusters of the total distances between cluster points and the best choice for a cluster center. All optimization problems that we consider are NP-hard to solve exactly even for $k = 2$.

Our results. For the Min-Sum All-Pairs objective function, we present algorithms for every k and for every $\epsilon > 0$ that compute a partition into k clusters $C_1, C_2, \ldots, C_k$ of cost at most $1+\epsilon$ times the cost of an optimum partition. In the metric case the algorithm is randomized and its running time is $O(n^{2k} + n^{k+1}2^{\tilde{O}(1/\epsilon^{3k+1})})$. In the case of the square of Euclidean distance, the algorithms are deterministic, and their running time is $n^{O(k/\epsilon^4)}$. Our algorithms can be modified to output, for all $\zeta > 0$, a clustering that excludes at most ζn outliers and has cost at most $1+\epsilon$ times the optimum cost. In the case of the square of Euclidean distance, we can do this in probabilistic time $O(f(k,\epsilon,\zeta) \cdot n^3 \log n)$, where f grows (rapidly) with k, $\frac{1}{\epsilon}$, and $\frac{1}{\zeta}$.

The Min-Sum Median objective function can be optimized in polynomial time for fixed k in finite metrics, because the number of choices for centers is polynomial. However, if the points are located in a larger space, such as $\mathbb{R}^d$, and the centers can be picked from this larger space, the problem may become hard. For points in $\mathbb{R}^d$ with distances measured by the square of Euclidean distance, we give Min-Sum Median algorithms that partition all points into k clusters of cost at most $1+\epsilon$ of the optimum cost in probabilistic time $O(g(k,\epsilon) \cdot n \cdot (\log n)^k)$, where g grows (rapidly) with k and $\frac{1}{\epsilon}$. Some of our ideas can be modified trivially to derive polynomial time approximation schemes for other objective functions, such as minimizing

[1] By "high dimensional" we mean that the dimension d should be treated as part of the input and not as a constant.

the maximum radius of a cluster. We do not elaborate on these modifications.

Related work. Schulman [56] initiated the study of approximation algorithms for Min-Sum All-Pairs k-Clustering. He gave probabilistic algorithms for clustering points in $\mathbb{R}^d$ with distance measured by the square of Euclidean distance. (Thus he also handled other interesting cases of metrics that embed isometrically into this distance space, such as Euclidean metrics or L^1 metrics.) His algorithms find a clustering such that either its cost is within a factor of $1+\epsilon$ of the optimum cost, or it can be converted into an optimum clustering by changing the assignment of at most an ϵ fraction of the points. The running time is linear if $d = o(\log n / \log\log n)$ and otherwise the running time is $n^{O(\log\log n)}$. Thus our results improve and extend Schulman's result, giving a true polynomial time approximation scheme for arbitrary dimension.

Earlier, Fernandez de la Vega and Kenyon [24] presented a polynomial time approximation scheme for Metric Max Cut, an objective function that is the complement of Metric Min-Sum All-Pairs 2-clustering. Indyk [35] later used this algorithm to derive a polynomial time approximation scheme for the latter problem. Thus our results extend Indyk's result to the case of arbitrary fixed k. Bartal, Charikar, and Raz [11] gave a polynomial time approximation algorithm with polylogarithmic performance guarantees for Metric Min-Sum All-Pairs k-Clustering where k is arbitrary (i.e., part of the input).

As mentioned above, instances of Min-Sum Median k-Clustering in finite metrics with fixed k are trivially solvable in polynomial time. (For arbitrary k, the problem is APX-hard [33] and has elicited much work and progress [8, 16, 37, 15].) This is not the case in geometric settings, including the square of Euclidean distance discussed in this paper. This case was considered by Drineas, Frieze, Kannan, Vempala, and Vinay [25], who gave a 2-approximation algorithm. Ostrovsky and Rabani [52] gave a polynomial time approximation scheme for this case and other geometric settings. Our results improve significantly the running time for the square of Euclidean distance case. Recently and independently of our work, Bădoiu, Har-Peled, and Indyk [10] gave a polynomial time approximation scheme for points in Euclidean space with much improved running time (as well as results on other clustering objectives). Their algorithm and analysis are in some respects similar to our algorithm (though it handles a different distance function).

It is interesting to note that both Schulman's algorithm for Min-Sum All-Pairs Clustering and the algorithm of Fernandez de la Vega and Kenyon for Mertic Max Cut use a similar idea of sampling data points at random from a biased distribution that depends on the pairwise distances. In recent research on clustering problems, sampling has been the core idea in the design of provably good algorithms for various objective functions. Examples include [5, 3, 51].

2 Preliminaries

In this section we introduce some notation and some tools that will be used to derive and analyze our algorithms.

Throughout the paper we use V to denote the input set of points and δ to denote the distance function over pairs of points in V. The function δ can be given explicitly or implicitly (for example, if $V \subset \mathbb{R}^d$ and δ is derived from a norm on $\mathbb{R}^d$). Our time bounds count arithmetic operations and assume that computing $\delta(x,y)$ is a single operation. The reader may assume that the input is rational to avoid having to deal with unrealistic computational models. We use k, a fixed constant, to denote the desired number of clusters. We omit the ceiling notation from expressions such as $\lceil \frac{1}{\epsilon} \rceil$. Our claims and proofs can be modified trivially to account for taking the ceiling of non-integers wherever needed.

Let $X, Y \subset V$ and $x \in V$. With a slight abuse of notation, we use $\delta(x,Y)$ to denote $\sum_{y \in Y} \delta(x,y)$, and we use $\delta(X,Y)$ to denote $\sum_{x \in X} \delta(x,Y)$ (notice that $\delta(\cdot,\cdot)$ is a symmetric bilinear form but is not a distance in the power set of V). We use $\delta(X)$ to denote $\delta(X,X)$. We put $W = \delta(V)$ and $w_x = \delta(x,V)$. Finally, we denote the diameter of X by $\mathrm{diam}(X) = \max_{x,y \in X} \delta(x,y)$.

Let $C_1, C_2, \ldots, C_k$ be a partition of V into k disjoint clusters. Then, for all $i = 1, 2, \ldots, k$, we use $\mathrm{cost}(C_i)$ to denote the cost of C_i. For most of the paper, we are concerned with the *all-pairs cost* of a cluster, putting $\mathrm{cost}(C_i) = \frac{1}{2}\delta(C_i)$. In some cases, our algorithms can be modified to apply to hard cases of the *median cost* of a cluster, putting $\mathrm{cost}(C_i) = \min_{x \in \mathbb{R}^d}\{\delta(x, C_i)\}$. In both cases, the cost of the clustering is $c = \mathrm{cost}(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} \mathrm{cost}(C_i)$. We use $C_1^*, C_2^*, \ldots, C_k^*$ to denote a clustering of V of minimum cost $c^* = \mathrm{cost}(C_1^*, C_2^*, \ldots, C_k^*)$.

Our polynomial time approximation schemes handle the case where δ induces an arbitrary metric on V, as well as the non-metric case of $V \subset \mathbb{R}^d$ and $\delta(x,y) = \|x-y\|_2^2$. The former case obviously includes instances where $V \subset \mathbb{R}^d$ and $\delta(x,y) = \|x-y\|_p$ for $p \in [1,\infty)$ or $p = \infty$. Instances of points in $\mathbb{R}^d$ are computationally hard if d is part of the input.

2.1 Properties of Metric Spaces

The main property of metrics that we use is the following proposition, which follows easily from the triangle inequality.

Proposition 1. Let $X, Y, Z \subseteq V$. Then,

$$|Z|\delta(X,Y) \leq |X|\delta(Y,Z) + |Y|\delta(Z,X).$$

Proof: For every x, y, z, we have $\delta(x,y) \leq \delta(y,z) + \delta(z,y)$. Summing over $X \times Y \times Z$ gives the desired result. □

Here are some corollaries which are used in our proofs in metric space.

Corollary 2. $\mathrm{diam}(V) \leq 2W/n$.

Proof: Let x, y be such that $\mathrm{diam}(V) = \delta(x,y)$, and apply Proposition 1 to $X = \{x\}$, $Y = \{y\}$, and $Z = V$. □

Corollary 3. Let $C \subseteq V$. For every vertex $v \in C$ we have

$$\delta(v,C) \geq \frac{\delta(C)}{2|C|}.$$

Proof: Apply Proposition 1 to $X = C$, $Y = C$ and $Z = \{v\}$. □

Our approximation scheme for min-sum all-pairs clustering in metric spaces uses as a tool an approximation scheme for Metric Max-k-Cut.

Definition: The Metric Max-k-Cut problem takes as input a set V of n points from an arbitrary metric space, and outputs a partition of V into k clusters $C_1, C_2, \ldots, C_k$ so as to maximize total distance between pairs of points in different clusters, i.e.

$$\max \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \delta(C_i, C_j).$$

.

For any partition, the sum of the Max-k-Cut value and of the min-sum all-pairs clustering value equals W. Thus the same partition is optimal for both objectives.

Theorem 4 ([24, 23]). There is a polynomial time approximation scheme for Metric Max-k-Cut.

Theorem 4 is actually an easy extension of the MaxCut approximation scheme of [24]. The same reduction which is used for MaxCut also applies to Max-k-Cut, and the resulting weighted dense graph is only a variant of dense graphs in the usual sense, so that the Max-k-Cut approximation schemes for dense graphs (see [32, 7]) apply. An alternate algorithm can be found in [23].

2.2 Properties of $\|\cdot\|_2^2$

Unless otherwise specified, all subsets and multi-subsets of $\mathbb{R}^d$ that we discuss are, for simplicity, finite. For a finite set $X \subset \mathbb{R}^d$ we denote by $\text{conv}(X)$ the *convex hull* of X, i.e., $\text{conv}(X) = \{y \in \mathbb{R}^d \mid \exists \alpha \in \mathbb{R}^{|X|} \text{ such that } \alpha \geq 0 \text{ and } \|\alpha\|_1 = 1 \text{ and } y = \sum_{x \in X} \alpha_x x\}$. We associate with every y in $\text{conv}(X)$ such that $y = \sum_{x \in X} \alpha_x x$ with rational coefficients α, a multi-subset Y of X as follows. For every $x \in X$, the number n_x of copies of x in Y is defined by $\alpha_x = n_x/|Y|$, where n_x is the number of times x appears in Y. We often use $\bar{Y}$ to denote the center of gravity of Y.

The following proposition characterizes the all-pairs cost of a cluster for the case that $\delta(x, y) = \|x - y\|_2^2$.

Proposition 5. For every cluster $C \subset V$, $\text{cost}(C) = |C|\delta(C, \bar{C})$.

Proof:

$$\begin{aligned}
|C| \cdot \delta(C, \bar{C}) &= |C| \cdot \sum_{x \in C} \left(x - \frac{1}{|C|}\sum_{y \in C} y\right) \cdot \left(x - \frac{1}{|C|}\sum_{y \in C} y\right) \\
&= |C| \cdot \sum_{x \in C} \left(\|x\|_2^2 + \frac{1}{|C|^2}\sum_{y \in C}\sum_{z \in C} y \cdot z - \frac{2}{|C|}\sum_{y \in C} x \cdot y\right) \text{ by bilinearity} \\
&= |C| \cdot \sum_{x \in C} \|x\|_2^2 - \frac{1}{|C|}\sum_{x \in C}\sum_{y \in C} x \cdot y \text{ by renaming and grouping}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\sum_{x\in C}\sum_{y\in C}\left(\|x\|_2^2+\|y\|_2^2-2x\cdot y\right) \text{ by renaming}\\
&= \frac{1}{2}\sum_{x\in C}\sum_{y\in C}\|x-y\|_2^2\\
&= \mathrm{cost}(C). \qquad \square
\end{aligned}
$$

The following simple propositions will come in handy.

Proposition 6. For every multi-subset Y of $\mathbb{R}^d$, the center of gravity of Y is such that $\bar{Y} = \arg\min_{z\in\mathbb{R}^d}\{\delta(Y,z)\}$.

Proof: Let $z \in \mathbb{R}^d$ be the point that minimizes the above expression. As $\delta(Y,z) = \sum_{i=1}^d \sum_{y\in Y}(y_i - z_i)^2$, we can determine z by minimizing each coordinate separately. We have $\frac{\partial \sum_{y\in Y}(y_i-z_i)^2}{\partial z_i} = -2\sum_{y\in Y}(y_i - z_i)$. The right hand side has a single zero at $z_i = \frac{1}{|Y|}\sum_{y\in Y} y_i$. As $\frac{\partial^2 \sum_{y\in Y}(y_i-z_i)^2}{\partial z_i^2} = 2|Y| > 0$, this point is the unique global minimum. □

Proposition 7. For every $x, y, z \in \mathbb{R}^d$, $\delta(x,z) \le \delta(x,y) + \delta(y,z) + 2\sqrt{\delta(x,y)\cdot\delta(y,z)}$.

Proof: By the triangle inequality for Euclidean distance, $\sqrt{\delta(x,z)} \le \sqrt{\delta(x,y)} + \sqrt{\delta(y,z)}$. Squaring this inequality gives the desired result. □

Proposition 8. For every $x \in \mathbb{R}^d$, for every multi-subset Y of $\mathbb{R}^d$, $\delta(x,\bar{Y}) \le \frac{1}{|Y|}\delta(x,Y)$.

Proof:

$$
\begin{aligned}
\delta(x,\bar{Y}) &= \left\| x - \frac{1}{|Y|}\sum_{y\in Y} y\right\|_2^2\\
&= \left\| \frac{1}{|Y|}\sum_{y\in Y}(x-y)\right\|_2^2\\
&= \sum_{i=1}^d\left(\frac{1}{|Y|}\sum_{y\in Y}(x_i-y_i)\right)^2\\
&\le \sum_{i=1}^d \frac{1}{|Y|}\sum_{y\in Y}(x_i-y_i)^2 \qquad (1)\\
&= \frac{1}{|Y|}\sum_{y\in Y}\sum_{i=1}^d (x_i-y_i)^2\\
&= \frac{1}{|Y|}\sum_{y\in Y}\delta(x,y),
\end{aligned}
$$

where (1) follows from the Cauchy-Schwarz inequality. □

The following lemma is attributed to Maurey [53, 14, 6]. We provide a proof for completeness.

Lemma 9 (Maurey). For every positive integer d, for every $Y \subset \mathbb{R}^d$, for every $\epsilon > 0$, and for every $x \in \mathrm{conv}(Y)$, there exists a multi-subset Z of Y containing $|Z| = \frac{1}{\epsilon}$ points such that $\delta(x, \bar{Z}) \le \epsilon \cdot (\mathrm{diam}(Y))$.

Proof: Put $t = \frac{1}{\epsilon}$. As $x \in \mathrm{conv}(Y)$, it can be expressed as a convex combination $x = \sum_{y \in Y} \alpha_y y$, where the coefficients α_y are non-negative reals that sum up to 1. Pick a multiset $Z = \{z^1, z^2, \ldots, z^t\}$ at random, where the z^i-s are independent, identically distributed, random points with $\Pr[z_i = y] = \alpha_y$. Now,

$$
\begin{aligned}
E\left[\delta(x, \bar{Z})\right] &= E\left[\left\| x - \frac{1}{t}\sum_{i=1}^{t} z^i \right\|_2^2\right] \\
&= E\left[\left\| \frac{1}{t}\sum_{i=1}^{t} \left(x - z^i\right) \right\|_2^2\right] \\
&= E\left[\frac{1}{t^2}\sum_{i=1}^{t}\sum_{j=1}^{t} \left(x - z^i\right) \cdot \left(x - z^j\right)\right] \\
&= \frac{1}{t^2}\sum_{i=1}^{t}\left(E\left[\|x - z^i\|_2^2\right] + \sum_{j \ne i} E\left[\left(x - z^i\right) \cdot \left(x - z^j\right)\right]\right) \qquad (2) \\
&= \frac{1}{t^2}\sum_{i=1}^{t} E\left[\|x - z^i\|_2^2\right] \qquad (3) \\
&\le \frac{1}{t}\mathrm{diam}(Y),
\end{aligned}
$$

where (2) follows from the linearity of expectation, and (3) follows from the fact that for every $i \ne j$, z^i and z^j are independent, so $E\left[(x - z^i) \cdot (x - z^j)\right] = \sum_{l=1}^{d} E\left[(x_l - z^i_l)\right] E\left[(x_l - z^j_l)\right] = 0$. As $E\left[\delta(x, \bar{Z})\right] \le \frac{1}{t}\mathrm{diam}(Y)$, there exists a choice of Z such that $\delta(x, \bar{Z}) \le \frac{1}{t}\mathrm{diam}(Y)$. □

Lemma 9 can be used to derive a high-probability argument as follows.

Lemma 10. There exists a universal constant κ such that for every integer d, for every $Y \subset \mathbb{R}^d$, for every $\epsilon > 0$, and for every $\rho > 0$, a multi-subset Z of Y that is generated by taking a sample of $\kappa \cdot \frac{1}{\epsilon^2} \cdot \log \frac{1}{\rho}$ independent, uniformly distributed, points from Y satisfies $\Pr\left[\delta(\bar{Y}, \bar{Z}) > \epsilon \cdot \mathrm{diam}(Y)\right] < \rho$.

Proof: Put $s = \frac{\kappa}{2} \cdot \frac{1}{\epsilon} \cdot \log(1/\rho)$ and $t = \frac{2}{\epsilon}$. Consider Z as s samples $Z_1, Z_2, \ldots, Z_s$ of size t each. By Proposition 8, $\delta(\bar{Y}, \bar{Z}) \le \frac{1}{s} \cdot \sum_{i=1}^{s} \delta(\bar{Y}, \bar{Z}_i)$. Therefore, $\Pr\left[\delta(\bar{Y}, \bar{Z}) > \epsilon \cdot \mathrm{diam}(Y)\right] \le \Pr\left[\sum_{i=1}^{s} \delta(\bar{Y}, \bar{Z}_i) > i\epsilon s \cdot \mathrm{diam}(Y)\right]$. Put $\chi_i = \delta(\bar{Y}, \bar{Z}_i)/\mathrm{diam}(Y)$ for all $i = 1, 2, \ldots s$. The χ_i are independent, identically distributed, random variables taking values in the range $[0, 1]$. By Lemma 9, $E[\chi_i] \le \frac{1}{2}\epsilon$ for all i. Using standard Chernoff bounds we get that $\Pr\left[\sum_{i=1}^{s} \chi_i > \epsilon s\right] < \left(\frac{e}{4}\right)^{\epsilon s/2}$. Putting $\kappa = 4/\log(4/e)$, the right hand side is equal to ρ. □

3 A PTAS for Metric Instances

In this section we present our algorithm for clustering metric spaces. We first describe a streamlined version of Indyk's algorithm [35] that solves the case of $k = 2$. It will help to motivate our approximation scheme for arbitrary fixed k.

Let (L, R) denote an optimal partition into 2 clusters. Run the following three algorithms, constructing three partitions into 2 clusters. Output the best of the three partitions.

1. First algorithm: Use the metric MaxCut approximation scheme of de la Vega and Kenyon with relative error ϵ^3.

2. Balanced clusters algorithm: By exhaustive search, guess $|L| \in (\epsilon n, n]$ and $|R| = n - |L|$. Repeat $O(1)$ times the following. Pick a random element $\ell \in V$ uniformly at random, and a random element $r \in V$ uniformly at random. For each vertex $v \in V$, let $\hat{\delta}(v, L) = |L| \cdot \delta(v, \ell)$ and $\hat{\delta}(v, R) = |R| \cdot \delta(v, r)$. Construct a partition (L', R') of V by placing v in L' if $\hat{\delta}(v, L) \leq \hat{\delta}(v, R)$, and placing v in R' otherwise.

3. Unbalanced clusters algorithm: By exhaustive search, guess $|L| \in (0, \epsilon n]$ and $|R| = n - |L|$. Repeat $O(1)$ times the following. Pick a random sample $r \in R$ uniformly at random. For each vertex $v \in V$, let $\hat{\delta}(v, R) = |R| \cdot \delta(v, r)$. Construct a partition (L', R') of V by placing in L' the $|L|$ vertices of V with largest value of $\hat{\delta}(v, R)$.

We now present our approximation scheme for arbitrary fixed k.

Definition: Given $\epsilon > 0$, two disjoint sets of points A and B are said to be well-separated if $\delta(A) + \delta(B) < \epsilon^{k+1}\delta(A \cup B)$.

Our algorithm consists of taking the best of all partitions that are generated as follows.

1. By exhaustive search, guess the optimal cluster sizes $|C_1| \geq |C_2| \geq \cdots \geq |C_k|$. Let i_0 be the largest i such that $|C_i| > \epsilon |C_{i-1}|$ for $i = 2, 3, \ldots, i_0$. Clusters C_1 through C_{i_0} are called *large clusters*, and the others are called *small clusters.* By exhaustive search, for each pair of large clusters C_i and C_j, guess whether clusters C_i and C_j are well-separated. Define *groups* of large clusters by taking the transitive closure of the relation "C_i and C_j are not well-separated".

2. Choose, uniformly at random, an element c_i in each large cluster C_i. (i.e. take i_0 points uniformly at random, and with constant probability the ith element will be in C_i). For each point x and for each large cluster C_i, define $\hat{\delta}(x, C_i) = |C_i|\delta(x, c_i)$.

3. For each x, consider the large cluster C_i which minimizes $\hat{\delta}(x, C_i)$. Place x in C_i's group and define its contribution to the group as $f(x) = \hat{\delta}(x, C_i)$. This defines a partition of V into groups.

4. By exhaustive search, for each group G thus constructed and for each small cluster C_j, guess $|G \cap C_i|$, and remove from G the $|G \cap C_i|$ elements with largest contribution $f(x)$. Recursively partition the removed elements into $(k - i_0)$ clusters.

5. Partition each group of h large clusters with $h > 1$ using Max-h-Cut with error parameter $\epsilon' = \epsilon^{3k+2}/h^2$.

4 Analysis of the Metric Algorithm

Lemma 11. Let $C \subseteq V$ and $r \in C$ be such that $\delta(r, C) \leq 2\delta(C)/|C|$. Let $\hat{\delta}(x, C) = |C| \cdot \delta(x, r)$, for $x \in V$. Then $|\delta(x, C) - \hat{\delta}(x, C)| \leq 2\delta(C)/|C|$.

Proof: Apply Proposition 1 to $X = \{x\}$, $Y = C$ and $Z = \{r\}$, and to $X = \{x\}$, $Y = \{r\}$ and $Z = |C|$. □

The following lemma is useful for analyzing balanced well-separated clusters.

Lemma 12. Consider two sets of points R and L which are both of size at least $\epsilon|L \cup R|$, and such that $\delta(R) + \delta(L) < \epsilon^2 \delta(R \cup L)$. Let r be such that $\delta(r, R) \leq 2\delta(R)/|R|$ and similarly ℓ be such that $\delta(\ell, L) \leq 2\delta(L)/|L|$. For any x, define $\hat{\delta}(x, R) = |R| \cdot \delta(x, r)$ and $\hat{\delta}(x, L) = |L| \cdot \delta(x, \ell)$. Let $F = \{x \in R | \hat{\delta}(x, L) \leq \hat{\delta}(x, R)\}$. Then,

- $|F| = O(\epsilon^2)|R \cup L|$; moreover, if $\delta(R) + \delta(L) < \epsilon^c \delta(R \cup L)$, then $|F| = O(\epsilon^c)|R \cup L|$.
- $\delta(F) \leq O(\epsilon)\delta(R)$, and
- $\delta(L, F) - \delta(R, F) \leq O(\epsilon)(\delta(R) + \delta(L))$.

Proof: If $x \in F$ then $\hat{\delta}(x, L) - \hat{\delta}(x, R) \leq 0$. Thus any point x in F must verify:

$$\begin{aligned}
&\delta(x, L) - \delta(x, R) \\
&\quad = \delta(x, L) - \hat{\delta}(x, L) + \hat{\delta}(x, L) - \hat{\delta}(x, R) + \hat{\delta}(x, R) - \delta(x, R) \\
&\quad \leq 2\delta(R)/|R| + 2\delta(L)/|L| \\
&\quad \leq \frac{2(\delta(R) + \delta(L))}{\epsilon|R \cup L|},
\end{aligned}$$

where the first inequality comes from Lemma 11 and the second one follows from $|R|, |L| \geq \epsilon|R \cup L|$.

We bound $|F|$ as follows.

$$\begin{aligned}
|F|\frac{\delta(R \cup L)}{2|R \cup L|} &\leq \sum_{x \in F} \delta(x, R \cup L) \text{ from Corollary 3 applied to } x \text{ in } R \cup L \\
&= \sum_{F} (2\delta(x, R) + (\delta(x, L) - \delta(x, R))) \\
&\leq 2\delta(F, R) + |F|\frac{2(\delta(R) + \delta(L))}{\epsilon|R \cup L|} \text{ from Equation 4} \\
&\leq 2\delta(R) + |F|\frac{2(\delta(R) + \delta(L))}{\epsilon|R \cup L|} \\
&\leq 2\epsilon^c \delta(R \cup L) + 2|F|\frac{\epsilon^2 \delta(R \cup L)}{|R \cup L|}.
\end{aligned}$$

Thus $|F| = O(\epsilon^2)|R \cup L|$, which proves the first statement of the Lemma.

Applying Proposition 1 to $X = Y = F$ and $Z = R$, we get

$$\delta(F) \leq 2\frac{|F|}{|R|}\delta(F,R) \leq O(\epsilon)\delta(R)$$

since $|F| = O(\epsilon^2)|R \cup L|$ and $|R| \geq \epsilon|R \cup L|$. This proves the second statement of the Lemma.

Finally, summing Equation (4) over every $x \in F$ gives

$$\delta(L,F) - \delta(R,F) \leq 2\frac{|E_ij|}{\epsilon|R \cup L|}(\delta(R) + \delta(L)) \leq O(\epsilon)(\delta(R) + \delta(L))$$

since $|F| \leq O(\epsilon^2)|R \cup L|$. This proves the last statement of the Lemma. □

The following lemma is useful to the analysis of unbalanced clusters.

Lemma 13. Consider two sets of points R and L such that $|L| < \epsilon|R|$ and such that $\delta(R) + \delta(L) < \epsilon^2\delta(R \cup L)$. Let $r \in R$ be such that $\delta(r,R) \leq 2\delta(R)/|R|$. For $x \in R \cup L$, let $\hat{\delta}(x,R) = |R| \cdot \delta(x,r)$. Let C_i' denote the $|R|$ points of $R \cup L$ with largest value of $\hat{\delta}(.,R)$, and $C_j' = R \cup L \setminus C_i'$. Let $F = R \cap C_j' = \{v_1, \ldots, v_m\}$ and $E = L \cap C_i' = \{v_1', \ldots, v_m'\}$. Then,

- $\delta(R,E) - \delta(R,F) = O(\epsilon)\delta(R)$.
- $\sum_{p=1}^{m} \delta(v_p, v_p') \leq O(1)\delta(R)/|R|$,
- $|\delta(L,F) - \delta(L,E)| \leq O(\epsilon)\delta(R)$,
- $\delta(F) \leq O(\epsilon)\delta(R)$, and
- $\delta(E) \leq O(\epsilon)\delta(R)$.

Proof: We pair up vertex v_p with vertex v_p'.

$$\delta(v_p,R) - \delta(v_p',R) = (\delta(v_p,R) - \hat{\delta}(v_p,R)) + (\hat{\delta}(v_p,R) - \hat{\delta}(v_p',R)) + (\hat{\delta}(v_p',R) - \delta(v_p',R)).$$

¿From Lemma 11 we have $\delta(v_p,R) - \hat{\delta}(v_p,R) \leq 2\delta(R)/|R|$ and $\hat{\delta}(v_p',R) - \delta(v_p',R) \leq 2\delta(R)/|R|$. By definition, the elements of C_i' (and hence of E) all have larger value of $\hat{\delta}(.,R)$ than the elements of C_j' (and hence of F). In particular, $\hat{\delta}(v_p,R) - \hat{\delta}(v_p',R) \leq 0$. Together, this implies that $\delta(v_p,R) - \delta(v_p',R) \leq 4\delta(R)/|R|$. Summing over p, we get

$$\begin{aligned}
\delta(E,R) - \delta(F,R) &\leq 4\frac{|F|}{|R|}\delta(R) \\
&= 4\frac{|E|}{|R|}\delta(R) \\
&\leq 4\frac{|L|}{|R|}\delta(R) \\
&= O(\epsilon)\delta(R),
\end{aligned}$$

hence the first statement of the Lemma.

Applying Proposition 1 to v_p, v'_p and R and summing over p, we get:

$$\begin{aligned} |R|\sum_p \delta(v_p, v'_p) &\leq \delta(F,R) + \delta(E,R) \\ &= \delta(E,R) - \delta(F,R)) + 2\delta(F,R) \\ &\leq O(\epsilon)\delta(R) + 2\delta(R) \\ &= O(1)\delta(R), \end{aligned}$$

hence the second statement of the Lemma.

Applying Proposition 1 to v_p, v'_p and L and to v'_p, v_p and L, we get

$$|\delta(v_p, L) - \delta(v'_p, L)| \leq |L| \cdot \delta(v_p, v'_p).$$

Summing over p, we get

$$|\delta(L,F) - \delta(L,E)| \leq \frac{|L|}{|R|}O(1)\delta(R) = O(\epsilon)\delta(R),$$

hence the third statement of the Lemma.

Applying Proposition 1 to F, F and R, we get

$$\delta(F) \leq \frac{2\delta(F,R)|F|}{|R|} \leq \frac{2\delta(R)|L|}{|R|} \leq 2\epsilon\delta(R),$$

hence the fourth statement of the Lemma.

Now, write $\delta(v'_p, v'_q) \leq \delta(v'_p, v_p) + \delta(v_p, v_q) + \delta(v_q, v'_q)$. When we sum over p and q, we obtain

$$\begin{aligned} \delta(E) &\leq 2\sum_p \delta(v_p, v'_p)|E| + \delta(F) \\ &\leq 2|L|O(1)\frac{\delta(R)}{|R|} + O(\epsilon)\delta(R) \\ &= O(\epsilon)\delta(R), \end{aligned}$$

hence the last statement of the Lemma. □

Now, let us analyze the 2-clustering algorithm.

Case 1: Assume that $c^* \geq \epsilon^2 W$. Then the MaxCut algorithm with error ϵ^3 produces a partition whose Cut value is at least OPT-Max-Cut$(1-\epsilon^3) \geq$ OPT-Max-Cut $- \epsilon^3 W$. The 2-cluster value of this partition is thus at most $W -$ OPT-Max-Cut $+ \epsilon^3 W$, which is $c^* + \epsilon^3 W$, hence at most $(1+\epsilon)\cdot c^*$.

Case 2: Assume that $c^* < \epsilon^2 W$ and that the optimal partition (L, R) is such that $|L|, |R| \geq \epsilon n$. We analyze the Balanced Clusters algorithm.

With probability at least $\epsilon/2$, the algorithm has picked $\ell \in L$ and $r \in R$. For ℓ picked uniformly at random in L, we have on average $E(\delta(\ell, L)) = \delta(L)/|L|$. By Markov's inequality, with probability at least $1/2$, it holds that $\delta(\ell, L) \leq 2\delta(L)/|L|$. Similarly, with probability at least 1/2, it holds that $\delta(r, R) \leq 2\delta(R)/|R|$. Moreover, the two events are independent. Thus, with probability at least $\epsilon(1-\epsilon)/4$, we have:

$$\ell \in L, \delta(\ell, L) \leq 2\delta(L)/|L|, r \in R, \text{ and } \delta(r, R) \leq 2\delta(R)/|R|.$$

We assume that ℓ and r satisfy these properties and that $|L|$ and $|R|$ have been guessed correctly.

Let $L' = L + F - E$ and $R' = R + E - F$. Then,

$$\begin{aligned} &\delta(L') + \delta(R') - \delta(L) - \delta(R) \\ &= \delta(L+F-E, L+F-E) + \delta(R+E-F, R+E-F) - \delta(L, L) - \delta(R, R) \\ &= 2(\delta(L,F) - \delta(R,F)) + 2(\delta(R,E) - \delta(L,E)) + 2\delta(E) + 2\delta(F) - 4\delta(E,F) \\ &= O(\epsilon)c^*, \end{aligned}$$

by Lemma 12.

Case 3: assume that $c^* < \epsilon^2 W$ and that the optimal partition (L, R) is such that $|L| < \epsilon n$. Then $|L| < \epsilon/(1-\epsilon)|R|$. We analyze the Unbalanced Clusters algorithm.

With probability at least $(1-\epsilon)/2$, we have $r \in R$ and $\delta(r, R) \leq 2\delta(R)/|R|$. We assume that this holds and that $|L|$ has been guessed correctly.

Let $E = L \cap R'$ and $F = R \cap L'$. The difference between the value of the cut constructed by the algorithm and the value of the optimal cut is

$$\begin{aligned} &\delta(L+F-E) + \delta(R+E-F) - \delta(L) - \delta(R) \\ &= 2(\delta(L,F) - \delta(L,E)) + 2(\delta(R,E) - \delta(R,F)) + 2\delta(E) + 2\delta(F) - 4\delta(E,F) \\ &= O(\epsilon)\delta(R), \end{aligned}$$

by Lemma 13.

Thus in all cases, one of the algorithms will output a near-optimal solution. This concludes the analysis of 2-clustering.

We now proceed with the analysis of the k-clustering algorithm.

We first analyze the mistakes made in step 3. For that, we focus on the large clusters. Consider two large clusters C_i and C_j which belong to different groups. let E_{ij} be the set of element of C_i which are mistakenly classified as belonging to C_j. Consider the intermediate k-cluster such that

$$C_i' = \begin{cases} C_i & \text{if } i > i_0 \\ C_i - \cup_j E_{ij} + \cup_j E_{ji} & \text{if } i \leq i_0. \end{cases}$$

We have:

$$\begin{aligned} &\sum_i \delta(C_i') - \sum_i \delta(C_i) \\ &\leq 2\sum_{i,j}(\delta(C_i, E_{ji}) - \delta(C_j, E_{ji})) + \sum_{i,j} \delta(E_{ij}) \\ &\quad +2\sum_{i,j,j'} \delta(E_{ji}, E_{j'i}) + 2\sum_{i,j,j'} \delta(E_{ij}, E_{ij'}). \end{aligned}$$

The first sum has only $O(k^2)$ terms, which are all small (i.e. $O(\epsilon)c^*$) by Lemma 12. The second sum also has only $O(k^2)$ terms, which are also all small by Lemma 12.

The third sum has only $O(k^3)$ terms. Consider one of them. Applying Proposition 1 to $X = E_{ji}$, $Y = E_{j'i}$ and $Z = C_i$, we get $|C_i| \cdot \delta(E_{ji}, E_{j'i}) \leq |E_{ji}|\delta(E_{ji}, C_i) + |E_{j'i}|\delta(E_{j'i}, C_i)$. We analyze the first of the two terms of this sum (by symmetry, our analysis will also hold for the second term of the sum). We have:

$$\begin{aligned}
&\frac{|E_{ji}|}{|C_i|}\delta(E_{ji}, C_i)\\
&\quad\leq \frac{|E_{ji}|}{|C_i|}(\delta(E_{ji}, C_j) + \delta(E_{ji}, C_i) - \delta(E_{ji}, C_j))\\
&\quad\leq \frac{|E_{ji}|}{|C_i|}(\delta(C_j) + O(\epsilon)(\delta(C_i) + \delta(C_j))) \text{ by Lemma 12}\\
&\quad= \frac{O(\epsilon^{k+1})|C_i \cup C_j|}{|C_i|}O(c^*)\\
&\quad= O(\epsilon)c^*,
\end{aligned}$$

where the previous-to-last equality follows from the definition of well-separated clusters, and the last equality follows from the definition of large clusters, which implies $|C_i| \geq \epsilon^k n$.

The last sum is analyzed similarly:

$$\begin{aligned}
&\delta(E_{ij}, E_{ij'})\\
&\quad\leq \frac{|E_{ij}|}{|C_i|}\delta(C_i, E_{ij'}) + \frac{|E_{ij'}|}{|C_i|}\delta(C_i, E_{ij})\\
&\quad\leq \frac{|E_{ij}| + |E_{ij'}|}{|C_i|}\delta(C_i)\\
&\quad\leq O(\epsilon)\delta(C_i).
\end{aligned}$$

Thus the partition (C_i') is a near-optimal k-clustering:

$$\sum_i \delta(C_i') \leq (1 + O(k^3\epsilon))c^*.$$

Unfortunately some mistakes are made in step 4 as well. We now need to bound the effect of those mistakes. For each large cluster C_i and each small cluster C_j, let F_{ij} denote the points of C_i' which mistakenly go into C_j, and F_{ji} denote the points of C_j which mistakenly go into C_i's group. By the guess made in step 4, we have $|F_{ij}| = |F_{ji}|$, and so we can pair up the vertices as in the analysis of the Unbalanced clustering algorithm. Let

$$C_i'' = \begin{cases} C_i' + \sum_{j>i_0} F_{ji} - \sum_{j>i_0} F_{ij} & \text{if } i \leq i_0 \\ C_i' + \sum_{j\leq i_0} F_{ji} - \sum_{j\leq i_0} F_{ij} & \text{if } i > i_0. \end{cases}$$

$$\sum_i \delta(C_i'') - \sum_i \delta(C_i')$$

$$\begin{aligned} &= \sum_i \delta(C'_i + \sum_j F_{ji} - \sum_j F_{ij}, C'_i + \sum_j F_{ji} - \sum_j F_{ij}) - \delta(C'_i, C'_i) \\ &= \sum_i \sum_j (\delta(C'_i, F_{ji}) - \delta(C'_i, F_{ij})) + \\ &\quad \sum_i \sum_j \delta(F_{ji}) + \sum_i \sum_j \delta(F_{ij}) + \\ &\quad \sum_i \sum_{j,j'} (\delta(F_{ji}, F_{j'i}) - \delta(F_{ji}, F_{ij'})) + \\ &\quad \sum_i \sum_{j,j'} (\delta(F_{ij}, F_{ij'}) - \delta(F_{ji}, F_{ij'})). \end{aligned}$$

Remember that F_{ab} is non-empty only if a refers to a small cluster and b to a large cluster, or if a refers to a large cluster and b to a small cluster.

The first term has $O(k^2)$ terms which are all small by Lemma 13. The next two terms also have $O(k^2)$ terms which are also all small by Lemma 13.

For the next term, remembering that $F_{j'i}$ is paired up with $F_{ij'}$ and using $\delta(x,y) - \delta(x,y') \leq \delta(y,y')$, we get

$$\delta(F_{ji}, F_{j'i}) - \delta(F_{ji}, F_{ij'}) \leq |F_{ji}| \sum_{(y,y') \text{ pair of } F_{j'i} \times F_{ij'}} \delta(y,y').$$

If C_i is large and $C_j, C_{j'}$ are small, then by Lemma 13 this is bounded by $|C_j| O(1) \delta(C_i)/|C_i|$, which is $O(\epsilon)\delta(C_i)$ because of the gap between sizes of large and small clusters.

If C_i is small and $C_j, C_{j'}$ are large, then by Lemma 13 this is bounded by $|C_i| O(1) \delta(C_{j'})/|C_{j'}|$, which is $O(\epsilon)\delta(C_{j'})$. Thus in all cases, this term, like the previous terms, is $O(\epsilon)c^*$. The last term can be dealt with similarly. Thus the partition (C''_i) is a near-optimal k-clustering:

$$\sum_i \delta(C''_i) \leq \sum_i \delta(C'_i) + O(k^3\epsilon)c^* \leq (1 + O(k^3\epsilon))c^*.$$

Finally, we need to analyze the use of Max-h-Cut in the last step of the algorithm; we will present the analysis as if the group was perfect, i.e. consisted of the clusters C_i. (It is easy to see that the proof also goes through when replacing the C_i by C''_i, at the cost of some bookkeeping of the small errors introduced at every step of the calculation.) In the groups of large clusters, the clusters are not well-separated. From this, we can deduce that c^* is $\Omega(W)$ as follows.

Consider a group $C_1 \cup C_2 \cup \cdots \cup C_h$. We have:

$$\delta(C_1 \cup \cdots \cup C_h) = \sum_i \delta(C_i) + \sum_{i \neq j} \delta(C_i, C_j). \tag{4}$$

For $i \neq j$, by definition of group, there exists a sequence of length $m \leq h$,

$$C_i = C_{i_0}, C_{i_1}, \ldots, C_{i_m} = C_j,$$

such that two consecutive clusters in that sequence are not well separated. Writing

$$\delta(x_{i_0}, x_{i_1}) \le \delta(x_{i_0}, x_{i_1}) + \delta(x_{i_1}, x_{i_2}) + \cdots + \delta(x_{i_{m-1}}, x_{i_m})$$

and summing over $C_{i_0} \times \cdots \times C_{i_m}$, we get

$$\frac{\delta(C_{i_0}, C_{i_m})}{|C_{i_0}| \times |C_{i_m}|} \le \frac{\delta(C_{i_0}, C_{i_1})}{|C_{i_0}| \times |C_{i_1}|} + \frac{\delta(C_{i_1}, C_{i_2})}{|C_{i_1}| \times |C_{i_2}|} + \cdots + \frac{\delta(C_{i_{m-1}}, C_{i_m})}{|C_{i_{m-1}}| \times |C_{i_m}|}.$$

Since the size of any two large clusters differ by a factor of ϵ^k at most, we deduce

$$\delta(C_i, C_j) \le \frac{1}{\epsilon^{2k}}(\delta(C_{i_0}, C_{i_1}) + \cdots + \delta(C_{i_{m-1}}, C_{i_m})).$$

By definition of well-separated clusters, we then obtain

$$\delta(C_i, C_j) \le \frac{1}{\epsilon^{3k+1}}((\delta(C_{i_0}) + \delta(C_{i_1})) + \cdots + (\delta(C_{i_{m-1}}) + \delta(C_{i_m})) \le \frac{2}{\epsilon^{3k+1}} c^*.$$

Plugging this into Equation (4) yields

$$\delta(C_1 \cup \cdots \cup C_h) \le (1 + \frac{2h(h-1)}{\epsilon^{3k+1}}) c^*. \tag{5}$$

Now, doing Max-h-Cut on $C_1 \cup \cdots \cup C_h$ with error parameter $\Theta(\epsilon^{3k+1}/h^2)$ will yield a partition whose cut value is within an additive $\Theta(\epsilon^{3k+1}/h^2)\delta(C_1 \cup \cdots \cup C_h)$ of optimal. Hence the value of the clustering will be off by

$$\Theta(\epsilon^{3k+1}/h^2)\delta(C_1 \cup \cdots \cup C_h) = \Theta(\epsilon)c^*$$

by Equation (5).

The algorithm then recursively finds a clustering of the removed elements. There are at most k levels of recursion, each inducing a mistake of order $1 + O(k^3\epsilon)$, for a total relative error of $O(k^4\epsilon)$.

Now, let us turn to the running time of the algorithm. The exhaustive search of the first step takes time $O(n^k 2^k)$. Sampling and computing $\hat{\delta}$ in the second step takes time $O(n+k) = O(n)$. The minimization in the third step takes time $O(nk)$. The fourth step takes time $O(n^k)$, excluding the recursive call. The final step uses Max-k-cut, which is a randomized algorithm and takes time $O(n^2 + nk2^{\tilde{O}(1/\epsilon'^3)})$ (in the version inspired from [32]). Overall, running the algorithm for $\epsilon' = (\epsilon/k^4)^{3k+1}/k^2$, the algorithm thus becomes a $(1 + O(\epsilon))$-approximation and has running time

$$O(n^k 2^k (n + nk + n^k + n^2 + nk2^{\tilde{O}(1/\epsilon'^3)})) \times k = O(k2^k n^{2k} + k^2 2^k n^{k+1} 2^{\tilde{O}(1/\epsilon'^3)}).$$

The above discussion proves the following theorem.

Theorem 14. For every fixed positive integer k and for every $\epsilon > 0$ there exists an algorithm for Metric Min-Sum All-Pairs k-clustering that computes a solution of cost within a factor of $1 + \epsilon$ of the optimum cost in time $O(n^{2k} + n^{k+1}2^{\tilde{O}(1/\epsilon^{3k+1})})$.

5 The Basic Algorithm for Squared Euclidean Distance

In this section we consider a finite input set $V \subset \mathbb{R}^d$ and distance function $\delta(x,y) = \|x-y\|_2^2$. We give, for every $\epsilon > 0$, an $n^{O(k/\epsilon^4)}$ time algorithm that produces a partition of the input space into k clusters with cost within a factor of $1+\epsilon$ of the cost of an optimum partition. Our algorithm can be modified to solve the min-sum median case. We indicate the changes needed at the end of the section.

We first present the algorithm, and then proceed to motivate and analyze it.

1. By exhaustive search, guess the optimal cluster sizes $|C_i| = n_i$, $n_1 + n_2 + \cdots + n_k = n$. By exhaustive search, for each $i = 1, \ldots, k$, consider all possible multisets A_i containing $\left(\frac{16}{\epsilon}\right)^4$ points.[2]

2. Consider the following weighted complete $n \times n$ bipartite graph G. The left side has n vertices, of which n_i are labelled A_i, and the right side has n vertices which correspond to the points of V. The edge between a vertex labelled A_i and a vertex x of V has weight $\hat{\delta}(x, C_i) = n_i \cdot \delta(x, \bar{A}_i)$.

3. Compute a minimum cost perfect matching in the graph G. This defines the following clustering $C_1, C_2, \ldots, C_k$: C_i is the set of points matched to the copies of A_i.

4. Output the best such clustering over all choices of $\mathcal{A} = (A_1, \ldots, A_k)$ and $N = (n_1, \ldots, n_k)$.

Our algorithm is motivated by the following bound.

Lemma 15. Let Y be any multi-subset of V. Then, for every ϵ such that $0 < \epsilon \leq 1$, there exists a multi-subset Z of Y of size $|Z| = \left(\frac{16}{\epsilon}\right)^4$ and such that

$$\left|\delta(Y, \bar{Z}) - \delta(Y, \bar{Y})\right| \leq \epsilon \cdot \delta(Y, \bar{Y}).$$

Proof: Let $\mu = \frac{1}{|Y|} \sum_{x \in Y} \delta(x, \bar{Y})$ denote the average distance between a point $x \in Y$ and $\bar{Y}$. Let $Y_c = \{x \in Y \mid \delta(x, \bar{Y}) \leq 64\mu/\epsilon^2\}$. By Proposition 7, $\operatorname{diam}(Y_c) \leq \left(2\sqrt{64\mu/\epsilon^2}\right)^2 = 256\mu/\epsilon^2$. By Lemma 9, there exists a multi-subset Z of Y_c such that $|Z| = (16/\epsilon)^4$ and $\delta(\bar{Z}, \bar{Y}_c) \leq \epsilon^4 \operatorname{diam}(Y_c)/16^4 \leq \epsilon^2\mu/256$. We complete the proof by proving the following claim.

Claim 16. If Z is a multiset such that $\delta(\bar{Z}, \bar{Y}_c) \leq \epsilon^2\mu/256$, then

$$\left|\delta(Y, \bar{Z}) - \delta(Y, \bar{Y})\right| \leq \epsilon \cdot \delta(Y, \bar{Y}).$$

Proof: We want to bound

$$\delta(Y, \bar{Z}) - \delta(Y, \bar{Y}_c) = \sum_{x \in Y} \left(\delta(x, \bar{Z}) - \delta(x, \bar{Y}_c)\right).$$

[2] The constant $16^4 = 65536$ was chosen to simplify our calculations. It can be improved significantly.

We bound each term of the right hand side separately using Proposition 7. This gives $\delta(x,\bar{Z}) - \delta(x,\bar{Y_c}) \leq \delta(\bar{Z},\bar{Y_c}) + 2\sqrt{\delta(\bar{Z},\bar{Y_c}) \cdot \delta(x,\bar{Y_c})}$. Let $Y_1 = \{x \in Y \mid \delta(x,\bar{Y_c}) \leq \mu\}$. If $x \in Y_1$, then

$$\delta(x,\bar{Z}) - \delta(x,\bar{Y_c}) \leq \left(\frac{1}{8}\epsilon + \frac{1}{256}\epsilon^2\right) \cdot \mu. \tag{6}$$

If $x \in Y \setminus Y_1$, then $\delta(\bar{Z},\bar{Y_c}) \leq \epsilon^2\mu/256 < \epsilon^2\delta(x,\bar{Y_c})/256$. Therefore,

$$\delta(x,\bar{Z}) - \delta(x,\bar{Y_c}) < \left(\frac{1}{8}\epsilon + \frac{1}{256}\epsilon^2\right) \delta(x,\bar{Y_c}). \tag{7}$$

By Proposition 6, $\sum_{x\in Y_c} \delta(x,\bar{Y_c}) \leq \sum_{x\in Y_c} \delta(x,\bar{Y})$. By Proposition 8,

$$\begin{aligned}\delta(\bar{Y},\bar{Y_c}) &\leq \frac{1}{|Y_c|}\sum_{y\in Y_c} \left\|\bar{Y} - y\right\|_2^2 \\ &\leq \mu, \end{aligned} \tag{8}$$

where (8) follows from the definition of Y_c. If $x \in Y \setminus Y_c$, then $\delta(x,\bar{Y}) > 64\mu/\epsilon^2$. Therefore, using Proposition 7 and (8) we get:

$$\begin{aligned}\delta(x,\bar{Y_c}) &\leq \delta(x,\bar{Y}) + \delta(\bar{Y},\bar{Y_c}) + 2\sqrt{\delta(x,\bar{Y}) \cdot \delta(\bar{Y},\bar{Y_c})} \\ &< \left(1 + \frac{1}{4}\epsilon + \frac{1}{64}\epsilon^2\right) \cdot \delta(x,\bar{Y}). \end{aligned} \tag{9}$$

Combining the bounds in (6), (7), and (9), we get

$$\begin{aligned}\sum_{x\in Y} \delta(x,\bar{Z}) &= \sum_{x\in Y_1} \delta(x,\bar{Z}) + \sum_{x\in Y\setminus Y_1} \delta(x,\bar{Z}) \\ &\leq \sum_{x\in Y_1} \delta(x,\bar{Y_c}) + \left(\frac{1}{8}\epsilon + \frac{1}{256}\epsilon^2\right) \cdot \mu \cdot |Y_1| + \left(1 + \frac{1}{8}\epsilon + \frac{1}{256}\epsilon^2\right) \cdot \sum_{x\in Y\setminus Y_1} \delta(x,\bar{Y_c}) \\ &\leq \left(1 + \frac{1}{4}\epsilon + \frac{1}{64}\epsilon^2\right) \cdot \left(1 + \frac{1}{8}\epsilon + \frac{1}{256}\epsilon^2\right) \cdot \sum_{x\in Y} \delta(x,\bar{Y}) + \left(\frac{1}{8}\epsilon + \frac{1}{256}\epsilon^2\right) \cdot \mu \cdot |Y| \\ &\leq \left(1 + \frac{1}{2}\epsilon + \frac{7}{128}\epsilon^2 + \frac{3}{1024}\epsilon^3 + \frac{1}{16384}\epsilon^4\right) \cdot \sum_{x\in Y} \delta(x,\bar{Y}) \\ &\leq (1+\epsilon) \cdot \sum_{x\in Y} \delta(x,\bar{Y}). \end{aligned}$$

On the other hand, by Proposition 6, $\sum_{x\in Y} \delta(x,\bar{Z}) \geq \sum_{x\in Y} \delta(x,\bar{Y})$. This completes the proof of Claim 16 and of Lemma 15. □

We are now ready for the analysis of our algorithm.

Theorem 17. The above algorithm computes a solution whose cost is within a factor of $(1+\epsilon)$ of the optimum cost in time $n^{O(k/\epsilon^4)}$.

Proof: By Lemma 15, for every $i = 1, 2, \ldots, k$, there exists a multi-subset Z_i of C_i^* of size $|Z_i| = (16/\epsilon)^4$ and such that

$$\left|\delta(C_i^*, \bar{Z}_i) - \delta(C_i^*, \bar{C}_i^*)\right| \leq \epsilon \cdot \delta(C_i^*, \bar{C}_i^*).$$

Consider the iteration of the algorithm where $A_i = Z_i$ and $n_i = |C_i^*|$ for every $i = 1, 2, \ldots, k$. Let C_i be the set of points matched to the nodes marked A_i in this iteration, for all $i = 1, 2, \ldots, k$. Then,

$$\begin{aligned}
\mathrm{cost}(C_1, C_2, \ldots, C_k) &= \sum_{i=1}^{k} |C_i| \cdot \sum_{x \in C_i} \delta(x, \bar{C}_i) \\
&\leq \sum_{i=1}^{k} n_i \cdot \sum_{x \in C_i} \delta(x, \bar{A}_i) \\
&\leq \sum_{i=1}^{k} n_i \cdot \sum_{x \in C_i^*} \delta(x, \bar{A}_i) \\
&\leq (1+\epsilon) \cdot \sum_{i=1}^{k} |C_i^*| \cdot \sum_{x \in C_i^*} \delta(x, \bar{C}_i^*).
\end{aligned}$$

The performance guarantee follows because the algorithm finds a partition whose cost is at least as good as $\mathrm{cost}(C_1, C_2, \ldots, C_k)$.

As for the running time of the algorithm, there are less than n^k possible representations of n as a sum $n_1 + n_2 + \cdots + n_k$. There are less than n^{65536k/ϵ^4} possible choices for $\mathcal{A}$. Computing a minimum cost perfect matching in G takes $O(n^3 \log n)$ time. □

To solve the min-sum median case, we modify the algorithm as follows. We remove the enumeration over the cluster sizes, and the multiplication of edges weights in G by those sizes. Instead of computing a minimum cost perfect matching in G, we assign each point to the closest set to it.

6 Outliers

In this section we present a much faster randomized algorithm that clusters at least $(1-\zeta)n$ points from V into k clusters $C_1, C_2, \ldots, C_k$, such that $\mathrm{cost}(C_1, C_2, \ldots, C_k)$ is within a factor of $1+\epsilon$ of the optimum cost to cluster all the points into k clusters (in fact, of the cost to cluster the points the algorithm chooses into k clusters), with probability at least $1-\rho$.

The algorithm differs from the previous algorithm in the way it enumerates over the choice of $\mathcal{A}$ and N. This is done as follows. Pick a sample Z of $\frac{\gamma}{\epsilon^8} \cdot \frac{k}{\zeta} \cdot \log(k/\rho)$ points, each chosen independently and uniformly at random from X (where γ is a sufficiently large constant). Enumerate over all choices for a list $\mathcal{A}$ of $t \leq k$ disjoint subsets $A_1, A_2, \ldots, A_t$ of Z, each containing $\frac{\lambda}{\epsilon^8} \cdot \log(k/\rho)$ points. For each choice of $\mathcal{A}$ enumerate over all choices for a list N of integers $n_1, n_2, \ldots, n_t$ such that for all $i = 1, 2, \ldots, t$, $n_i = \left(1 + \frac{\zeta}{2}\right)^{j_i} \cdot \frac{\zeta n}{2k}$, for some non-negative integer j_i, and furthermore $\left(1 - \frac{\zeta}{2}\right) n \leq \sum_{i=1}^{t} n_i \leq n$. Proceed to compute a clustering using the graph $G(\mathcal{A}, N)$ as in the previous algorithm. (Notice that the two sides of the graph need not be equal, so a minimum cost maximum matching may fail to assign some of the points to clusters.) Output the best clustering computed over all choices of $\mathcal{A}$ and N.

Theorem 18. With probability at least $1-\rho$, the above algorithm computes a solution containing at least $(1-\zeta)n$ points, whose cost is within a factor of $1+\epsilon$ of the optimum cost. The algorithm runs in time $O\left(g(k,\epsilon,\zeta,\rho)\cdot n^3\log n\right)$, where $g(k,\epsilon,\zeta,\rho) = \exp\left(\frac{1}{\epsilon^8}\cdot k\cdot\log(k/\rho)\cdot(\log k+\log(1/\epsilon)+\log(1/\zeta)+\log\log(1/\rho))\right)$.

Proof: If there are any clusters among $C_1^*, C_2^*, \ldots, C_k^*$ that contain less than $\frac{\zeta}{2}\cdot\frac{n}{k}$ points, then by removing them we remove at most $\frac{\zeta}{2}\cdot n$ points and we do not increase the cost of clustering the remaining points into k clusters. So, consider a cluster C_i^* that contains at least $\frac{\zeta}{2}\cdot\frac{n}{k}$ points. Let $\mu_i = \frac{1}{|C_i^*|}\sum_{x\in C_i^*}\delta(x,\bar{C}_i^*)$, and let $Y_i = \{x\in C_i^* \mid \delta(x,\bar{C}_i^*)\le 64\mu_i/\epsilon^2\}$. By Markov's inequality, $|Y_i| \ge \left(1-\frac{\epsilon^2}{64}\right)\cdot\frac{\zeta}{2}\cdot\frac{n}{k}$. Therefore, for every sufficiently large λ there exists $\gamma > 0$ such that

$$\Pr\left[|Z\cap Y_i| < \frac{\lambda}{\epsilon^8}\log(k/\rho)\right] < \frac{\rho}{2k}. \tag{10}$$

(In the above expression we consider the intersection $Z\cap Y_i$ as a multiset.)

Conditioned on the event $|Z\cap Y_i| \ge \frac{\lambda}{\epsilon^8}\log(k/\rho)$, the multiset Z_i containing the first $\frac{\lambda}{\epsilon^8}\log(k/\rho)$ points in $Z\cap Y_i$ is a sample of $|Z_i|$ points picked independently and uniformly at random from Y_i. By Lemma 10, assuming λ is sufficiently large,

$$\Pr\left[\delta(\bar{Z}_i,\bar{Y}_i) > \left(\frac{\epsilon}{16}\right)^4\cdot\mathrm{diam}(Y_i)\right] < \frac{\rho}{2k} \tag{11}$$

If $\delta(\bar{Z}_i,\bar{Y}_i) \le \left(\frac{\epsilon}{16}\right)^4\cdot\mathrm{diam}(Y_i)$, then by Claim 16

$$\left|\sum_{x\in C_i^*}\delta(x,\bar{Z}_i) - \sum_{x\in C_i^*}\delta(x,\bar{C}_i^*)\right| \le \epsilon\cdot\sum_{x\in C_i^*}\delta(x,\bar{C}_i^*). \tag{12}$$

Let $I\subset\{1,2,\ldots,k\}$ be the set of indices i such that $C_i^* \ge \frac{\zeta}{2}\cdot\frac{n}{k}$. Without loss of generality, let $I = \{1,2,\ldots,|I|\}$. Consider the event $\mathcal{E}$ that for every $i\in I$ we have $|Z\cap Y_i| \ge \frac{\lambda}{\epsilon^8}\log(k/\rho)$ and furthermore $\delta(\bar{Z}_i,\bar{Y}_i) \le \left(\frac{\epsilon}{16}\right)^4\cdot\mathrm{diam}(Y_i)$. Summing (10) and (11) over all $i\in I$, $\Pr[\mathcal{E}] \ge 1-\rho$. Assuming $\mathcal{E}$ holds, consider the iteration of the algorithm where $t = |I|$, for all $i\in I$, $A_i = Z_i$, and $\left(1-\frac{\zeta}{2}\right)|C_i^*| \le n_i \le |C_i^*|$. Let $C_1, C_2, \ldots, C_t$ be the clustering produced by the algorithm in this iteration. Then,

$$\begin{aligned}
\mathrm{cost}(C_1,C_2,\ldots,C_t) &\le \sum_{i=1}^{t} n_i\cdot\sum_{x\in C_i^*}\delta(x,\bar{A}_i)\\
&\le (1+\epsilon)\cdot\sum_{i=1}^{t}|C_i^*|\cdot\sum_{x\in C_i^*}\delta(x,\bar{C}_i^*)\\
&\le (1+\epsilon)\cdot\mathrm{cost}(C_1^*,C_2^*,\ldots,C_k^*).
\end{aligned}$$

Furthermore, the number of points clustered is

$$\begin{aligned}\sum_{i=1}^{t} n_i &\geq \left(1-\frac{\zeta}{2}\right)\cdot\sum_{i=1}^{t}|C_i^*| \\ &\geq \left(1-\frac{\zeta}{2}\right)^2\cdot\sum_{i=1}^{k}|C_i^*| \\ &> (1-\zeta)\cdot n.\end{aligned}$$

It remains to analyze the time complexity of the algorithm. The number of possible choices for $\mathcal{A}$ is

$$2^{O\left(\frac{1}{\epsilon^8}\cdot k\cdot\log(k/\rho)\cdot(\log k+\log(1/\epsilon)+\log(1/\zeta)+\log\log(1/\rho))\right)}.$$

The number of possible choices for N is

$$2^{O(k\cdot(\log k+\log(1/\zeta)))}.$$

Each iteration requires the computation of a minimum cost maximum bipartite matching. □

7 A Faster Min-Sum Median Algorithm

In this section we present an improved polynomial time approximation scheme for min-sum median k-clustering, building on the ideas of the previous section. We give a randomized polynomial time approximation scheme for min-sum median clustering of a finite input set $V\subset\mathbb{R}^d$ with distance function $\delta(x,y)=\|x-y\|_2^2$. The running time of our algorithms, for fixed k, ϵ, and ρ, is just $O(n\text{poly}\log n)$ (ρ is the failure probability).

The approximation scheme works as follows. Enumerate over all possible monotonically non-increasing integer sequences $n_1,n_2,\ldots,n_k$ such that for all $i=1,2,\ldots,k$, $n_i=(1+\epsilon)^{j_i}$ for a non-negative integer j_i, and $n\leq\sum_{i=1}^{k}n_i\leq(1+\epsilon)\cdot n$.[3] Partition $\{1,2,\ldots,k\}$ into segments $B_1,B_2,\ldots,B_t$ as follows. The first segment begins with 1 and every consecutive segment begins with the index following the last index of the previous segment. A segment B_i that starts with a_i ends with the first $s=b_i$, $s\geq a_i$, such that $s=k$ or $n_{s+1}<\left(\frac{\epsilon}{16k}\right)^2\cdot n_s$. Compute a set of candidate clusterings using a depth-t recursion. It is convenient to think of the recursion as a depth-t rooted tree T, where every node of T is labelled by a clustering of a subset of X into at most k clusters. The candidate clusterings are the labels of the leaves of T. Output the best candidate clustering.

To proceed with our description, we need some notation. Put $m_i=n_{a_i}$, for all $i=1,2,\ldots,t$. Put $m_{t+1}=0$. For every $i=1,2,\ldots,t$, every depth-i node of T corresponds to a clustering into b_i clusters, excluding $\frac{16k^2}{\epsilon}m_{i+1}$ points. (The root of T corresponds to an empty clustering.) The label on a node of T is an extension of the label on its parent. I.e., it is a clustering that adds points and clusters to the label of its parent, but does not change the assignment of points already clustered.

[3] In fact, a coarser approximation by a factor of 2 would suffice.

Let $\mathcal{C}_{i-1}$ be a label on a depth-$(i-1)$ node of T, where $1 \le i \le t$. We describe how to compute the labels of the children of this node. Denote by R_{i-1} the set of points that are not clustered in $\mathcal{C}_{i-1}$. Pick a sample Z of R_{i-1} of $\left(\frac{k}{16\epsilon}\right)^{2k} \cdot \frac{\gamma}{\epsilon^{10}} \ln k$ points drawn independently and uniformly at random, where $\gamma > 0$ is a constant. Enumerate over all choices for an ordered list of $|B_i|$ disjoint subsets $A_{a_i} \ldots, A_{b_i}$ of Z, each containing $\frac{\lambda}{\epsilon^8} \ln k$ points, where $\lambda > 0$ is a constant. (Both γ and λ are determined in the analysis below.) Every such choice generates a child of $\mathcal{C}_{i-1}$. (In the analysis it will be convenient to assume that every depth-i node of T includes, in addition to its label, the list $A_1, A_2, \ldots, A_{b_i}$, where its prefix $A_1, A_2, \ldots, A_{b_{i-1}}$ is inherited from its parent.) Augment $\mathcal{C}_{i-1}$ by finding a minimum cost assignment of $|R_{i-1}| - \frac{16k^2}{\epsilon} \cdot m_{i+1}$ points[4] from R_{i-1} to $C_1, C_2, \ldots, C_{b_i}$, where the cost of assigning $x \in R_{i-1}$ to C_j is $\delta(x, \bar{A}_j)$. This completes the specification of the algorithm. We now proceed with its analysis.

Claim 19. For all $i = 1, 2, \ldots, t$, $n_{b_i} \ge \left(\frac{\epsilon}{16k}\right)^{2(k-1)} m_i$.

Proof: By construction, for every $j \in \{a_i + 1, \ldots, b_i\}$, $n_j \ge \left(\frac{\epsilon}{16k}\right)^2 n_{j-1}$. Therefore, putting $s = b_i - a_i$, $n_{b_i} \ge \left(\frac{\epsilon}{16k}\right)^{2s} n_{a_i}$. As $s < k$, the claim follows. □

Claim 20. Among the sequences $n_1, n_2, \ldots, n_k$ that the algorithm enumerates over there exists one such that for every $j = 1, 2, \ldots, k$, $\left|C_j^*\right| \le n_j \le (1+\epsilon) \cdot \left|C_j^*\right|$.

Proof: Clearly for every j there is a valid choice of n_j that satisfies the bounds in the claim. Because for these values $n \le \sum_{j=1}^k n_j \le (1+\epsilon) \cdot n$, there is an iteration where the whole sequence is considered. □

Thus, from now on we analyze the iteration of the algorithm for which the bounds in Claim 20 hold. Consider a depth-$(i-1)$ node u of T with label $C_1, C_2, \ldots, C_{b_{i-1}}$, list $A_1, A_2, \ldots, A_{b_{i-1}}$, and set of unclustered points R_{i-1}. To generate a child v of u, we add to the list sets A_j, for $j = a_i, \ldots, b_i$. We are interested in a particular choice of those sets. Let $K_{a_i}, \ldots, K_{b_i} \subset R_{i-1}$ be mutually disjoint sets such that $K_j = R_{i-1} \cap C_j^*$ if $\left|R_{i-1} \cap C_j^*\right| \ge \left(\frac{\epsilon}{16}\right)^3 \cdot n_j$, and otherwise K_j is an arbitrary set of size n_j. (Notice that as $|R_{i-1}| = \frac{16k^2}{\epsilon} \cdot m_i > k \cdot m_i \ge \sum_{j \in B_i} n_j$, such a choice of sets exists.)

Claim 21. For every $\rho > 0$ and for every sufficiently large $\lambda > 0$, there exists $\gamma > 0$ such that with probability at least $1 - \frac{\rho}{k}$, the sample Z from R_{i-1} has the following property. For every $j \in B_i$, $|Z \cap K_j| \ge \frac{\lambda}{\epsilon^8} \ln k$.

Proof: The sets K_j, $j \in B_i$, are disjoint. There are at most k such sets, and each set has size at least $\left(\frac{\epsilon}{16}\right)^3 n_{b_i} \ge \frac{\epsilon^2}{256} \cdot \left(\frac{\epsilon}{16k}\right)^{2k} \cdot |R_{i-1}|$. Then, $|Z \cap K_j|$ is the sum of $\left(\frac{16k}{\epsilon}\right)^{2k} \cdot \frac{\gamma}{\epsilon^{10}} \ln k$ Bernouli trials with success probability $\frac{\epsilon^2}{256} \cdot \left(\frac{\epsilon}{16k}\right)^{2k}$. Thus, by standard Chernoff bounds, for γ sufficiently large, the probability that $|Z \cap K_j| < \frac{\lambda}{\epsilon^8} \ln k$ is at most $\frac{\rho}{k^2}$. Summing this probability for $j \in B_i$ completes the proof. □

[4] Notice that this is a positive number of points, and in fact, almost all the points in R_{i-1} get assigned at depth i.

Claim 22. For every $\rho > 0$ there exist $\lambda > 0$ and $\gamma > 0$ such that with probability at least $1 - \frac{\rho}{k}$, u has a child v with list $A_1, A_2, \ldots, A_{b_i}$ such that for every $j \in B_i$,

$$\left| \sum_{x \in K_j} \delta(x, \bar{A}_j) - \sum_{x \in K_j} \delta(x, \bar{K}_j) \right| \leq \frac{\epsilon}{8} \cdot \sum_{x \in K_j} \delta(x, \bar{K}_j).$$

Proof: Following the proof of Theorem 18 put, for every $j \in B_i$, $\mu_j = \frac{1}{|K_j|} \sum_{x \in K_j} \delta(x, \bar{K}_j)$, and $Y_j = \{x \in K_j \mid \delta(x, \bar{K}_j) \leq 64\mu_j/\epsilon^2\}$. Set λ so that the following property holds. For every $j \in B_i$, a multi-subset Z_j of $\frac{\lambda}{2\epsilon^8} \ln k$ independent, uniformly distributed, points of Y_j satisfies $\Pr\left[\delta(\bar{Z}_j, \bar{Y}_j) > \left(\frac{\epsilon}{128}\right)^4 \cdot \mathrm{diam}(Y_j)\right] < \frac{\rho}{3k}$. (This is possible by Lemma 10.) Set γ so that with probability at least $1 - \frac{\rho}{3k}$ the bound in Claim 21 holds. Conditioned on this event, for every $j \in B_i$ Z contains a sample of $\frac{\lambda}{\epsilon^8} \ln k$ independent, uniformly distributed, points from K_j. Notice that Y_j contains more than two-thirds of the points in K_j. If λ is sufficiently large, then the probability that $Z_j = Z \cap Y_j$ has at least $\frac{\lambda}{2\epsilon^8} \ln k$ points is at least $1 - \frac{\rho}{3k}$. Conditioned on this assumption, Z_j is a sample of independent, uniformly distributed, points of Y_j as discussed above. If $\delta(\bar{Z}_j, \bar{Y}_j) \leq \left(\frac{\epsilon}{128}\right)^4 \cdot \mathrm{diam}(Y_j)$, then, by Claim 16, $\left|\sum_{x \in K_j} \delta(x, \bar{Z}_j) - \sum_{x \in K_j} \delta(x, \bar{K}_j)\right| \leq \frac{\epsilon}{8} \cdot \sum_{x \in K_j} \delta(x, \bar{K}_j)$. The probability that all our assumptions are true is at least $1 - \frac{\rho}{k}$. In this case, v is the child of u corresponding to the choice $A_j = Z_j$, for all $j \in B_i$. □

Claim 23. With constant probability, T contains a depth-t node l with label $C_1, C_2, \ldots, C_t$ and list $A_1, A_2, \ldots, A_t$ such that the directed path p in T from its root to l has the property that every parent-child pair along p satisfies the bound in Claim 22.

Proof: By a trivial induction on the level i. □

Assume from now that the event in Claim 23 occurs. Denote, for every $x \in X$, by j_x the index of the cluster that x gets assigned to by the algorithm, and denote by j_x^* the index for which $x \in C_{j_x^*}^*$. Let J be the set of indices j such that $K_j \subseteq C_j^*$. For $i = 1, 2, \ldots, t$, let $J_i = \{j \in J \mid j \leq b_i\}$. For $i = 1, 2, \ldots, t$, let D_i be the set of points assigned to clusters at the depth-i node of p. A point $x \in D_i$ is *premature* iff $j_x^* > b_i$. Let P_i denote the set of premature points in D_i. A point $x \in D_i$ is *leftover* iff $K_{j_x^*} \not\subseteq C_{j_x^*}^*$ and $j_x^* \leq b_i$. Notice that in this case, almost all points from $C_{j_x^*}^*$ must be premature at some depth less than i. Let L_i denote the set of leftover points in D_i.

Let $j \notin J$. Let L^j denote the set of leftover points from C_j^*. By definition, $|L^j| < \left(\frac{\epsilon}{16}\right)^3 n_j$. Sort the points in $C_j^* \backslash L^j$ by non-decreasing order of $w(x) = \max\{\delta(x, \bar{C}_j^*), \delta(x, \bar{A}_{j_x})\}$. (These are all premature points.) Assign these points to the points in L^j in round-robin fashion. Let $Q(x)$ be the set of points assigned to $x \in L^j$. Let $q(x)$ be a point in $Q(x)$ with smallest $w(x)$. (Notice that $\{q(x) \mid x \in L^j\}$ is a set of $|L^j|$ points with smallest $w(\cdot)$ value in $C_j^* \setminus L^j$.) For every $x \in X$ let

$$\phi(x) = \begin{cases} \text{undefined} & \text{if } x \in P_i; \\ j_{q(x)} & \text{if } x \text{ is leftover}; \\ j_x^* & \text{otherwise.} \end{cases}$$

Claim 24. For every $j \notin J$,

$$\sum_{x\in L^j} \delta(x, \bar{A}_{\phi(x)}) \le \left(1+\frac{\epsilon}{3}\right) \cdot \sum_{x\in L^j} \delta(x, \bar{C}_j^*) + \frac{\epsilon}{6} \cdot \sum_{x\in L^j} \sum_{y\in Q(x)} w(y).$$

Proof: Let $x \in L^j$. By the triangle inequality,

$$\sqrt{\delta(x, \bar{A}_{\phi(x)})} \le \sqrt{\delta(x, \bar{C}_j^*)} + \sqrt{\delta(q(x), \bar{C}_j^*)} + \sqrt{\delta(q(x), \bar{A}_{\phi(x)})}.$$

By definition, $j = j_x^* = j_{q(x)}^*$. If $w(q(x)) \le \left(\frac{\epsilon}{16}\right)^2 \cdot \delta(x, \bar{C}_j^*)$, we get that

$$\begin{aligned}
\delta(x, \bar{A}_{\phi(x)}) &\le \left(\sqrt{\delta(x, \bar{C}_j^*)} + \sqrt{\delta(q(x), \bar{C}_j^*)} + \sqrt{\delta(q(x), \bar{A}_{\phi(x)})}\right)^2 \\
&\le \left(\sqrt{\delta(x, \bar{C}_j^*)} + 2 \cdot \sqrt{\left(\frac{\epsilon}{16}\right)^2 \cdot \delta(x, \bar{C}_j^*)}\right)^2 \\
&= \left(\left(1+\frac{\epsilon}{8}\right) \cdot \sqrt{\delta(x, \bar{C}_j^*)}\right)^2 \\
&\le \left(1+\frac{\epsilon}{3}\right) \cdot \delta(x, \bar{C}_j^*).
\end{aligned}$$

Otherwise, for every $y \in Q(x)$, $w(y) > \left(\frac{\epsilon}{16}\right)^2 \cdot \delta(x, \bar{C}_j^*)$. Moreover,

$$\begin{aligned}
|Q(x)| &= \frac{|C_j^* \setminus L^j|}{|L^j|} \\
&\ge \frac{(1-\epsilon)\cdot n_j - \left(\frac{\epsilon}{16}\right)^3 \cdot n_j}{\left(\frac{\epsilon}{16}\right)^3 \cdot n_j} \\
&> \frac{1}{2} \cdot \left(\frac{16}{\epsilon}\right)^3
\end{aligned}$$

Therefore, in this case,

$$\sum_{y\in Q(x)} w(y) > \frac{1}{2} \cdot \left(\frac{16}{\epsilon}\right)^3 \cdot \left(\frac{\epsilon}{16}\right)^2 \cdot \delta(x, \bar{C}_j^*) = \frac{8}{\epsilon} \cdot \delta(x, \bar{C}_j^*),$$

and

$$w(q(x)) < 2 \cdot \left(\frac{\epsilon}{16}\right)^3 \cdot \sum_{y\in Q(x)} w(y).$$

Thus, we get

$$\begin{aligned}
\delta(x, \bar{A}_{\phi(x)}) &\le \left(\sqrt{\delta(x, \bar{C}_j^*)} + \sqrt{\delta(q(x), \bar{C}_j^*)} + \sqrt{\delta(q(x), \bar{A}_{\phi(x)})}\right)^2 \\
&< \left(\sqrt{\frac{\epsilon}{8} \cdot \sum_{y\in Q(x)} w(y)} + 2 \cdot \sqrt{2 \cdot \left(\frac{\epsilon}{16}\right)^3 \cdot \sum_{y\in Q(x)} w(y)}\right)^2 \\
&< \frac{\epsilon}{6} \cdot \sum_{y\in Q(x)} w(y).
\end{aligned}$$

Summing these bounds over all $x \in L^j$ completes the proof. □

Put $Q_i = \{x \in X \mid j_x^* \in J_i \text{ and } x \in R_i\}$. Let S_i be a set of $|P_i|$ points in Q_i with smallest $\delta(x, \bar{A}_{\phi(x)})$ value (notice that by definition Q_i cannot contain premature points, so $\phi(x)$ is defined for every $x \in Q_i$).

Claim 25.

$$\sum_{i=1}^{t} \sum_{x \in S_i} \delta(x, \bar{A}_{\phi(x)}) \leq \frac{\epsilon}{8} \cdot \sum_{i=1}^{t} \sum_{x \in D_i \setminus P_i} \delta(x, \bar{A}_{\phi(x)}).$$

Proof: Fix $i \in \{1, 2, \ldots, t\}$. The set P_i is a subset of $\bigcup_{j > b_i} C_j^*$ and therefore $|P_i| \leq k \cdot m_{i+1}$. On the other hand, $|Q_i| \geq |R_i| - k \cdot m_{i+1} = \left(\frac{16k^2}{\epsilon} - k\right) \cdot m_{i+1} > \frac{8k^2}{\epsilon} \cdot m_{i+1}$. Thus,

$$\frac{\sum_{x \in S_i} \delta(x, \bar{A}_{\phi(x)})}{\sum_{x \in Q_i} \delta(x, \bar{A}_{\phi(x)})} \leq \frac{|S_i|}{|Q_i|} = \frac{|P_i|}{|Q_i|} \leq \frac{\epsilon}{8k}.$$

Moreover, $Q_i \subset X \setminus (\bigcup_{i'} P_{i'})$, so $\sum_{x \in Q_i} \delta(x, \bar{A}_{\phi(x)}) \leq \sum_{i'} \sum_{x \in D_{i'} \setminus P_{i'}} \delta(x, \bar{A}_{\phi(x)})$. Summing over i, which takes $t \leq k$ values, completes the proof. □

Claim 26.

$$\sum_{i=1}^{t} \sum_{x \in D_i \setminus P_i \setminus L_i} \delta(x, \bar{A}_{\phi(x)}) \leq \left(1 + \frac{\epsilon}{8}\right) \cdot \sum_{j \in J} \sum_{x \in K_j} \delta(x, \bar{C}_j^*).$$

Proof: Notice that the lhs sums precisely over the points in $\bigcup_{j \in J} K_j$. Moreover, for $j \in J$, $x \in K_j$, $\phi(x) = j_x^* = j$. As we are assuming that the bound in Claim 22 holds, the proof is complete. □

Theorem 27. With constant probability the above algorithm computes a solution whose cost is within a factor of $1 + \epsilon$ of the optimum cost. The running time of the algorithm is $O(g(k, \epsilon) \cdot n \cdot (\log n)^k)$, where $g(k, \epsilon) = \exp\left(\frac{1}{\epsilon^8} \cdot k^3 \ln k \cdot \left(\ln \frac{1}{\epsilon} + \ln k\right)\right)$

Proof: With constant probability the recurrence T will contain a computation path p as per Claim 23. Assuming this occurs, consider the clustering $C_1, C_2, \ldots, C_k$ computed at the leaf l reached by the path p.

For every $i = 1, 2, \ldots, t$, the set $D_i \bigcup S_i \setminus P_i$ is a subset of R_{i-1} of size $|R_{i-1}| - |R_i|$. Therefore, assigning every $x \in D_i \bigcup S_i \setminus P_i$ to $C_{\phi(x)}$ is a feasible augmentation of $\mathcal{C}_{i-1}$, so its cost $\sum_{x \in D_i \bigcup S_i \setminus P_i} \delta(x, \bar{A}_{\phi(x)})$ cannot be smaller than the cost of the augmentation that the algorithm chooses which is $\sum_{x \in D_i} \delta(x, \bar{A}_{j_x})$. Therefore,

$$\begin{aligned} \sum_{x \in X} \delta(x, \bar{A}_{j_x}) &= \sum_{i=1}^{t} \sum_{x \in D_i} \delta(x, \bar{A}_{j_x}) \\ &\leq \sum_{i=1}^{t} \sum_{x \in D_i \bigcup S_i \setminus P_i} \delta(x, \bar{A}_{\phi(x)}) \end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{j=1}^{k} \sum_{x \in L^j} \delta(x, \bar{A}_{\phi(x)}) + \sum_{i=1}^{t} \sum_{x \in S_i} \delta(x, \bar{A}_{\phi(x)}) + \sum_{i=1}^{t} \sum_{x \in D_i \setminus P_i \setminus L_i} \delta(x, \bar{A}_{\phi(x)}) \\
&\leq \left(1 + \frac{\epsilon}{8}\right) \cdot \left(\sum_{j=1}^{k} \sum_{x \in L^j} \delta(x, \bar{A}_{\phi(x)}) + \sum_{i=1}^{t} \sum_{x \in D_i \setminus P_i \setminus L_i} \delta(x, \bar{A}_{\phi(x)}) \right) \\
&\leq \left(1 + \frac{\epsilon}{8}\right) \cdot \left(\left(1 + \frac{\epsilon}{3}\right) \cdot \sum_{j \notin J} \sum_{x \in L^j} \delta(x, \bar{C}_j^*) + \frac{\epsilon}{6} \cdot \sum_{j \notin J} \sum_{x \in L^j} \sum_{y \in Q(x)} w(y) \right) + \\
&\quad + \left(1 + \frac{\epsilon}{8}\right)^2 \cdot \sum_{j \in J} \sum_{x \in K_j} \delta(x, \bar{C}_j^*) \\
&\leq \left(1 + \frac{\epsilon}{2}\right) \cdot \sum_{x \in X} \delta(x, \bar{C}^*_{j_x^*}) + \frac{\epsilon}{5} \sum_{i=1}^{t} \sum_{x \in P_i} \delta(x, \bar{A}_{j_x}).
\end{aligned}
$$

Moving terms around, we get

$$
\begin{aligned}
\sum_{x \in X} \delta(x, \bar{A}_{j_x}) &\leq \frac{1 + \epsilon/2}{1 - \epsilon/5} \cdot \sum_{x \in X} \delta(x, \bar{C}^*_{j_x^*}) \\
&< (1 + \epsilon) \cdot \sum_{x \in X} \delta(x, \bar{C}^*_{j_x^*}).
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
\operatorname{cost}(C_1, C_2, \ldots, C_k) &= \sum_{j=1}^{k} \sum_{x \in C_j} \delta(x, \bar{C}_j) \\
&\leq \sum_{j=1}^{k} \sum_{x \in C_j} \delta(x, \bar{A}_j) \\
&= \sum_{x \in X} \delta(x, \bar{A}_{j_x}).
\end{aligned}
$$

As the algorithm outputs a clustering which is at least as good as $C_1, C_2, \ldots, C_k$, this establishes the performance guarantee of the algorithm.

As for the running time of the algorithm, the number of sequences $n_1, n_2, \ldots, n_k$ that the algorithms has to enumerate over is $O\left(\left(\log_{1+\epsilon} n\right)^k\right)$. The size of T is at most

$$
2^{\left(\frac{1}{\epsilon^8} \cdot k^3 \ln k \cdot \left(\ln \frac{1}{\epsilon} + \ln k\right)\right)}.
$$

Computing the augmentation at each node of T requires $O(n)$ distance computations, where the hidden constant depends mildly on k and ϵ. □

References

[1] P.K. Agarwal and C.M. Procopiuc. Exact and approximation algorithms for clustering. In *Proc. of the 9th Ann. ACM-SIAM Symp. on Discrete Algorithms*, January 1998, pages 658–667.

[2] P.K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Computing Surveys*, 30(4):412–458, 1998.

[3] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. In *Proc. of the 41th Ann. IEEE Symp. on Foundations of Computer Science* 2000.

[4] N. Alon and J. Spencer. *The Probabilistic Method.* Wiley, 1992.

[5] N. Alon and B. Sudakov. On two segmentation problems. *Journal of Algorithms*, 33:173–184, 1999.

[6] M. Anthony and P.L. Bartlett. *Neural Network Learning: Theoretical Foundations.* Cambridge University Press, 1999.

[7] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *J. Comp. System. Sci.*, 58:193–210, 1999.

[8] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. In *Proc. of the 30th Ann. ACM Symp. on Theory of Computing*, 1998.

[9] K. Azuma. Weighted sum of certain dependent random variables. *Tôhoku Math. J.* 3:357–367, 1967.

[10] M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via Core-Sets. To appear in *STOC 2002.*

[11] Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum k-clustering in metric spaces. In *Proc. of the 33rd Ann. ACM Symp. on Theory of Computing*, July 2001, pages 11–20.

[12] A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the Web. In *Proc. of the 6th Int'l World Wide Web Conf.*, 1997, pages 391–404.

[13] M. Bern and D. Eppstein. Approximation algorithms for geometric problems. In D. Hochbaum, Editor, *Approximation Algorithms for Hard Problems.* PWS Publishing, 1996.

[14] B. Carl and I. Stephani. *Entropy, Compactness and the Approximation of Operators.* Cambridge University Press, 1990.

[15] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proc. of the 40th Ann. IEEE Symp. on Foundations of Computer Science*, 1999.

[16] M. Charikar, S. Guha, D.B. Shmoys, and É. Tardos. A constant factor approximation algorithm for the k-median problem. In *Proc. of the 31st Ann. ACM Symp. on Theory of Computing*, 1999.

[17] D.R. Cutting, D.R. Karger, and J.O. Pedersen. Constant interaction-time scatter-gather browsing of very large document collections. In *Proc. of the 16th Ann. Int'l SIGIR Conf. on Research and Development in Information Retrieval*, 1993.

[18] D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. of the 15th Ann. Int'l SIGIR Conf. on Research and Development in Information Retrieval*, 1992, pages 318–329.

[19] S. Dasgupta. Learning mixtures of Gaussians. In *Proc. of the 40th Ann. IEEE Symp. on Foundations of Computer Science*, 1999.

[20] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[21] W. Fernandez de la Vega. MAX-CUT has a randomized approximation scheme in dense graphs. *Random Structures and Algorithms* 8:187–199, 1996.

[22] W. Fernandez de la Vega and M. Karpinski. Polynomial time approximation of dense weighted instances of MAX-CUT. *Random Structures and Algorithms*, 2000.

[23] W. Fernandez de la Vega, M. Karpinski, and C. Kenyon. A polynomial time approximation scheme for metric MIN-BISECTION. Manuscript, 2002.

[24] , W. Fernandez de la Vega and C. Kenyon. A randomized approximation scheme for metric MAX CUT. In *Proc. of the 39th Ann. IEEE Symp. on Foundations of Computer Science*, 1998, pages 468–471.

[25] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proc. of the 10th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 1999, pages 291–299.

[26] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3):231–262, 1994.

[27] U. Feige and R. Kranthgammer. A polylogharithmic approximation of minimum bisection. In *Proc. of the 41th Ann. IEEE Symp. on Foundations of Computer Science* 2000, pages 349–358.

[28] P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. of Combinatorial Theory B*, 44:355–362, 1988.

[29] A.M. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–200, 1999.

[30] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proc. of the 39th Ann. IEEE Symp. on Foundations of Computer Science*, 1998, pages 370–378.

[31] M.X. Goemans and D.P. Williamson. .878-approximation algorithms for MAX-CUT and MAX-2SAT. In *Proc. of the 26th Ann. ACM Symp. on Theory of Computing*, 1994, pages 422–431.

[32] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *J. of the ACM*, 45:653–750, 1998.

[33] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *Proc. of the 9th Ann. ACM-SIAM Symp. on Discrete Algorithms*, January 1998.

[34] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[35] P. Indyk. A sublinear time approximation scheme for clustering in metric spaces. In *Proc. of the 40th Ann. IEEE Symp. on Foundations of Computer Science*, 1999.

[36] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. of the 30th Ann. ACM Symp. on Theory of Computing*, 1998, pages 604–613.

[37] K. Jain and V.V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proc. of the 40th Ann. IEEE Symp. on Foundations of Computer Science*, 1999.

[38] K. Jansen, M. Karpinski, A. Lingas, and E. Seidel. Polynomial time approximation schemes for MAX-BISECTION on planar and geometric graphs. In *Proc. of 18th STACS*, LNCS 2010, Springer, pages 365–375, 2001.

[39] W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into Hilbert space, *Contemporary Mathematics*, 26:189–206, 1984.

[40] R. Kannan, S. Vempala, and A. Vetta. On clusterings: good, bad and spectral. In *Proc. of the 41st Ann. IEEE Symp. on Foundations of Computer Science*, 2000.

[41] R.M. Karp. The genomics revolution and its challenges for algorithmic research. *Bulletin of the EATCS*, 71:151–159, June 2000.

[42] S. Khanna, M. Sudan, and D. Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In *Proc. of the 29th Ann. ACM Symp. on Theory of Computing*, 1997, pages 11–20.

[43] J. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proc. of the 29th Ann. ACM Symp. on Theory of Computing*, 1997, pages 599–608.

[44] J. Kleinberg. Authoritative sources in a hyperlinked environment. *J. of the ACM*, 46, 1999.

[45] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. In *Proc. of the 30th Ann. ACM Symp. on Theory of Computing*, 1998, pages 473–482.

[46] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000.

[47] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[48] R. Lupton, F.M. Maley, and N. Young. Data collection for the Sloan digital sky survey: A network-flow heuristic. In *Proc. of the 7th Ann. ACM-SIAM Symp. on Discrete Algorithms*, January 1996, pages 296–303.

[49] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*, North Holland, Amsterdam, 1977.

[50] Manjara. `http://cluster.cs.yale.edu/`

[51] N. Mishra, D. Oblinger, and L. Pitt. Sublinear time approximate clustering. In *Proc. of the 12th Ann. ACM-SIAM Symp. on Discrete Algorithms*, January 2001, pages 439–447.

[52] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric clustering problems. *J. of the ACM*, 49(2):139–156, March 2002.

[53] G. Pisier. Remarques sur un résultat non publié de B. Maurey. In *Séminaire d'Analyse Fonctionelle 1980–1981*, 1981, Ecole Polytechnic, Centre de Mathématiques, Palaiseau.

[54] E. Rasmussen. Clustering algorithms. In W.B. Frakes, R. Baeza-Yates, eds., *Information Retrieval.* Prentice Hall, 1992.

[55] J. O'Rourke and G. Toussaint. Pattern recognition. In J. Goodman, J. O'Rourke, eds., *Handbook of Discrete and Computational Geometry.* CRS Press, 1997.

[56] L.J. Schulman. Clustering for edge-cost minimization. In *Proc. of the 32nd Ann. ACM Symp. on Theory of Computing*, 2000, pages 547–555.

[57] R. Shamir and R. Sharan. Algorithmic approaches to clustering gene expression data. In T. Jiang, T. Smith, Y. Xu, M.Q. Zhang eds., *Current Topics in Computational Biology*, MIT Press, to appear.

[58] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[59] S.J. Szarek. On the best constants in the Khinchin Inequality. *Studia Math.* 58:197–208, 1976.

[60] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[61] O. Zamir, O. Etzioni, O. Madani, and R.M. Karp. Fast and intuitive clustering of web documents. In Proc. KDD '97, 1997, pages 287–290.