A Unified Framework for Clustering Constrained Data Without Locality Property

Hu Ding[1] · Jinhui Xu[2]

Abstract

In this paper, we consider a class of constrained clustering problems of points in $\mathbb{R}^d$, where d could be rather high. A common feature of these problems is that their optimal clusterings no longer have the locality property (due to the additional constraints), which is a key property required by many algorithms for their unconstrained counterparts. To overcome the difficulty caused by the loss of locality, we present in this paper a unified framework, called *Peeling-and-Enclosing*, to iteratively solve two variants of the constrained clustering problems, *constrained k-means clustering* (*k*-CMeans) and *constrained k-median clustering* (*k*-CMedian). Our framework generalizes Kumar et al.'s (J ACM 57(2):5, 2010) elegant *k*-means clustering approach from unconstrained data to constrained data, and is based on two standalone geometric techniques, called *Simplex Lemma* and *Weaker Simplex Lemma*, for *k*-CMeans and *k*-CMedian, respectively. The simplex lemma (or weaker simplex lemma) enables us to efficiently approximate the mean (or median) point of an unknown set of points by searching a small-size grid, independent of the dimensionality of the space, in a simplex (or the surrounding region of a simplex), and thus can be used to handle high dimensional data. If k and $\frac{1}{\epsilon}$ are fixed numbers, our framework generates, in nearly linear time (i.e., $O(n(\log n)^{k+1}d)$), $O((\log n)^k)$ *k*-tuple candidates for the k mean or median points, and one of them induces a $(1 + \epsilon)$-approximation for *k*-CMeans or *k*-CMedian, where n is the number of points. Combining this unified framework with a problem-specific selection algorithm (which determines the best *k*-tuple candidate), we obtain a $(1 + \epsilon)$-approximation for each of the constrained clustering problems. Our framework improves considerably the best known results for these problems. We expect that our technique will be applicable to other variants of *k*-means and *k*-median clustering problems without locality.

Extended author information available on the last page of the article

1 Introduction

Clustering is one of the most fundamental problems in computer science, and finds numerous applications in many different areas, such as data management, machine learning, bioinformatics, networking, etc. [38]. The common goal of many clustering problems is to partition a set of given data items into a number of clusters so as to minimize the total cost measured by a certain objective function. For example, the popular *k-means* (or *k-median*) clustering seeks k mean (or median) points to induce a partition of the given data items so that the average squared distance (or the average distance) from each data item to its closest mean (or median) point is minimized. Most existing clustering techniques assume that the data items are independent from each other and therefore can "freely" determine their memberships in the resulting clusters (i.e., a data item does not need to pay attention to the clustering of others). However, in many real-world applications, data items are often constrained or correlated, which require a great deal of effort to handle such additional constraints. In recent years, considerable attention has been paid to various types of constrained clustering problems, such as *l-diversity clustering* [46], *r-gather clustering* [2,27,36], *capacitated clustering* [5,20,41], *chromatic clustering* [7,23], and *probabilistic clustering* [19,33,45]. In this paper, we study a class of constrained clustering problems of points in Euclidean space.

Given a set of points P in $\mathbb{R}^d$, a positive integer k, and a constraint $\mathbb{C}$, the constrained k-means (or k-median) clustering problem is to partition P into k clusters so as to minimize the objective function of the ordinary k-means (or k-median) clustering and satisfy the constraint $\mathbb{C}$. In general, the problems are denoted by k-CMeans and k-CMedian, respectively.

The detailed definition for each individual problem is given in Sect. 4. Roughly speaking, data constraints can be imposed at either *cluster* or *item* level. Cluster level constraints are restrictions on the resulting clusters, such as the size of the clusters [2] or their mutual differences [59], while item level constraints are mainly on data items inside each cluster, such as the coloring constraint which prohibits items of the same color being clustered into one cluster [7,23,46].

The additional constraints could considerably change the nature of the clustering problems. For instance, one key property exhibited in many unconstrained clustering problems is the so called *locality* property, which indicates that each cluster is located entirely inside the Voronoi cell of its center (e.g., the mean, median, or center point) in the Voronoi diagram of all the centers [37] (see Fig. 1a). Existing algorithms for these clustering problems often rely on such a property [8,10,17,31,37,43,48,50]. However, due to the additional constraints, the locality property may no longer exist (see Fig. 1b). Therefore, we need new techniques to overcome this challenge.
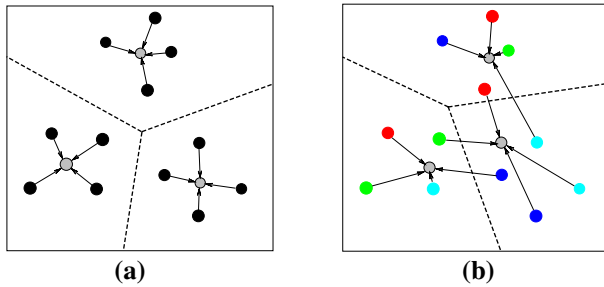
Fig. 1 a The Voronoi diagram induced by the mean points (i.e. the grey points) of k-means clustering for $k = 3$; **b** the Voronoi diagram induced by the mean points of chromatic k-means clustering, where the points sharing the same color should be in different clusters

1.1 Our Main Results

In this paper we present a unified framework called *Peeling-and-Enclosing (PnE)*, based on two standalone geometric techniques called *Simplex Lemma* and *Weaker Simplex Lemma*, to solve a class of constrained clustering problems without the locality property in Euclidean space, where the dimensionality of the space could be rather high and the number k of clusters is assumed to be some fixed number. Particularly, we investigate the constrained k-means (k-CMeans) and k-median (k-CMedian) versions of these problems. For the k-CMeans problem, our unified framework generates in $O(n(\log n)^{k+1}d)$ time a set of k-tuple candidates of cardinality $O((\log n)^k)$ for the to-be-determined k mean points. We show that among the set of candidates, one of them induces a $(1 + \epsilon)$-approximation for k-CMeans. To find out the best k-tuple candidate, a problem-specific selection algorithm is needed for each individual constrained clustering problem (note that due to the additional constraints, the selection problems may not be trivial). Combining the unified framework with the selection algorithms, we obtain a $(1 + \epsilon)$-approximation for each constrained clustering problem in the considered class. Our results considerably improve (in various ways) the best known algorithms for all these problems (see the table in Sect. 1.2). Below is a list of the constrained clustering problems considered in this paper. Our techniques can also be extended to achieve $(1 + \epsilon)$-approximations for the k-CMedian version of these constrained clustering problems. We expect that our technique will be applicable to other variants of k-means and k-median clustering problems without locality property, as long as the corresponding selection problems can be solved.

1. *l*-**Diversity Clustering** In this problem, each input point is associated with a color, and each cluster has no more than a fraction $\frac{1}{l}$ (for some constant $l > 1$) of its points sharing the same color. The problem is motivated by a widely-used privacy preserving model called *l-diversity* [46,47] in data management, which requires that each block contains no more than a fraction $\frac{1}{l}$ of items sharing the same sensitive attribute.
2. **Chromatic Clustering** In [23], Ding and Xu introduced a new clustering problem called *chromatic clustering*, which requires that the points with the same color should be clustered in different clusters. It is motivated by a biological application

for clustering chromosome homologs in a population of cells, where homologs from the same cell should be clustered into different clusters. Similar problem also appears in applications related to transportation system design [7].

3. **Fault Tolerant Clustering** The problem of *fault tolerant clustering* assigns each point p to its l nearest cluster centers for some $l \geq 1$, and counts all the l distances as its cost. The problem has been extensively studied in various applications for achieving better fault tolerance [16,35,40,44,54].

4. r-**Gather Clustering** This clustering problem requires each of the clusters to contain at least r points for some $r > 1$. It is motivated by the *k-anonymity* model for privacy preserving [2,55], where each block contains at least k items.[1]

5. **Capacitated Clustering** This clustering problem has an upper bound on the size of each cluster, and finds various applications in data mining and resource assignment [20,41].

6. **Semi-Supervised Clustering** Many existing clustering techniques, such as *ensemble clustering* [52,53] and *consensus clustering* [3,18], make use of a priori knowledge. Since such clusterings are not always based on the geometric cost (e.g., k-means cost) of the input, thus a more accurate way of clustering is to consider both the priori knowledge and the geometric cost. We consider the following *semi-supervised clustering* problem: given a set P of points and a clustering $\overline{S}$ of P (based on the priori knowledge), partition P into k clusters so as to minimize the sum (or some function) of the geometric cost and the difference with the given clustering $\overline{S}$. Another related problem is *evolutionary clustering* [15], where the clustering in each time point needs to minimize not only the geometric cost, but also the total shifting from the clustering in the previous time point (which can be viewed as $\overline{S}$).

7. **Uncertain Data Clustering** Due to the unavoidable error, data for clustering are not always precise. This motivates us to consider the following *probabilistic clustering* problem [19,33,45] : given a set of "nodes" with each represented as a probabilistic distribution over a point set in $\mathbb{R}^d$, group the nodes into k clusters so as to minimize the expected cost with respect to the probabilistic distributions.

Note Following our work published in [25], Bhattacharya et al. [14] improved the running time for finding the candidates of k-cluster centers from nearly linear to linear based on the elegant D^2-sampling. Their work also follows the framework of clustering constrained data, i.e., generating the candidates and selecting the best one by a problem-specific selection algorithm, presented in this paper. Our paper represents the first systematically theoretical study of the constrained clustering problems. Some of the underlying techniques, such as Simplex Lemma and Weaker Simplex Lemma, are interesting in their rights, which have already been used to solve other problems [26] (e.g., the popular "truth discovery" problem in data mining).

1.2 Related Works

The above 7 constrained clustering problems have been extensively studied in the past and a number of theoretical results have been obtained (in addition to many

---

[1] We use r here, instead of k, since k often denotes the number of clusters in a clustering problem.

Table 1 Existing and our new results for the class of constrained clustering problems

Problems	Existing results
l-Diversity clustering	2-Approx. for metric *k*-centers [46] (only for a restricted version of *l*-diversity clustering)
Chromatic clustering	$(1 + \epsilon)$-Approx. for chromatic *k*-ones clustering in $\mathbb{R}^d$ [23]; $(1 + \epsilon)$-approx. for 2-center in $\mathbb{R}^2$ [7]
Fault tolerant clustering	4 and 93-Approx. for uniform and non-uniform metric *k*-median [35,54]; 2-approx. for metric *k*-centers [16,40];
r-Gather clustering	2-Approx. for metric *k*-centers and 4-approx. for metric *k*-cellulars [2]; $(4 + \epsilon)$-approx. for *k*-centers in constant dimensional space [27,36]
Capacitated clustering	6 and 7-Approx. for metric *k*-centers with uniform and non-uniform capacities [20,41]
Semi-supervised clustering	Heuristic algorithms [13,32,56,57]
Uncertain data clustering	$(1 + \epsilon)$-Approx. for *k*-means and unassigned *k*-median [19]; $(3 + \epsilon)$-approx. for assigned *k*-median [19,58]; $(1 + \epsilon)$-approx. for assigned *k*-median in constant dimensional space [45]; $O(1)$-approx. for *k*-centers [33]

Our results: $(1 + \epsilon)$-approx. of *k*-means and *k*-median for all the 7 problems in $\mathbb{R}^d$ where *d* could be rather high

heuristic/practical solutions). Table 1 lists the best known theoretical results for each of them. It is clear that most existing results are either constant approximations or only for some restricted versions (e.g., constant dimensional space, etc.), and therefore can be improved by our techniques.

For the related traditional Euclidean *k*-means and *k*-median clustering problems, extensive research has been done in the past. Inaba et al. [37] showed that an exact *k*-means clustering can be computed in $O(n^{O(dk)})$ time for *n* points in $\mathbb{R}^d$. Arthur and Vassilvitskii [8] presented the *k*-means++ algorithm that achieves the expected $O(\log k)$ approximation ratio. Ostrovsky et al. [50] provided a $(1 + \epsilon)$-approximation for well-separated points. Based on the concept of stability, Awasthi et al. [9] presented the PTAS for the problems of *k*-means and *k*-median clustering. Matousek [48] obtained a nearly linear time $(1 + \epsilon)$-approximation for any fixed *d* and *k*. Similar result for *k*-median has also been achieved by Kolliopoulos and Rao [42]. Later, Fernandez de la Vega et al. [31] and Bǎdŏiu et al. [10] achieved nearly linear time $(1 + \epsilon)$-approximations for high dimensional *k*-means and *k*-median clustering, respectively, for fixed *k*. Kumar et al. [43] showed that linear-time randomized $(1 + \epsilon)$-approximation algorithms can be obtained for several Euclidean clustering problems (such as *k*-means and *k*-median) in any dimensional space. Recently, this technique has been further extended to several clustering problems with non-metric distance functions [1]. Later, Jaiswal et al. [39] applied a non-uniform sampling technique, which is called D^2-sampling, to simplify and improve the result in [43]; their algorithm can also handle the non-metric distance clustering problems studied in [1]. Using the *core-set* technique, a series of improvements have been achieved for high dimensional clustering problems [30].

As for the hardness of the problem, Dasgupta [21] showed that it is NP-hard for k-means clustering in high dimensional space even if $k = 2$; Awasthi et al. [4] proved that there is no PTAS for k-means clustering if both d and k are large, unless $P = NP$. Guruswami and Indyk [34] showed that it is NP-hard to obtain any PTAS for k-median clustering if k is not a constant and d is $\Omega(\log n)$.

Besides the traditional clustering models, Balcan et al. considered the problem of finding the clustering with small difference from the unknown ground truth [11,12].

1.3 Our Main Ideas

Most existing k-means or k-median clustering algorithms in Euclidean space consist of two main steps: (1) identify the set of k mean or median points and (2) partition the input points into k clusters based on these mean or median points (we call this step **Partition**). Note that for unconstrained clustering problems, the Partition step can be done by using the Voronoi diagram of the obtained mean or median points; however, for some constrained clustering problems, the Partition step may not be trivial (see Fig. 1a, b). More formally, we have the following definition.

Definition 1 (*Partition Step*) Given an instance P of k-CMeans (or k-CMedian) and k cluster centers (i.e., the mean or median points), the Partition step is to form k clusters of P, where the clusters should satisfy the constraint and each cluster is assigned to an individual cluster center, such that the objective function of the ordinary k-means (or k-median) clustering is minimized.

To determine the candidate set of the k mean or median points in step (1), most existing algorithms (either explicitly or implicitly) rely on the locality property. To shed some light on this, consider a representative and elegant approach by Kumar et al. [43] for k-means clustering. Let $\{Opt_1, \ldots, Opt_k\}$ be the set of k unknown optimal clusters in non-increasing order of their sizes. Their approach uses random sampling and sphere peeling to iteratively find k mean points. At the jth iterative step, it draws $j - 1$ peeling spheres centered at the $j - 1$ already obtained mean points, and takes a random sample from the points outside the peeling spheres to find the jth mean point. Due to the locality property, the points belonging to the first $j - 1$ clusters lie inside their corresponding $j - 1$ Voronoi cells; that is, for each peeling sphere, most of the covered points belong to their corresponding cluster, and thus ensures the correctness of the peeling step.

However, when the additional constraint (such as coloring or size) is imposed on the points, the locality property may no longer exist (see Fig. 1b), and thus the correctness of the peeling step cannot always be guaranteed. In this scenario, the core-set technique [30] is also unlikely to be able to resolve the issue. The main reason is that although the core-set can greatly reduce the size of the input points, it is quite challenging to impose the constraint through the core-set.

To overcome this challenge, we present a unified framework, called *Peeling-and-Enclosing (PnE)*, in this paper, based on a standalone new geometric technique called *Simplex Lemma*. The simplex lemma aims to address the major obstacle encountered by the peeling strategy in [43] for constrained clustering problems. More specifically,

due to the loss of locality, at the jth peeling step, the points of the jth cluster Opt_j could be scattered over all the Voronoi cells of the first $j-1$ mean points, and therefore their mean point can no longer be simply determined by the sample outside the $j-1$ peeling spheres. To resolve this issue, our main idea is to view Opt_j as the union of j unknown subsets, $Q_1, \ldots, Q_j$, with each Q_l, $1 \le l \le j\text{-}1$, being the set of points inside the Voronoi cell (or peeling sphere) of the obtained lth mean point and Q_j being the set of remaining points of Opt_j. After approximating the mean point of each unknown subset by using random sampling, we build a simplex to enclose a region which contains the mean point of Opt_j, and then search the simplex region for a good approximation of the jth mean point. To make this approach work, we need to overcome two difficulties: (a) how to generate the desired simplex to contain the jth mean point, and (b) how to efficiently search the (approximate) jth mean point inside the simplex.

For difficulty (a), our idea is to use the already determined $j-1$ mean points (which can be shown that they are also the approximate mean points of $Q_1, \ldots, Q_{j-1}$, respectively) and another point, which is the mean of those points in Opt_j outside the peeling spheres (or Voronoi cells) of the first $j-1$ mean points (i.e., Q_j), to build a $(j-1)$-dimensional simplex to contain the jth mean point. Since we do not know how Opt_j is partitioned (i.e., how Opt_j intersects the $j-1$ peeling spheres), we vary the radii of the peeling spheres $O(\log n)$ times to guess the partition and generate a set of simplexes, where the radius candidates are based on an upper bound of the optimal value determined by a novel estimation algorithm (in Sect. 3.4). We show that among the set of simplexes, one of them contains the jth (approximate) mean point.

For difficulty (b), our simplex lemma (in Sect. 2) shows that if each vertex v_l of the simplex $\mathcal{V}$ is the (approximate) mean point of Q_l, then we can find a good approximation of the mean point of Opt_j by searching a small-size grid inside $\mathcal{V}$. A nice feature of the simplex lemma is that the grid size is independent of the dimensionality of the space and thus can be used to handle high dimensional data. In some sense, our simplex lemma can be viewed as a considerable generalization of the well-known sampling lemma (i.e., Lemma 4 in this paper) in [37], which has been widely used for estimating the mean of a point set through random sampling [29,37,43]. Different from Lemma 4, which requires a global view of the point set (meaning that the sample needs to be taken from the point set), our simplex lemma only requires some partial views (e.g., sample sets are taken from those unknown subsets whose size might be quite small). If Opt_j is the point set, our simplex lemma enables us to bound the error by the variance[2] of Opt_j (i.e., a local measure) and the optimal value of the clustering problem on the whole instance P (i.e., a global measure), and thus helps us to ensure the quality of our solution.

For the k-CMedian problem, we show that although the simplex lemma no longer holds since the median point may lie outside the simplex, a weaker version (in Sect. 5.1) exists, which searches a surrounding region of the simplex. Thus our Peeling-and-Enclosing framework works for both k-CMeans and k-CMedian. It generates in total $O((\log n)^k)$ k-tuple candidates for the constrained k mean or median points. To deter-

---

[2] Given a point set in Euclidean space, its "variance" is the average of the squared distances from the points to their mean point.
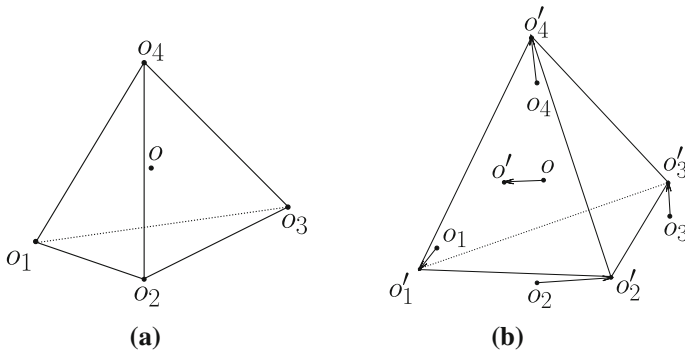
Fig. 2 Examples for Lemmas 1 and 3 with $j = 4$ respectively

mine the best k mean or median points, we need to use the property of each individual problem to design a selection algorithm. The selection algorithm takes each k-tuple candidate, computes a clustering (i.e., completing the Partition step) satisfying the additional constraint, and outputs the k-tuple with the minimum cost. We present a selection algorithm for each considered problem in Sects. 4 and 5.4.

2 Simplex Lemma

In this section, we present the *Simplex Lemma* for approximating the mean point of an **unknown** point set Q, where the only known information is a set of j points with each of them being an approximate mean point of an unknown subset of Q. In Sect. 5.1, we show how to extend the idea to approximate median point by the *Weaker Simplex Lemma*. These two lemmas are crucially used to solve the k-CMeans and k-CMedian problems.

Lemma 1 (Simplex Lemma I) *Let Q be a set of points in $\mathbb{R}^d$ with a partition of $Q = \cup_{l=1}^{j} Q_l$ and $Q_{l_1} \cap Q_{l_2} = \emptyset$ for any $l_1 \neq l_2$. Let o be the mean point of Q, and o_l be the mean point of Q_l for $1 \leq l \leq j$. Let the variance of Q be $\delta^2 = \frac{1}{|Q|} \sum_{q \in Q} ||q - o||^2$, and $\mathcal{V}$ be the simplex determined by $\{o_1, \ldots, o_j\}$. Then for any $0 < \epsilon \leq 1$, it is possible to construct a grid of size $O((8j/\epsilon)^j)$ inside $\mathcal{V}$ such that at least one grid point τ satisfies the inequality $||\tau - o|| \leq \sqrt{\epsilon}\delta$.*

Figure 2a gives an example for Lemma 1. To prove Lemma 1, we first introduce the following lemma.

Lemma 2 *Let Q be a set of points in $\mathbb{R}^d$, and Q_1 be a subset containing $\alpha|Q|$ points for some $0 < \alpha \leq 1$. Let o and o_1 be the mean points of Q and Q_1, respectively. Then $||o_1 - o|| \leq \sqrt{\frac{1-\alpha}{\alpha}}\delta$, where $\delta^2 = \frac{1}{|Q|} \sum_{q \in Q} ||q - o||^2$.*

Proof of Lemma 2 The following claim has been proved in [43].

Claim 1 *Let Q be a set of points in $\mathbb{R}^d$ space, and o be the mean point of Q. For any point $\tilde{o} \in \mathbb{R}^d$, $\sum_{q \in Q} ||q - \tilde{o}||^2 = \sum_{q \in Q} ||q - o||^2 + |Q| \times ||o - \tilde{o}||^2$.*

Let $Q_2 = Q \backslash Q_1$, and o_2 be its mean point. By Claim 1, we have the following two equalities.

$$\sum_{q \in Q_1} ||q - o||^2 = \sum_{q \in Q_1} ||q - o_1||^2 + |Q_1| \times ||o_1 - o||^2, \tag{1}$$

$$\sum_{q \in Q_2} ||q - o||^2 = \sum_{q \in Q_2} ||q - o_2||^2 + |Q_2| \times ||o_2 - o||^2. \tag{2}$$

Let $L = ||o_1 - o_2||$. By the definition of mean point, we have

$$o = \frac{1}{|Q|} \sum_{q \in Q} q = \frac{1}{|Q|} \left(\sum_{q \in Q_1} q + \sum_{q \in Q_2} q \right) = \frac{1}{|Q|} (|Q_1| o_1 + |Q_2| o_2). \tag{3}$$

Thus the three points $\{o, o_1, o_2\}$ are collinear, while $||o_1 - o|| = (1 - \alpha)L$ and $||o_2 - o|| = \alpha L$. Meanwhile, by the definition of δ, we have

$$\delta^2 = \frac{1}{|Q|} \left(\sum_{q \in Q_1} ||q - o||^2 + \sum_{q \in Q_2} ||q - o||^2 \right). \tag{4}$$

Combining (1) and (2), we have

$$\begin{aligned}
\delta^2 &= \frac{1}{|Q|} \left(\sum_{q \in Q_1} ||q - o_1||^2 + |Q_1| \times ||o_1 - o||^2 \right. \\
&\qquad \left. + \sum_{q \in Q_2} ||q - o_2||^2 + |Q_2| \times ||o_2 - o||^2 \right) \\
&\geq \frac{1}{|Q|} \left(|Q_1| \times ||o_1 - o||^2 + |Q_2| \times ||o_2 - o||^2 \right) \\
&= \alpha((1 - \alpha)L)^2 + (1 - \alpha)(\alpha L)^2 \\
&= \alpha(1 - \alpha)L^2. \tag{5}
\end{aligned}$$

Thus, we have $L \leq \frac{\delta}{\sqrt{\alpha(1-\alpha)}}$, which means that $||o_1 - o|| = (1 - \alpha)L \leq \sqrt{\frac{1-\alpha}{\alpha}} \delta$.   □

Proof of Lemma 1 We prove this lemma by induction on j.

Base case For $j = 1$, since $Q_1 = Q$, $o_1 = o$. Thus, the simplex $\mathcal{V}$ and the grid are all simply the point o_1. Clearly $\tau = o_1$ satisfies the inequality.

Induction step Assume that the lemma holds for any $j \leq j_0$ for some $j_0 \geq 1$ (i.e., the induction hypothesis). Now we consider the case of $j = j_0 + 1$. First, we assume that $\frac{|Q_l|}{|Q|} \geq \frac{\epsilon}{4j}$ for each $1 \leq l \leq j$. Otherwise, we can reduce the problem to the case of a smaller j in the following way. Let $I = \{l | 1 \leq l \leq j, \frac{|Q_l|}{|Q|} < \frac{\epsilon}{4j}\}$ be the

index set of small subsets. Then, $\frac{\sum_{l \in I} |Q_l|}{|Q|} < \frac{\epsilon}{4}$, and $\frac{\sum_{l \notin I} |Q_l|}{|Q|} \geq 1 - \frac{\epsilon}{4}$. By Lemma 2, we know that $||o' - o|| \leq \sqrt{\frac{\epsilon/4}{1-\epsilon/4}}\delta$, where o' is the mean point of $\cup_{l \notin I} Q_l$. Let $(\delta')^2$ be the variance of $\cup_{l \notin I} Q_l$. Then, we have $(\delta')^2 \leq \frac{|Q|}{|\cup_{l \notin I} Q_l|}\delta^2 \leq \frac{1}{1-\epsilon/4}\delta^2$. Thus, if we replace Q and ϵ by $\cup_{l \notin I} Q_l$ and $\frac{\epsilon}{16}$, respectively, and find a point τ such that $||\tau - o'||^2 \leq \frac{\epsilon}{16}(\delta')^2 \leq \frac{\epsilon/16}{1-\epsilon/4}\delta^2$, then we have

$$||\tau - o||^2 \leq (||\tau - o'|| + ||o' - o||)^2 \leq \frac{\frac{9}{16}\epsilon}{1 - \epsilon/4}\delta^2 \leq \epsilon\delta^2, \tag{6}$$

where the last inequality is due to the fact $\epsilon < 1$. This means that we can reduce the problem to a problem with the point set $\cup_{l \notin I} Q_l$ and a smaller j (i.e., $j - |I|$). By the induction hypothesis, we know that the reduced problem can be solved, where the new simplex would be a subset of $\mathcal{V}$ determined by $\{o_l \mid 1 \leq l \leq j, l \notin I\}$, and therefore the induction step holds for this case. Note that in general, we do not know I, but we can enumerate all the 2^j possible combinations to guess I if j is a fixed number as is the case in the algorithm in Sect. 3.2. Thus, in the following discussion, we can assume that $\frac{|Q_l|}{|Q|} \geq \frac{\epsilon}{4j}$ for each $1 \leq l \leq j$.

For each $1 \leq l \leq j$, since $\frac{|Q_l|}{|Q|} \geq \frac{\epsilon}{4j}$, by Lemma 2, we know that $||o_l - o|| \leq \sqrt{\frac{1-\frac{\epsilon}{4j}}{\frac{\epsilon}{4j}}}\delta \leq 2\sqrt{\frac{j}{\epsilon}}\delta$. This, together with triangle inequality, implies that for any $1 \leq l, l' \leq j$,

$$||o_l - o_{l'}|| \leq ||o_l - o|| + ||o_{l'} - o|| \leq 4\sqrt{j/\epsilon}\delta. \tag{7}$$

Thus, if we pick any index l_0, and draw a ball $\mathcal{B}$ centered at o_{l_0} and with radius $r = \max_{1 \leq l \leq j}\{||o_l - o_{l_0}||\} \leq 4\sqrt{j/\epsilon}\delta$ [by (7)], the whole simplex $\mathcal{V}$ will be inside $\mathcal{B}$. Note that $o = \sum_{l=1}^{j} \frac{|Q_l|}{|Q|}o_l$, so o lies inside the simplex $\mathcal{V}$. To guarantee that o is contained by the ball $\mathcal{B}$, we can construct $\mathcal{B}$ only in the $(j - 1)$-dimensional space spanned by $\{o_1, \ldots, o_j\}$, rather than the whole $\mathbb{R}^d$ space. Also, if we build a grid inside $\mathcal{B}$ with grid length $\frac{\epsilon r}{4j}$, i.e., generating a uniform mesh with each cell being a $(j - 1)$-dimensional hypercube of edge length $\frac{\epsilon r}{4j}$, the total number of grid points is no more than $O((\frac{8j}{\epsilon})^j)$. With this grid, we know that for any point p inside $\mathcal{V}$, there exists a grid point g such that $||g - p|| \leq \sqrt{j(\frac{\epsilon r}{4j})^2} = \frac{\epsilon}{4\sqrt{j}}r \leq \sqrt{\epsilon}\delta$. This means that we can find a grid point τ inside $\mathcal{V}$, such that $||\tau - o|| \leq \sqrt{\epsilon}\delta$. Thus, the induction step holds, and the lemma is true for any $j \geq 1$. $\qquad\square$

In the above lemma, we assume that the exact positions of $\{o_1, \ldots, o_j\}$ are known (see Fig. 2a). However, in some scenarios (*e.g.*, in the Algorithm in Sect. 3.2), we only know an approximate position of each mean point o_i (see Fig. 2b). The following lemma shows that an approximate position of o can still be similarly determined (see Sect. 7.1 for the proof).
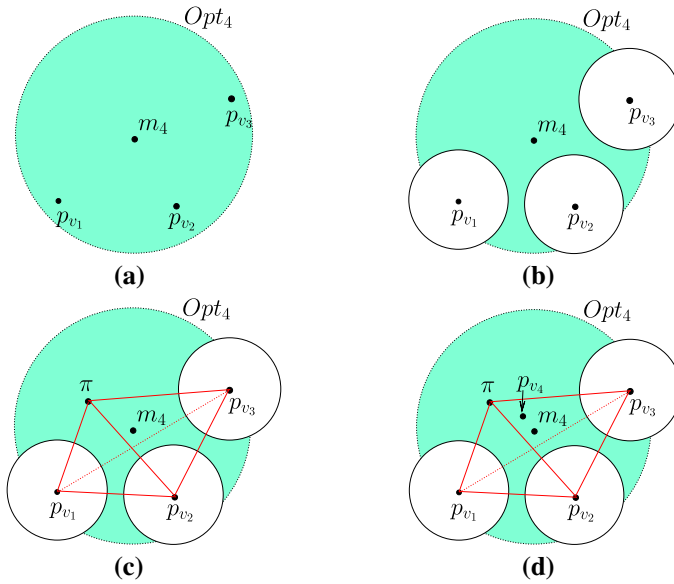
Fig. 3 Illustration for one iteration of Peeling-and-Enclosing. **a** Beginning of iteration 4; **b** generate 3 spheres (in white) to peel the optimal cluster Opt_4 (in green); **c** build a simplex (in red) to contain m_4; **d** find an approximate mean point p_{v_4} for m_4 (Color figure online)

Lemma 3 (Simplex Lemma II) *Let Q, o, Q_l, o_l, $1 \leq l \leq j$, and δ be defined as in Lemma 1. Let $\{o'_1, \ldots, o'_j\}$ be j points in $\mathbb{R}^d$ such that $||o'_l - o_l|| \leq L$ for $1 \leq l \leq j$ and $L > 0$, and $\mathcal{V}'$ be the simplex determined by $\{o'_1, \ldots, o'_j\}$. Then for any $0 < \epsilon \leq 1$, it is possible to construct a grid of size $O((8j/\epsilon)^j)$ inside $\mathcal{V}'$ such that at least one grid point τ satisfies the inequality $||\tau - o|| \leq \sqrt{\epsilon}\delta + (1 + \epsilon)L$.*

3 Peeling-and-Enclosing Algorithm for *k*-CMeans

In this section, we present a new Peeling-and-Enclosing (PnE) algorithm for generating a set of candidates for the mean points of k-CMeans. Our algorithm uses peeling spheres and the simplex lemma to iteratively find a good candidate center for each unknown cluster. An overview of the algorithm is given in Sect. 3.1.

Some notations Let $P = \{p_1, \ldots, p_n\}$ be the set of $\mathbb{R}^d$ points in k-CMeans, and $\mathcal{OPT} = \{Opt_1, \ldots, Opt_k\}$ be the k unknown optimal constrained clusters with m_j being the mean point of Opt_j for $1 \leq j \leq k$. Without loss of generality, we assume that $|Opt_1| \geq |Opt_2| \geq \cdots \geq |Opt_k|$. Denote by δ_{opt}^2 the optimal objective value, i.e., $\delta_{opt}^2 = \frac{1}{n} \sum_{j=1}^{k} \sum_{p \in Opt_j} ||p - m_j||^2$. We also set $\epsilon > 0$ as the parameter related to the quality of the approximate clustering result.

3.1 Overview of the Peeling-and-Enclosing Algorithm

Our Peeling-and-Enclosing algorithm needs an upper bound Δ on the optimal value δ_{opt}^2. Specifically, δ_{opt}^2 satisfies the condition $\Delta/c \leq \delta_{opt}^2 \leq \Delta$ for some constant $c \geq 1$. In Sect. 3.4, we will present a novel algorithm to determine such an upper bound for general constrained k-means clustering problems. Then, it searches for a $(1 + \epsilon)$-approximation δ^2 of δ_{opt}^2 in the set

$$H = \{\Delta/c, (1 + \epsilon)\Delta/c, (1 + \epsilon)^2\Delta/c, \ldots, (1 + \epsilon)^{\lceil \log_{1+\epsilon} c \rceil}\Delta/c \geq \Delta\}. \quad (8)$$

Obviously, there exists one element of H lying inside the interval $[\delta_{opt}^2, (1 + \epsilon)\delta_{opt}^2]$, and the size of H is $O(\frac{1}{\epsilon} \log c)$.

At each searching step, our algorithm performs a sphere-peeling and simplex-enclosing procedure to iteratively generate k approximate mean points for the constrained clusters. Initially, our algorithm uses Lemmas 4 and 5 to find an approximate mean point p_{v_1} for Opt_1 (note that since Opt_1 is the largest cluster, $|Opt_1|/n \geq 1/k$ and the sampling lemma applies). At the $(j + 1)$th iteration, it already has the approximate mean points $p_{v_1}, \ldots, p_{v_j}$ for $Opt_1, \ldots, Opt_j$, respectively (see Fig. 3a). Due to the lack of locality, some points of Opt_{j+1} could be scattered over the regions (e.g., Voronoi cells or peeling spheres) of $Opt_1, \ldots, Opt_j$ and are difficult to be distinguished from the points in these clusters. Since the number of such points could be small (comparing to that of the first j clusters), they need to be handled differently from the remaining points. Our idea is to separate them using j peeling spheres, $B_{j+1,1}, \ldots, B_{j+1,j}$, centered at the j approximate mean points respectively and with some properly guessed radius (see Fig. 3b). Let $\mathcal{A}$ be the set of unknown points in $Opt_{j+1} \backslash (\cup_{l=1}^{j} B_{j+1,l})$. Our algorithm considers two cases, (a) $|\mathcal{A}|$ is large enough and (b) $|\mathcal{A}|$ is small. For case (a), since $|\mathcal{A}|$ is large enough, we can use Lemma 4 and Lemma 5 to find an approximate mean point π of $\mathcal{A}$, and then construct a simplex determined by π and $p_{v_1}, \ldots, p_{v_j}$ to contain the $j + 1$th mean point (see Fig. 3c). Note that $\mathcal{A}$ and $Opt_{j+1} \cap B_{j+1,l}$, $1 \leq l \leq j$, can be viewed as a partition of Opt_{j+1} where the points covered by multiple peeling spheres can be assigned to anyone of them, and p_{v_l} can be shown as an approximate mean point of $Opt_{j+1} \cap B_{j+1,l}$; thus the simplex lemma applies. For case (b), it directly constructs a simplex determined just by $p_{v_1}, \ldots, p_{v_j}$. For either case, our algorithm builds a grid inside the simplex and uses Lemma 3 to find an approximate mean point for Opt_{j+1} (i.e., $p_{v_{j+1}}$, see Fig. 3d). The algorithm repeats the Peeling-and-Enclosing procedure k times to generate the k approximate mean points.

3.2 Peeling-and-Enclosing Algorithm

Before presenting our algorithm, we first introduce two basic lemmas from [24,37] for random sampling. Let S be a set of n points in $\mathbb{R}^d$ space, and T be a randomly selected subset of size t from S. Denote by $m(S)$ and $m(T)$ the mean points of S and T respectively.

Lemma 4 [37] *With probability* $1 - \eta$, $||m(S) - m(T)||^2 < \frac{1}{\eta t}\delta^2$, *where* $\delta^2 = \frac{1}{n}\sum_{s \in S}||s - m(S)||^2$ *and* $0 < \eta < 1$.

Lemma 5 [24] *Let* Ω *be a set of elements, and* S *be a subset of* Ω *with* $\frac{|S|}{|\Omega|} = \alpha$ *for some* $\alpha \in (0, 1)$. *If we randomly select* $\frac{t \ln \frac{t}{\eta}}{\ln(1+\alpha)} = O(\frac{t}{\alpha} \ln \frac{t}{\eta})$ *elements from* Ω, *then with probability at least* $1 - \eta$, *the sample contains at least* t *elements from* S *for* $0 < \eta < 1$ *and* $t \in \mathbb{Z}^+$.

Our Peeling-and-Enclosing algorithm is shown in Algorithm 1.

Algorithm 1 Peeling-and-Enclosing for k-CMeans

Input: $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$, $k \geq 2$, a constant $\epsilon \in (0, \frac{1}{4k^2})$, and an upper bound $\Delta \in [\delta_{opt}^2, c\delta_{opt}^2]$ with $c \geq 1$.
Output: A set of k-tuple candidates for the k constrained mean points.

1. For $i = 0$ to $\lceil \log_{1+\epsilon} c \rceil$ do

   (a) Set $\delta = \sqrt{(1 + \epsilon)^i \Delta / c}$, and run Algorithm 2.
   (b) Let $\mathcal{T}_i$ be the output tree.

2. For each root-to-leaf path of every $\mathcal{T}_i$, build a k-tuple candidate using the k points associated with the path.

Algorithm 2 Peeling-and-Enclosing-Tree

Input: $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$, $k \geq 2$, a constant $\epsilon \in (0, \frac{1}{4k^2})$, and $\delta > 0$.

1. Initialize $\mathcal{T}$ as a single root node v associated with no point.
2. Recursively grow each node v in the following way

   (a) If the height of v is already k, then it is a leaf.
   (b) Otherwise, let j be the height of v. Build the radius candidate set $\mathcal{R} = \cup_{t=0}^{\log n} \{\frac{1+l\frac{\epsilon}{2}}{2(1+\epsilon)} j2^{t/2}\sqrt{\epsilon}\delta \mid 0 \leq l \leq 4 + \frac{2}{\epsilon}\}$. For each $r \in \mathcal{R}$, do

      i. Let $\{p_{v_1}, \ldots, p_{v_j}\}$ be the j points associated with the nodes on the root-to-v path.
      ii. For each p_{v_l}, $1 \leq l \leq j$, construct a ball $B_{j+1,l}$ centered at p_{v_l} and with radius r.
      iii. Take a random sample from $P \setminus \cup_{l=1}^{j} B_{j+1,l}$ of size $s = \frac{8k^3}{\epsilon^9} \ln \frac{k^2}{\epsilon^6}$. Compute the mean points of all the subsets of the sample, and denote them by $\Pi = \{\pi_1, \ldots, \pi_{2^s-1}\}$.
      iv. For each $\pi_i \in \Pi$, construct a simplex using $\{p_{v_1}, \ldots, p_{v_j}, \pi_i\}$ as its vertices. Also construct another simplex using $\{p_{v_1}, \ldots, p_{v_j}\}$ as its vertices. For each simplex, build a grid with size $O((32j/\epsilon^2)^j)$ inside itself and each of its 2^j possible degenerated sub-simplices.
      v. In total, there are $2^{s+j}(32j/\epsilon^2)^j$ grid points inside the 2^s simplices. For each grid point, add one child to v, and associate it with the grid point.

3. Output $\mathcal{T}$.

Theorem 1 *Let* P *be the set of* n $\mathbb{R}^d$ *points and* $k \in \mathbb{Z}^+$ *be a fixed constant. Given a constant* $\epsilon \in (0, \frac{1}{4k^2})$, *in* $O(2^{poly(\frac{k}{\epsilon})}n(\log n)^{k+1}d)$ *time, Algorithm 1 outputs* $O(2^{poly(\frac{k}{\epsilon})}(\log n)^k)$ k-*tuple candidate mean points. With constant probability,*

there exists one k-tuple candidate in the output which is able to induce a $(1 + O(\epsilon))$-approximation of k-CMeans.

Remark 1 To increase the success probability to be close to 1, e.g., $1 - \frac{1}{n}$, one only needs to repeatedly run the algorithm $O(\log n)$ times; both the time complexity and the number of k-tuple candidates increase by a factor of $O(\log n)$.

3.3 Proof of Theorem 1

Let $\beta_j = |Opt_j|/n$, and $\delta_j^2 = \frac{1}{|Opt_j|} \sum_{p \in Opt_j} ||p - m_j||^2$, where m_j is the mean point of Opt_j. By our assumption in the beginning of Sect. 3, we know that $\beta_1 \geq \cdots \geq \beta_k$. Clearly, $\sum_{j=1}^k \beta_j = 1$ and the optimal objective value $\delta_{opt}^2 = \sum_{j=1}^k \beta_j \delta_j^2$.

Proof Synopsis Instead of directly proving Theorem 1, we consider the following Lemma 6 and Lemma 7 which jointly ensure the correctness of Theorem 1. In Lemma 6, we show that there exists such a root-to-leaf path in one of the returned trees that its associated k points along the path, denoted by $\{p_{v_1}, \ldots, p_{v_k}\}$, are close enough to the mean points $m_i, \ldots, m_k$ of the k optimal clusters, respectively. The proof is based on mathematical induction; each step needs to build a simplex, and applies Simplex Lemma II to bound the error, i.e., $||p_{v_j} - m_j||$ in (9). The error is estimated by considering both the local (i.e., the variance of cluster Opt_j) and global (i.e., the optimal value δ_{opt}) measurements. This is a more accurate estimation, comparing to the widely used Lemma 4 which considers only the local measurement. Such an improvement is due to the increased flexibility in the Simplex Lemma II, and is a key to our proof. In Lemma 7, we further show that the k points, $\{p_{v_1}, \ldots, p_{v_k}\}$, in Lemma 6 induce a $(1 + O(\epsilon))$-approximation of k-CMeans.

Lemma 6 *Among all the trees generated by Algorithm 1, with constant probability, there exists at least one tree, $\mathcal{T}_i$, which has a root-to-leaf path with each of its nodes v_j at level j $(1 \leq j \leq k)$ associating with a point p_{v_j} and satisfying the inequality*

$$||p_{v_j} - m_j|| \leq \epsilon \delta_j + (1 + \epsilon) j \sqrt{\frac{\epsilon}{\beta_j}} \delta_{opt}. \tag{9}$$

Before proving this lemma, we first show its implication.

Lemma 7 *If Lemma 6 is true, then $\{p_{v_1}, \ldots, p_{v_k}\}$ is able to induce a $(1 + O(\epsilon))$-approximation of k-CMeans.*

Proof We assume that Lemma 6 is true. Then for each $1 \leq j \leq k$, we have

$$\sum_{p \in Opt_j} ||p - p_{v_j}||^2 = \sum_{p \in Opt_j} ||p - m_j||^2 + |Opt_j| \times ||m_j - p_{v_j}||^2$$

$$\leq \sum_{p \in Opt_j} ||p - m_j||^2 + |Opt_j| \times 2 \left(\epsilon^2 \delta_j^2 + (1 + \epsilon)^2 j^2 \frac{\epsilon}{\beta_j} \delta_{opt}^2 \right)$$

$$= (1 + 2\epsilon^2)|Opt_j|\delta_j^2 + 2(1 + \epsilon)^2 j^2 \epsilon n \delta_{opt}^2, \tag{10}$$

where the first equation follows from Claim 1 in the proof of Lemma 2 (note that m_j is the mean point of Opt_j), the inequality follows from Lemma 6 and the fact that $(a+b)^2 \leq 2(a^2+b^2)$ for any two real numbers a and b, and the last equality follows from the fact that $\frac{|Opt_j|}{\beta_j} = n$. Summing both sides of (10) over j, we have

$$\sum_{j=1}^{k} \sum_{p \in Opt_j} ||p - p_{v_j}||^2 \leq \sum_{j=1}^{k} ((1 + 2\epsilon^2)|Opt_j|\delta_j^2 + 2(1+\epsilon)^2 j^2 \epsilon n \delta_{opt}^2)$$

$$\leq (1 + 2\epsilon^2) \sum_{j=1}^{k} |Opt_j|\delta_j^2 + 2(1+\epsilon)^2 k^3 \epsilon n \delta_{opt}^2$$

$$= (1 + O(k^3)\epsilon)n\delta_{opt}^2, \tag{11}$$

where the last equation follows from the fact that $\sum_{j=1}^{k} |Opt_j|\delta_j^2 = n\delta_{opt}^2$. By (11), we know that $\{p_{v_1}, \ldots, p_{v_k}\}$ will induce a $(1 + O(k^3)\epsilon)$-approximation for k-CMeans. Note that k is assumed to be a fixed number. Thus the lemma is true. $\qquad\square$

Lemma 7 implies that Lemma 6 is indeed sufficient to ensure the correctness of Theorem 1 (except for the number of candidates and the time complexity). Now we prove Lemma 6.

Proof of Lemma 6 Let $\mathcal{T}_i$ be the tree generated by Algorithm 2 when δ falls in the interval of $[\delta_{opt}, (1 + \epsilon)\delta_{opt}]$. We will focus our discussion on $\mathcal{T}_i$, and prove the lemma by mathematical induction on j.

Base case For $j = 1$, since $\beta_1 = \max\{\beta_j | 1 \leq j \leq k\}$, we have $\beta_1 \geq \frac{1}{k}$. By Lemma 4 and Lemma 5, we can find the approximate mean point through random sampling. Let Ω and S (in Lemma 5) be P and Opt_1, respectively. Also, p_{v_1} is the mean point of the random sample from P. Lemma 5 ensures that the sample contains enough number of points from Opt_1, and Lemma 4 implies that $||p_{v_1} - m_1|| \leq \epsilon\delta_1 \leq \epsilon\delta_1 + (1+\epsilon)\sqrt{\frac{\epsilon}{\beta_1}}\delta_{opt}$.

Induction step Suppose $j > 1$. We assume that there is a path in $\mathcal{T}_i$ from the root to the $(j-1)$th level, such that for each $1 \leq l \leq j - 1$, the level-l node v_l on the path is associated with a point p_{v_l} satisfying the inequality $||p_{v_l} - m_l|| \leq \epsilon\delta_l + (1+\epsilon)l\sqrt{\frac{\epsilon}{\beta_l}}\delta_{opt}$ (i.e., the induction hypothesis). Now we consider the case of j. Below we will show that there is one child of v_{j-1}, i.e., v_j, such that its associated point p_{v_j} satisfies the inequality $||p_{v_j} - m_j|| \leq \epsilon\delta_j + (1+\epsilon)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}$. First, we have the following claim (see Sect. 7.2 for the proof).

Claim 2 *In the set of radius candidates in Algorithm 2, there exists one value $r_j \in \mathcal{R}$ such that*

$$r_j \in \left[j\sqrt{\epsilon/\beta_j}\delta_{opt}, \left(1 + \frac{\epsilon}{2}\right) j\sqrt{\epsilon/\beta_j}\delta_{opt} \right]. \tag{12}$$

Now, we construct the $j - 1$ peeling spheres, $\{B_{j,1}, \ldots, B_{j,j-1}\}$ as in Algorithm 2. For each $1 \leq l \leq j - 1$, $B_{j,l}$ is centered at p_{v_l} and with radius r_j. By Markov's inequality and the induction hypothesis, we have the following claim (see Sect. 7.3 for the proof).

Claim 3 *For each* $1 \leq l \leq j - 1$, $|Opt_l \setminus (\bigcup_{w=1}^{j-1} B_{j,w})| \leq \frac{4\beta_j n}{\epsilon}$.

Claim 3 shows that $|Opt_l \setminus (\bigcup_{w=1}^{j-1} B_{j,w})|$ is bounded for $1 \leq l \leq j-1$, which helps us to find the approximate mean point of Opt_j. Induced by the $j - 1$ peeling spheres $\{B_{j,1}, \ldots, B_{j,j-1}\}$, Opt_j is divided into j subsets, $Opt_j \cap B_{j,1}, \ldots, Opt_j \cap B_{j,j-1}$ and $Opt_j \setminus (\bigcup_{w=1}^{j-1} B_{j,w})$. For ease of discussion, let P_l denote $Opt_j \cap B_{j,l}$ for $1 \leq l \leq j-1$, P_j denote $Opt_j \setminus (\bigcup_{w=1}^{j-1} B_{j,w})$, and τ_l denote the mean point of P_l for $1 \leq l \leq j$. Note that the peeling spheres may intersect with each other. For any two intersecting spheres B_{j,l_1} and B_{j,l_2}, we arbitrarily assign the points in $Opt_j \cap (B_{j,l_1} \cap B_{j,l_2})$ to either P_{l_1} or P_{l_2}. Thus, we can assume that $\{P_l \mid 1 \leq l \leq j\}$ are pairwise disjoint.

Now consider the size of P_j. We have the following two cases: (a) $|P_j| \geq \epsilon^3 \frac{\beta_j}{j} n$ and (b) $|P_j| < \epsilon^3 \frac{\beta_j}{j} n$. We show how, in each case, Algorithm 2 can obtain an approximate mean point for Opt_j by using the simplex lemma (i.e., Lemma 3).

For case (a), by Claim 3, together with the fact that $\beta_l \leq \beta_j$ for $l > j$, we know that

$$\sum_{l=1}^{k} |Opt_l \setminus \left(\bigcup_{w=1}^{j-1} B_{j,w} \right)| \leq \sum_{l=1}^{j-1} |Opt_l \setminus \left(\bigcup_{w=1}^{j-1} B_{j,w} \right)| + |P_j| + \sum_{l=j+1}^{k} |Opt_l|$$

$$\leq \frac{4(j-1)\beta_j}{\epsilon} n + |P_j| + (k-j)\beta_j n, \tag{13}$$

where the second inequality follows from Claim 3. So we have

$$\frac{|P_j|}{\sum_{l=1}^{k} |Opt_l \setminus \left(\bigcup_{w=1}^{j-1} B_{j,w} \right)|} \geq \frac{|P_j|}{\frac{4(j-1)\beta_j}{\epsilon} n + |P_j| + (k-j)\beta_j n}. \tag{14}$$

We view the right-hand side as a function of $|P_j|$. Given any $h > 0$, the function $f(x) = \frac{x}{x+h}$ is an increasing function on the variable $x \in [0, +\infty)$. Note that we assume $|P_j| \geq \epsilon^3 \frac{\beta_j}{j} n$. Thus

$$\frac{|P_j|}{\sum_{l=1}^{k} |Opt_l \setminus \left(\bigcup_{w=1}^{j-1} B_{j,w} \right)|} \geq \frac{\frac{\epsilon^3}{j}\beta_j n}{\frac{4(j-1)\beta_j}{\epsilon} n + \frac{\epsilon^3}{j}\beta_j n + (k-j)\beta_j n}$$

$$> \frac{\epsilon^4}{8kj} \geq \frac{\epsilon^4}{8k^2}, \tag{15}$$

(15) implies that P_j is large enough, comparing to the set of points outside the peeling spheres. Hence, we can obtain an approximate mean point π for P_j in the following

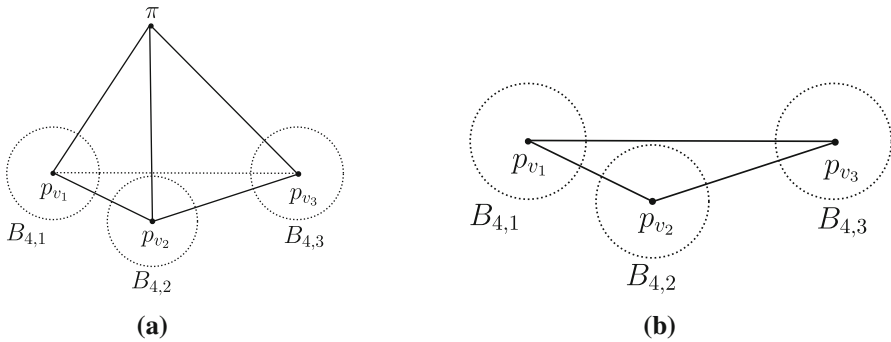**Fig. 4** **a**, **b** The Simplexes of case (a) and case (b) with $j = 4$ respectively

way. First, we set $t = \frac{k}{\epsilon^5}$, $\eta = \frac{\epsilon}{k}$, and take a sample of size $\frac{t \ln(t/\eta)}{\epsilon^4/8k^2} = \frac{8k^3}{\epsilon^9} \ln \frac{k^2}{\epsilon^6}$. By Lemma 5, we know that with probability $1 - \frac{\epsilon}{k}$, the sample contains at least $\frac{k}{\epsilon^5}$ points from P_j. Then we let π be the mean point of the $\frac{k}{\epsilon^5}$ points from P_j, and a^2 be the variance of P_j. By Lemma 4, we know that with probability $1 - \frac{\epsilon}{k}$, $||\pi - \tau_j||^2 \leq \epsilon^4 a^2$. Also, since $\frac{|P_j|}{|Opt_j|} = \frac{|P_j|}{\beta_j n} \geq \frac{\epsilon^3}{j}$ (because $|P_j| \geq \epsilon^3 \frac{\beta_j}{j} n$ for case (a)), we have $a^2 \leq \frac{|Opt_j|}{|P_j|} \delta_j^2 \leq \frac{j}{\epsilon^3} \delta_j^2$. Thus,

$$||\pi - \tau_j||^2 \leq \epsilon j \delta_j^2. \tag{16}$$

Once obtaining π, we can apply Lemma 3 to find a point p_{v_j} satisfying the condition of $||p_{v_j} - m_j|| \leq \epsilon \delta_j + (1 + \epsilon) j \sqrt{\frac{\epsilon}{\beta_j}} \delta_{opt}$. We construct a simplex $\mathcal{V}'_{(a)}$ with vertices $\{p_{v_1}, \ldots, p_{v_{j-1}}\}$ and π (see Fig. 4b). Note that Opt_j is partitioned by the peeling spheres into j disjoint subsets, $P_1, \ldots, P_j$. Each P_l ($1 \leq l \leq j - 1$) lies inside $B_{j,l}$, which implies that τ_l, i.e., the mean point of P_l, is also inside $B_{j,l}$. Further, by Claim 2, for $1 \leq l \leq j - 1$, we have

$$||p_{v_l} - \tau_l|| \leq r_j \leq \left(1 + \frac{\epsilon}{2}\right) j \sqrt{\epsilon/\beta_j} \delta_{opt}. \tag{17}$$

Recall that $\beta_j \delta_j^2 \leq \delta_{opt}^2$. Thus, together with (16), we have

$$||\pi - \tau_j|| \leq \sqrt{\epsilon j} \delta_j \leq \sqrt{\epsilon j/\beta_j} \delta_{opt}. \tag{18}$$

By (17) and (18), if setting the value of L (in Lemma 3) to be

$$\max\{r_j, ||\pi - \tau_j||\} \leq \max\left\{\left(1 + \frac{\epsilon}{2}\right) j \sqrt{\epsilon/\beta_j} \delta_{opt}, \sqrt{\epsilon j/\beta_j} \delta_{opt}\right\}$$
$$= \left(1 + \frac{\epsilon}{2}\right) j \sqrt{\epsilon/\beta_j} \delta_{opt}, \tag{19}$$

and the value of ϵ (in Lemma 3) to be $\epsilon_0 = \epsilon^2/4$, by Lemma 3 we can construct a grid inside the simplex $\mathcal{V}'_{(a)}$ with size $O((8j/\epsilon_0)^j)$ to ensure the existence of the grid point τ satisfying the inequality of

$$||\tau - m_j|| \leq \sqrt{\epsilon_0}\delta_j + (1 + \epsilon_0)L \leq \epsilon\delta_j + (1 + \epsilon)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}. \qquad (20)$$

Hence, let p_{v_j} be the grid point τ, and the induction step holds for this case.

For case (b), we can also apply Lemma 3 to find an approximate mean point in a way similar to case (a); the difference is that we construct a simplex $\mathcal{V}'_{(b)}$ with vertices $\{p_{v_1}, \ldots, p_{v_{j-1}}\}$ (see Fig. 4b). Roughly speaking, since $|P_j|$ is small, the mean points of $Opt_j \backslash P_j$ and Opt_j are very close to each other (by Lemma 2). Thus, we can ignore P_j and just consider $Opt_j \backslash P_j$.

Let a^2 and m'_j denote the variance and mean point of $Opt_j \backslash P_j$ respectively. We know that $\{P_1, P_2, \ldots, P_{j-1}\}$ is a partition on $Opt_j \backslash P_j$. Thus, similar with case (a), we construct a simplex $\mathcal{V}'_{(b)}$ determined by $\{p_{v_1}, \ldots, p_{v_{j-1}}\}$ (see Fig. 4b), set the value of L to be $r_j \leq (1 + \frac{\epsilon}{2})j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}$, and then build a grid inside $\mathcal{V}'_{(b)}$ with size $O((\frac{8j}{\epsilon_0})^j)$, where $\epsilon_0 = \epsilon^2/4$. By Lemma 3, we know that there exists one grid point τ satisfying the condition of

$$||\tau - m'_j|| \leq \sqrt{\epsilon_0}a + (1 + \epsilon_0)L \leq \frac{\epsilon}{2}a + (1 + \epsilon)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}. \qquad (21)$$

Meanwhile, we know that $|Opt_j \backslash P_j| \geq (1 - \epsilon^3/j)|Opt_j|$, since $|P_j| \leq \frac{\epsilon^3}{j}|Opt_j|$. Thus, we have $a^2 \leq \frac{|Opt_j|}{|Opt_j \backslash P_j|}\delta_j^2 \leq \frac{1}{1-\epsilon^3/j}\delta_j^2$, and $||m'_j - m_j|| \leq \sqrt{\frac{\epsilon^3/j}{1-\epsilon^3/j}}\delta_j$ (by Lemma 2). Together with (21), we have

$$
\begin{aligned}
||\tau - m_j|| &\leq ||\tau - m'_j|| + ||m'_j - m_j|| \\
&\leq \frac{\epsilon}{2}a + (1 + \epsilon)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} + \sqrt{\frac{\epsilon^3/j}{1-\epsilon^3/j}}\delta_j \\
&\leq \frac{\epsilon}{2}\sqrt{\frac{1}{1-\epsilon^3/j}}\delta_j + (1 + \epsilon)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} + \sqrt{\frac{\epsilon^3/j}{1-\epsilon^3/j}}\delta_j \\
&\leq \left(\frac{\epsilon}{2}\sqrt{\frac{1}{1-\epsilon^3/j}} + \sqrt{\frac{\epsilon^3/j}{1-\epsilon^3/j}}\right)\delta_j + (1 + \epsilon)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} \\
&\leq \epsilon\delta_j + (1 + \epsilon)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}. \qquad (22)
\end{aligned}
$$

Hence, let p_{v_j} be the grid point τ, and the induction step holds for this case.

Since Algorithm 2 executes every step in our above discussion, the induction step, as well as the lemma, is true. $\qquad\square$

Success Probability From the above analysis, we know that in the jth iteration, only case (a) (i.e., $|P_j| \geq \epsilon^3 \frac{\beta_j}{j} n$) needs to consider the success probability of random sampling. Recall that in case (a), we take a sample of size $\frac{8k^3}{\epsilon^9} \ln \frac{k^2}{\epsilon^6}$. Thus with probability $1 - \frac{\epsilon}{k}$, it contains at least $\frac{k}{\epsilon^5}$ points from P_j. Meanwhile, with probability $1 - \frac{\epsilon}{k}$, $||\pi - \tau_j||^2 \leq \epsilon^4 a^2$. Hence, the success probability in the jth iteration is $(1 - \frac{\epsilon}{k})^2$. By taking the union bound, the success probability in all k iterations is $(1 - \frac{\epsilon}{k})^{2k} \geq 1 - 2\epsilon$.

Number of Candidates and Running Time Algorithm 1 calls Algorithm 2 $O(\frac{1}{\epsilon} \log c)$ times (in Sect. 3.4, we will show that c can be a constant number). It is easy to see that each node in the returned tree has $|\mathcal{R}|2^{s+j}(\frac{32j}{\epsilon^2})^j$ children, where $|\mathcal{R}| = O(\frac{\log n}{\epsilon})$, and $s = \frac{8k^3}{\epsilon^9} \ln \frac{k^2}{\epsilon^6}$. Since the tree has the height of k, the complexity of the tree is $O(2^{poly(\frac{k}{\epsilon})}(\log n)^k)$. Consequently, the number of candidates is $O(2^{poly(\frac{k}{\epsilon})}(\log n)^k)$. Further, since each node takes $O(|\mathcal{R}|2^{s+j}(\frac{32j}{\epsilon^2})^j nd)$ time, the total time complexity of the algorithm is $O(2^{poly(\frac{k}{\epsilon})} n(\log n)^{k+1}d)$.

3.4 Upper Bound Estimation

As mentioned in Sect. 3.1, our Peeling-and-Enclosing algorithm needs an upper bound Δ on the optimal value δ_{opt}^2. To compute this, our main idea is to use some unconstrained k-means clustering algorithm $\mathcal{A}_*$ (e.g., the linear time $(1+\epsilon)$-approximation algorithm in [43]) on the input points P without considering the constraint, to obtain a λ-approximation to the k-means clustering for some constant $\lambda > 1$. Let $\mathcal{C} = \{c_1, \ldots, c_k\}$ be the set of mean points returned by algorithm $\mathcal{A}_*$.[3] Below, we show that the Cartesian product $[\mathcal{C}]^k = \underbrace{\mathcal{C} \times \cdots \times \mathcal{C}}_{k}$ contains one k-tuple, which is an $(18\lambda + 16)$-approximation of k-CMeans on the same input P. Clearly, to select the k-tuple from $[\mathcal{C}]^k$ with the smallest objective value, we still need to solve the Partition step on each k-tuple to form the desired clusters. Similar to Remark 1, we refer the reader to Sect. 4 for the selection algorithms for the considered constrained clustering problems.

Theorem 2 *Let $P = \{p_1, \ldots, p_n\}$ be the input points of k-CMeans, and $\mathcal{C} = \{c_1, \ldots, c_k\}$ be the mean points of a λ-approximation of the k-means clustering on P (without considering the constraint) for some constant $\lambda \geq 1$. Then $[\mathcal{C}]^k$ contains at least one k-tuple which is able to induce an $(18\lambda + 16)$-approximation of k-CMeans.*

Proof Synopsis Let ω be the objective value of the k-means clustering on P corresponding to the k mean points in $\mathcal{C}$. To prove Theorem 2, we create a new instance of k-CMeans: for each point $p_i \in P$, move it to its nearest point, say c_t, in $\{c_1, \ldots, c_k\}$; let $\tilde{p}_i$ denote the new p_i (note that c_t and $\tilde{p}_i$ coincide with each other; see Fig. 5a). The set $\tilde{P} = \{\tilde{p}_1, \ldots, \tilde{p}_n\}$ forms a new instance of k-CMeans. Let $\tilde{\delta}_{opt}^2$ be the optimal value of k-CMeans on $\tilde{P}$, and $\delta_{opt}^2([\mathcal{C}]^k)$ be the minimum cost of k-CMeans on P by

---

[3] Note that they are different from $\{m_1, \ldots, m_k\}$, which are the mean points of the k optimal constrained clusters $\{Opt_1, \ldots, Opt_k\}$ of P defined in the beginning of Sect. 3.
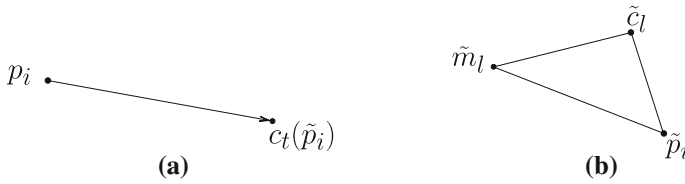
Fig. 5 **a** p_i is moved to c_t and becomes $\tilde{p}_i$; **b** $||\tilde{m}_l - \tilde{c}_l|| \leq ||\tilde{m}_l - \tilde{p}_i||$

restricting its mean points to be one k-tuple in $[\mathcal{C}]^k$. We show that $\tilde{\delta}_{opt}^2$ is bounded by a combination of ω and δ_{opt}^2, and $\delta_{opt}^2([\mathcal{C}]^k)$ is bounded by a combination of ω and $\tilde{\delta}_{opt}^2$ (see Lemma 8). Together with the fact that ω is no more than $\lambda\delta_{opt}^2$, we consequently obtain that $\delta_{opt}^2([\mathcal{C}]^k) \leq (18\lambda + 16)\delta_{opt}^2$, which implies Theorem 2.

Lemma 8 $\tilde{\delta}_{opt}^2 \leq 2\omega + 2\delta_{opt}^2$, and $\delta_{opt}^2([\mathcal{C}]^k) \leq 2\omega + 8\tilde{\delta}_{opt}^2$.

Proof We first prove the inequality of $\tilde{\delta}_{opt}^2 \leq 2\omega + 2\delta_{opt}^2$. Consider any point $p_i \in P$. Let Opt_l be the optimal cluster containing p_i. Then, we have

$$\begin{aligned}
||\tilde{p}_i - m_l||^2 &\leq (||\tilde{p}_i - p_i|| + ||p_i - m_l||)^2 \\
&\leq 2||\tilde{p}_i - p_i||^2 + 2||p_i - m_l||^2,
\end{aligned} \tag{23}$$

where the first inequality follows from triangle inequality, and the second inequality follows from the fact that $(a + b)^2 \leq 2a^2 + 2b^2$ for any two real numbers a and b. For both sides of (23), we take the averages over all the points in P, and obtain

$$\frac{1}{n}\sum_{l=1}^{k}\sum_{p_i \in Opt_l}||\tilde{p}_i - m_l||^2 \leq \frac{2}{n}\sum_{i=1}^{n}||\tilde{p}_i - p_i||^2 + \frac{2}{n}\sum_{l=1}^{k}\sum_{p_i \in Opt_l}||p_i - m_l||^2. \tag{24}$$

Note that the left-hand side of (24) is not smaller than $\tilde{\delta}_{opt}^2$, since $\tilde{\delta}_{opt}^2$ is the optimal objective value of k-CMeans on $\tilde{P}$. For the right-hand side of (24), the first term $2\frac{1}{n}\sum_{i=1}^{n}||\tilde{p}_i - p_i||^2 = 2\omega$ (by the construction of $\tilde{P}$), and the second term $2\frac{1}{n}\sum_{l=1}^{k}\sum_{p_i \in Opt_l}||p_i - m_l||^2 = 2\delta_{opt}^2$. Thus, we have $\tilde{\delta}_{opt}^2 \leq 2\omega + 2\delta_{opt}^2$.

Next, we show the inequality $\delta_{opt}^2([\mathcal{C}]^k) \leq 2\omega + 8\tilde{\delta}_{opt}^2$. Consider k-CMeans clustering on $\tilde{P}$. Let $\{\tilde{m}_1, \ldots, \tilde{m}_k\}$ be the optimal constrained mean points of $\tilde{P}$, and $\{\tilde{O}_1, \ldots, \tilde{O}_k\}$ be the corresponding optimal clusters. Let $\{\tilde{c}_1, \ldots, \tilde{c}_k\}$ be the k-tuple in $[\mathcal{C}]^k$ with $\tilde{c}_l$ being the nearest point in $\mathcal{C}$ to $\tilde{m}_l$. Thus, by an argument similar to the one for (23), we have

$$||\tilde{p}_i - \tilde{c}_l||^2 \leq 2||\tilde{p}_i - \tilde{m}_l||^2 + 2||\tilde{m}_l - \tilde{c}_l||^2 \leq 4||\tilde{p}_i - \tilde{m}_l||^2. \tag{25}$$

for each $\tilde{p}_i \in \tilde{O}_l$. In (25), the last one follows from the facts that $\tilde{c}_l$ is the nearest point in $\mathcal{C}$ to $\tilde{m}_l$ and $\tilde{p}_i \in \mathcal{C}$, which implies that $||\tilde{m}_l - \tilde{c}_l|| \leq ||\tilde{m}_l - \tilde{p}_i||$ (see Fig. 5b).

Summing both sides of (25) over all the points in $\tilde{P}$, we have

$$\sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||\tilde{p}_i - \tilde{c}_l||^2 \le 4 \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||\tilde{p}_i - \tilde{m}_l||^2. \tag{26}$$

Now, consider the following clustering on P. For each p_i, if $\tilde{p}_i \in \tilde{O}_l$, we cluster it to the corresponding $\tilde{c}_l$.

Then the objective value of the clustering is

$$\frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||p_i - \tilde{c}_l||^2 \le \frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} (2||p_i - \tilde{p}_i||^2 + 2||\tilde{p}_i - \tilde{c}_l||^2)$$

$$\le 2\frac{1}{n} \sum_{i=1}^{n} ||p_i - \tilde{p}_i||^2 + 8\frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||\tilde{p}_i - \tilde{m}_l||^2. \tag{27}$$

The left-hand side of (27), $\frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||p_i - \tilde{c}_l||^2$, is no smaller than $\delta_{opt}^2([\mathcal{C}]^k)$ (by the definition), and the right-hand side of (27) is equal to $2\omega + 8\tilde{\delta}_{opt}^2$. Thus, we have $\delta_{opt}^2([\mathcal{C}]^k) \le 2\omega + 8\tilde{\delta}_{opt}^2$. □

Proof of Theorem 2 By the two inequalities in Lemma 8, we know that $\delta_{opt}^2([\mathcal{C}]^k) \le 18\omega + 16\tilde{\delta}_{opt}^2$. It is obvious that the optimal objective value of the k-means clustering is no larger than that of k-CMeans on the same set of input points P. This implies that $\omega \le \lambda\delta_{opt}^2$. Thus, we have

$$\delta_{opt}^2([\mathcal{C}]^k) \le (18\lambda + 16)\delta_{opt}^2. \tag{28}$$

So there exists one k-tuple in $[\mathcal{C}]^k$, which is able to induce an $(18\lambda + 16)$-approximation. □

4 Selection Algorithms for k-CMeans

As shown in Sect. 3, a set of k-tuple candidates for the mean points of k-CMeans can be obtained by our Peeling-and-Enclosing algorithm. To determine the best candidate, we need a selection algorithm to compute the clustering for each k-tuple candidate, and select the one with the smallest objective value. Clearly, the key to designing a selection algorithm is to solve the Partition step (i.e., generating the clustering) for each k-tuple candidate. We need to design a problem-specific algorithm for the Partition step, to satisfy the constraint of each individual problem.

We consider all the constrained k-means clustering problems which are mentioned in Sect. 1.1, except for the uncertain data clustering, since Cormode and McGregor [19] have showed that it can be reduced to an ordinary k-means clustering problem.
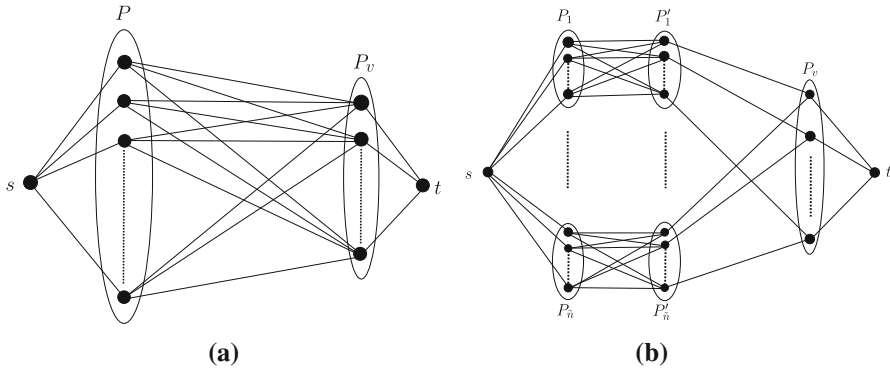
Fig. 6 Minimum cost circulations for *r*-gather clustering (**a**) and *l*-diversity clustering (**b**)

However, the *k*-median version of the uncertain data clustering does not have such a property. In Sect. 5.4, we will discuss how to obtain the $(1 + \epsilon)$-approximation by applying our Peeling-and-Enclosing framework.

4.1 *r*-Gather *k*-Means Clustering

Let P be a set of n points in $\mathbb{R}^d$. *r-Gather k-means clustering* (denoted by (r, k)-GMeans) on P is the problem of clustering P into k clusters with size at least r, such that the average squared Euclidean distance from each point in P to the mean point of its cluster is minimized [2].

To solve the Partition problem of (r, k)-GMeans, we adopt the following strategy. For each k-tuple candidate $P_v = \{p_{v_1}, \ldots p_{v_k}\}$ returned by Algorithm 1, build a complete bipartite graph G (see Fig. 6a): each vertex in the left column corresponds to a point in P, and each vertex in the right column represents a candidate mean point in P_v; for each pair of vertices in different partite sets, connect them by an edge with the weight equal to their squared Euclidean distance. We can solve the Partition problem by finding the minimum cost matching in G: each vertex in the left has the supply 1, and each vertex in the right has the demand r and capacity n. After adding a source node s connecting to all the verities in the left and a sink node t connecting to all the vertices in the right, we can reduce the Partition problem to a *minimum cost circulation* problem, and solve it by using the algorithm in [28]. Denote by V and E the sets of vertices and edges of G. The running time for solving the minimum cost circulation problem is $O(|E|^2 \log |V| + |E| \cdot |V| \log^2 |V|)$ [49]. In our case, $|E| = O(n)$ and $|V| = O(n)$ if k is a fixed constant. Also, the time complexity for building G is $O(nd)$. Thus, the total time for solving the Partition problem is $O\left(n\left(n(\log n)^2 + d\right)\right)$.[4] Together with the time complexity in Theorem 1, we have the following theorem.

---

[4] In our problem, an integral solution is necessary for generating the clusters on P. Actually, since the demands and capacities are all integers, any optimal solution of the minimum cost circulation problem can be transformed to an integral solution without loss of the quality in $O(|P_v| \cdot |E|) = O(n)$ time [22].

Theorem 3 *There exists an algorithm yielding a* $(1 + \epsilon)$*-approximation for* (r, k)*-GMeans with constant probability, in* $O\left(2^{poly(\frac{k}{\epsilon})}(\log n)^{k+1}n\left(n \log n + d\right)\right)$ *time.*

4.2 *r*-Capacity *k*-Means Clustering

r-Capacity k-means clustering (denoted by (r, k)-CaMeans) [41] on a set P of n points in $\mathbb{R}^d$ is the problem of clustering P into k clusters with size at most r, such that the average squared Euclidean distance from each point in P to the mean point of its cluster is minimized.

We can solve the Partition problem of (r, k)-CaMeans in a way similar to that of (r, k)-GMeans; the only difference is that the demand r is replaced by the capacity r.

Theorem 4 *There exists an algorithm yielding a* $(1 + \epsilon)$*-approximation for* (r, k)*-CaMeans with constant probability, in* $O\left(2^{poly(\frac{k}{\epsilon})}(\log n)^{k+1}n\left(n \log n + d\right)\right)$ *time.*

4.3 *l*-Diversity *k*-Means Clustering

Let $P = \bigcup_{i=1}^{\tilde{n}} P_i$ be a set of colored points in $\mathbb{R}^d$ and $\sum_{i=1}^{\tilde{n}} |P_i| = n$, where the points in each P_i share the same color. *l-Diversity k-means clustering* (denoted by (l, k)-DMeans) on P is the problem of clustering P into k clusters such that the points sharing the same color inside each cluster have a fraction no more than $\frac{1}{l}$ for some $l > 1$, and the average squared Euclidean distance from each point in P to the mean point of its cluster is minimized.

Similar to (r, k)-GMeans, we reduce the Partition problem of (l, k)-DMeans to a minimum cost circulation problem for each k-tuple candidate $P_v = \{p_{v_1}, \ldots p_{v_k}\}$. The challenge is that we do not know the size of each resulting cluster, and therefore it is difficult to control the flow on each edge if directly using the bipartite graph built in Fig. 6a. Instead, we add a set of "gates" between the input P and the k-tuple P_v (see Fig. 6b). First, following the definition of (l, k)-DMeans, we partition the "vertices" P into $\tilde{n}$ groups $\{P_1, \ldots, P_{\tilde{n}}\}$. For each P_i, we generate a new set of vertices (i.e., the gates) $P'_i = \{c^i_1, \ldots, c^i_k\}$, and connect each pair of $p \in P_i$ and $c^i_j \in P'_i$ by an edge with weight $||p - p_{v_j}||^2$. We also connect each pair of c^i_j and p_{v_j} by an edge with weight 0. In Fig. 6b, the size of vertices $|V| = n + k\tilde{n} + k + 2 = O(kn)$, and the size of edges $|E| = n + kn + k\tilde{n} + k = O(kn)$. Below we show that we can use c^i_j to control the flow from P_i to p_{v_j} by setting appropriate capacities and demands.

Let $t = \max_{1 \le i \le \tilde{n}} |P_i|$. We consider the value $\lfloor |Opt_j|/l \rfloor$ that is the upper bound on the number of points with the same color in Opt_j (recall Opt_j is the jth optimal cluster defined in Sect. 3). The upper bound $\lfloor |Opt_j|/l \rfloor$ can be either between 1 and t or larger than t. Clearly, if the upper bound is larger than t, there is no need to consider the upper bound anymore. Thus, we can enumerate all the $(t + 1)^k$ cases to guess the upper bound $\lfloor |Opt_j|/l \rfloor$ for $1 \le j \le k$. Let u_j be the guessed upper bound for Opt_j. If u_j is no more than t, then each c^i_j, $1 \le i \le \tilde{n}$, has the capacity u_j, and p_{v_j} has the demand $l \times u_j$ and capacity $l \times (u_j + 1) - 1$. Otherwise (i.e., $u_j > t$), set the capacity of each c^i_j, $1 \le i \le \tilde{n}$, to be n, and the demand and capacity of p_{v_j} to be

$l \times (t + 1)$ and n, respectively. By using the algorithm in [49], we solve the minimum cost circulation problem for each of the $(t + 1)^k$ guesses.

Theorem 5 *For any colored point set* $P = \bigcup_{i=1}^{\tilde{n}} P_i$ *in* $\mathbb{R}^d$ *with* $n = |P|$ *and* $t = \max_{1 \le i \le \tilde{n}} |P_i|$, *there exists an algorithm yielding, in* $O\left(2^{poly(\frac{k}{\epsilon})}(\log n)^{k+1}(t + 1)^k n(n \log n + d)\right)$ *time, a* $(1 + \epsilon)$-*approximation for* (l, k)-*DMeans with constant probability.*

Note We can solve the problem in [46] by slightly changing the above Partition algorithm. In [46], it requires that the size of each cluster is at least l and the points inside each cluster have distinct colors, which means that the upper bound u_j is always equal to 1 for each $1 \le j \le k$. Thus, there is no need to guess the upper bounds in our Partition algorithm. We can simply set the capacity for each c_j^i to be 1, and the demand for each p_{v_j} to be l. With this change, our algorithm yields a $(1 + \epsilon)$-approximation with constant probability in $O\left(2^{poly(\frac{k}{\epsilon})}(\log n)^{k+1} n(n \log n + d)\right)$ time.

4.4 Chromatic *k*-Means Clustering

Let $P = \bigcup_{i=1}^{\tilde{n}} P_i$ be a set of colored points in $\mathbb{R}^d$ and $\sum_{i=1}^{\tilde{n}} |P_i| = n$, where the points in each P_i share the same color. *Chromatic k-means clustering* (denoted by k-ChMeans) [7,23] on P is the problem of clustering P into k clusters such that no pair of points with the same color is clustered into the same cluster, and the average squared Euclidean distance from each point in P to the mean point of its cluster is minimized.

To satisfy the chromatic requirement, each P_i should have a size no more than k. Given a k-tuple candidate $P_v = \{p_{v_1}, \ldots, p_{v_k}\}$, we can consider the partition problem for each P_i independently, since there is no mutual constraint among them. It is easy to see that finding a partition of P_i is equivalent to computing a minimum cost one-to-one matching between P_i and P_v, where the cost of the matching between any $p \in P_i$ and $p_{v_j} \in P_v$ is their squared Euclidean distance. We can build this bipartite graph in $O(k^2 d)$ time, and solve this matching problem by using Hungarian algorithm in $O(k^3)$ time. Thus, the running time of the Partition step for each P_v is $O(k^2(k+d)n)$.

Theorem 6 *There exists an algorithm yielding a* $(1+\epsilon)$-*approximation for k-ChMeans with constant probability, in* $O\left(2^{poly(\frac{k}{\epsilon})}(\log n)^{k+1} nd\right)$ *time.*

4.5 Fault Tolerant *k*-Means Clustering

Fault Tolerant k-means clustering (denoted by (l, k)-FMeans) [54] on a set P of n points in $\mathbb{R}^d$ and a given integer $1 \le l \le k$ is the problem of finding k points $\mathcal{C} = \{c_1, \ldots, c_k\} \subset \mathbb{R}^d$, such that the average of the total squared distances from each point in P to its l nearest points in $\mathcal{C}$ is minimized.

To solve the Partition problem of (l, k)-FMeans, our idea is to reduce (l, k)-FMeans to k-ChMeans, and use the Partition algorithm for k-ChMeans to generate the desired

clusters. The reduction simply makes l monochromatic copies $\{p_i^1, \ldots, p_i^l\}$ for each $p_i \in P$. The following lemma shows the relation of the two problems.

Lemma 9 *For any constant $\lambda \geq 1$, a λ-approximation of (l, k)-FMeans on P is equivalent to a λ-approximation of k-ChMeans on $\bigcup_{i=1}^{n}\{p_i^1, \ldots, p_i^l\}$.*

Proof We build a bijection between the solutions of (l, k)-FMeans and k-ChMeans. First, we consider the mapping from (l, k)-FMeans to k-ChMeans. Let $\mathcal{C} = \{c_1, \ldots, c_k\}$ be the k mean points of (l, k)-FMeans, and $\{c_{i(1)}, \ldots, c_{i(l)}\} \subset \mathcal{C}$ be the l nearest mean points to each $p_i \in P$. If using $\mathcal{C}$ as the k mean points of k-ChMeans on $\bigcup_{i=1}^{n}\{p_i^1, \ldots, p_i^l\}$, the l copies $\{p_i^1, \ldots, p_i^l\}$ of p_i will be respectively clustered to the l clusters of $\{c_{i(1)}, \ldots, c_{i(l)}\}$ to minimize the cost.

Now consider the mapping from k-ChMeans to (l, k)-FMeans. Let $\mathcal{C}' = \{c_1', \ldots, c_k'\}$ be the k mean points of k-ChMeans. For each i, $\{c_{i(1)}', \ldots, c_{i(l)}'\}$ are the mean points of the l clusters that $\{p_i^1, \ldots, p_i^l\}$ are clustered to. It is easy to see that the l nearest mean points of p_i are $\{c_{i(1)}', \ldots, c_{i(l)}'\}$ if we use $\mathcal{C}'$ as the k mean points of (l, k)-FMeans.

With this bijection, we can pair up the solutions to the two problems. Clearly, each pair of solutions to (l, k)-FMeans and k-ChMeans formed by the bijection have the equal objective value. Further, their optimal objective values are equal to each other, and for any pair of solutions, their approximation ratios are the same. Thus, Lemma 9 is true. □

With Lemma 9, we immediately have the following theorem.

Theorem 7 *There exists an algorithm yielding a $(1 + \epsilon)$-approximation for (l, k)-FMeans with constant probability, in $O\left(2^{poly(\frac{k}{\epsilon})}(\log n)^{k+1}nd\right)$ time.*

Note As mentioned in [35], a more general version of fault tolerant clustering problem is to allow each point $p_i \in P$ to have an individual l-value l_i. From the above discussion, it is easy to see that this general version can also be solved in the same way (i.e., through reduction to k-ChMeans) and achieve the same approximation result.

4.6 Semi-supervised *k*-Means Clustering

As shown in Sect. 1.1, semi-supervised clustering has various forms. In this paper, we consider the semi-supervised k-means clustering problem which takes into account the geometric cost and priori knowledge. Let P be a set of n points in $\mathbb{R}^d$, and $\overline{\mathcal{S}} = \{\overline{S}_1, \ldots, \overline{S}_k\}$ be a given clustering of P. *Semi-supervised k-means clustering* (denoted by k-SMeans) on P and $\overline{\mathcal{S}}$ is the problem of finding a clustering $\mathcal{S} = \{S_1, \ldots, S_k\}$ of P such that the following objective function is minimized,

$$\alpha \frac{Cost(\mathcal{S})}{E_1} + (1 - \alpha)\frac{dist\{\mathcal{S}, \overline{\mathcal{S}}\}}{E_2}, \tag{29}$$

where $\alpha \in [0, 1]$ is a given constant, E_1 and E_2 are two given scalars to normalize the two terms, $Cost(\mathcal{S})$ is the k-means clustering cost of $\mathcal{S}$, and $dist\{\mathcal{S}, \overline{\mathcal{S}}\}$ is the

distance between $\mathcal{S}$ and $\overline{\mathcal{S}}$ defined in the same way as in [11]. For any pair of S_j and $\overline{S}_i$, $1 \leq j, i \leq k$, their difference is $|S_j \backslash \overline{S}_i|$. Given a bipartite matching σ between $\mathcal{S}$ and $\overline{\mathcal{S}}$, $dist\{\mathcal{S}, \overline{\mathcal{S}}\}$ is defined as $\sum_{j=1}^{k} |S_j \backslash \overline{S}_{\sigma(j)}|$.

The challenge is that the bipartite matching σ is unknown in advance. We fix the k-tuple candidate $P_v = \{p_{v_1}, \ldots p_{v_k}\}$. To find the desired σ to minimize the objective function (29), we build a bipartite graph, where the left (resp., right) column contains k vertices corresponding to $p_{v_1}, \ldots, p_{v_k}$ (resp., $\overline{S}_1, \ldots, \overline{S}_k$), respectively. For each pair $(p_{v_j}, \overline{S}_i)$, we connect them by an edge; we calculate the edge weight $w_{(i,j)}$ in the following way. For each $p \in \overline{S}_i$, it could be potentially assigned to any of the k clusters in $\mathcal{S}$; if $i = \sigma(j)$, the induced k costs of p will be $\{c_p^1, c_p^2, \ldots, c_p^k\}$, where $c_p^l = \alpha \frac{||p - p_{v_l}||^2}{E_1}$ if $l = j$, or $c_p^l = \alpha \frac{||p - p_{v_l}||^2}{E_1} + (1 - \alpha) \frac{1}{E_2}$ otherwise. Thus, we set

$$w_{(i,j)} = \sum_{p \in \overline{S}_i} \min_{1 \leq l \leq k} c_p^l. \tag{30}$$

We solve the minimum cost bipartite matching problem to determine σ. To build such a bipartite graph, we need to first compute all the kn distances from the points in P to the k-tuple P_v; then, we calculate the k^2 edge weights via (30). The bipartite graph can be built in a total of $O(knd + k^2 n)$ time, and the optimal matching can be obtained via Hungarian algorithm in $O(k^3)$ time.

Theorem 8 *There exists an algorithm yielding a $(1 + \epsilon)$-approximation for k-SMeans with constant probability, in $O\left(2^{poly(\frac{k}{\epsilon})} (\log n)^{k+1} nd\right)$ time.*

5 Constrained k-Median Clustering (k-C Median)

In this section, we extend our approach for k-CMeans to the constrained k-median clustering problem (k-CMedian). Similar to k-CMeans, we show that the Peeling-and-Enclosing framework can be used to construct a set of candidates for the constrained median points. Combining this with the selection algorithms (with trivial modification) in Sect. 4, we achieve the $(1 + \epsilon)$ approximations for a class of k-CMedian problems.

To solve k-C Median, a straightforward idea is to extend the simplex lemma to median points and combine it with the Peeling-and-Enclosing framework to achieve an approximate solution. However, due to the essential difference between mean and median points, such an extension for the simplex lemma is not always possible. The main reason is that the median point (i.e., Fermat point) does not necessarily lie inside the simplex, and thus there is no guarantee to find the median point by searching inside the simplex. Below is an example showing that the median point actually can lie outside the simplex.

Let $P = \{p_1, p_2, \ldots, p_9\}$ be a set of points in $\mathbb{R}^d$. We consider the following partition of P, $P_1 = \{p_i \mid 1 \leq i \leq 5\}$ and $P_2 = \{p_i \mid 6 \leq i \leq 9\}$. Assume that all the points of P locate at the three vertices of a triangle $\triangle abc$. Particularly, $\{p_1, p_2, p_6\}$ coincide with vertex a, $\{p_3, p_4, p_5\}$ with vertex b, and $\{p_7, p_8, p_9\}$ with vertex c (see Fig. 7). It is easy to see that the median points of P_1 and P_2 are b and c, respectively.
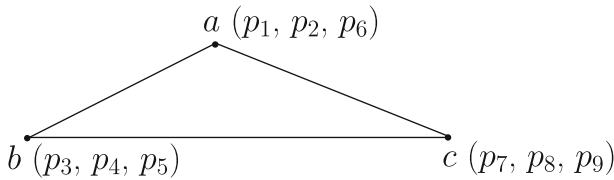
Fig. 7 An example showing non-existence of a simplex lemma for k-CMedian

If the angle $\angle bac \geq \frac{2\pi}{3}$, the median point of P is vertex a (note that the median point can be viewed as the Fermat point of Δabc with each vertex associated with weight 3). This means that the median point of P is outside the simplex formed by the median points of P_1 and P_2 (i.e., segment $\overline{bc}$). Thus, a good approximation of the median point cannot be obtained by searching a grid inside $\overline{bc}$.

To overcome this difficulty, we show that a weaker version of the simplex lemma exists for median, which enables us to achieve similar results for k-CMedian.

5.1 Weaker Simplex Lemma for Median Point

Comparing to the simplex lemma in Sect. 2, the following Lemma 10 has two differences. One is that the lemma requires a partial partition on a significantly large subset of P, rather than a complete partition on P. Secondly, the grid is built in the flat spanned by $\{o_1, \ldots, o_j\}$, instead of the simplex. Later, we will show that the grid is actually built in a surrounding region of the simplex, and thus the lemma is called "*weaker simplex lemma*".

Lemma 10 (Weaker Simplex Lemma) *Let P be a set of n points in $\mathbb{R}^d$, and $\bigcup_{l=1}^{j} P_l \subset P$ be a partial partition of P with $P_{l_1} \cap P_{l_2} = \emptyset$ for any $l_1 \neq l_2$. Let o_l be the median point of P_l for $1 \leq l \leq j$, and $\mathcal{F}$ be the flat spanned by $\{o_1, \ldots, o_j\}$. If $|P \backslash (\bigcup_{l=1}^{j} P_l)| \leq \epsilon |P|$ for some constant $\epsilon \in (0, 1/5)$ and each P_l is contained inside a ball $\mathcal{B}(o_l, L)$ centered at o_l and with radius $L \geq 0$, then it is possible to build a grid in $\mathcal{F}$ with size $O(j^2(\frac{j\sqrt{j}}{\epsilon})^j)$ such that at least one grid point τ satisfies the following inequality, where o is the median point of P (see Fig. 8).*

$$\frac{1}{|P|} \sum_{p \in P} ||\tau - p|| \leq \left(1 + \frac{9}{4}\epsilon\right) \frac{1}{|P|} \sum_{p \in P} ||p - o|| + (1 + \epsilon)L. \tag{31}$$

Proof Synopsis To prove Lemma 10, we let $\tilde{o}$ be the orthogonal projection of o to $\mathcal{F}$ (see Fig. 8). In Claim 4, we show that the distance between o and $\tilde{o}$ is bounded, and consequently, the induced cost of $\tilde{o}$, i.e., $\frac{1}{|P|} \sum_{p \in P} ||p - \tilde{o}||$, is also bounded according to Claim 5. Thus, $\tilde{o}$ is a good approximation of o, and we can focus on building a grid inside $\mathcal{F}$ to approximate $\tilde{o}$. Since $\mathcal{F}$ is unbounded, we need to determine a range for the grid. Claim 6 resolves the issue. It considers two cases. One is that there are at least two subsets in the partial partition, $\{P_1, \ldots, P_j\}$, having large enough fractions of P; the other is that only one subset is large enough. For either case, Claim 6 shows that
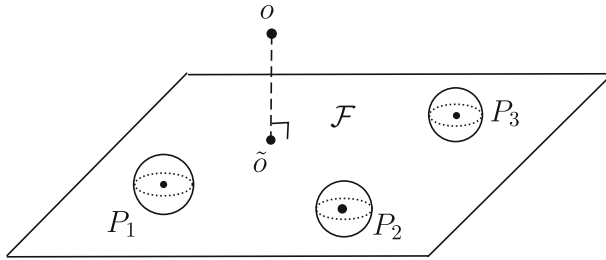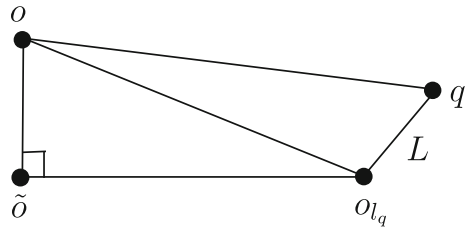
Fig. 8 An illustration for Lemma 10

Fig. 9 An illustration for
Claim 4

we can determine the range of the grid using the location information of $\{o_1, \ldots, o_j\}$. Finally, we can obtain the desired grid point τ in the following way: draw a set of balls centered at $\{o_1, \ldots, o_j\}$ with proper radii; build the grids inside each of the balls, and find the desired grid point τ in one of these balls. Note that since all the balls are inside $\mathcal{F}$, the complexity of the union of the grids is independent of the dimensionality d.

Claim 4

$$||o - \tilde{o}|| \leq L + \frac{1}{1 - \epsilon} \frac{1}{|P|} \sum_{p \in P} ||o - p||. \tag{32}$$

Proof Lemma 10 assumes that $\bigcup_{l=1}^{j} P_l \geq (1 - \epsilon)|P|$. By Markov's inequality, we know that there exists one point $q \in \bigcup_{l=1}^{j} P_l$ such that
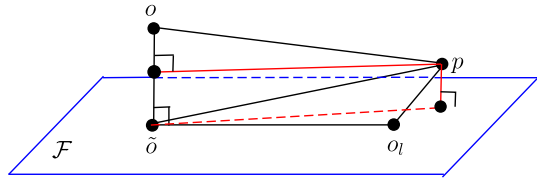
$$||q - o|| \leq \frac{1}{1 - \epsilon} \frac{1}{|P|} \sum_{p \in P} ||o - p||. \tag{33}$$

Let P_{l_q} be the subset containing q. Then from (33), we immediately have

$$
\begin{aligned}
||o - \tilde{o}|| &\leq ||o_{l_q} - o|| \\
&\leq ||o_{l_q} - q|| + ||q - o|| \\
&\leq L + \frac{1}{1 - \epsilon} \frac{1}{|P|} \sum_{p \in P} ||o - p||.
\end{aligned}
\tag{34}
$$

This implies Claim 4 (see Fig. 9). $\qquad\qquad\square$

Fig. 10 An illustration for
Claim 5

Claim 5

$$\frac{1}{|P|} \sum_{p \in P} ||p - \tilde{o}|| \leq \frac{1}{1 - \epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o|| + L. \tag{35}$$

Proof For any point $p \in P_l$, let $dist\{\overline{o\tilde{o}}, p\}$ (resp., $dist\{\overline{\mathcal{F}}, p\}$) denote its distance to the line $\overline{o\tilde{o}}$ (resp., flat $\mathcal{F}$). See Fig. 10. Then we have

$$||p - \tilde{o}|| = \sqrt{dist^2\{\overline{o\tilde{o}}, p\} + dist^2\{\mathcal{F}, p\}}, \tag{36}$$

$$||p - o|| \geq dist\{\overline{o\tilde{o}}, p\}. \tag{37}$$

Combining (36) and (37), we have

$$
\begin{aligned}
||p - \tilde{o}|| - ||p - o|| &\leq \sqrt{dist^2\{\overline{o\tilde{o}}, p\} + dist^2\{\mathcal{F}, p\}} - dist\{\overline{o\tilde{o}}, p\} \\
&\leq dist\{\mathcal{F}, p\} \\
&\leq ||p - o_l|| \leq L.
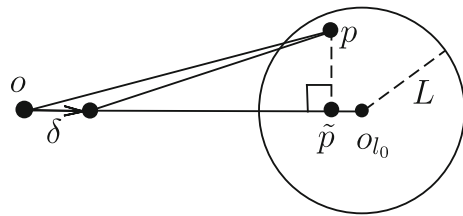\end{aligned} \tag{38}
$$

For any point $p \in P \backslash (\bigcup_{l=1}^{j} P_l)$, we have

$$||p - \tilde{o}|| \leq ||p - o|| + ||o - \tilde{o}||. \tag{39}$$

Combining (38), (39), and (32), we have

$$
\begin{aligned}
\frac{1}{|P|} \sum_{p \in P} ||p - \tilde{o}|| &= \frac{1}{|P|} \left(\sum_{p \in \bigcup_{l=1}^{j} P_l} ||p - \tilde{o}|| + \sum_{p \in P \backslash \left(\bigcup_{l=1}^{j} P_l \right)} ||p - \tilde{o}|| \right) \\
&\leq \frac{1}{|P|} \left(\sum_{p \in \bigcup_{l=1}^{j} P_l} (L + ||p - o||) + \sum_{p \in P \backslash \left(\bigcup_{l=1}^{j} P_l \right)} (||p - o|| + ||o - \tilde{o}||) \right) \\
&\leq (1 - \epsilon)L + \frac{1}{|P|} \sum_{p \in P} ||p - o|| + \epsilon L + \frac{\epsilon}{1 - \epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o|| \\
&= \frac{1}{1 - \epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o|| + L.
\end{aligned} \tag{40}
$$

Fig. 11 An illustration for
Claim 6

Thus the claim is true.   □

Claim 6 *At least one of the following two statements is true.*

1. *There exist at least two points in $\{o_1, \ldots, o_j\}$ whose distances to $\tilde{o}$ are no more than $L + \frac{3j}{1-\epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o||$.*
2. *There exists one point in $\{o_1, \ldots, o_j\}$, say o_{l_0}, whose distance to $\tilde{o}$ is no more than $(1 + \frac{1+2\epsilon}{\sqrt{3-12\epsilon}})L.$*[5]

Proof We consider two cases: (i) there are two subsets P_{l_1} and P_{l_2} of P with size at least $\frac{1-\epsilon}{3j}|P|$, and (ii) no such pair of subsets exists.

For case (i), by Markov's inequality, we know that there exist two points $q \in P_{l_1}$ and $q' \in P_{l_2}$ such that

$$||q - o|| \le \frac{3j}{1-\epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o||; \tag{41}$$

$$||q' - o|| \le \frac{3j}{1-\epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o||. \tag{42}$$

This, together with triangle inequality, indicates that both $||o_{l_1} - o||$ and $||o_{l_2} - o||$ are no more than $L + \frac{3j}{1-\epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o||$. Since $\tilde{o}$ is the orthogonal projection of o to $\mathcal{F}$, we have $||o_{l_1} - \tilde{o}|| \le ||o_{l_1} - o||$ and $||o_{l_2} - \tilde{o}|| \le ||o_{l_2} - o||$. Thus, the first statement is true in this case.

For case (ii), i.e., no two subsets with size at least $\frac{1-\epsilon}{3j}|P|$, since $\sum_{l=1}^{j} |P_l| \ge (1-\epsilon)|P|$, by pigeonhole principle we know that there must exist one P_{l_0}, $1 \le l_0 \le j$, with size

$$|P_{l_0}| \ge \left(1 - (j-1)\frac{1}{3j}\right)(1-\epsilon)|P| \ge \frac{2}{3}(1-\epsilon)|P|. \tag{43}$$

Let $x = ||o - o_{l_0}||$. We assume that $x > L$, since otherwise the second statement is automatically true.

Now imagine moving o slightly toward o_{l_0} by a small distance δ. See Fig. 11. For any point $p \in P_{l_0}$, let $\tilde{p}$ be its orthogonal projection to the line $\overline{oo_{l_0}}$, and a and b be

---

[5] Note that we assume $\epsilon < \frac{1}{5}$ in Lemma 10, so $(1 + \frac{1+2\epsilon}{\sqrt{3-12\epsilon}})L$ is a finite real number.

the distances $||o - \tilde{p}||$ and $||p - \tilde{p}||$, respectively. Then, the distance between p and o is decreased by $\sqrt{a^2 + b^2} - \sqrt{(a - \delta)^2 + b^2}$. Also, we have

$$\lim_{\delta \to 0} \frac{\sqrt{a^2 + b^2} - \sqrt{(a - \delta)^2 + b^2}}{\delta} = \lim_{\delta \to 0} \frac{2a - \delta}{\sqrt{a^2 + b^2} + \sqrt{(a - \delta)^2 + b^2}}$$

$$= \frac{(a/b)}{\sqrt{(a/b)^2 + 1}}. \tag{44}$$

Since p is inside ball $\mathcal{B}(o_{l_0}, L)$, we have $a/b \geq (x - L)/L$. For any point $p \in P \setminus P_{l_0}$, the distance to o is non-increased or increased by at most δ. Thus, the average distance from the points in P to o is decreased by at least

$$\frac{2}{3}(1 - \epsilon) \frac{((x - L)/L)\delta}{\sqrt{((x - L)/L)^2 + 1}} - \left(1 - \frac{2}{3}(1 - \epsilon)\right)\delta. \tag{45}$$

Since the original position of o is the median point of P, the value of (45) should be non-positive. With simple calculation, we have

$$(x - L)/L \leq \frac{1 + 2\epsilon}{\sqrt{3 - 12\epsilon}} \implies x \leq \left(1 + \frac{1 + 2\epsilon}{\sqrt{3 - 12\epsilon}}\right)L. \tag{46}$$

By the same argument in case (i), we know that $||o_{l_0} - \tilde{o}|| \leq ||o_{l_0} - o||$. This, together with (46), implies that the second statement is true for case (ii). This completes the proof for this claim. □

With the above claims, we now prove Lemma 10.

Proof of Lemma 10 We build a grid in $\mathcal{F}$ as follows. First, draw a set of balls.

– For each o_l, $1 \leq l \leq j$, draw a ball (called *type-1 ball*) centered at o_l and with radius $(1 + \frac{1 + 2\epsilon}{\sqrt{3 - 12\epsilon}})L$.
– For each pair of o_l and $o_{l'}$, $1 \leq l, l' \leq j$, draw a ball (called *type-2 ball*) centered at o_l and with radius $(1 + \frac{1 + 2\epsilon}{\sqrt{3 - 12\epsilon}})(||o_l - o_{l'}|| + L)$.

We claim that among the above balls, there must exist one ball that contains $\tilde{o}$. If there is only one subset in $\{P_1, \ldots, P_j\}$ with size no smaller than $\frac{1 - \epsilon}{3j}|P|$, it corresponds to the second case in Claim 6, and thus there exists a type-1 ball containing $\tilde{o}$. Now consider the case that there are multiple subsets, say $\{P_{l_1}, \ldots, P_{l_t}\}$ for some $t \geq 2$, all with size no smaller than $\frac{1 - \epsilon}{3j}|P|$. Without loss of generality, assume that $||o_{l_1} - o_{l_2}|| = \max\{||o_{l_1} - o_{l_s}|| \mid 1 \leq s \leq t\}$. Then, we can view $\bigcup_{s=1}^{t} P_{l_s}$ as a big subset of P bounded by a ball centered at o_{l_1} and with radius $||o_{l_1} - o_{l_2}|| + L$. By the same argument given in the proof of Claim 6 for (43), we know that $|\bigcup_{s=1}^{t} P_{l_s}| \geq \frac{2}{3}(1 - \epsilon)|P|$. This also means that we can reduce this case to the second case in Claim 6, i.e., replacing P_{l_0}, o_{l_0} and L by $|\bigcup_{s=1}^{t} P_{l_s}|$, o_{l_1} and $||o_{l_1} - o_{l_2}|| + L$ respectively. Thus, there is a type-2 ball containing $\tilde{o}$.

Next, we discuss how to build the grids inside these balls. For type-1 balls with radius $(1 + \frac{1+2\epsilon}{\sqrt{3-12\epsilon}})L$, we build the grids inside them with grid length $\frac{\epsilon}{\sqrt{j}}L$. For type-2 balls with radius $r_{l,l'} = (1 + \frac{1+2\epsilon}{\sqrt{3-12\epsilon}})(||o_l - o_{l'}|| + L)$ for some l and l', we build the grids inside them with grid length

$$\frac{1}{1 + \frac{1+2\epsilon}{\sqrt{3-12\epsilon}}} \frac{(1-\epsilon)\epsilon}{6j\sqrt{j}} r_{l,l'}. \tag{47}$$

If $\tilde{o}$ is contained in a type-1 ball, then there exists one grid point τ whose distance to $\tilde{o}$ is no more than ϵL. If $\tilde{o}$ is contained in a type-2 ball, such a distance is no more than

$$\frac{(1-\epsilon)\epsilon}{6j}(||o_l - o_{l'}|| + L) \tag{48}$$

by (47). By the first statement in Claim 6 and triangle inequality, we know that

$$||o_l - o_{l'}|| \le ||o_l - \tilde{o}|| + ||\tilde{o} - o_{l'}|| \le 2\left(L + \frac{3j}{1-\epsilon} \frac{1}{|P|} \sum_{p \in P} ||p - o||\right). \tag{49}$$

Equations (48) and (49) imply that there exists one grid point τ whose distance to $\tilde{o}$ is no more than

$$\epsilon \frac{1}{|P|} \sum_{p \in P} ||p - o|| + \frac{(1-\epsilon)\epsilon}{2j}L \le \epsilon \frac{1}{|P|} \sum_{p \in P} ||p - o|| + \epsilon L. \tag{50}$$

Thus in both types of ball-containing, by triangle inequality and Claim 5, we have

$$\frac{1}{|P|} \sum_{p \in P} ||p - \tau|| \le \frac{1}{|P|} \sum_{p \in P} (||p - \tilde{o}|| + ||\tilde{o} - \tau||)$$

$$\le \left(\frac{1}{1-\epsilon} + \epsilon\right) \frac{1}{|P|} \sum_{p \in P} ||p - o|| + (1+\epsilon)L$$

$$\le \left(1 + \frac{9}{4}\epsilon\right) \frac{1}{|P|} \sum_{p \in P} ||p - o|| + (1+\epsilon)L, \tag{51}$$

where the second inequality follows from the assumption that $\epsilon \le \frac{1}{5}$.

As for the grid size, since we build the grids inside the balls in the $(j-1)$-dimensional flat $\mathcal{F}$, through simple calculation, we know that the grid size is $O(j^2(\frac{j\sqrt{j}}{\epsilon})^j)$. This completes the proof. □

Fig. 12 The gray area is U
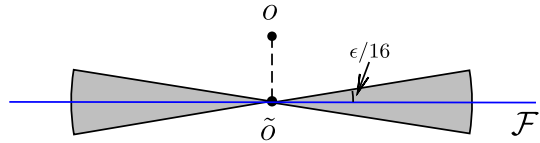
5.2 Peeling-and-Enclosing Algorithm for *k*-CMedian Using Weaker Simplex Lemma

In this section, we present a unified *Peeling-and-Enclosing* algorithm for generating a set of candidate median points for k-CMedian. Similar to the algorithm for k-CMeans, our algorithm iteratively determines the k median points. At each iteration, it uses a set of peeling spheres and a simplex to search for an approximate median point. Since the simplex lemma no longer holds for k-CMedian, we use the weaker simplex lemma as a replacement. Thus a number of changes are needed to accommodate the differences.

Before presenting our algorithm, we first introduce the following lemma proved by Badŏiu et al. in [10] for finding an approximate median point of a given point set.

Theorem 9 [10] *Let P be a normalized set of n points in $\mathbb{R}^d$ space, $1 > \epsilon > 0$, and R be a random sample of $O(1/\epsilon^3 \log 1/\epsilon)$ points from P. Then one can compute, in $O(d2^{O(1/\epsilon^4)} \log n)$ time, a point set $S(P, R)$ of cardinality $O(2^{O(1/\epsilon^4)} \log n)$, such that with constant probability (over the choices of R), there is a point $\tau \in S(P, R)$ such that $\sum_{p \in P} ||\tau - p|| \leq (1 + \epsilon) \sum_{p \in P} ||o - p||$, where o is the optimal median point of P.*

Sketch of the proof of Theorem 9. Since our algorithm uses some ideas in Theorem 9, we sketch its proof for completeness. First, by Markov's inequality, we know that there exists one point, say s_1, from R whose distance to o is no more than $2\frac{1}{|P|} \sum_{p \in P} ||o - p||$ with certain probability. Then the sampling procedure can be viewed as an incremental process starting with s_1; a flat $\mathcal{F}$ spanned by all previously obtained sample points is maintained; at each time that a new sample point is added, $\mathcal{F}$ is updated. Let $\tilde{o}$ be the projection of o on $\mathcal{F}$, and

$$U = \left\{ p \in \mathbb{R}^d \mid \frac{\pi}{2} - \frac{\epsilon}{16} \leq \angle o\tilde{o}p \leq \frac{\pi}{2} + \frac{\epsilon}{16} \right\}. \tag{52}$$

See Fig. 12. It has been shown that this incremental sampling process stops before at most $O(1/\epsilon^3 \log 1/\epsilon)$ points are taken, and one of the following two events happens with constant probability: (1) $\mathcal{F}$ is close enough to o, and (2) $|P \backslash U|$ is small enough. For either event, a grid can be built inside $\mathcal{F}$, and one of the grid points τ is the desired approximate median point.

Below we give an overview of our Peeling-and-Enclosing algorithm for k-CMedian. Let $P = \{p_1, \ldots, p_n\}$ be the set of $\mathbb{R}^d$ points in k-CMedian, and $\mathcal{OPT} = \{Opt_1, \ldots, Opt_k\}$ be the k (unknown) optimal clusters with m_j being the median point of cluster Opt_j for $1 \leq j \leq k$. Without loss of generality, we assume that $|Opt_1| \geq |Opt_2| \geq \cdots \geq |Opt_k|$. Denote by μ_{opt} the optimal objective value, i.e., $\mu_{opt} = \frac{1}{n} \sum_{j=1}^{k} \sum_{p \in Opt_j} ||p - m_j||$.

Algorithm overview We mainly focus on the differences with the k-CMeans algorithm. First, our algorithm uses Theorem 9 (instead of Lemma 4) to find an approximation p_{v_1} for m_1. Then, it iteratively finds the approximate median points for $\{m_2, \ldots, m_k\}$ using the Peeling-and-Enclosing strategy. At the $(j + 1)$th iteration, it has already obtained the approximate median points $p_{v_1}, \ldots, p_{v_j}$ for clusters $Opt_1, \ldots, Opt_j$, respectively. To find the approximate median point $p_{v_{j+1}}$ for Opt_{j+1}, the algorithm draws j peeling spheres $B_{j+1,1}, \ldots, B_{j+1,j}$ centered at $\{p_{v_1}, \ldots, p_{v_j}\}$, respectively, and considers the size of $\mathcal{A} = Opt_{j+1} \backslash (\bigcup_{l=1}^{j} B_{j+1,l})$. If $|\mathcal{A}|$ is small, it builds a flat (instead of a simplex) spanned by $\{p_{v_1}, \ldots, p_{v_j}\}$, and finds $p_{v_{j+1}}$ by using the weaker simplex lemma where the j peeling spheres can be viewed as a partial partition on Opt_{j+1}. If $|\mathcal{A}|$ is large, it adopts a strategy similar to the one in Theorem 9 to find $p_{v_{j+1}}$: start with the flat $\mathcal{F}$ spanned by $\{p_{v_1}, \ldots, p_{v_j}\}$, and grow $\mathcal{F}$ by repeatedly adding a sample point in $\mathcal{A}$ to it. As it will be shown in Lemma 11, $\mathcal{F}$ will become close enough to m_{j+1}, and $p_{v_{j+1}}$ can be obtained by searching a grid (built in a way similar to Lemma 10) in $\mathcal{F}$. By choosing a proper value (i.e., $O(\epsilon)\mu_{opt}$) for L in Lemmas 10 and 11, we can achieve the desired $(1 + \epsilon)$-approximation.

As for the running time, although Theorem 9 introduces an extra factor of $\log n$ for estimating the optimal cost of each Opt_{j+1}, our algorithm actually does not need it as such estimations have already been obtained during the Peeling-and-Enclosing step (see Claim 2 in the proof of Lemma 6). Thus, the running time is still $O(n(\log n)^{k+1}d)$, which is the same as k-CMeans.

The algorithm is shown in Algorithm 3. The following lemma is needed to ensure the correctness of our algorithm.

Lemma 11 *Let $\mathcal{F}$ be a flat in $\mathbb{R}^d$ containing $\{p_{v_1}, \ldots, p_{v_j}\}$ and having a distance to m_{j+1} no more than $\frac{2}{|Opt_{j+1}|} \sum_{p \in Opt_{j+1}} ||p - m_{j+1}||$. Assume that all the peeling spheres $B_{j+1,1}, \ldots, B_{j+1,j}$ are centered at $\{p_{v_1}, \ldots, p_{v_j}\}$, respectively, and have a radius $L \geq 0$. Then if $|Opt_{j+1} \backslash ((\bigcup_{w=1}^{j} B_{j+1,w}) \bigcup U)| \leq \epsilon |Opt_{j+1}|$, we have*

$$\frac{1}{|Opt_{j+1}|} \sum_{p \in Opt_{j+1}} ||p - \tilde{m}_{j+1}||$$

$$\leq (1 + 2\epsilon)\frac{1}{|Opt_{j+1}|} \sum_{p \in Opt_{j+1}} ||p - m_{j+1}|| + L \qquad (53)$$

for any $0 \leq \epsilon \leq 1$, where $\tilde{m}_{j+1}$ is the projection of m_{j+1} on $\mathcal{F}$ and U is defined in (52) (after replacing o and $\tilde{o}$ by m_{j+1} and $\tilde{m}_{j+1}$, respectively).

Proof To prove this lemma, we first compare it with Lemma 10. The main difference is that there is an extra part $U \cap Opt_{j+1}$ in Opt_{j+1}, where Opt_{j+1} can be viewed as the point set P in Lemma 10. Thus, Opt_{j+1} can be viewed as having three subsets, $(\bigcup_{w=1}^{j} B_{j+1,w}) \bigcap Opt_{j+1}$, $U \bigcap Opt_{j+1}$ and $Opt_{j+1} \backslash ((\bigcup_{w=1}^{j} B_{j+1,w}) \bigcup U)$.

For each point $p \in (\bigcup_{w=1}^{j} B_{j+1,w}) \bigcap Opt_{j+1}$, similar to (38) in Claim 5, we know that the cost increases by at most L if the median point moves from m_{j+1} to $\tilde{m}_{j+1}$.

Thus we have

$$\sum_{p \in Opt_{j+1} \cap \left(\bigcup_{w=1}^{j} B_{j+1,w} \right)} ||p - \tilde{m}_{j+1}||$$

$$\leq \sum_{p \in Opt_{j+1} \cap \left(\bigcup_{w=1}^{j} B_{j+1,w} \right)} (||p - m_{j+1}|| + L). \qquad (54)$$

For the part $Opt_{j+1} \backslash ((\bigcup_{w=1}^{j} B_{j+1,w}) \bigcup U)$, by triangle inequality we have

$$\sum_{p \in Opt_{j+1} \backslash \left(\left(\bigcup_{w=1}^{j} B_{j+1,w} \right) \bigcup U \right)} ||p - \tilde{m}_{j+1}||$$

$$\leq \sum_{p \in Opt_{j+1} \backslash \left(\left(\bigcup_{w=1}^{j} B_{j+1,w} \right) \bigcup U \right)} (||p - m_{j+1}|| + ||m_{j+1} - \tilde{m}_{j+1}||)$$

$$\leq \sum_{p \in Opt_{j+1} \backslash \left(\left(\bigcup_{w=1}^{j} B_{j+1,w} \right) \bigcup U \right)} ||p - m_{j+1}|| + 2\epsilon \sum_{p \in Opt_{j+1}} ||p - m_{j+1}||, (55)$$

where the second inequality follows from the assumption that $\mathcal{F}$'s distance to m_{j+1} is no more than $\frac{2}{|Opt_{j+1}|} \sum_{p \in Opt_{j+1}} ||p - m_{j+1}||$ and

$$|Opt_{j+1} \backslash \left(\left(\bigcup_{w=1}^{j} B_{j+1,w} \right) \bigcup U \right)| \leq \epsilon |Opt_{j+1}|.$$

For each point $p \in Opt_{j+1} \cap U$, recall that the angle $\angle m_{j+1} \tilde{m}_{j+1} p \in [\frac{\pi}{2} - \frac{\epsilon}{16}, \frac{\pi}{2} + \frac{\epsilon}{16}]$ in (52). In Theorem 3.2 of [10], it showed that $||p - \tilde{m}_{j+1}|| \leq (1+\epsilon)||p - m_{j+1}||$. Therefore,

$$\sum_{p \in Opt_{j+1} \cap U} ||p - \tilde{m}_{j+1}|| \leq (1 + \epsilon) \sum_{p \in Opt_{j+1} \cap U} ||p - m_{j+1}||. \qquad (56)$$

Combining (54), (55) and (56), we obtain (53). □

To complete the Peeling-and-Enclosing algorithm for k-CMedian, we also need an upper bound for the optimal objective value. In Sect. 5.3, we will show how to obtain such an estimation. For this moment, we assume that the upper bound is available.

Using the same idea for proving Theorem 1, we obtain the following theorem for k-CMedian.

Theorem 10 *Let P be a set of n points in $\mathbb{R}^d$ and $k \in \mathbb{Z}^+$ be a fixed constant. Given a constant $\epsilon \in (0, \frac{1}{4k^2})$, in $O(2^{poly(\frac{k}{\epsilon})} n (\log n)^{k+1} d)$ time, Algorithm 3 outputs $O(2^{poly(\frac{k}{\epsilon})} (\log n)^k)$ k-tuple candidate median points. With constant probability,*

Algorithm 3 Peeling-and-Enclosing for k-CMedian

Input: $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$, $k \geq 2$, a constant $\epsilon \in (0, \frac{1}{4k^2})$, and an upper bound $\Delta \in [\mu_{opt}, c\mu_{opt}]$ with $c \geq 1$.
Output: A set of k-tuple candidates for the k constrained median points.

1. For $i = 0$ to $\lceil \log_{1+\epsilon} c \rceil$ do

   (a) Set $\mu = (1 + \epsilon)^i \Delta/c$, and run Algorithm 4.
   (b) Let $\mathcal{T}_i$ be the output tree.

2. For each root-to-leaf path of every $\mathcal{T}_i$, build a k-tuple candidate using the k points associated with the path.

Algorithm 4 Peeling-and-Enclosing-Tree II

Input: $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$, $k \geq 2$, a constant $\epsilon \in (0, \frac{1}{4k^2})$, and $\mu > 0$.

1. Initialize $\mathcal{T}$ with a single root node v associated with no point.
2. Recursively grow each node v in the following way

   (a) If the height of v is already k, then it is a leaf.
   (b) Otherwise, let j be the height of v. Build the radius candidates set $\mathcal{R} = \cup_{t=0}^{\log n} \{\frac{1+l\frac{\epsilon}{2}}{2(1+\epsilon)} j 2^t \epsilon \mu \mid 0 \leq l \leq 4 + \frac{2}{\epsilon}\}$. For each $r \in \mathcal{R}$, do

      i. Let $\{p_{v_1}, \ldots, p_{v_j}\}$ be the j points associated with nodes on the root-to-v path.
      ii. For each p_{v_l}, $1 \leq l \leq j$, construct a ball $B_{j+1,l}$ centered at p_{v_l} and with radius r.
      iii. Compute a flat spanned by $\{p_{v_1}, \ldots, p_{v_j}\}$, and build a grid inside it by Lemma 10.
      iv. Take a random sample from $P \setminus \cup_{l=1}^{j} B_{j+1,l}$ with size $s = \frac{k^3}{\epsilon^{11}} \ln \frac{k^2}{\epsilon^6}$, and compute the flat determined by these sample points and $\{p_{v_1}, \ldots, p_{v_j}\}$. Build a grid inside the flat by Theorem 9.
      v. In total, there are $O(2^{poly(\frac{k}{\epsilon})})$ grid points inside these two flats. For each grid point, add one child to v, and associate it with the grid point.

3. Output $\mathcal{T}$.

---

there exists one k-tuple candidate in the output which is able to induce a $(1 + O(\epsilon))$-approximation of k-CMedian.

5.3 Upper Bound Estimation for *k*-CMedian

In this section, we show how to obtain an upper bound of the optimal objective value of k-CMedian.

Theorem 11 *Let $P = \{p_1, \ldots, p_n\}$ be the input points of k-CMedian, and $\mathcal{C}$ be the set of k median points of a λ-approximation of k-median on P (without considering the constraint) for some constant $\lambda \geq 1$. Then the Cartesian product $[\mathcal{C}]^k$ contains at least one k-tuple which is able to induce a $(3\lambda + 2)$-approximation of k-CMedian.*

Let $\{c_1, \ldots, c_k\}$ be the k median points in $\mathcal{C}$, and ω be the corresponding objective value of the k-median approximate solution on P. Recall that $\{m_1, \ldots, m_k\}$ are the k unknown optimal constrained median points of P, and $\mathcal{OPT} = \{Opt_1, \ldots, Opt_k\}$ are the corresponding k optimal constrained clusters. To prove Theorem 11, we create a new instance of k-CMedian in the following way: for each point $p_i \in P$, move it

to its nearest point, say c_t, in $\{c_1, \ldots, c_k\}$; let $\tilde{p}_i$ denote the new p_i (note that c_t and $\tilde{p}_i$ overlap with each other). Then the set $\tilde{P} = \{\tilde{p}_1, \ldots, \tilde{p}_n\}$ forms a new instance of k-CMedian. Let μ_{opt} and $\tilde{\mu}_{opt}$ be the optimal cost of P and $\tilde{P}$ respectively, and $\mu_{opt}([\mathcal{C}]^k)$ be the minimum cost of P by restricting its k constrained median points to being a k-tuple in $[\mathcal{C}]^k$. The following two lemmas are keys to proving Theorem 11.

Lemma 12 $\tilde{\mu}_{opt} \leq \omega + \mu_{opt}$.

Proof For each $p_i \in Opt_l$, by triangle inequality we have

$$||\tilde{p}_i - m_l|| \leq ||\tilde{p}_i - p_i|| + ||p_i - m_l||. \tag{57}$$

For both sides of (57), taking the averages over i and l, we get

$$\frac{1}{n} \sum_{l=1}^{k} \sum_{p_i \in Opt_l} ||\tilde{p}_i - m_l|| \leq \frac{1}{n} \sum_{i=1}^{n} ||\tilde{p}_i - p_i|| + \frac{1}{n} \sum_{l=1}^{k} \sum_{p_i \in Opt_l} ||p_i - m_l||. \tag{58}$$

Note that the left-hand side of (58) is not smaller than $\tilde{\mu}_{opt}$, since $\tilde{\mu}_{opt}$ is the optimal object value of k-CMedian on $\tilde{P}$. For the right-hand side of (58), the first term $\frac{1}{n} \sum_{i=1}^{n} ||\tilde{p}_i - p_i|| = \omega$ (by the construction of $\tilde{P}$), and the second term $\frac{1}{n} \sum_{l=1}^{k} \sum_{p_i \in Opt_l} ||p_i - m_l|| = \mu_{opt}$. Thus, we have $\tilde{\mu}_{opt} \leq \omega + \mu_{opt}$. □

Lemma 13 $\mu_{opt}([\mathcal{C}]^k) \leq \omega + 2\tilde{\mu}_{opt}$.

Proof Consider k-CMedian on $\tilde{P}$. Let $\{\tilde{m}_1, \ldots, \tilde{m}_k\}$ be the optimal constraint median points, and $\{\tilde{O}_1, \ldots, \tilde{O}_k\}$ be the corresponding optimal constraint clusters of $\tilde{P}$. Let $\{\tilde{c}_1, \ldots, \tilde{c}_k\}$ be the k-tuple in $[\mathcal{C}]^k$ with $\tilde{c}_l$ being the nearest point in $\mathcal{C}$ to $\tilde{m}_l$. Thus, by an argument similar to the one for (57), we have the following inequality, where $\tilde{p}_i$ is assumed to be clustered in $\tilde{O}_l$.

$$||\tilde{p}_i - \tilde{c}_l|| \leq ||\tilde{p}_i - \tilde{m}_l|| + ||\tilde{m}_l - \tilde{c}_l|| \leq 2||\tilde{p}_i - \tilde{m}_l||. \tag{59}$$

In (59), the last one follows from the facts that $\tilde{c}_l$ is the nearest point in $\mathcal{C}$ to $\tilde{m}_l$ and $\tilde{p}_i \in \mathcal{C}$, which implies $||\tilde{m}_l - \tilde{c}_l|| \leq ||\tilde{m}_l - \tilde{p}_i||$. For both sides of (59), taking the averages over i and l, we have

$$\frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||\tilde{p}_i - \tilde{c}_l|| \leq 2 \frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||\tilde{p}_i - \tilde{m}_l||. \tag{60}$$

Now, consider the following k-CMedian on P. For each p_i, if $\tilde{p}_i \in \tilde{O}_l$, we cluster it to the corresponding median point $\tilde{c}_l$. Then the objective value of the clustering is

$$\frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||p_i - \tilde{c}_l|| \leq \frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} (||p_i - \tilde{p}_i|| + ||\tilde{p}_i - \tilde{c}_l||)$$

$$\leq \frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||p_i - \tilde{p}_i|| + 2\frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||\tilde{p}_i - \tilde{m}_l||. \tag{61}$$

The left-hand side of (61), i.e., $\frac{1}{n} \sum_{l=1}^{k} \sum_{\tilde{p}_i \in \tilde{O}_l} ||p_i - \tilde{c}_l||$, is no smaller than $\mu_{opt}([\mathcal{C}]^k)$ (by the definition), and the right-hand side of (61) is equal to $\omega + 2\tilde{\mu}_{opt}$. Thus, we have $\mu_{opt}([\mathcal{C}]^k) \leq \omega + 2\tilde{\mu}_{opt}$.   □

Proof of Theorem 11 By Lemmas 12 and 13, we know that $\mu_{opt}([\mathcal{C}]^k) \leq 3\omega + 2\mu_{opt}$. It is obvious that the optimal objective value of the k-median clustering is no larger than that of k-CMedian on the same set of points in P. This implies that $\omega \leq \lambda\mu_{opt}$. Thus, we have

$$\mu_{opt}([\mathcal{C}]^k) \leq (3\lambda + 2)\mu_{opt}. \tag{62}$$

The above inequality means that there exists one k-tuple in $[\mathcal{C}]^k$, which is able to induce a $(3\lambda + 2)$-approximation.   □

5.4 Selection Algorithms for k-CMedian

For each of the six constrained clustering problems studied in Sect. 4, the same results (including the approximation ratio and time complexity) can be extended to the corresponding constrained k-median version with slight modification (e.g., assigning the edge cost to be the Euclidean distance rather than squared Euclidean distance when computing the minimum cost circulation on the graph G). Thus, we only focus on the probabilistic clustering problem.

Probabilistic k-Median Clustering (k-PMedian) [19]. Let $V = \{v_1, \ldots, v_n\}$ be a set of nodes; each node v_i is associated with a point set $D_i = \{p_1^i, \ldots, p_h^i\} \subset \mathbb{R}^d$, where each p_l^i has a probability $t_l^i \geq 0$ satisfying the condition $\sum_{l=1}^{h} t_l^i \leq 1$. Let $w_i = \sum_{l=1}^{h} t_l^i$ for $1 \leq i \leq n$ and $W = \sum_{i=1}^{n} w_i$. k-PMedian is the problem of finding k points $\{m_1, \ldots, m_k\}$ in $\mathbb{R}^d$ such that $\sum_{i=1}^{n} \min_{1 \leq j \leq k} dist\{v_i, m_j\}$ is minimized, where $dist\{v_i, m_j\} = \sum_{l=1}^{h} t_l^i ||p_l^i - m_j||$.

Note that for the k-means version of probabilistic clustering, Cormode and McGregor [19] have showed that it can be reduced to an ordinary k-means clustering problem after replacing each D_i by its weighted mean point. However, this strategy can only yield a $(3 + \epsilon)$-approximation for the k-median version [19,58]. We briefly sketch our idea for solving k-PMedian below.

Actually, k-PMedian is equivalent to the k-median clustering on the weighted point set $\bigcup_{i=1}^{n} D_i$ with the constraint that for each $1 \leq i \leq n$, all the points in D_i should be clustered into the same cluster. Thus, we can use our Peeling-and-Enclosing algorithm

for k-CMedian in Sect. 5.2 to generate a set of candidates for the constrained k median points; the difference is that the points have weights, and thus in each sampling step we sample points with probabilities proportional to their weights. To accommodate such a difference, several minor modifications need to be made to Lemma 10 and Lemma 11: all distances are changed to weighted distances, and the involved set sizes (such as $|P|$) are changed to nh.

As for the running time of the Peeling-and-Enclosing algorithm, it still builds the trees with heights equal to k. But the number of children for each node is different. Recall that in the proof of Claim 2, in order to obtain an estimation for $\beta_j = \frac{|Opt_j|}{n}$, we need to try $O(\log n)$ times since $\frac{1}{n} \leq \beta_j \leq 1$; but for k-PMedian, the range of β_j becomes $[\frac{w_{min}}{W}, 1]$ where $w_{min} = \min_{1 \leq i \leq n} w_i$ (note that $W = \sum_{i=1}^{n} w_i \leq n$). Thus, the running time of Peeling-and-Enclosing algorithm becomes $O(nh(\log \frac{n}{w_{min}})^{k+1}d)$. Furthermore, for each k-tuple candidate, we perform the Partition step through assigning each D_i to the m_j with the smallest $dist\{v_i, m_j\}$. Obviously, the Partition step can be finished within linear time. Thus we have the following theorem.

Theorem 12 *There exists an algorithm yielding a* $(1 + \epsilon)$*-approximation for* k*-PMedian with constant probability, in* $O(2^{poly(\frac{k}{\epsilon})} nh (\log \frac{n}{w_{min}})^{k+1} d)$ *time, where* $w_{min} = \min_{1 \leq i \leq n} w_i$.

6 Future Work

Following this work, some interesting problems deserve to be further studied in the future. For example, we reduce the partition step to the minimum cost circulation problem for several constrained clustering problems in Sect. 4; however, since the goal is to find an approximate solution, one may consider using the geometric information to solve the Partition step approximately. In Euclidean space, several techniques have been developed for solving approximate matching problems efficiently [6,51]. But it is still not clear whether such techniques can be extended to solve the constrained matching problems (such as the r-gather or l-diversity) considered in this paper, especially in high dimensional space. We leave it as an open problem for future work.

7 Appendix

7.1 Proof of Lemma 3

Similar to Lemma 1, we prove this lemma by mathematical induction on j.

Base case For $j = 1$, since $o_1 = o$, we just need to let $\tau = o_1'$. Then, we have

$$||\tau - o|| = ||o_1' - o|| = ||o_1' - o_1|| \leq L \leq \sqrt{\epsilon}\delta + (1 + \epsilon)L. \qquad (63)$$

Thus, the base case holds.

Induction step Assume that the lemma holds for any $j \leq j_0$ for some $j_0 \geq 1$ (i.e., the induction hypothesis). Now we consider the case of $j = j_0 + 1$. Similar to

the proof of Lemma 1, we assume that $\frac{|Q_l|}{|Q|} \geq \frac{\epsilon}{4j}$ for each $1 \leq l \leq j$. Otherwise, through a similar idea from Lemma 1, it can be reduced to the case with smaller j, and solved by the induction hypothesis. Hence, in the following discussion, we assume that $\frac{|Q_l|}{|Q|} \geq \frac{\epsilon}{4j}$ for each $1 \leq l \leq j$.

First, we know that $o = \sum_{l=1}^{j} \frac{|Q_l|}{|Q|} o_l$. Let $o' = \sum_{l=1}^{j} \frac{|Q_l|}{|Q|} o'_l$. Then, we have

$$||o - o'|| = ||\sum_{l=1}^{j} \frac{|Q_l|}{|Q|} o_l - \sum_{l=1}^{j} \frac{|Q_l|}{|Q|} o'_l|| \leq \sum_{l=1}^{j} \frac{|Q_l|}{|Q|} ||o_l - o'_l|| \leq L. \qquad (64)$$

Thus, if we can find a grid point τ having $||\tau - o'|| \leq \sqrt{\epsilon}\delta + \epsilon L$, by inequality (64), we will have $||\tau - o|| \leq ||\tau - o'|| + ||o' - o|| \leq \sqrt{\epsilon}\delta + (1 + \epsilon)L$. So we only need to find a grid point close enough to o'.

To find such a τ, we consider the distance from o'_l to o' for any $1 \leq l \leq j$. We have

$$||o'_l - o'|| \leq ||o'_l - o_l|| + ||o_l - o|| + ||o - o'|| \leq 2\sqrt{\frac{j}{\epsilon}}\delta + 2L, \qquad (65)$$

where the first inequality follows from triangle inequality, and the second inequality follows from the facts that $||o'_l - o_l||$ and $||o - o'||$ are both bounded by L, and $||o_l - o|| \leq 2\sqrt{\frac{j}{\epsilon}}\delta$ (by Lemma 2).

This implies that we can use a similar idea in Lemma 1 to construct a ball $\mathcal{B}$ centered at any o'_{l_0} and with radius $r = \max_{1 \leq l \leq j}\{||o'_l - o'_{l_0}||\}$. Also, the simplex $\mathcal{V}'$ is inside $\mathcal{B}$. Note that

$$||o'_l - o'_{l_0}|| \leq ||o'_l - o'|| + ||o' - o'_{l_0}|| \leq 4\sqrt{\frac{j}{\epsilon}}\delta + 4L \qquad (66)$$

by (65), which implies $r \leq 4\sqrt{\frac{j}{\epsilon}}\delta + 4L$. Similar to Lemma 1, we can build a grid inside $\mathcal{B}$ with grid length $\frac{\epsilon r}{4j}$, and the number of grid points is $O((8j/\epsilon)^j)$. Moreover, o' must lie inside $\mathcal{V}'$ by the definition. In this grid, we can find a grid point τ such that $||\tau - o'|| \leq \frac{\epsilon}{4\sqrt{j}}r \leq \sqrt{\epsilon}\delta + \epsilon L$. Thus, $||\tau - o|| \leq ||\tau - o'|| + ||o' - o|| \leq \sqrt{\epsilon}\delta + (1+\epsilon)L$, and the induction step, as well as the lemma, holds.

7.2 Proof of Claim 2 for Lemma 6

Since $1 \geq \beta_j \geq \frac{1}{n}$, there is one integer t between 1 and $\log n$, such that $2^{t-1} \leq \frac{1}{\beta_j} \leq 2^t$. Thus $2^{t/2-1}\sqrt{\epsilon}\delta_{opt} \leq \sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} \leq 2^{t/2}\sqrt{\epsilon}\delta_{opt}$. Together with $\delta \in [\delta_{opt}, (1 + \epsilon)\delta_{opt}]$, we have

$$2^{t/2-1}\sqrt{\epsilon}\frac{\delta}{1 + \epsilon} \leq \sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} \leq 2^{t/2}\sqrt{\epsilon}\delta. \qquad (67)$$

Thus if setting $\hat{r}_j = 2^{t/2}\sqrt{\epsilon}\delta$, we have

$$\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} \leq \hat{r}_j \leq 2(1+\epsilon)\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}. \tag{68}$$

We consider the interval $\mathcal{I} = [\frac{j}{2(1+\epsilon)}\hat{r}_j, j\hat{r}_j]$. Equation (68) ensures that $j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} \in \mathcal{I}$. Also, we build a grid in the interval with grid length $\frac{\epsilon}{2}\frac{1}{2(1+\epsilon)}j\hat{r}_j$, i.e., $\mathcal{R}_j = \{\frac{1+l\frac{\epsilon}{2}}{2(1+\epsilon)}j\hat{r}_j \mid 0 \leq l \leq 4+\frac{2}{\epsilon}\}$. Moreover, the grid length $\frac{\epsilon}{2}\frac{1}{2(1+\epsilon)}j\hat{r}_j \leq \frac{\epsilon}{2}j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}$, which implies that there exists $r_j \in \mathcal{R}_j$ such that

$$j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} \leq r_j \leq \left(1+\frac{\epsilon}{2}\right)j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}. \tag{69}$$

Note that $\mathcal{R}_j \subset \mathcal{R}$, where $\mathcal{R} = \cup_{t=0}^{\log n}\{\frac{1+l\frac{\epsilon}{2}}{2(1+\epsilon)}j2^{t/2}\sqrt{\epsilon}\delta \mid 0 \leq l \leq 4+\frac{2}{\epsilon}\}$. Thus, the Claim is true.

7.3 Proof of Claim 3 for Lemma 6

Note that $\delta_{opt}^2 = \sum_{j=1}^k \beta_j\delta_j^2$, and $\beta_j \leq \beta_l$ for each $1 \leq l \leq j-1$. Thus, we have $\delta_l \leq \sqrt{\frac{1}{\beta_l}}\delta_{opt} \leq \sqrt{\frac{1}{\beta_j}}\delta_{opt}$. Together with $j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} \leq r_j$ (Claim 2) and $||p_{v_l} - m_l|| \leq \epsilon\delta_l + (1+\epsilon)l\sqrt{\frac{\epsilon}{\beta_l}}\delta_{opt}$ (by the induction hypothesis), we have

$$\begin{aligned}
r_j - ||p_{v_l} - m_l|| &\geq j\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} - \left(\epsilon\delta_l + (1+\epsilon)(j-1)\sqrt{\frac{\epsilon}{\beta_l}}\delta_{opt}\right) \\
&\geq (1-(j-1)\epsilon)\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt} - \epsilon\delta_l \\
&\geq (1-(j-1)\epsilon - \sqrt{\epsilon})\sqrt{\frac{\epsilon}{\beta_j}}\delta_{opt}. \tag{70}
\end{aligned}$$

Since $\epsilon \in (0, \frac{1}{4k^2})$ in the input of Algorithm 1, we know $r_j - ||p_{v_l} - m_l|| > 0$. That is, m_l is covered by the ball $B_{j,l}$.

For each $1 \leq l \leq j-1$, we have $|Opt_l\backslash(\cup_{w=1}^{j-1} B_{j,w})| \leq |Opt_l\backslash B_{j,l}|$. For any $p \in Opt_l\backslash B_{j,l}, ||p - m_l|| \geq r_j - ||p_{v_l} - m_l||$. By Markov's inequality, we have

$$|Opt_l\backslash B_{j,l}| \leq \frac{\delta_l^2}{(r_j - ||p_{v_l} - m_l||)^2}|Opt_l|. \tag{71}$$

Together with (70), we have

$$|Opt_l\backslash B_{j,l}| \leq \frac{\delta_l^2}{(1-(j-1)\epsilon - \sqrt{\epsilon})^2\frac{\epsilon}{\beta_j}\delta_{opt}^2}|Opt_l|$$

$$\leq \frac{\delta_l^2}{(1-(j-1)\epsilon-\sqrt{\epsilon})^2 \frac{\epsilon}{\beta_j}\beta_l\delta_l^2}|Opt_l|$$

$$= \frac{\beta_j}{(1-(j-1)\epsilon-\sqrt{\epsilon})^2\epsilon\beta_l}|Opt_l|$$

$$= \frac{\beta_j n}{(1-(j-1)\epsilon-\sqrt{\epsilon})^2\epsilon} \leq \frac{\beta_j n}{(1-j\sqrt{\epsilon})^2\epsilon}, \tag{72}$$

where the second inequality follows from the fact that $\beta_l\delta_l^2 \leq \delta_{opt}^2$, and the fourth equation follows from that $\frac{|Opt_l|}{\beta_l} = n$. Again, $\epsilon \in (0, \frac{1}{4k^2})$ implies that $\frac{\beta_j n}{(1-j\sqrt{\epsilon})^2\epsilon} \leq \frac{4\beta_j n}{\epsilon}$. Thus, in total, we have

$$|Opt_l \backslash B_{j,l}| \leq \frac{4\beta_j n}{\epsilon}. \tag{73}$$

Hence, the Claim is true.

References

1. Ackermann, M.R., Blömer, J., Sohler, C.: Clustering for metric and non-metric distance measures. ACM Trans. Algorithms **6**(4), 59 (2010)
2. Aggarwal, G., Panigrahy, R., Feder, T., Thomas, D., Kenthapadi, K., Khuller, S., Zhu, A.: Achieving anonymity via clustering. ACM Trans. Algorithms **6**(3), 49 (2010)
3. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. J. ACM **55**(5), 23 (2008)
4. Awasthi, P., Charikar, M., Krishnaswamy, R., Sinop, A.K.: The hardness of approximation of euclidean k-means. In: Proceedings of 31st International Symposium on Computational Geometry (SoCG'15), pp. 754–767 (2015)
5. An, H.-C., Bhaskara, A., Chekuri, C., Gupta, S., Madanand, V., Svensson, O.: Centrality of Trees for Capacitated k-Center, Proc. 7th International Conference Integer Programming and Combinatorial Optimization (IPCO'14), pp. 52-63, (2014)
6. Andoni, A., Onak, K., Nikolov, A., Yaroslavtsev, G.: Parallel algorithms for geometric graph problems. In: Proceedings of 46th Symposium on Theory of Computing Conference (STOC'14) (2014)
7. Arkin, E.M., Díaz-Báñez, J.M., Hurtado, F., Kumar, P., Mitchell, J.S.B., Palop, B., Pérez-Lantero, P., Saumell, M., Silveira, R.I.: Bichromatic 2-center of pairs of points. In: Proceedings of 10th Latin American Theoretical Informatics Symposium (LATIN'12), pp. 25–36 (2012)
8. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of 18th ACM-SIAM Symposium on Discrete Algorithms (SODA'07), pp. 1027–1035 (2007)
9. Awasthi, P., Blum, A., Sheffet, O.: Stability yields a PTAS for k-median and k-means clustering. In: Proceedings of 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS'10), pp. 309–318 (2010)
10. Badŏiu, M., Har-Peled, S., Indyk, P.: Approximate clustering via core-sets. In: Proceedings of 34th ACM Symposium on Theory of Computing (STOC'02), pp. 250–257 (2002)
11. Balcan, M.-F., Blum, A., Gupta, A.: Clustering under approximation stability. J. ACM **60**(2), 8 (2013)
12. Balcan, M.-F., Braverman, M.: Finding low error clusterings. In: Proceedings of 22nd Conference on Learning Theory (COLT'09) (2009)
13. Basu, S., Bilenko, M., Mooney, R.J.: A Probabilistic framework for semi-supervised clustering. In: Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04), pp. 59–68 (2004)

14. Bhattacharya, A., Jaiswal, R., Kumar, A.: Faster algorithms for the constrained k-means problem. In: Proceedings of 33rd Symposium on Theoretical Aspects of Computer Science (STACS'16), pp. 16:1–16:13 (2016)
15. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06), pp. 554–560 (2006)
16. Chaudhuri, S., Garg, N., Ravi, R.: The p-neighbor k-center problem. Inf. Proc. Lett. **65**(3), 131–134 (1998)
17. Chen, K.: On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. SIAM J. Comput. **39**(3), 923–947 (2009)
18. Coleman, T., Wirth, A.: A polynomial time approximation scheme for k-consensus clustering. In: Proceedings of 21st ACM-SIAM Symposium on Discrete Algorithms (SODA'10), pp. 729–740 (2010)
19. Cormode, G., McGregor, A.: Approximation algorithms for clustering uncertain data. In: Proceedings of 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'08), pp. 191–200 (2008)
20. Cygan, M., Hajiaghayi, M.T., Khuller, S.: LP rounding for k-centers with non-uniform hard capacities. In: Proceedings of 53rd IEEE Symposium on Foundations of Computer Science (FOCS'12), pp. 273–282 (2012)
21. Dasgupta, S.: The hardness of k-means clustering. Technical Report (2008)
22. Ding, H.: Faster balanced clusterings in high dimension (2018). arXiv preprint arXiv:1809.00932
23. Ding, H., Xu, J.: Solving chromatic cone clustering via minimum spanning sphere. In: Proceedings of 38th International Colloquium on Automata, Languages and Programming (ICALP'11), pp. 773–784 (2011)
24. Ding, H., Xu, J.: Sub-linear time hybrid approximations for least trimmed squares estimator and related problems. In: Proceedings of 30th Annual Symposium on Computational Geometry (SoCG'14), p. 110 (2014)
25. Ding, H., Xu, J.: A unified framework for clustering constrained data without locality property. In: Proceedings of 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'15), pp. 1471–1490 (2015)
26. Ding, H., Gao, J., Xu, J.: Finding global optimum for truth discovery: entropy based geometric variance. In: Proceedings of 32nd Symposium on Computational Geometry (SoCG'16), pp. 34:1–34:16 (2016)
27. Ene, A., Har-Peled, S., Raichel, B.: Fast clustering with lower bounds: no customer too far, no shop too small (2013). CoRR arXiv:1304.7318
28. Erickson, J.: Course lecture: extensions of maximum flow (2019). http://jeffe.cs.illinois.edu/teaching/algorithms/index.html
29. Feldman, D., Monemizadeh, M., Sohler, C.: A PTAS for k-means clustering based on weak coresets. In: Proceedings of 23rd ACM Symposium on Computational Geometry (SoCG'07), pp. 11–18 (2007)
30. Feldman, D., Langberg, M.: A unified framework for approximating and clustering data. In: Proceedings of 43rd ACM Symposium on Theory of Computing (STOC'11), pp. 569–578 (2011)
31. Fernandez de la Vega, W., Karpinski, M., Kenyon, C., Rabani, Y.: Approximation schemes for clustering problems. In: Proceedings of 35th ACM Symposium on Theory of Computing (STOC'03), pp. 50–58 (2003)
32. Gao, J., Tan, P., Cheng, H.: Semi-supervised clustering with partial background information. In: Proceedings of 6th SIAM International Conference on Data Mining (SDM'06), pp. 489–493 (2006)
33. Guha, S., Munagala, K.: Exceeding expectations and clustering uncertain data. In: Proceedings of 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'09), pp. 269–278 (2009)
34. Guruswami, V., Indyk, P.: Embeddings and non-approximability of geometric problems. In: Proceedings of 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03), pp. 537–538 (2003)
35. Hajiaghayi, M.T., Hu, W., Li, J., Li, S., Saha, B.: A constant factor approximation algorithm for fault-tolerant k-median. In: Proceedings of 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14), pp. 1–12 (2014)
36. Har-Peled, S., Raichel, B.A.: Net and Prune: a linear time algorithm for euclidean distance problems. In: Proceedings of 45th ACM Symposium on Theory of Computing (STOC'13), pp. 605–614 (2013)
37. Inaba, M., Katoh, N., Imai, H.: Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In: Proceedings of 10th ACM Symposium on Computational Geometry (SoCG'94), pp. 332–339 (1994)

38. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. (CSUR) **31**(3), 264–323 (1999)
39. Jaiswal, R., Kumar, A., Sen, S.: A simple D2-sampling based PTAS for k-means and other clustering problems. Algorithmica **70**(1), 22–46 (2014)
40. Khuller, S., Pless, R., Sussmann, Y.J.: Fault tolerant k-center problems. Theor. Comput. Sci. **242**(1–2), 237–245 (2000)
41. Khuller, S., Sussmann, Y.J.: The capacitated k-center problem. SIAM J. Discrete Math. (SIAMDM) **13**(3), 403–418 (2000)
42. Kolliopoulos, S.G., Rao, S.: A nearly linear-time approximation scheme for the euclidean k-median problem. SIAM J. Comput. **37**(3), 757–782 (2007)
43. Kumar, A., Sabharwal, Y., Sen, S.: Linear-time approximation schemes for clustering problems in any dimensions. J. ACM **57**(2), 5 (2010)
44. Kumar, N., Raichel, B.: Fault tolerant clustering revisited. In: Proceedings of 25th Canadian Conference on Computational Geometry (CCCG) (2013)
45. Lammersen, C., Schmidt, M., Sohler, C.: Probabilistic k-median clustering in data streams. In: Proceedings of 10th International Workshop on Approximation and Online Algorithms (WAOA'12), pp. 70–81 (2012)
46. Li, J., Yi, K., Zhang, Q.: Clustering with diversity. In: Proceedings of 37th International Colloquium on Automata, Languages and Programming (ICALP'10), pp. 188–200 (2010)
47. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-Diversity: privacy beyond k-anonymity. In: Proceedings of 22nd International Conference on Data Engineering (ICDE'06), p. 24 (2006)
48. Matousek, J.: On approximate geometric k-clustering. Discrete Comput. Geom. **24**(1), 61–84 (2000)
49. Orlin, J.B.: A faster strongly polynomial minimum cost flow algorithm. Oper. Res. **41**(2), 338–350 (1993)
50. Ostrovsky, R., Rabani, Y.,Schulman, L.J., Swamy, C.: The effectiveness of Lloyd-type methods for the k-means problem. In: Proceedings of 47th IEEE Symposium on Foundations of Computer Science (FOCS'06), pp. 165–174 (2006)
51. Sharathkumar, R., Agarwal, P.K.: A Near-linear time ϵ-approximation algorithm for geometric bipartite matching. In: Proceedings of 44th Symposium on Theory of Computing Conference (STOC'12), pp. 385–394 (2012)
52. Singh, V., Mukherjee, L., Peng, J., Xu, Jinhui: Ensemble clustering using semidefinite programming with applications. Mach. Learn. **79**(1–2), 177–200 (2010)
53. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
54. Swamy, C., Shmoys, D.B.: Fault-tolerant facility location. ACM Trans. Algorithms (TALG) **4**(4), 51 (2008)
55. Sweeney, L.: k-Anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(5), 557–570 (2002)
56. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: Proceedings of 17th International Conference on Machine Learning (ICML'00), pp. 1103–1110 (2000)
57. Wagstaff, K., Cardie, C., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of 8th International Conference on Machine Learning (ICML'01), pp. 577–584 (2001)
58. Xu, G., Xu, J.: Efficient approximation algorithms for clustering point-sets. Comput. Geom. **43**(1), 59–66 (2010)
59. Zhu, Z.A., Lattanzi, S., Mirrokni, V.S.: A local algorithm for finding well-connected clusters. In: Proceedings of 30th International Conference on Machine Learning (ICML'13), pp. 396–404 (2013)

Affiliations

Hu Ding[1] **· Jinhui Xu[2]**

✉   Hu Ding
     huding@ustc.edu.cn

     Jinhui Xu
     jinhui@buffalo.edu

[1]   School of Computer Science and Technology, University of Science and Technology of China,
      Hefei, China

[2]   Department of Computer Science and Engineering, State University of New York at Buffalo,
      Buffalo, USA