TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/14 pp 1050–1062
DOI: 10.26599/TST.2022.9010056
Volume 28, Number 6, December 2023

Efficient Algorithm for the k-Means Problem with Must-Link and Cannot-Link Constraints

Chaoqi Jia, Longkun Guo*,†, Kewen Liao†, and Zhigang Lu†

Abstract: Constrained clustering, such as k-means with instance-level Must-Link (ML) and Cannot-Link (CL) auxiliary information as the constraints, has been extensively studied recently, due to its broad applications in data science and AI. Despite some heuristic approaches, there has not been any algorithm providing a non-trivial approximation ratio to the constrained k-means problem. To address this issue, we propose an algorithm with a provable approximation ratio of $O(\log k)$ when only ML constraints are considered. We also empirically evaluate the performance of our algorithm on real-world datasets having artificial ML and disjoint CL constraints. The experimental results show that our algorithm outperforms the existing greedy-based heuristic methods in clustering accuracy.

Key words: Constrained k-means; Must-Link (ML) and Cannot-Link (CL) constraints; approximation algorithm; constrained clustering

1 Introduction

As a well-known clustering problem, given n data samples/points, the k-means clustering[1–4] aims to partition the data points into k clusters, such that the overall squared Euclidean distance between each data point and its closest cluster centroid (mean of the cluster) is minimized. Due to the Non-deterministic Polynomial (NP)-hardness of the k-means problem[5–7], several approximation algorithms were proposed[8–11]. Among these studies, Kanungo et al.[9] presented a local improvement heuristic based on swapping centers and proved that this yields a $(9 + \epsilon)$-approximation algorithm. Ahmadian et al.[11] provided a 6.357-approximation algorithm with respect to the standard Linear Programming (LP) relaxation, achieving the state-of-the-art approximation ratio for the standard k-means problem. With a different line of methodology, Arthur and Vassilvitskii[8] provided a k-means++ algorithm achieving an approximation ratio of $O(\log k)$ by improving the initialization step.

In addition to the standard k-means clustering, the constrained k-means clustering problem, first introduced by Wagstaff and Cardie[12] and Wangstaff et al.[13], has attracted research interests in recent years[14–19]. Specifically, the constrained k-means clustering problem considers additional Must-Link (ML) and Cannot-Link (CL) constraints to the dataset on top of the standard k-means setting, where any data points belong to the same ML/CL set must/cannot be placed in the same cluster. There are many real-life cases arising from the constrained k-means setting. For example, we consider a set of surveillance cameras laid out at different locations (not closed to each other) to capture human faces. The captured images can then be clustered by

• Chaoqi Jia and Longkun Guo are with Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China. E-mail: chaoqi.jia@foxmail.com; longkun.guo@gmail.com.

• Kewen Liao is with HilstLab, Peter Faber Business School, Australian Catholic University, Sydney 2060, Australia. E-mail: kewen.liao@acu.edu.au.

• Zhigang Lu is with Macquarie University Cyber Security Hub, Macquarie University, Sydney 2109, Australia. E-mail: zhigang.lu@mq.edu.au.

∗ To whom correspondence should be addressed.

† Longkun Guo, Kewen Liao, and Zhigang Lu contribute equally to this paper.

Manuscript received: 2022-08-09; revised: 2022-10-24; accepted: 2022-11-20

personal identity for face categorization. Face images continuously captured from a single camera must come from the same person, while images from different cameras at the same time must be different persons. We can use this background information to improve the accuracy of clustering.

Davidson and Ravi[20] proved that it is NP-complete even only to determine whether the constrained k-means problem is feasible regarding general CL constraints; whereas, it is polynomial solvable when only considering ML constraints. Their results indicate that feasible approximation solutions may exist to a simplified constrained k-means problem considering only ML constraints. However, to the best of our knowledge, there is no bounded approximation even to such a simplified problem.

Contributions. In this paper, we fill the aforementioned research gap by providing an $O(\log k)$ approximation algorithm leveraging the initialization step of k-means++. Based on our approximation algorithm, we further propose a heuristic-based convergent k-means algorithm for the problem with both ML constraints and Disjoint CL (DCL) constraints, where the DCL simplification requires the CL sets are disjoint from each other. We also provide a counterexample for the problem with ML constraints, which fails to achieve an approximation ratio when uniformly selecting a data point to be the representative data point for the ML set at random. Instead, we show that it is desirable for our algorithm to set a mass center as the representative point. Finally, we extensively evaluate the clustering accuracy of our algorithm against the greedy-based heuristic algorithm[13] on real-world datasets. The practical results validate that our algorithm consistently achieves a better performance.

2 Related Work

Wagstaff et al.[13] first introduced the constrained k-means clustering, where two additional constraints, ML and CL, were considered in addition to the classic k-means problem. They proposed a heuristic greedy algorithm to solve the constrained k-means problem. Their algorithm firstly randomly picks the centers from the set of data points, then assigns each data point to its nearest center or to the center where its ML fellow data points belong to. Note that a data point cannot be assigned to any center taken by the point's CL sets. This work also demonstrated in their experiments that the accuracy of k-means clustering with constraints can be significantly improved. However, the greedy solution is unconcerned about the loss/objective function of the problem, and cannot guarantee the convergence of the algorithm.

Due to the NP-completeness of the standard k-means problem with the CL and ML constraints, the aforementioned works cannot provide an approximation ratio in their solutions. Different from solving the original constrained k-means clustering, the work in Ref. [20] introduced two new constraint types to make a feasible constrained k-means clustering and provide a corresponding algorithm that satisfies the given constraints.

Recently, Baumann[21] provided a binary linear programming-based k-means algorithm to solve the "soft" constrained k-means problem. Instead of attempting to satisfy all constraints, he imposed punishment to the objective function if the constraints are not satisfied. To ensure that must-link and cannot-link constraints are respected, his work formulated a binary linear program and solved it with mathematical programming. However, the solution is inapplicable in the case where the constraints must be satisfied. Thus, in this paper, we aim to provide an approximation algorithm for the constrained k-means problem and achieve a ratio analysis on ML-constrained k-means.

3 Preliminary

Given a set of n data points P in d-dimensional space and several sets of ML and CL, the constrained k-means problem aims to partition n data points into k clusters (same as the traditional k-means), however, subject to the ML and CL constraints. In this paper, we study a special constrained k-means problem having ML constraints (the same as Ref. [13]) and DCL constraints.

In this paper, the ML constraint is a set of data point sets $\mathcal{X} = \{X_1, X_2, \dots, X_h\}$, where every $X \in \mathcal{X}$ is an ML set. Given data points p_i and p_j, if p_i, $p_j \in X \subseteq P$, then p_i, $p_j \in A_m$, where A_m denotes cluster m. Similarly, the DCL constraint is a set of data point sets $\mathcal{Y} = \{Y_1, Y_2, \dots, Y_l\}$, where $Y_i \subset P$ is a DCL set with $|Y_i| \leqslant k$, and $\forall i \neq j, Y_i \cap Y_j = \varnothing$ holds. Given data points p_i and p_j, if p_i, $p_j \in Y \subseteq P$ and $p_i \in A_m$ then $p_j \notin A_m$ must hold, where A_m denotes cluster m.

Then finding the k centers C of an instance of the constrained k-means problem and partitioning the data points into k clusters $\mathcal{A} = \{A_1, A_2, \dots, A_k\}$

accordingly can be formulated as solving the problem below:

$$\{C,\mathcal{A}\} = \arg\min \sum_{i=1}^{k} \sum_{\substack{p\in A_i\in\mathcal{A}\\ c_i\in C}} \|p-c_i\|_2^2,$$

$$\forall X\in\mathcal{X}, \exists A\in\mathcal{A}, \text{ s.t., } X\subseteq A,$$

$$\forall u,v\in Y\in\mathcal{Y}, \nexists A\in\mathcal{A}, \text{ s.t., } u,v\in A,$$

where $\mathcal{X}$ and $\mathcal{Y}$ are ML and DCL sets, respectively. Moreover, $\|p-c_i\|_2$ is the Euclidean distance between data points p and c_i, and $\sum_{i=1}^{k}\sum_{p\in A_i}\|p-c_i\|_2^2$ is the same cost function of the k-means clustering.

In addition to the above, in this paper, we also use $d_{\min}(p,C)$ to denote the shortest distance from a data point to its closest center we have already selected,

$$d_{\min}(p,C) = \min_{c\in C}\|p-c\|_2,$$

and use $D^2(p)$ to denote the cost of the data point p by the constrained k-means algorithm. In particular, if the data point p is from an ML set X_i, it will be assigned to the nearest center of $\bar{X}_i$. Moreover, we set $\phi(A_i)$ as the cost of $A_i\in\mathcal{A}$. For briefness, we denote $\psi=\sum_{i=1}^{k}\phi(A_i)$ as the cost of the whole dataset.

The optimal center of a given cluster is the average or mean of the cluster, which is the same characteristic as the k-means clustering problem. Thus, Lemma 1 is also useful for the constrained k-means problem.

Lemma 1[8] Let S be a set of data points with mass center $c(S)$, and let z be an arbitrary data point. Then, $\sum_{x\in S}\|x-z\|_2^2-\sum_{x\in S}\|x-c(S)\|_2^2 = |S|\cdot\|c(S)-z\|_2^2$.

4 Factor-$O(\log k)$ Approximation

In this section, we provide a "D^2 weighting" sampling initialization method to select k centers and then show that the center set has a cost at most $O(\log k)$ times of an optimal solution.

4.1 Initialization algorithm

We devise a new initialization algorithm to select the first k centers for constrained k-means clustering with ML sets, and show that the initialization actually deserves an approximation ratio $O(\log k)$.

To properly assign the ML sets, we consider each ML set as a "big point". For each "big point", we use the ML set mass center to represent the set and use the number of data points in ML as its weight. By Lemma 1, we get the following property for ML sets.

Lemma 2 For a must-link set X_i, we denote its mass center by $\bar{X}_i$. Then, for the data point x in X_i and an arbitrary data point p in $\mathbf{R}^d$, we get

$$\sum_{x\in X_i}\|x-p\|_2^2 = \sum_{x\in X_i}\|x-\bar{X}_i\|_2^2 + |X_i|\cdot\|\bar{X}_i-p\|_2^2.$$

Following Lemma 1 and setting S as X_i, z as p, and $c(S)$ as $\bar{X}_i$, which is the mass center of the ML set X_i, we immediately have the following equation:

$$\sum_{x\in X_i}\|x-p\|_2^2 - \sum_{x\in X_i}\|x-\bar{X}_i\|_2^2 = |X_i|\cdot\|\bar{X}_i-p\|_2^2.$$

To implement the k-means++ sampling and select the data points of P in an uniform way, we set the possibility of selecting a data point proportional to its squared distance to the current center set. Note that compared with the traditional k-means problem, we have to focus on the cost according to the constraints.

Therefore, for the current dataset P and center set C with $|C|<k$, the initialization samples a random data point $p_c\in P$ with the following possibility $\frac{D^2(p_c)}{\sum_{p'\in P}D^2(p')}$, where the data point p can be an unconstrained point or a mass center from an ML set. Then we use the data point p_c or $\bar{X}_i$ (the mass center of the ML set where $p_c\in X_i$) as a new center. In particular, when C is empty, the first step of the initialization uniformly selects a data point p_0 from all data points in the entire dataset P at random. The procession of adding the sampled data point as a center to C repeats until $|C|\geqslant k$. The detailed algorithm is depicted in Algorithm 1.

The performance guarantee of the algorithm can be stated as in the following theorem whose proof is postponed to Section 4.2.

Algorithm 1 Initialization for center selection

```
Input: Database $P$ of size $n$ with a family of ML sets $\mathcal{X}$, a
       positive integer $k$.
Output: Set of center $C$.
Uniformly select a data point $p_0 \in P$ at random, and set
  $C \leftarrow p_0$;
while $|C| < k$ do
    Select a data point $p_c$ from $P$ with probability
    $\frac{D^2(p_c)}{\sum_{p'\in P} D^2(p')}$, where $D^2(p_c)$ is computed by the
    following rule:
    if $p_c \in X \in \mathcal{X}$ then
        Set $D^2(p_c) = d_{\min}^2(\bar{X}, C) + \|p_c - \bar{X}\|_2^2$;
        Set $p_c \leftarrow \bar{X}$, where $\bar{X}$ is the mass center of $X$;
    else
        Set $D^2(p_c) = d_{\min}^2(p_c, C)$;
    end
    Let $C \leftarrow p_c$;
end
Return $C$
```

Theorem 1 If C is a set of centers selected by Algorithm 1, then the expectation cost of C satisfies $E[\psi] \leqslant 8(\ln k + 2)\psi_{\text{OPT}}$, where ψ_{OPT} is the cost of optimal solution for the clustering.

4.2 Proof of Theorem 1

In the following, we argue that the Algorithm 1 deserves an approximation ratio $O(\log k)$ for the ML constrained k-means problem.

We first analyze that the first step is simple yet important in the case, which uniformly selects one center at random. Below is the Lemma 3 with a similar form to Lemma 3.1 in Ref. [8], whose proof also follows a similar line but with different details.

Lemma 3 Let A be an arbitrary cluster from the optimal cluster C_{OPT}, where C is an empty center set for the clustering. We select uniformly a random center p_0 and add it to C from A. Then $E[\phi(A)] = 2\phi_{\text{OPT}}(A)$.

Proof Let $c(A)$ denote the mass center of A, which is an optimal cluster from C_{OPT}. In addition, the cluster A contains two kinds of data points, which are unconstrained points and ML points. We denote the unconstrained points in A as $P(A)$, and a family of ML sets belonging to A as $\mathcal{X}(A)$. Because there is only one center, so all points of the dataset are assigned to it. Considering an ML set as a "big point" with a relatively large weight and defining $V_{\mathcal{X}(A)} = \bigcup_{X\in\mathcal{X}(A)} X$, $E[\phi(A)]$ can be computed as

$$E[\phi P(A)] = \frac{1}{|A|}\left(\sum_{p_0\in P(A)}\Big(\sum_{p\in P(A)}\|p-p_0\|_2^2 + \sum_{x\in V_{\mathcal{X}(A)}}\|x-p_0\|_2^2\Big) + \sum_{X_i\in\mathcal{X}(A)}|X_i|\Big(\sum_{p\in P(A)}\|p-\bar{X}_i\|_2^2 + \sum_{x\in V_{\mathcal{X}(A)}}\|x-\bar{X}_i\|_2^2\Big)\right) =$$

$$\frac{1}{|A|}\sum_{a\in A}\Big(\sum_{p_0\in P(A)}\|a-p_0\|_2^2 + \sum_{X_i\in\mathcal{X}(A)}|X_i|\cdot\|a-\bar{X}_i\|_2^2\Big) =$$

$$\frac{1}{|A|}\sum_{a\in A}\sum_{p_0\in P(A)}\|a-p_0\|_2^2 + \frac{1}{|A|}\sum_{a\in A}\sum_{X_i\in\mathcal{X}(A)}|X_i|\cdot\|a-\bar{X}_i\|_2^2 \tag{1}$$

where $\bar{X}_i$ is the mass center for the ML set X_i.

Then by Lemma 2, for an ML set X_i and an arbitrary data point a, we know that $|X_i|\cdot\|a-\bar{X}_i\|_2^2 = \sum_{x\in X_i}\|a-x\|_2^2 - \sum_{x\in X_i}\|x-\bar{X}_i\|_2^2$. Summing over all data points $a\in A$, we have

$$\sum_{a\in A}|X_i|\cdot\|a-\bar{X}_i\|_2^2 = \sum_{a\in A}\sum_{x\in X_i}\|a-x\|_2^2 - \sum_{a\in A}\sum_{x\in X_i}\|x-\bar{X}_i\|_2^2.$$

By simplifying the above results and incorporating them into the Eq. (1), we have the expectation $E[\phi(A)]$ in the following:

$$E[\phi(A)] = \frac{1}{|A|}\sum_{a\in A}\sum_{p_0\in P(A)}\|a-p_0\|_2^2 + \frac{1}{|A|}\sum_{a\in A}\sum_{x\in V_{\mathcal{X}(A)}}\|a-x\|_2^2 - \frac{1}{|A|}\sum_{a\in A}\sum_{x\in V_{\mathcal{X}(A)}}\|x-\bar{X}_i\|_2^2 = \frac{1}{|A|}\sum_{a\in A}\sum_{p_0\in A}\|a-p_0\|_2^2 - \frac{1}{|A|}\sum_{a\in A}\sum_{x\in V_{\mathcal{X}(A)}}\|x-\bar{X}_i\|_2^2.$$

Thus, by Lemma 1 we get $\sum_{a\in A}\|a-p_0\|_2^2 = \sum_{a\in A}\|a-c(A)\|_2^2 + |A|\cdot\|p_0-c(A)\|_2^2$, where p_0 can be an arbitrary data point in A. So we have $E[\phi(A)]$ as given by the following formulation and complete the proof:

$$E[\phi(A)] = \frac{1}{|A|}\sum_{p_0\in A}\Big(\sum_{a\in A}\|a-c(A)\|_2^2 + |A|\cdot\|p_0-c(A)\|_2^2\Big) - \sum_{X_i\in\mathcal{X}(A)}\sum_{x\in X_i}\|x-\bar{X}_i\|_2^2 \leqslant \frac{2}{|A|}\sum_{p_0\in A}\sum_{a\in A}\|a-c(A)\|_2^2 = 2\sum_{a\in A}\|a-c(A)\|_2^2.$$ ■

The next step is based on one center to prove an analog of Lemma 3, but is designated for the other data points belonging to different optimal clusters that are not yet covered, with D^2 weighting probability of section.

Lemma 4 Let A be an arbitrary cluster in C_{OPT} with $A\cap C = \varnothing$, where C is a center set for the current clustering. If we add a random center p_c to C from A, selected with D^2 weighting probability, then according to the new clustering center the expectation potential function satisfies $E[\phi(A)] \leqslant 8\phi_{\text{OPT}}(A)$.

Proof According to the algorithm, the probability, at which we select an unconstrained point p_c as our new center from an arbitrary optimal cluster A, is precisely $\frac{D^2(p_c)}{\sum_{a\in A} D^2(a)}$. For the ML sets in A, we select $\bar{X}_i$ (mass center of X_i) at a probability precisely $\frac{|X_i|\cdot d_{\min}^2(\bar{X}_i,C)+\sum_{x\in X_i}\|x-\bar{X}_i\|_2^2}{\sum_{a\in A} D^2(a)}$. After selecting the new center, we process the constraints to compute the cost for the constrained k-means problem. For each ML set X from the cluster A, we require every data points $x \in X$ to be assigned to the center, which is the nearest one for $\bar{X}$.

Therefore, we sum the cost for different kinds of data points selected as centers. Let $P(A)$ and $\mathcal{X}(A)$ denote the unconstrained points in A and a family of ML constraints from A, respectively.

Firstly, we consider that the center p_c is selected by the unconstrained points that $E[\phi(P(A))]$ is at most

$$\begin{gathered}
E[\phi(P(A))] \leqslant \sum_{p_c\in P(A)} \frac{D^2(p_c)}{\sum_{a\in A} D^2(a)} \times \\
\Bigg(\sum_{p\in P(A)} \min(D(p), \|p-p_c\|_2)^2 + \\
\sum_{X\in\mathcal{X}(A)} (|\bar{X}| \min(d_{\min}(\bar{X},C), \|\bar{X}-p_c\|_2)^2 + \\
\sum_{x\in X} \|x-\bar{X}\|_2^2) \Bigg) \qquad (2)
\end{gathered}$$

Then we show the center is selected by a mass center for an ML set $X_i \in \mathcal{X}(A)$. We use $\bar{X}_i$ on behalf of the new center in Eq. (3), and have the bound of $E[\phi(\mathcal{X}(A))]$ as follows:

$$\begin{gathered}
E[\phi(\mathcal{X}(A))] \leqslant \sum_{X_i\in\mathcal{X}(A)} \frac{D^2(X_i)}{\sum\limits_{a\in A} D^2(a)} \times \\
\Bigg(\sum_{p\in P(A)} \min(D(p), \|p-\bar{X}_i\|_2)^2 + \\
\sum_{X\in\mathcal{X}(A)} (|\bar{X}| \min(d_{\min}(\bar{X}_i,C), \|\bar{X}-\bar{X}_i\|_2)^2 + \\
\sum_{x\in X} \|x-\bar{X}\|_2^2) \Bigg) \qquad (3)
\end{gathered}$$

Next, we will sum up the two cases for A to obtain $E[\phi(A)] = E[\phi(P(A))] + E[\phi(\mathcal{X}(A))]$, and establish its bound as the sum of the right sides of Formulas (2) and (3).

Below we will separately compute the results for the two cases in cluster A. According to Eq. (2), we calculate the expected value of the cases where the new center is an unconstrained point or a mass center of an ML set. Thus, we gain the expectations for the corresponding clustering as in the following.

For Eq. (2), assume that we select an unconstrained point as the new center. Due to the feature that data point in the k-means algorithm will be assigned to its closest center, we can obtain $D(p_c) \leqslant \|a-p_c\|_2 + D(a)$ to all p_c, where a follows the property of triangle inequality. Thus, the power-mean inequality deduces that $D^2(p_c) \leqslant 2\|a-p_c\|_2^2 + 2D^2(a)$. Summing over all $a \in A$, we then have that $D^2(p_c) \leqslant \frac{2}{|A|}\sum_{a\in A}\|p_c - a\|_2^2 + \frac{2}{|A|}\sum_{a\in A} D^2(a)$, and hence $E[\phi(P(A))]$ is at most

$$\begin{gathered}
E[\phi(p(A))] \leqslant \sum_{p_c\in P(A)} \frac{\frac{2}{|A|}\sum\limits_{a\in A}\|a-p_c\|_2^2 + \frac{2}{|A|}\sum\limits_{a\in A} D^2(a)}{\sum\limits_{a\in A} D^2(a)} \times \\
\Bigg(\sum_{p\in P(A)} \min(D(p), \|p-p_c\|_2)^2 + \\
\sum_{X\in\mathcal{X}(A)} (|\bar{X}| s \min(d_{\min}(\bar{X},C), \|\bar{X}-p_c\|_2)^2 + \\
\sum_{x\in X} \|x-\bar{X}\|_2^2) \Bigg) \leqslant \\
\sum_{p_c\in P(A)} \frac{\frac{2}{|A|}\sum_{a\in A}\|p_c-a\|_2^2}{\sum_{a\in A} D^2(a)} \cdot \Bigg(\sum_{p\in P(A)} D^2(p) + \\
\sum_{X\in\mathcal{X}(A)} (|\bar{X}| \cdot d_{\min}^2(\bar{X},C) + \sum_{x\in X}\|x-\bar{X}\|_2^2) \Bigg) + \\
\sum_{p_c\in P(A)} \frac{\frac{2}{|A|}\sum_{a\in A} D^2(a)}{\sum_{a\in A} D^2(a)} \cdot \Bigg(\sum_{p\in P(A)} \|p-p_c\|_2^2 + \\
\sum_{X\in\mathcal{X}(A)} (|\bar{X}| \cdot \|\bar{X}-p_c\|_2^2 + \sum_{x\in X}\|x-\bar{X}\|_2^2) \Bigg).
\end{gathered}$$

Furthermore, we clearly have $\sum_{a\in A} D^2(a) = \sum_{p_c\in P(A)} D^2(p_c) + \sum_{X\in\mathcal{X}(A)}\sum_{x\in X} D^2(x)$. Therefore, by Lemma 2, we know $E[\phi(P(A))]$ is bounded by

$$\begin{gathered}
\frac{2}{|A|}\sum_{p_c\in P(A)}\sum_{a\in A}\|a-p_c\|_2^2 + \frac{2}{|A|}\sum_{p_c\in P(A)}\sum_{a\in A}\|a-p_c\|_2^2 \leqslant \\
\frac{4}{|A|}\sum_{p_c\in P(A)}\sum_{a\in A}\|a-p_c\|_2^2.
\end{gathered}$$

That is, Formula (2) could be

$$E[\phi(P(A))] \leqslant \frac{4}{|A|}\sum_{p_c\in P(A)}\sum_{a\in A}\|a-p_c\|_2^2 \qquad (4)$$

In addition, by Formula (3), we assume that the new center is selected by forming an ML set X_i. According to the algorithm, the must-link set has a possibility $\sum_{x\in X_i} D^2(x) = |X_i| d^2_{\min}(\bar{X}_i, C) + \sum_{x\in X_i} \|x - \bar{X}_i\|_2^2$ to get its mass center $\bar{X}_i$ selected as the new center.

Compared with Formula (2), the case also satisfies triangle inequality $d_{\min}(\bar{X}_i, C) \leqslant d_{\min}(a, C) + \|a - \bar{X}_i\|$ for any a, and $\bar{X}_i$ follows a similar way as the above case. Due to the influence for the constrained relationships, the data point a has $d^2_{\min}(a, C) \leqslant D^2(a)$. Thus, summing over all $a \in A$, we have

$$|X_i| d^2_{\min}(\bar{X}_i, C) \leqslant \frac{2|X_i|}{|A|} \sum_{a\in A} D^2(a) + \frac{2|X_i|}{|A|} \sum_{a\in A} \|a - \bar{X}_i\|_2^2$$

by the power-means inequality as well. Thus for $D^2(X_i)$, we have

$$\begin{aligned} D^2(X_i) = \sum_{x\in X_i} D^2(x) = \\ |X_i| \cdot d^2_{\min}(\bar{X}_i, C) + \sum_{x\in X_i} \|x - \bar{X}_i\|_2^2 \leqslant \\ \frac{2|X_i|}{|A|} \sum_{a\in A} D^2(a) + \frac{2|X_i|}{|A|} \sum_{a\in A} \|a - \bar{X}_i\|_2^2 + \\ \sum_{x\in X_i} \|x - \bar{X}_i\|_2^2 = \\ \frac{2|X_i|}{|A|} \sum_{a\in A} D^2(a) + \frac{2|X_i|}{|A|} \sum_{a\in A} \|a - \bar{X}_i\|_2^2 + \\ \frac{1}{|A|} \sum_{a\in A} \sum_{x\in X_i} \|x - \bar{X}_i\|_2^2. \end{aligned}$$

Then by Lemma 1, the above equation can be deduced to

$$\begin{aligned} \frac{2|X_i|}{|A|} \sum_{a\in A} \|a - \bar{X}_i\|_2^2 + \frac{2}{|A|} \sum_{a\in A} \sum_{x\in X_i} \|x - \bar{X}_i\|_2^2 = \\ \frac{2}{|A|} \sum_{a\in A} \sum_{x\in X_i} \|a - x\|_2^2. \end{aligned}$$

Thus, combining the above equation with Formula (2), we get the bound of $E[\phi(\mathcal{X}(A))]$ as

$$\begin{aligned} \sum_{X_i\in\mathcal{X}(A)} \Big(\frac{2|X_i|}{|A|} \sum_{a\in A} D^2(a) + \frac{2}{|A|} \sum_{a\in A} \sum_{x\in X_i} \|a - x\|_2^2\Big) / \\ \sum_{a\in A} D^2(a) \Bigg(\sum_{p\in P(A)} \min(D(p), \|p - \bar{X}_i\|_2)^2 + \\ \sum_{X\in\mathcal{X}(A)} |\bar{X}| \cdot \min(d_{\min}(\bar{X}_i, C), \|\bar{X} - \bar{X}_i\|_2)^2 + \\ \sum_{x\in X} \|x - \bar{X}\|_2^2 \Bigg) \leqslant \\ \sum_{X_i\in\mathcal{X}(A)} \frac{2|X_i| \sum_{a\in A} D^2(a)}{|A| \sum_{a\in A} D^2(a)} \Bigg(\sum_{p\in P(A)} \|p - \bar{X}_i\|_2^2 + \\ \sum_{X\in\mathcal{X}(A)} (|\bar{X}| \cdot \|\bar{X} - \bar{X}_i\|_2^2 + \sum_{x\in X} \|x - \bar{X}\|_2^2) \Bigg) + \\ \sum_{X_i\in\mathcal{X}(A)} \frac{\frac{2}{|A|} \sum_{a\in A} \sum_{x\in X_i} \|a - x\|_2^2}{\sum_{a\in A} D^2(a)} \Bigg(\sum_{p\in P(A)} D^2(p) + \\ \sum_{X\in\mathcal{X}(A)} (|\bar{X}| \cdot d^2_{\min}(\bar{X}_i, C) + \sum_{x\in X} \|x - \bar{X}\|_2^2) \Bigg). \end{aligned}$$

In the next step, we deal with Formula (3) in a similar way as for Formula (2). Also defining $V_{\mathcal{X}(A)} = \bigcup_{\mathcal{X}\in X(A)} X$, we obtain

$$\begin{aligned} E[\phi(\mathcal{X}(A))] \leqslant \\ \frac{4}{|A|} \sum_{X_i\in\mathcal{X}(A)} \sum_{x\in X_i} \sum_{a\in A} \|a - x\|_2^2 = \\ \frac{4}{|A|} \sum_{x\in V_{\mathcal{X}(A)}} \sum_{a\in A} \|a - x\|_2^2. \end{aligned}$$

Finally, combining with the two aforementioned cases, we obtain the bound of $E[\phi(A)]$ and complete the proof,

$$\begin{aligned} \frac{4}{|A|} \sum_{p_c\in P(A)} \sum_{a\in A} \|a - p_c\|_2^2 + \frac{4}{|A|} \sum_{x\in V_{\mathcal{X}(A)}} \sum_{a\in A} \|a - x\|_2^2 \leqslant \\ 8\phi_{\mathrm{OPT}}(A). \end{aligned}$$ ■

Based on Lemmas 3 and 4, we shall show the total error is at most $O(\log k)$ times of the optimum. We first give an even more general bound by Lemma 5 as in the following, whose special case will yield the approximation ratio of our algorithm.

Lemma 5 Let A_u be a set of clusters in C_{OPT} with $A_u \cap C = \varnothing$, where C is a center set for the current clustering and u is the number of "uncovered" optimal clusters. Let A_c denote the set of data points in these clusters $A_c = C_{\mathrm{OPT}} - A_u$. Suppose we add $t \leqslant u$ random centers to C, selected with "D^2 weighting". Let ψ' denote the corresponding potential for the resulting clustering after selecting a new center in an iteration. Then,

$$\begin{aligned} E[\psi'] \leqslant (\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u)) \cdot (1 + H_t) + \\ \frac{u - t}{u} \cdot \phi(A_u), \end{aligned}$$

where H_t denotes the harmonic sum $H_t = 1 + \frac{1}{2} + \cdots + \frac{1}{t}$.

Proof We prove the lemma by induction. The 2-tuple (t, u) means that the clustering needs to select t centers from u optimal clusters that are "uncovered". In fact, the key to inductive is that (t, u) selects an arbitrary data point and arrives to the two cases $(t-1, u)$ or $(t-1, u-1)$. Therefore, it is obvious that $(0, u)$ and $(1, 1)$ are the base of induction.

Due to $1 + H_t = (u-t)/u = 1$, for the case $(0, u)$, the clustering result satisfies

$$E(\psi') = \psi \leqslant \phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u) + \phi(A_u).$$

Next, provided that $(1, 1)$, we will select the latest center from the uncovered cluster with probability exactly $\frac{\phi(A_u)}{\psi}$. In this case, if the new center is selected by the uncovered cluster, then the last uncovered cluster will gain at most $8\phi_{\mathrm{OPT}}(A_u)$ by Lemma 4. So we have

$$E(\psi') \leqslant \frac{\phi(A_u)}{\psi}\cdot(\phi(A_c)+8\phi_{\mathrm{OPT}}(A_u)) + \frac{\phi(A_c)}{\psi}\cdot\psi \leqslant 2\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u).$$

Then, we proceed to prove the inductive step by considering two cases. Based on the hypotheses $(t-1, u)$ and $(t-1, u-1)$, we shall show the lemma holds for the (t, u) case. The first stage is supposed for our selected center to come from a covered cluster. So the above case happens with probability exactly $\frac{\phi(A_c)}{\psi}$. Note that if any point becomes the new center, the value of ψ must be decreased.

Note that in our algorithm, we use the cost of each ML set to be the probability of selection. The cost of ML sets is included in $\phi(A)$, so we will briefly describe the proof in the inductive step.

Beginning from $(t-1, u)$ to (t, u), the algorithm selects a new center from a covered optimal cluster. So the contribution of the case $(t-1, u)$ to $E[\psi']$ is bounded by

$$\frac{\phi(A_c)}{\psi}\cdot((\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u))\cdot(1 + H_{t-1}) + \frac{u-t+1}{u}\cdot\phi(A_u)).$$

Then, we assume the algorithm selects a new center from an uncovered optimal cluster, which has a probability of $\frac{\phi(A_c)}{\psi}$. By the power-mean inequality, $\frac{1}{u}\phi^2(A_u) \leqslant \sum_{A\subseteq A_u}\phi^2(A)$ holds. So, the contribution of the case $(t-1, u-1)$ to $E[\psi']$ is bounded by

$$\frac{\phi(A_u)}{\psi}\cdot(\phi(A_c) + \phi(A) + 8\phi_{\mathrm{OPT}}(A_u) - 8\phi_{\mathrm{OPT}}(A))\cdot(1 + H_{t-1}) + \frac{u-t}{u-1}\cdot(\phi(A_u) - \phi(A)) \leqslant \frac{\phi(A_u)}{\phi}\cdot((\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u))(1 + H_{t-1}) + \frac{u-t}{u}\cdot\phi(A_u)).$$

Finally, we combine both cases of $(t-1, u)$ and $(t-1, u-1)$ to obtain the following bound and complete the proof,

$$\begin{aligned} E[\psi'] \leqslant\ & \frac{\phi(A_c)}{\psi}\cdot\Big((\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u)) \times (1 + H_{t-1}) + \frac{u-t+1}{u}\cdot\phi(A_u)\Big) + \\ & \frac{\phi(A_u)}{\psi}\cdot\Big((\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u)) \times (1 + H_{t-1}) + \frac{u-t}{u}\cdot\phi(A_u)\Big) \leqslant \\ & \Big(\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u)\Big)(1 + H_{t-1}) + \frac{u-t}{u}\cdot\phi(A_u) + \frac{\phi(A_c)}{\psi}\cdot\frac{1}{u}\cdot\phi(A_u) \leqslant \\ & \Big(\phi(A_c) + 8\phi_{\mathrm{OPT}}(A_u)\Big)(1 + H_t) + \frac{u-t}{u}\cdot\phi(A_u). \end{aligned}$$

■

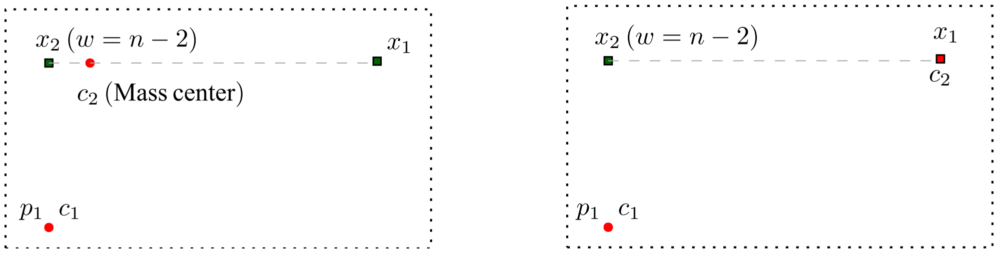
Next, we use Lemma 5 to generate the desired bound $E[\psi] \leqslant 8(\ln k + 2)\phi_{\mathrm{OPT}}$. Consider that the center set C has a center after the completion of the first step of Algorithm 1. Let A_0 be a cluster of C_{OPT} that contains the first selected center, so that $\phi(A_0)$ is at most $2\phi_{\mathrm{OPT}}(A_0)$ by Lemma 3. We combine the case of $(k-1, k-1)$ as above with $\phi(A_0)$, then from Lemma 5 we have

$$\begin{aligned} E[\psi] \leqslant\ & \Big(\phi(A_0) + 8\phi_{\mathrm{OPT}}(A\setminus A_0)\Big)\cdot(1 + H_{k-1}) \leqslant \\ & \Big(2\phi_{\mathrm{OPT}}(A_0) + 8\phi_{\mathrm{OPT}}(A\setminus A_0)\Big)(1 + \ln k). \end{aligned}$$

Then the approximation ratio immediately follows from Theorem 1 combining with the fact that $H_{k-1} \leqslant 1 + \ln k$.

4.3 Counterexample

In this subsection, we depict a counterexample to show the importance of using the mass center as the representative point (as the strategy of our algorithm). Figure 1 demonstrates two examples carried out on the same dataset, where p is an unconstrained point and $X = \{x_1, x_2\}$ is an ML set, of which x_2 is a data point with the weight of $n-2$. For the parameter $k = 2$ and $\{c_1\}$ as the current center set, $C = \{c_1, c_2\}$ is then produced by Algorithm 1 for the constrained k-means problem (as depicted in Fig. 1a). In contrast,



(a) Solution of Algorithm 1 with a cost of approximately $\|x_1 - c_2\|_2^2 + (n-2)\|c_2 - x_2\|_2^2$

(b) Output of Algorithm 1 without Steps 4–6 that has a cost of approximately $(n-2)\|x_1 - x_2\|_2^2$ with high possibility

Fig. 1 Counterexample for the ratio of Algorithm 1 that selects a point of the ML set (instead of its mass center) as a center. The red points (i.e., c_1 and c_2) are the selected centers.

for Algorithm 1 but without using mass center as in Steps 4–6, $C = \{p, x_1\}$ is a produced solution with a high probability as shown in Fig. 1b. Provided $\{c_1\}$ as the current center set, the probability of selecting x_1 as a new center is as below:

$$\frac{\|x_1 - c_1\|_2^2}{(n-2)\|x_2 - c_1\|_2^2 + \|x_1 - c_1\|_2^2}.$$

Note that if $(n-2)\|x_2 - c_1\|_2^2 \ll \|x_1 - c_1\|_2^2$ holds, $(n-2)\|x_2 - c_1\|_2^2 = 0.01 \times \|x_1 - c_1\|_2^2$, then the above probability can be relatively high. i.e.,

$$\frac{\|x_1 - c_1\|_2^2}{(n-2)\|x_2 - c_1\|_2^2 + \|x_1 - c_1\|_2^2} \geqslant 0.99 \qquad (5)$$

In that case, we have the set of centers as $C = \{p, x_1\}$ in Fig. 1b at the probability 0.99, whose cost is

$$\frac{\|x_1 - c_1\|_2^2}{(n-2)\|x_2 - c_1\|_2^2 + \|x_1 - c_1\|_2^2}(n-2)\|x_1 - x_2\|_2^2 + \frac{(n-2)\|x_2 - c_1\|_2^2}{(n-2)\|x_2 - c_1\|_2^2 + \|x_1 - c_1\|_2^2}\|x_1 - x_2\|_2^2 \geqslant 0.99 \times (n-2)\|x_1 - x_2\|_2^2,$$

where the inequality holds due to Formula (5). That is, when $\|x_1 - x_2\|_2^2$ dominates $\|c_2 - x_2\|_2^2$ (c_2 in Fig. 1a), the cost of the solution output by our algorithm is less than $\|x_1 - x_2\|_2^2$, and hence the solution of using x_1 instead of the mass center consumes at least $0.99 \times (n-2)$ times of our cost. Therefore, it is essential to use the mass center instead of immediately using the discrete data points of P as centers in Algorithm 1.

5 Grand Algorithm for Constrained k-Means with Convergence

In the section, we propose an iterative phase to fast decrease ψ of the k-means problem with DCL and ML constraints. Observing that the conventional k-means algorithm is convergent, the key idea of our algorithm is to deal with the constrained points, which is shown in Algorithm 2. To reassign the constrained points, we divide them into two cases: DCL constraints and ML constraints. For DCL constraints, we use min-sum matching[22] to assign the data point of each DCL set to the current centers at minimum cost. Following the key idea of Algorithm 1, we consider an ML set to be a "big point" and then assign the data points following the rule designated for unconstrained points or DCL sets (e.g., the data point in this ML set is also included in

Algorithm 2 Assignment step of iterative phase

```
Input: Database $P$ of size $n$ with CL sets $\mathcal{Y}$ and ML sets $\mathcal{X}$
       and a set of center $C^t$.
Output: Set of clusters $\mathcal{A}^t$.
Set $\mathcal{A}^t \leftarrow \{A_i^t = \varnothing \mid i = 1, 2, \ldots, k\}$;
/*$\mathcal{A}^t$ is $k$ clusters that partition the data points of $P$ regarding
$C^t$. */
for each $p \in P$ do
    Assign $p$ to its nearest center $c_i^t$;
    $A_i^t \leftarrow p$;
end
for each $X \in \mathcal{Y}$ do
    Let $\bar{X}$ as the mass center for $X$ with $|X|$ weight on
    behalf of these data points to assign to the its nearest center
    $c_i^t$ of $\bar{X}$;
    $A_i^t \leftarrow X$;
end
for each $Y \in \mathcal{Y}$ do
    for each $q \in X \cap Y$ do
        Compute $\bar{X}$ that is the mass centers of $X$ and is with $X$
        weight, and the use $\bar{X}$ to replace $q \in Y$;
    end
    Compute the corresponding center $c(p)$ within $C^t$ for every
    data point $p \in Y$ by the min-sum matching method[22],
    such that the total square distance is minimized,
    $\min \sum_{p \in Y} d^2(p, c(p))$;
    if $\exists q \in (X \cap Y)$ for some $X \in \mathcal{X}$ then
        Assign every data point in $X$ to the corresponding
        cluster regarding $q$;
    end
end
Return $\mathcal{A}^t$
```

DCL set). Then we update the new center set by the mass centers for the current k clusters. Repeating the two steps, the iteration terminates until the cost is less than a very small constant. The detailed algorithm is depicted in Algorithm 3.

Next, we shall show the convergence of the cost of our algorithm's clustering, which indicates that our algorithm always terminates.

Lemma 6 Algorithm 3 always terminates and the cost of the clustering is convergent to a local minimum.

Proof The key idea of the proof is to treat the iterative phase of Algorithm 3 following a similar line for the convergence analysis of traditional k-means. In essence, our proof incorporates the behavior of the algorithm against the constraints with the analysis process of Refs. [23, 24]. Assume that C^t is the center set of the t-th iteration, and $\mathcal{A}^t$ denotes the clustering for the center sets C^t.

By definition, the cost of the clustering for C^t is as follows:

$$cost(\mathcal{A}^t, C^t) = \sum_{\substack{A_i^t \in \mathcal{A}^t \\ c_i^t \in C^t}} \sum_{a \in A_i^t} \|a - c_i^t\|_2^2 \tag{6}$$

where the data point a is assigned against the clustering result of $\mathcal{A}^t$. Moreover, let $\overline{P}(\mathcal{X}, \mathcal{Y})$ denote the set of the unconstrained points. Then we use $cost(\mathcal{X}, \mathcal{A}^t, C^t)$, $cost(\mathcal{Y}, \mathcal{A}^t, C^t)$, and $cost(\overline{P}(\mathcal{X}, \mathcal{Y}), \mathcal{A}^t, C^t)$ to denote the costs regarding the points of all ML constraints $\mathcal{X}$, DCL constraints $\mathcal{Y}$, and the unconstrained points when assigning to C^t for $\mathcal{A}^t$, respectively.

Algorithm 3 Iterative phase for constrained *k*-means

```
   Input: Database $P$ of size $n$ with CL sets $\mathcal{Y}$ and ML sets $\mathcal{X}$,
          and a positive integer $k$.
   Output: Set of centers $C$ and set of clusters $\mathcal{A}$.
   // Initialization
1  Set $t \leftarrow 0$;
2  Compute $C^0 = \{c_1^0, c_2^0, \ldots, c_k^0\}$ by Algorithm 1 respecting
   $P$ and $k$;
3  Compute the clusters $\mathcal{A}^0 \leftarrow \{A_i^0 \mid i = 1, 2, \ldots, k\}$ by
   employing Algorithm 2 against $P$ and $C^0$;
   // re-centroid step
4  for each $A_i^t \in \mathcal{A}^t$ do
5      Update the center set $C^{t+1}$ by $c_i^{t+1} \leftarrow \frac{1}{|A_i^t|} \sum_{p \in A_i^t} p$;
6  end
   // re-assignment step
7  Update the assignment $\mathcal{A}^{t+1}$ of the dataset $P$ by
   Algorithm 2 respecting $C^{t+1}$;
8  if $cost(\mathcal{A}^t, C^t) - cost(\mathcal{A}^{t+1}, C^{t+1}) \geqslant 0$ then
9      Set $t \leftarrow t + 1$ and then go to Step 4;
10 end
11 Return $C^{t+1}$ and $\mathcal{A}^{t+1}$.
```

Following the procession of Algorithm 3, it terminates once $cost(\mathcal{A}^t, C^t) - cost(\mathcal{A}^{t+1}, C^{t+1}) < 0$. Then to prove the convergence, we only need to show the value of $cost(\mathcal{A}, C)$ strictly decreases during the iterations of Algorithm 3 (excepting the last iteration), provided that $cost(\mathcal{A}, C) \geqslant 0$ always holds. In fact, we will actually do the conversely equivalent task that shows $cost(\mathcal{A}, C)$ never increases in either the re-centroid step or the re-assignment step.

First, for the re-centroid step, as C^{t+1} is the set of mass centers for $\mathcal{A}^t$, we clearly have

$$cost(\mathcal{A}^t, C^{t+1}) \leqslant cost(\mathcal{A}^t, C^t) \tag{7}$$

Then for the re-assignment step, Algorithm 2 re-clusters the data points of P as $\mathcal{A}^{t+1}$ according to the set of centers C^{t+1}. We only need to show the distance sums of both unconstrained and constrained points are non-increasing. Since the unconstrained points are assigned to their nearest centers, we have

$$cost(\overline{P}(\mathcal{X}, \mathcal{Y}), \mathcal{A}^{t+1}, C^{t+1}) \leqslant$$
$$cost(\overline{P}(\mathcal{X}, \mathcal{Y}), \mathcal{A}^t, C^{t+1}).$$

Similarly, because each set of ML constraints $X \in \mathcal{X}$ with the same mass center is assigned to its nearest center, the cost sum regarding all ML sets satisfies

$$cost(\mathcal{X}, \mathcal{A}^{t+1}, C^{t+1}) \leqslant cost(\mathcal{X}, \mathcal{A}^t, C^{t+1}).$$

Lastly, as min-sum matching on each set of CL constraints $Y \in \mathcal{Y}$ attains the minimum cost, we have

$$cost(\mathcal{Y}, \mathcal{A}^{t+1}, C^{t+1}) \leqslant cost(\mathcal{Y}, \mathcal{A}^t, C^{t+1}).$$

Therefore, combining the above cases on the re-assignment step yields

$$cost(\mathcal{A}^{t+1}, C^{t+1}) \leqslant cost(\mathcal{A}^t, C^{t+1}) \tag{8}$$

Then, we have

$$cost(\mathcal{A}^{t+1}, C^{t+1}) - cost(\mathcal{A}^t, C^t) =$$
$$cost(\mathcal{A}^{t+1}, C^{t+1}) - cost(\mathcal{A}^t, C^{t+1}) +$$
$$cost(\mathcal{A}^t, C^{t+1}) - cost(\mathcal{A}^t, C^t).$$

Eventually, combining Formulas (7) and (8), we derive the following inequality:

$$cost(\mathcal{A}^{t+1}, C^{t+1}) - cost(\mathcal{A}^t, C^t) \leqslant 0,$$

where the equation is active when $cost(\mathcal{A}^t, C^t) - cost(\mathcal{A}^{t+1}, C^{t+1}) = 0$ holds, which immediately results in the termination of the algorithm. ■

6 Numerical Evaluation

In this section, we carry out Algorithm 1 to select k centers, where DCL-constrained points are treated as unconstrained. The selected centers are then input into Algorithm 3 that completes the cluster assignment of P.

The final solution from Algorithm 3 is compared with a baseline method in terms of the clustering accuracy.

6.1 Experimental setup

6.1.1 Baseline method

In a nutshell, the heuristic method from Ref. [13] works as follows:

(1) Select k centers from P uniformly at random;

(2) For each ML set X of $\mathcal{X}$, link the entire ML set to the nearest center of the first data point encountered/processed (in input order) in the ML set;

(3) For each CL set Y of $\mathcal{Y}$, assign data points (in input order) to their nearest centers. If a center is already used by another data point in the CL set, then connect the current data point to its second nearest center, and so on.

We name this method as the "greedy" baseline for ML and DCL constrained k-means.

6.1.2 Datasets

In our experiments, following the study of Ref. [13], we also use four real-world datasets from the UCI repository to evaluate the clustering accuracy of our proposed algorithm against the greedy baseline. A summary of the datasets is shown in Table 1. Detailed descriptions of these datasets can be found in Ref. [25]. For a fair comparison, we randomly shuffle items in each dataset so the input order to an algorithm is not biased.

6.1.3 Clustering constraints

The datasets described previously do not contain any constraints. In the following, we describe a simple way to generate the disjoint cannot-link and must-link constraints in real-world datasets with labels. We randomly sample r data points from a given label to produce each ML set. To produce each DCL set, we randomly select $r \leqslant k$ labels, and then randomly sample one data point from each of the r labels. Note that by construction, the established constraints are formulated as sets instead of the pairwise constrained used in Ref. [13]. For a DCL/ML set $\{p_1, p_2, p_3\}$, the constraints are the same as the three pairwise DCL/ML relationships $\{p_1, p_2\}$, $\{p_1, p_3\}$, and $\{p_2, p_3\}$. In addition, we need to take into account the implicit transitivity of ML sets such as each ML set can intersect with at most one DCL set, as otherwise, DCL sets will not hold disjointness. For simplicity, when sampling a DCL set, we regard each constructed ML set as a merged single point and the sampling is done without replacement.

Table 1 Datasets summary.

Dataset	Number of instances	Number of dimensions	k
Soybean	43	35	4
Iris	150	4	3
Wine	178	13	3
Tic-Tac-Toe	958	9	2

6.1.4 Evaluation metrics

For measuring the clustering accuracy on real-world datasets, we again follow the previous study in Ref. [13] to use the Rand Index (RI)[26]. The metric is to calculate the agreement degree between an algorithm's clustering result and the ground truth labeled clusters by treating the instances with the same label as belonging to the same cluster. For a dataset D with n data points, there can be at most $n \times (n-1)/2$ pairs of matching relations on any two nodes belonging to the same cluster or not. Provided we have two clustering/partition results R_1 and R_2 over D with one possibly being the ground truth partition. Let α be the total number of pairs of data points that belong to two different clusters in both R_1 and R_2, and β belonging to the same cluster in both R_1 and R_2. The RI measuring the total agreement can then be calculated as

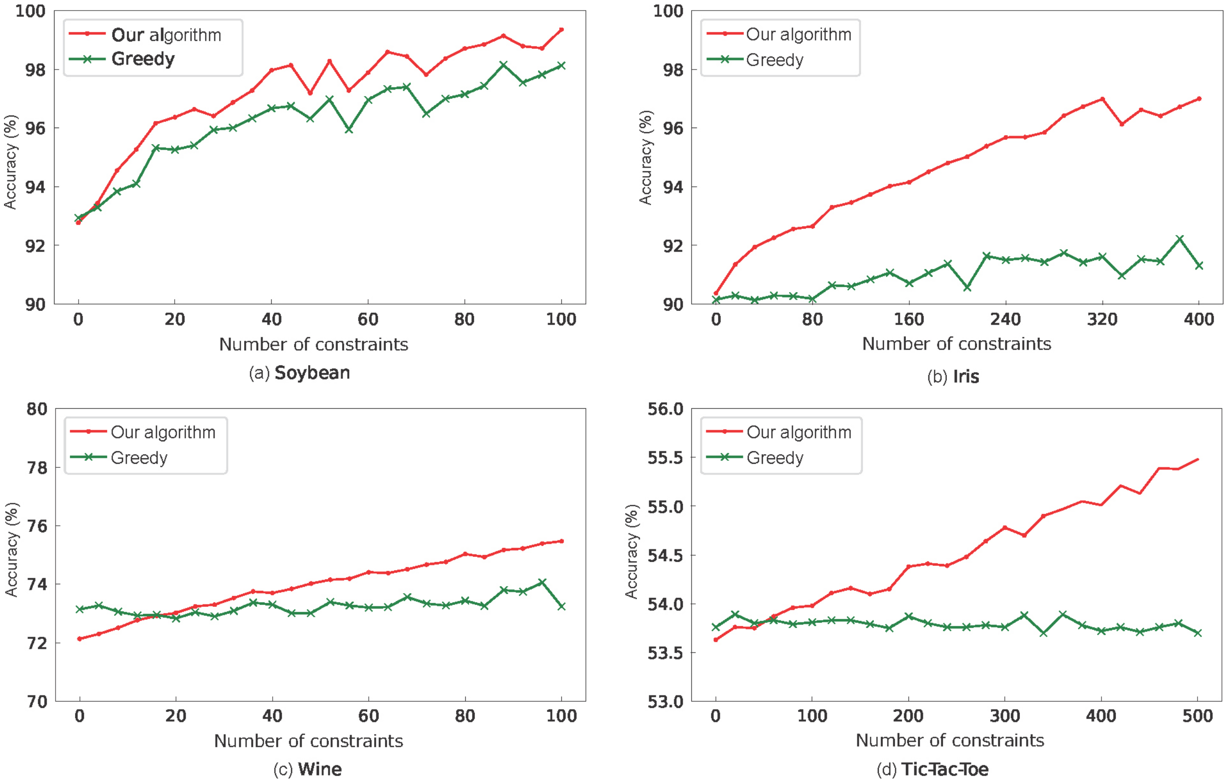
$$\mathrm{RI}(R_1, R_2) = \frac{\alpha + \beta}{n \times (n-1)/2}.$$

6.2 Experimental results and analysis

In this subsection, we report the experimental results on the four real-world datasets (Soybean, Iris, Wine, and Tic-Tac-Toe) as depicted in Fig. 2. For each respective dataset, k is set to the number of labeled classes. Each subfigure reports the clustering accuracy for a single dataset (with the dataset name labeled in a yellow box) against an increasing number of pairwise DCL and ML constraints that are equally portioned. It is intuitive that adding more constraints or background knowledge can lead to a higher accuracy as in the case of semi-supervised learning. The settings on the number of constraints (i.e., the values on the x-axises of Fig. 2) are aligned the same as that in Ref. [13]. To mitigate the bias introduced by random sampling of the constraints, from each dataset, 100 instances incorporating the same number of different random constraints are produced. In addition, each algorithm gets to run 100 times to mitigate the bias introduced from different random k-means initialization. Therefore, every obtained accuracy value is calculated by averaging the scores of RI across different dataset instances and runs of algorithms.

6.2.1 Clustering accuracy

Overall speaking, although with some fluctuations, all four subfigures of Fig. 2 display a consistent trend in



(a) **Soybean**

(b) **Iris**

(c) **Wine**

(d) **Tic-Tac-Toe**

Fig. 2 Clustering accuracy of our algorithm vs. greedy across on four UCI datasets.

the growth of accuracy along with the increase in the number of clustering constraints. In particular, the largest gain is on the Iris dataset, where our algorithm attains an accuracy of 97.0% (when number of constraints = 400). This is in contrast with the highest accuracy rate of 92.2% of the greedy algorithm. While on the Soybean dataset, greedy and our algorithm have a similar growth trend, which demonstrates the impact of background knowledge on both algorithms. Comparing the improvement of accuracy on the Tic-Tac-Toe and Wine datasets, the accuracy of the greedy algorithm only fluctuates without much change whereas our algorithm's accuracy grows steadily. We consider the reason for this phenomenon is that the greedy algorithm depends too much on the input order of data. Moreover, different from other three datasets, the Tic-Tac-Toe dataset seems difficult to cluster correctly with our algorithm only achieving an accuracy of about 55% and greedy around 53.5%. We reason that some real-world datasets are normally used for supervised classification tasks (with labels), so the optimal clustering does not necessarily map to the optimal classification. In theory, the different performance between our algorithm and the greedy baseline is mainly due to more optimized/cost-effective ways of center selection and cluster assignment.

Overall, our algorithm consistently outperforms the baseline method in clustering accuracy as the number of constraints grows. This leads to the conclusion that our empirical results reflect the theoretical proofs and yield the expected results.

6.2.2 Runtime

We also compare the average runtime across the different number of constraints on the four real-world datasets. The detailed comparison is shown in Table 2, despite the significant improvement in clustering accuracy as shown in Fig. 2, we can see that the practical runtime of our algorithm is still comparable or not much inferior to that of the greedy method. The runtime increase is mainly due to the matching method that we adopt to effectively deal with DCL sets.

7 Conclusion

In this paper, we constructed an algorithm for k-means with disjoint cannot-link and must-link constraints, where the constraints correspond to background information that can be used to improve the accuracy of clustering. For initialization of the algorithm, we devised an approximation algorithm for the k-means problem with ML constraints and showed that the algorithm deserves an approximation

Table 2 Runtime for the greedy & our algorithm on four UCI datasets.

($\times 10^{-3}$ ms)

	Algorithm	Number of constraints					
Soybean		0	20	40	60	80	100
	Greedy	866.59	857.25	871.05	860.39	865.94	854.57
	Our algorithm	877.62	1076.90	1064.58	1104.16	1114.40	1210.11
	Algorithm	Number of constraints					
Wine		0	20	40	60	80	100
	Greedy	1258.31	1290.80	1295.54	1279.67	1322.70	1301.08
	Our algorithm	1258.02	1511.00	1612.88	1878.56	1807.56	1886.52
	Algorithm	Number of constraints					
Iris		0	80	160	240	320	400
	Greedy	402.96	445.21	476.77	498.59	525.45	502.06
	Our algorithm	240.90	743.84	944.50	1171.33	1481.17	1191.72
	Algorithm	Number of constraints					
Tic-Tac-Toe		0	100	200	300	400	500
	Greedy	3306.44	2992.15	2872.69	2867.99	2886.96	2952.98
	Our algorithm	3335.62	3680.43	4093.57	4680.94	5248.77	5924.88

ratio of $O(\log k)$ by mathematical induction. Then we proved the convergence of the iterative phase of the algorithm considering both DCL and ML constraints. Lastly, we carried out experiments on several prevalent real-world datasets and demonstrated that our approximation algorithm can achieve a significantly improved clustering accuracy, which validates our theoretical findings.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 12271098 and 61772005) and the Outstanding Youth Innovation Team Project for Universities of Shandong Province (No. 2020KJN008).

References

[1] X. Li and H. Liu, Greedy optimization for K-means-based consensus clustering, *Tsinghua Science and Technology*, vol. 23, no. 2, pp. 184–194, 2018.

[2] S. Ji, D. Xu, L. Guo, M. Li, and D. Zhang, The seeding algorithm for spherical K-means clustering with penalties, in *Proc. 13th Int. Conf. Algorithmic Aspects in Information and Management*, Beijing, China, 2019, pp. 149–158.

[3] S. Har-Peled and S. Mazumdar, On coresets for k-means and k-median clustering, in *Proc. 36th Annu. ACM Symp. Theory of Computing*, Chicago, IL, USA, 2004, pp. 291–300.

[4] Z. Lu and H. Shen, Differentially private k-means clustering with convergence guarantee, *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1541–1552, 2021.

[5] S. Dasgupta, The Hardness of k-Means Clustering, Report, Department of Computer Science and Engineering University of California, https://cseweb.ucsd.edu/∼dasgupta/papers/kmeans.pdf, 2008.

[6] A. Vattani, The hardness of k-means clustering in the plane, https://cseweb.ucsd.edu/∼avattani/papers/kmeans_hardness.pdf, 2009.

[7] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, The planar k-means problem is NP-hard, in *Proc. 3rd Int. Workshop on Algorithms and Computation*, Kolkata, India, 2009, pp. 274–285.

[8] D. Arthur and S. Vassilvitskii, k-means++: The advantages of careful seeding, in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, New Orleans, LA, USA, 2007, pp. 1027–1035.

[9] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, A local search approximation algorithm for k-means clustering, in *Proc. 18th Annu. Symp. Computational Geometry*, Barcelona, Spain, 2002, pp. 10–18.

[10] S. Lattanzi and C. Sohler, A better k-means++ algorithm via local search, in *Proc. 36th Int. Conf. Machine Learning*, Long Beach, CA, USA, 2019, pp. 3662–3671.

[11] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward, Better guarantees for k-means and Euclidean k-median by primal-dual algorithms, in *Proc. of the 2017 IEEE 58th Annu. Symp. Foundations of Computer Science* (*FOCS*), Berkeley, CA, USA, 2017, pp. 61–72.

[12] K. Wagstaff and C. Cardie, Clustering with instance-level constraints, in *Proc. 17th Int. Conf. Machine Learning*, San Francisco, CA, USA, 2000, pp. 1103–1110.

[13] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, Constrained k-means clustering with background knowledge, in *Proc. 18th Int. Conf. Machine Learning*, San Francisco, CA, USA, 2001, pp. 577–584.

[14] T. Lange, M. H. C. Law, A. K. Jain, and J. M. Buhmann, Learning with constrained and unlabelled data, in *Proc. of 2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition* (*CVPR'05*), San Diego, CA, USA, 2005, pp. 731–738.

[15] S. Basu, I. Davidson, and K. Wagstaff, *Constrained*

Clustering: Advances in Algorithms, Theory, and Applications. Boca Raton, FL, USA: CRC Press, 2008.

[16] Z. Li, J. Liu, and X. Tang, Constrained clustering via spectral regularization, in *Proc. of 2009 IEEE Conf. Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 421–428.

[17] X. Zhai, Y. Peng, and J. Xiao, Cross-modality correlation propagation for cross-media retrieval, in *Proc. of 2012 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), Kyoto, Japan, 2012, pp. 2337–2340.

[18] H. Ding and J. Xu, A unified framework for clustering constrained data without locality property, in *Proc. 26th Annu. ACM-SIAM Symp. Discrete Algorithms*, San Diego, CA, USA, 2015, pp. 1471–1490.

[19] Q. Feng, J. Hu, N. Huang, and J. Wang, Improved PTAS for the constrained k-means problem, *J. Comb. Optim.*, vol. 37, no. 4, pp. 1091–1110, 2019.

[20] I. Davidson and S. S. Ravi, Clustering with constraints: Feasibility issues and the k-means algorithm, in *Proc. 2005 SIAM Int. Conf. Data Mining*, Newport Beach, CA, USA, 2005, pp. 138–149.

[21] P. Baumann, A binary linear programming-based k-means algorithm for clustering with must-link and cannot-link constraints, in *Proc. of 2020 IEEE Int. Conf. Industrial Engineering and Engineering Management* (*IEEM*), Singapore, 2020, pp. 324–328.

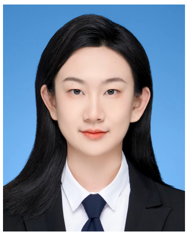
[22] J. Edmonds, Maximum matching and a polyhedron with 0,1-vertices, *Journal of Research of the National Bureau of Standards B*, vol. 69, nos. 1&2, pp. 125–130, 1965.

[23] J. B. MacQuuen, Some methods for classification and analysis of multivariate observations, in *Proc. 5th Berkley Symp. Mathematical Statistics and Probability*, Berkeley, CA, USA, 1967, pp. 281–297.

[24] L. Bottou and Y. Bengio, Convergence properties of the k-means algorithms, in *Proc. 7th Int. Conf. Neural Information Processing Systems*, Denver, CO, USA, 1994, pp. 585–592.

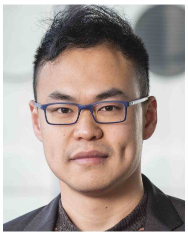
[25] M. Lichman, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2013.

[26] W. M. Rand, Objective criteria for the evaluation of clustering methods, *J. Am. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.

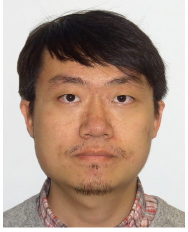


Chaoqi Jia received the BEng degree in computer science from Beijing Information Science and Technology University, China in 2019. She is currently a master student at the Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences, China). Her research interests include machine learning and algorithm design.

Longkun Guo received the BEng and PhD degrees in computer science from University of Science and Technology of China (USTC), China in 2005 and 2011, respectively. He joined Fuzhou University in 2011 and became a professor there. He is currently a full professor at the Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences, China). His research interests include efficient algorithm design and computational complexity analysis, particularly for optimization problems in high performance computing systems and networks, VLSI, etc. He has published more than 100 academic papers in reputable journals/conferences, such as *IEEE TMC*, *IEEE TC*, *Algorithmica*, *IEEE TPDS*, IEEE ICDCS, IJCAI, and ACM SPAA.



Kewen Liao received the PhD degree in computer science from The University of Adelaide, Australia in 2014. He is currently a senior lecturer in information technology and the director of the HilstLab, Australian Catholic University. He was a postdoctoral researcher at The University of Melbourne and Swinburne University of Technology. He has published at premier venues of ICDE, WSDM, WWW, CHI, IJCAI, CIKM, SPAA, ICSOC, etc. His research interests include approximation algorithms, data science, and machine learning.



Zhigang Lu is a postdoctoral researcher at Macquarie University Cyber Security Hub, Macquarie University, Australia. Prior to that, he received the PhD degree in 2020 and the MPhil degree in 2015 both from The University of Adelaide Australia, and the BEng degree from Xidian University, China in 2011. His papers were published in top-tier journals and conferences, such as *IEEE TIFS*, *IEEE TDSC*, and ACM CCS. His research interests include differential privacy, machine learning, and information theory.