Efficient Constrained *K*-center Clustering with Background Knowledge

Longkun Guo[1, 2*], **Chaoqi Jia**[2*], **Kewen Liao**[3*], **Zhigang Lu**[4*], **Minhui Xue**[5*]

[1]School of Mathematics and Statistics, Fuzhou University, Fuzhou 350116, China
[2]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250316, China
[3]HilstLab, Peter Faber Business School, Australian Catholic University, North Sydney 2060, Australia
[4]College of Science and Engineering, James Cook University, Townsville 4810, Australia
[5]CSIRO's Data61, Sydney 2015, Australia
{longkun.guo, chaoqi1.jia}@gmail.com, kewen.liao@acu.edu.au, zhigang.lu@jcu.edu.au, jason.xue@data61.csiro.au

Abstract

Center-based clustering has attracted significant research interest from both theory and practice. In many practical applications, input data often contain background knowledge that can be used to improve clustering results. In this work, we build on widely adopted *k*-center clustering and model its input background knowledge as must-link (ML) and cannot-link (CL) constraint sets. However, most clustering problems including *k*-center are inherently NP-hard, while the more complex constrained variants are known to suffer severer approximation and computation barriers that significantly limit their applicability. By employing a suite of techniques including reverse dominating sets, linear programming (LP) integral polyhedron, and LP duality, we arrive at the first efficient approximation algorithm for constrained *k*-center with the best possible ratio of 2. We also construct competitive baseline algorithms and empirically evaluate our approximation algorithm against them on a variety of real datasets. The results validate our theoretical findings and demonstrate the great advantages of our algorithm in terms of clustering cost, clustering quality, and running time.

Introduction

Center-based clustering is a fundamental unsupervised task in machine learning that aims to group similar data points into subsets or clusters based on different distance metrics. Classical clustering problems such as k-means (Lloyd 1982), k-center (Gonzalez 1985; Hochbaum and Shmoys 1985), and k-median (Charikar et al. 1999) are all $\mathcal{NP}$-hard. Among those, the k-center formulation stands out in the perspectives of robustness to outliers, guaranteed low approximation ratio, and computational efficiency and scalability. In addition, unlike k-means, k-center produces deterministic clustering results and seeks distance-balanced clusters. Its objective is to minimize the maximum distance between any data point and its closest cluster center, or equivalently, minimize the maximum covering *radius* of the clusters. Extended from this basic k-center objective, there has been significant research interest in devising k-center based applications and their associated optimization models, including but not limited to capacitated k-center (Khuller and Sussmann 2000), k-center with outliers (Charikar et al. 2001), minimum coverage k-center (Lim et al. 2005), connected k-center (Ester et al. 2006), fault-tolerant k-center (Khuller, Pless, and Sussmann 2000), and recently fair k-center (Chierichetti et al. 2017; Kleindessner, Awasthi, and Morgenstern 2019; Bera et al. 2022) and distributed k-center (Huang et al. 2023).

In various machine learning applications (Zhang et al. 2013; Liu, Li, and Li 2017), there is often an abundance of unlabeled data and limited labeled data due to the cost of labeling. Even worse, the ground truth information can be completely missing or hidden (Basu, Banerjee, and Mooney 2004), while we may obtain some background knowledge among the data points to be clustered, that is, whether pairs of data points should belong to the same cluster or not. Unsurprisingly, utilizing such background knowledge achieves improved results on center-based clustering (Basu, Davidson, and Wagstaff 2008). However, this requires more complex clustering models with instance-level must-link (ML) and cannot-link (CL) constraints to encode the auxiliary input background knowledge. Clustering with instance-level constraints was first introduced in (Wagstaff and Cardie 2000), where pairwise ML and CL constraints between data instances were considered. Instances of data points with an ML constraint must belong to the same cluster, while with a CL constraint must be placed into different clusters.

Problem Formulation

Formally, given an input dataset $P = \{p_1, \ldots, p_n\}$ in the metric space, k-center aims to locate a cluster center set $C \subseteq P$ such that the maximum distance between points in P and their assigned cluster centers in C is minimized. Furthermore, if we denote the distance function between any two data points as $d(\cdot, \cdot)$, then the distance between a point $p \in P$ and a center set C can be defined as $d(p, C) = \min_{c \in C} d(p, c)$. Hence, the objective of k-center is to seek an optimal C^* with min-max radius or solution cost of $r^* = \max_{p \in P} d(p, C^*)$, i.e., to seek

$$C^* = \arg\min_{C \subseteq P} \max_{p \in P} d(p, C).$$

[*] These authors contributed equally.

Built on the k-center's min-max optimization objective, our considered constrained k-center adds predefined ML and CL constraints over a subset of input data points as the background knowledge. Without loss of generality, we adopt an equivalent set formulation of ML and CL constraints, instead of the usual setting of pairwise instance-level constraints (Wagstaff et al. 2001). Specifically, we treat the ML constraints as a family of point sets $\mathcal{X} = \{X_1, \ldots, X_h\}$ where every $X_i \subseteq P$ is a maximal ML set of points bounded together by mutual ML sets. Similarly, the CL constraints are treated as $\mathcal{Y} = \{Y_1, \ldots, Y_l\}$ with every $Y_i \subseteq P$ and $|Y_i| \leq k$ comprised of mutual CL sets. Now let $\sigma(p)$ denote the cluster center of a point p is assigned to in the clustering result, then the ML and CL constraints are essentially the clustering conditions to be met on $\sigma(\cdot)$ such that each $X \in \mathcal{X}$ satisfies $\forall(p,q) \in X$ iff $\sigma(p) = \sigma(q)$ and each $Y \in \mathcal{Y}$ satisfies $\forall(p,q) \in Y$ iff $\sigma(p) \neq \sigma(q)$. In addition, ML sets in $\mathcal{X}$ are mutually disjoint by construction since otherwise any two intersected ML sets can be merged into a single ML set due to the transitivity of ML sets.

Whereas the more interesting CL sets in $\mathcal{Y}$ can have arbitrary intersection or disjointness, for a reduced mutually disjoint setting $\mathcal{Y}$, we have $Y_i \cap Y_j = \emptyset$ for $\forall Y_i, Y_j \in \mathcal{Y}$. Note that intersected CL constraints can be reduced to disjoint via a process of (i) equivalent data points reduction/merging; or (ii) intersected data points removal. For instance, w.r.t. (i), if there are two singular CL sets $\{u, v\}$ and $\{v, w\}$ and a singular ML set $\{u, w\}$, then the CL sets can be reduced into a new CL set $\{v, z\}$ with z being a merged point of u and w. For (ii), given intersected example sets $\{u, v, w\}$ and $\{w, x, y\}$, they can be reduced to $\{u, v, w\}$ and $\{x, y\}$ or $\{u, v\}$ and $\{w, x, y\}$ via a pre-processing step. Apparently, such a removal process would ignore a portion of background knowledge. Nevertheless, as shown in the sequel of the paper, designing and analyzing our algorithm in a guided disjoint setting dramatically alleviates the problem's computational complexity, while imposing minimal impact on solving the arbitrary intersected case.

Challenge and Motivation

The biggest challenge faced when adopting constrained clustering with background knowledge is the computational complexity of the problem. As noted, most clustering problems are inherently $\mathcal{NP}$-hard, despite that there usually exist either efficient heuristic algorithms with no performance guarantee or approximation algorithms with non-practical performance ratio and high runtime complexity. Moreover, clustering models with instance-level ML and CL constraints (Wagstaff et al. 2001) leads to the following severe approximation and computation barriers (introduced by the CL constraints) that significantly limit their use.

Theorem 1. *(Davidson and Ravi 2007) It is $\mathcal{NP}$-complete even only to determine whether an instance of the CL-constrained clustering problem is feasible.*

Overcoming theoretical barriers. Arbitrarily intersected CL constraints were known to be problematic to clustering as their inclusion leads to a computationally intractable feasibility problem as stated in the above theorem. That means, under the assumption of $\mathcal{P} \neq \mathcal{NP}$, it is impossible to devise an efficient polynomial time algorithm to even determine, for an instance of the CL-constrained clustering problem (e.g., CL-constrained k-center or k-means), whether there exists a clustering solution satisfying all arbitrary CL constraints irrespective of the optimization objective. This inapproximability result can be obtained via a reduction from the k-coloring problem and, we believe, has hindered the development of efficient approximation algorithms for constrained clustering despite their many useful applications (Basu, Davidson, and Wagstaff 2008). For instance, for the closely related constrained k-means problem with both ML and CL constraints, only heuristic algorithms (Wagstaff et al. 2001; Davidson and Ravit 2005) without performance guarantee are known. Therefore, a strong motivation for us is to algorithmically overcome the long-standing theoretical barriers on constrained k-center that have been prohibiting its wide adoption (like k-center) in practice. We also hope that our proposed techniques can inspire novel solutions to other more intricate problems like constrained k-means.

Enhancing practical applications. The reward of our theoretical breakthrough will be the enhanced capability for utilizing additional information or domain knowledge (i.e., in the form of ML and CL constraints) in a range of practical applications (Basu, Davidson, and Wagstaff 2008) such as GPS lane finding, text mining, interactive visual clustering, distance metric learning, privacy-preserving data publishing, and video object classification. Among these applications, clustering results can be further used to infer data classification, where both ML and CL constraints were shown to improve classification performance.

Results and Contribution

In this paper, we gradually develop an efficient solution to constrained k-center clustering with guaranteed performance both in theory and practice. Inspired by the heuristic algorithm (Wagstaff et al. 2001) for constrained k-means, we started by adapting it to construct optimized heuristic algorithms for constrained k-center, namely Greedy and Matching, both aiming at reducing cluster connection costs. We use these algorithms as competitive baselines in our experiments. Then we focus on choosing the optimal cluster centers, which leads to our main contribution of an efficient approximation algorithm for solving ML/CL-constrained k-center problem.

Specifically, assuming the optimal radius is known, our algorithm leverages the structure of disjoint CL sets and then the construction of a Reverse Dominating Set (RDS) to result in a constant factor approximation ratio of 2 that is the best possible. For an efficient computation of RDS, we first propose a Linear Programming (LP)-based formulation and show the equivalence between the LP solution and RDS via the integral polyhedron. Instead of directly solving the LP for RDS using slower industry solvers, we devise a fast primal-dual algorithm that exploits LP duality to achieve time $O(k^{2.5})$. Finally, dropping the assumption of knowing the optimal radius, the algorithm can be recovered from combining with binary searched radius-thresholding, which eventually consumes a total time of $O(nk^{3.5} \log n)$.

Extensive experiments are carried out on a variety of real-world datasets to demonstrate the clustering effectiveness and efficiency of our proposed approximation algorithm with theoretical guarantees.

Other Related Work

Constrained clustering. Instance-level or pairwise ML and CL constraints have been widely adopted in clustering problems such as k-means clustering (Wagstaff et al. 2001; Jia et al. 2023), spectral clustering (Coleman, Saunderson, and Wirth 2008), and hierarchical clustering (Davidson and Ravi 2009). Basu et al. (Basu, Davidson, and Wagstaff 2008) have collated an extensive list of constrained clustering problems and applications. As confirmed by (Xing et al. 2002; Wagstaff et al. 2001), instance-level constraints are beneficial for improving the clustering quality.

Constrained k-center. Several studies have included k-center clustering with instance-level constraints in rather limited settings. Davidson et al. (Davidson, Ravi, and Shamis 2010) were the pioneers to consider constrained k-center when k is 2. They incorporated an SAT-based framework to obtain an approximation scheme with $(1 + \epsilon)$-approximation for this extreme case. Brubach et al. (Brubach et al. 2021) studied k-center only with ML constraints and achieved an approximation ratio 2. In contrast, we (including our constructed baseline methods) neither consider a limited special case (i.e., with a very small cluster number k) nor only the much simpler ML constraints.

Algorithm for CL-Constrained k-Center

Algorithm overview. In this section, we propose a threshold-based algorithm for CL k-center and show it deserves an approximation ratio of 2. The key idea of our algorithm is to incrementally expand a set of centers while arguably ensuring that each center is in a distinct cluster of the optimal solution. In the following, we first introduce a structure called *reverse dominating set* (RDS) and propose an algorithm using the structure to grow the desired center set; then we propose a linear programming (LP) relaxation and use it to find a maximum RDS. Moreover, we accelerate the algorithm by devising a faster LP primal-dual algorithm for finding the maximum RDS.

For briefness, we first assume that the optimal radius (r^*) for the constrained k-center problem is already known and utilize it as the threshold. Aligns with previous studies on threshold-based algorithms (Badanidiyuru et al. 2014), we discuss the problem with both ML and CL constraints without knowing r^* in the next section.

Reverse Dominating Set and the Algorithm

To facilitate our description, we introduce the following auxiliary bipartite graph, denoted as $G(Y, C; E)$, which allows us to represent the relationships between a CL set Y and C.

Definition 2. *The auxiliary bipartite graph $G(Y, C; E)$ regarding the threshold η is a graph with vertex sets Y and C, where the edge set E is as follows: an edge $e(y, z)$ is included in E iff the metric distance $d(y, z)$ between $y \in Y$ and $z \in C$ is bounded by η, i.e., $d(y, z) \leq \eta$.*

Algorithm 1: Approximating CL k-center via RDS.

```
Input: A family of $l$ disjoint CL sets $\mathcal{Y}$, a positive integer
       $k$, and a distance bound $\eta = 2r^*$.
Output: A set of centers $C$.
Initialization: Set $C \leftarrow \emptyset$, $C' \leftarrow \emptyset$ and $Y' \leftarrow \emptyset$;
while true do
    for each $Y \in \mathcal{Y}$ do
        Select a CL set $Y \in \mathcal{Y}$ and construct an auxiliary
          graph $G(Y, C; E)$ regarding $\eta$ according to
          Def. 2;
        if $G(Y, C; E)$ contains an RDS $(Y', C')$ then
            Update the center set using a maximum RDS:
              $C \leftarrow C \cup Y' \setminus C'$;
        end
    end
    if $G(Y, C; E)$, $\forall Y \in \mathcal{Y}$, contains no RDS then
        Return $C$.
    end
end
```

Based on the above auxiliary graph, we define *reverse dominating set* (RDS) as below:

Definition 3. *For a center set C, a CL set Y, and the auxiliary bipartite graph $G(Y, C; E)$ regarding $\eta = 2r^*$, we say (Y', C'), $Y' \subseteq Y$ and $C' = N(Y') \subseteq C$, is a reverse dominating set (RDS) iff $|C'| < |Y'|$, where $N(Y')$ denotes the set of neighboring points of Y' in G.*

In particular, for a point $y \in Y$ that is with distance $d(y, z) > 2r^*$ to any center $z \in C$ in the auxiliary graph $G(Y, C; E)$, $(\{y\}, \emptyset)$ is an RDS.

Note that RDS is a special case of Hall violator that it considers only the case $|C'| < |Y'|$. Anyhow, it obviously inherits the $\mathcal{NP}$-hardness of computing a minimum Hall violator (Cygan et al. 2015):

Lemma 4. *It is $\mathcal{NP}$-hard to compute an RDS with minimum cardinality.*

In comparison with the above-mentioned $\mathcal{NP}$-hardness, we discover that a maximum RDS (i.e. an RDS with maximized $|Y'| - |C'|$) can be computed in polynomial time. Consequently, our algorithm proceeds in iterations, where in each iteration it computes a maximum RDS regarding the current center set C and a CL set $Y \in \mathcal{Y}$ (if there exists any), and uses the computed RDS to increase C towards a desired solution. The formal layout of our algorithm is as in Alg. 1. The correctness of Alg. 1 is as below:

Theorem 5. *Alg. 1 always outputs a center set C that satisfies the following two conditions: (1) all the CL constraints are satisfied; (2) the size of C is bounded by k.*

For proving Condition (1), we have the following property which is actually a restatement of Hall's marriage theorem:

Lemma 6. *(Hall 1935) There exists a perfect matching in $G(Y, C; E)$, iff there exists no RDS in $G(Y, C; E)$.*

Recall that Alg. 1 terminates once it is unable to find any RDS between the current C and any $Y \in \mathcal{Y}$. Then by the above lemma, this implies the existence of a perfect matching in each $G(Y, C; E)$ regarding the current C and each

$Y \in \mathcal{Y}$. Consequently, this ensures the satisfaction of each CL set in $\mathcal{Y}$. Lastly, for Condition (2), we have the following lemma, which immediately indicates $|C| \le k$:

Lemma 7. *Assume that $\mathcal{V}^* = \{V_1, \ldots, V_k\}$ is the set of clusters in an optimum solution. Let C be the output of Alg. 1. Then the centers of C appear in pairwise different clusters of $\mathcal{V}^*$.*

Efficient Computations of RDS

It remains to give a method to compute a maximum RDS. For the task, we propose a linear program (LP) relaxation and then show that any basic solution of the LP is integral. Let C be the set of current centers in an iteration and Y be a CL set. Then we can relax the task of finding a maximum RDS $(Y',\, C')$ (i.e. an RDS with maximized $|Y'| - |C'|$) as in the following linear program (LP (1)):

$$\begin{aligned} \max \quad & \sum_{y\in Y} y - \sum_{z\in C} z \\ s.t. \quad & y - z \le 0 && \forall e(y,\, z) \in E \\ & 0 \le y, z \le 1 && \forall y \in Y,\, z \in C \end{aligned}$$

Note that when we force $y, z \in \{0, 1\}$, the above LP (1) is an integer linear program that exactly models the task of finding a maximum RDS. Moreover, we observe that any basic solution of LP (1) is integral, as stated below:

Lemma 8. *In any feasible basic solution of LP (1), the values of every y and z must be integers.*

From the above lemma, we can obtain a maximum RDS by computing an optimal basic (fractional) solution to LP (1). It is worth noting that LPs can be efficiently solved in polynomial time using widely-used LP solvers like CPLEX (IBM 2022). Consequently, we can immediately achieve a polynomial time 2 approximation for CL k-center.

Next, we devise an LP primal-dual algorithm that accelerates the computation of RDS and consequently improves the theoretical runtime of Alg. 1 mainly incurred by solving the LP formulation. The dual of LP (1) can be easily obtained as in the following (LP (2)):

$$\begin{aligned} \min \quad & \sum_{y\in Y} \alpha_y + \sum_{z\in C} \beta_z \\ s.t. \quad & \alpha_y + \sum_{e\in\delta(y)} \gamma_e \ge 1 && \forall y \in Y \\ & \beta_z - \sum_{e\in\delta(z)} \gamma_e \ge -1 && \forall z \in C \\ & \alpha_y,\, \beta_z \ge 0 && \forall y \in Y,\, \forall z \in C \\ & \gamma_e \ge 0 && \forall e \in E \end{aligned}$$

Our algorithm first constructs a special feasible solution to the dual LP (2) based on a maximal matching in $G(Y, C; E)$, and then uses the dual solution to construct an RDS. To utilize the maximal matching, we need the relationship between the primal LP and its dual as in the following:

Algorithm 2: A fast algorithm for maximum RDS.

```
Input: $C$ and $Y$.
Output: An RDS $(Y',\, C')$.
Construct the auxiliary graph $G(Y, C; E)$ according to
  Def. 2 and set $C' = \emptyset$;
Find a maximal matching in $G$, say $M$, and set $\gamma_e = 1$ for
  each $e \in M$;
Set $Y' = Y \setminus M$, set $\alpha_y = 1$ for each $y \in Y'$, and set
  $\alpha_y = 0$ and $\beta_z = 0$ for every $y \in Y \setminus Y'$ and $z \in C$;
  // Construction of an initial solution
     to the dual LP (2).
while $C'$ does not equal $N_G(Y')$ do
    /* $N_G(Y')$ denotes the set of
       neighbors of $Y'$ in $G$.              */
    Set $C' \leftarrow C' \cup N_G(Y')$ and then
      $Y' \leftarrow Y' \cup N_M(C')$, where $N_M(C')$ is the set of
      neighbours of $C'$ in $M$;
end
if $|Y'| > |C'|$ then
    // Otherwise no RDS exists.
    Set $\alpha_y = 1$ for $\forall y \in Y'$, $\beta_z = -1$ for $\forall z \in C'$, and
      $\gamma_e = 0$ for edge $e$ adjacent to each $z \in C'$;
    // Update the initial solution to
       LP (2).
    Return $(Y', C')$ as the maximum RDS.
end
```

Lemma 9. *When there exists no RDS, we have*

$$\max\left\{\sum_{y\in Y} y - \sum_{z\in C} z\right\} = \min\left\{\sum_{y\in Y} \alpha_y + \sum_{z\in C} \beta_z\right\} = 0.$$

Moreover, when there exists no RDS, $\alpha_y = \beta_z = 0$ holds. Hence, the polyhedron of LP (2) becomes

$$\left\{\gamma \in [0,1]^{E(G)} : \sum_{e\in\delta(y)} \gamma_e \ge 1, y \in Y;\ \sum_{e\in\delta(z)} \gamma_e \le 1, z \in C\right\}$$

Notably, this is exactly the LP polyhedron for the maximal matching problem. So any feasible integral solution to the above polyhedron means a perfect matching. In other words, when $G(Y, C; E)$ contains no RDS, there must exist a perfect matching between Y and C in $G(Y, C; E)$, conforming to Lem. 6.

To construct an RDS in cases where a perfect matching does not exist in $G(Y, C; E)$, we employ a two-step approach: (1) utilize maximal matching to construct a feasible dual solution; (2) employ the dual solution to obtain the RDS, and meanwhile constructing a corresponding dual solution.

For the first, we demonstrate that an initial feasible solution of LP (2) can be easily constructed from a maximal matching M of $G(C, Y; E)$ as in the following: (1) For each $e \in M$, set $\gamma_e = 1$; (2) For each $y \in Y \setminus M$, set $\alpha_y = 1$; (3) Set $\alpha_y = 0$ and $\beta_z = 0$ for every other $y \in Y$ and $z \in C$. It can be easily verified that the above solution is feasible since all the constraints of LP (2) remain satisfied.

Secondly, according to the RDS returned by the algorithm, we update the initial feasible solution such that it corresponds to the RDS and remains a feasible solution to the dual LP (2). The updating simply proceeds as: set $\alpha_y = 1$ for each $y \in Y'$, $\beta_z = -1$ for each $z \in C'$, set $\gamma_e = 0$ for each edge adjacent to each $z \in C'$. It is easy to verify that such an updated solution satisfies the constraints of LP (2). The details of the algorithm are as depicted in Alg. 2.

Lemma 10. *During the iterations of Alg. 2, $N(Y') \subseteq M \cap C$ always holds. When it terminates, Alg. 2 produces an RDS (Y', C') with maximum $|Y'| - |C'|$.*

Lastly, recalling that Alg. 1 employs Alg. 2 to compute RDS, we have their runtimes as below:

Lemma 11. *Alg. 2 runs in time $O(k^{2.5})$, and consequently Alg. 1 runs in $O(nk^{3.5})$.*

The Whole Algorithm for ML/CL *k*-Center

We will firstly demonstrate that Alg. 1 can be extended to approximate k-center with both ML and CL constraints. Secondly, when the value of r^* is unknown, we will show the algorithm can be easily tuned by employing a binary search.

For the first, the key idea is to contract the ML sets into a set of "big" points. That is, for each X, we remove all the points from P that belong to X and replace them with a "big" point x. Then, for distances involving the points resulting from this contraction, we use the following definition:

Definition 12. *For two points x_i and x_j that result from contraction and correspond to X_i and X_j respectively, the refined distance between them is*

$$\hat{d}(x_i,\, x_j) = \hat{d}(X_i,\, X_j) = \max_{p \in X_i,\, q \in X_j} d(p, q).$$

Treating a single point as a singleton CL set, we can calculate the distance between x (representing X) and a point $q \notin X$ using Def. 12 as follows:

$$\hat{d}(x,\, q) = \hat{d}(X,\, q) = \max_{p \in X} d(p, q).$$

Then the distance between x and the center set C is

$$\hat{d}(x,\, C) = \hat{d}(X,\, C) = \max_{p \in X} d(p, C) = \max_{p \in X} \min_{q \in C} d(p, q).$$

By Def. 12, we can simply extend Alg. 1 to solve ML/CL k-center.

For the second, we show the same ratio can be achieved even without knowing r^* based on the following observation:

Lemma 13. *Let $\Psi = \{d(p_i, p_j) | p_i, p_j \in P\}$. Then we have $r^* \in \Psi$. In other words, the optimum radius r^* must be a distance between two points of P.*

That is, we need only to find the smallest $r \in \Psi$, such that regarding $2r$, Alg. 1 can successfully return C with $|C| \leq k$. By employing a binary search on the distances in Ψ, we can find in $O(\log n)$ iterations the smallest r under which Alg. 2 can find a feasible solution. Therefore, by combining the two aforementioned techniques, the whole algorithm for k-center with both ML and CL sets without known r^*, is depicted in Alg. 3. Eventually, we have runtime and performance guarantees for Alg. 3 as follows:

Algorithm 3: Whole algorithm for ML/CL k-center.

```
Input: Database $P$ of size $n$ with ML sets $\mathcal{X}$ and CL sets
       $\mathcal{Y}$ and a positive integer $k$.
Output: A set of centers $C$.
Initialization: Set $C \leftarrow \emptyset$, $C' \leftarrow \emptyset$, $Y' \leftarrow \emptyset$, and shrink
  each ML set $X \in \mathcal{X}$ as $x$;
Compute $\Psi = \{d(p_i, p_j) \mid p_i,\, p_j \in P\}$, the set of
  distances between each pair points of $P$;
while true do
    Assign the value of the median of $\Psi$ to $\eta$;
    for each $Y \in \mathcal{Y}$ do
        Construct an auxiliary graph $G(Y, C; E)$
          according to Def. 2 wrt $\eta$, and the distances
          concerning the shrunken point $x$ are computed
          according to Def. 12;
        if $G(Y, C; E)$ contains an RDS $(Y', C')$ then
            Update the center set using the RDS:
              $C \leftarrow C \cup Y' \setminus C'$;
        end
    end
    if $|C| > k$ then
        Remove each $d \leq \eta$ (except $\eta$) from $\Psi$;
    else
        Remove each $d \geq \eta$ (except $\eta$) from $\Psi$;
    end
    /* Note that $|C| \leq k$ indicates $\eta$ is
       sufficiently large while $|C| > k$
       for otherwise.                                  */
    if $|\Psi| = 1$ then
        Return $\eta$ together with the corresponding $C$.
    end
end
```

Datasets	#Rec.	#Dim.	k
Wine	178	13	3
Cnae-9	1,080	856	9
NLS-KDD	22,544	41	2
Skin	245,057	3	2
Wide09	570,223	21	13
Covertype	581,012	54	7
Simulated	10,000	50	5/10/50/100

Table 1: Datasets summary.

Theorem 14. *Alg. 3 solves the ML/CL k-center within runtime $O(nk^{3.5} \log n)$ and outputs a center set C, such that: (1) $d(p, \sigma(p)) \leq 2r^*$ holds for $\forall p \in P$ where $\sigma(p)$ is the center for p in C; (2) All the ML and CL constraints are satisfied; (3) $|C| \leq k$.*

Experimental Evaluation

Experimental Configurations

This section includes a brief description of the experimental configurations.

Real-world datasets. We follow existing studies on constrained clustering (Wagstaff et al. 2001; Malkomes et al. 2015) to use the four real-world datasets (Wine, Cnae-9, Skin and Covertype (Bache and Lichman 2013)) and

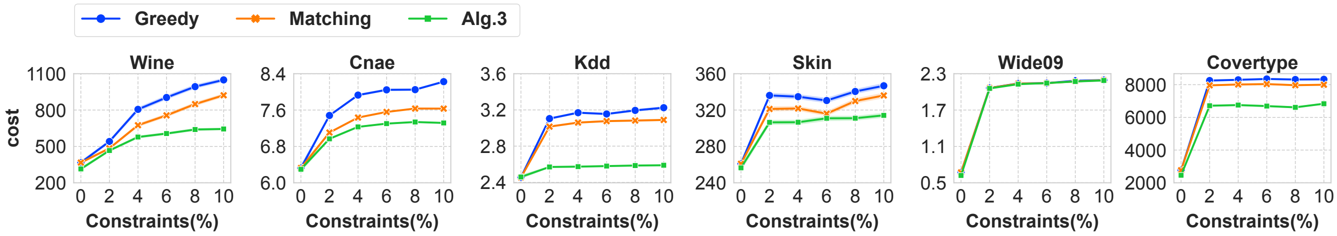

Figure 1: Cost (Disjoint ML/CL).

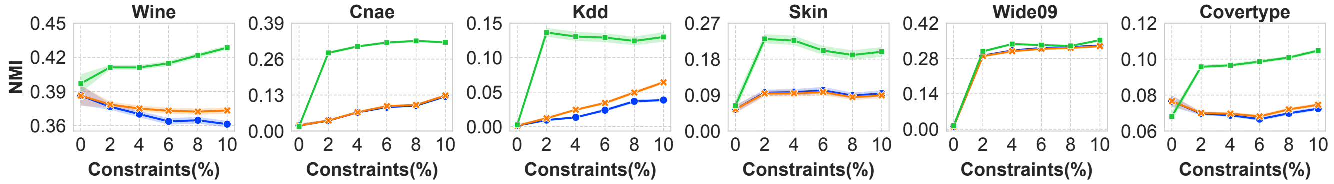

Figure 2: Normalized Mutual Information (Disjoint ML/CL).

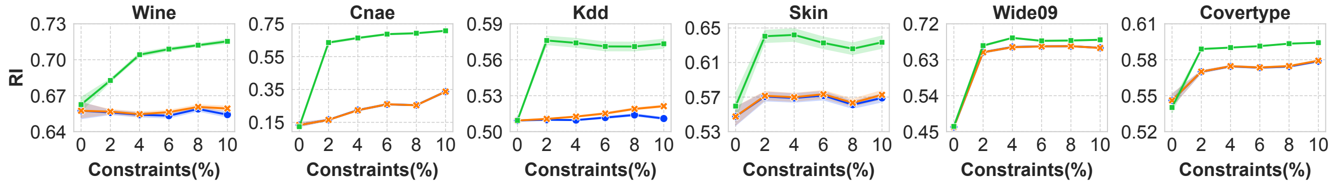

Figure 3: Rand Index (Disjoint ML/CL).

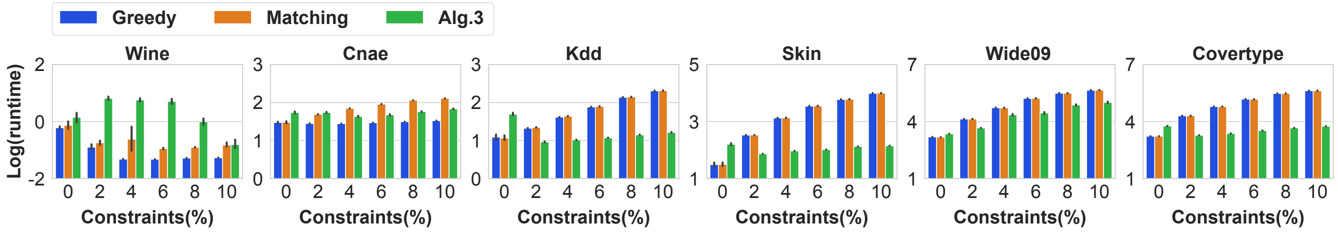

Figure 4: Runtime(ms) (Disjoint ML/CL).

two famous network traffic datasets (KDDTest+ of NLS-KDD (Tavallaee et al. 2009) and Wide09 of MAWI (Group 2020)) to evaluate the algorithms on the Internet traffic classification problem.

Simulated datasets. It is challenging for us to evaluate the practical approximation ratio of Alg. 3 on real-world datasets due to the lack of optimal solution costs as a benchmark for the constrained k-center problem. To overcome this problem, we construct simulated datasets using given parameters (e.g., k, n, and r^*). Tab. 1 gives brief statistics of the aforementioned datasets.

Constraints construction. We construct both disjoint and intersected ML/CL constraints for the real-world and the simulated datasets in accordance with the Introduction and Algorithm to evaluate the clustering performance of our approximation algorithm against baselines. In short, for a given dataset, a given number of constrained data points, and a given number of participants (who have their own background knowledge of ML/CL), we uniformly sample data points from the raw dataset into different ML and CL sets.

Baselines. Since this is the *first non-trivial* work with a 2-opt algorithm for the constrained k-center problem with disjoint CL sets, we propose two baseline algorithms - a greedy algorithm (Greedy) and a matching-based algorithm (Matching). In brief, Greedy is adapted from a constrained k-means algorithm (Wagstaff et al. 2001) to handle the CL sets while considering the ML constraints as "big" points, and matching is a simple improvement of Greedy by overall matching points to closer centers that incurs smaller covering radiuses.

Evaluation metrics. Following existing studies on clustering (Bera et al. 2022; Wang, Nie, and Huang 2014; Lingam, Rout, and Das 2020; Rand 1971), we use the common clustering quality metrics in the experiments, which are *Cost* (Epasto, Esfandiari, and Mirrokni 2019; Bera et al. 2022), *Normalized Mutual Information* (NMI) (Lingam, Rout, and Das 2020) and *Rand Index* (Rand 1971). For *runtime*, we report it in the base ten logarithms.

Implementation details. We implemented all algorithms in Java 1.8.0 on a 64-bit Linux 3.10 high-performance computer, where each node equips an Intel Xeon Gold 6240 CPU and 32 GB RAM.

Clustering Quality and Efficiency with Disjoint ML/CL Constraints

In this section, we show and analyze the experimental clustering quality and efficiency (averaged over 40 runs on each dataset) of our algorithms and the two baselines, when applied to disjoint ML/CL settings.

Given disjoint ML/CL constraints, Alg. 3 outperforms the baseline algorithms as guaranteed. From Fig. 1 to Fig. 3, while varying the number of constrained data points from 0% to 10% of all data points, we observe that Alg. 3 demonstrates promising clustering accuracy across all performance metrics, varying 10% to 300% clustering quality improvement. Since the experimental clustering accuracy is highly related to the theoretical approximation ratio, the guaranteed 2-opt for the constrained k-center problem with disjoint ML/CL constraints provides Alg. 3 with a significant advantage.

As the number of constraints increases, all three algorithms show a general upward trend in clustering accuracy. However, Alg. 3 stabilizes once the number of constraints reaches 4%. We argue the reason behind this behavior is that adding constraints effectively reduces the feasible solution space in optimization problems. Introducing more constraints is expected to improve the solution quality of an approximation algorithm. Consequently, we anticipate a higher clustering accuracy for a constrained k-center algorithm as the number of constraints increases.

Alg. 3 demonstrates significantly better performance on sparse datasets. An example of such a dataset is Cnae-9, which is highly sparse with 99.22% of its entries being zeros. Among all the datasets we examined, we observed the largest discrepancy in terms of cost/accuracy between Alg. 3 and the two baseline algorithms when applied to the Cnae-9 dataset. We attribute this disparity to the fact that the traditional k-center problem struggles with sparse high-dimensional datasets, as highlighted in the study (Steinbach, Ertóz, and Kumar 2013). However, the introduction of constraints proves beneficial in adjusting misclustering and center selection, offering improved performance in these cases.

The crucial factor in bounding the approximation ratio for the constrained k-center problem with disjoint ML/CL constraints is the selection of centers. Analyzing the results depicted in the figures, we observe that Greedy and Matching algorithms often yield similar clustering outcomes despite employing distinct strategies to handle the CL constraints. We contend that the cluster assignment methods have minimal influence on certain datasets, as the improvement in the experiment hinges on the correction of center selection facilitated by the approximation algorithm with constraints. This correction not only enhances the approximation ratio but also contributes to improved clustering accuracy.

#CL/ML	Algorithm	$k=5$	$k=10$	$k=50$	$k=100$
$1,000$	Alg. 3	**1.9901**	**1.9971**	**1.9964**	**1.9997**
	Matching	2.8678	2.8295	2.9281	2.7805
	Greedy	2.9542	2.9793	2.9308	2.9286
$2,000$	Alg. 3	**1.9892**	**1.9931**	**1.9942**	**1.9986**
	Matching	2.7537	2.8047	2.9246	2.6884
	Greedy	2.9992	2.9648	3.0400	2.9073
$5,000$	Alg. 3	**1.9941**	**1.9908**	**1.9958**	**1.9952**
	Matching	2.5500	2.7392	3.0239	2.8794
	Greedy	3.0555	3.0647	3.2250	3.1741
$10,000$	Alg. 3	**1.9965**	**1.9938**	**1.9979**	**1.9983**
	Matching	2.5140	2.7952	3.0236	3.1212
	Greedy	3.3707	3.5421	3.4695	3.4143

Table 2: Empirical Approximation Ratio.

Alg. 3 exhibits superior efficiency when applied to larger datasets. Fig. 4 depicts the (log) runtime of all three algorithms. As a larger number of points are designated as constrained points, the likelihood of encountering a scenario where a CL set contains k points increases. In such cases, Alg. 3 can rapidly determine the center set, whereas Greedy and Matching algorithms must rely on center sets determined by traditional methods.

Empirical Approximation Ratio

In this experiment, **approximation ratio** is measured based on the clustering radius ratio obtained on the simulated dataset, i.e., Approx Ratio $= r_{\max}/r^*$, where $r_{\max}$ represents the maximum radius (worst solution cost) obtained from $1,000$ runs of the algorithm (10 simulated datasets $\times$ 10 distinct constrained cases $\times$ 10 repeated runs) and r^* denotes the optimal cost by construction in the simulated dataset. Tab. 2 presents the empirical ratios, demonstrating that Alg. 3 consistently produces better approximation ratios below two, which aligns with the theoretical results presented in the previous Algorithm section.

Conclusion

In this paper, we confirmed the existence of an efficient approximation algorithm for the constrained k-center problem with instance-level ML/CL constraints. Despite the known inapproximability barrier from the arbitrary CL constraints, we made a significant breakthrough by uncovering that the reducible disjoint set structure of CL constraints on k-center can lead to constant factor approximation in our analysis. To achieve the best possible 2-approximation, we introduced a structure called *reverse dominating set* (RDS) for obtaining the desired set of cluster centers. For efficient RDS computation, we employ a suite of linear programming-based techniques. Our work sheds light on devising efficient approximation algorithms for solving more complex clustering problems involving constraints. For instance, it opens avenues for further investigation such as approximations with inconsistent, stochastic, and/or active constraints.

Acknowledgments

This work is supported by the Taishan Scholars Young Expert Project of Shandong Province (No. tsqn202211215), the National Science Foundation of China (Nos. 12271098 and 61772005), and the Outstanding Youth Innovation Team Project for Universities of Shandong Province (No. 2020KJN008). The complete version of this paper can be found at https://arxiv.org/abs/2401.12533 with datasets and codes at https://github.com/ChaoqiJia/ML-CL-k-Center.

References

Bache, K.; and Lichman, M. 2013. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

Badanidiyuru, A.; Mirzasoleiman, B.; Karbasi, A.; and Krause, A. 2014. Streaming Submodular Maximization: Massive Data Summarization on the Fly. In *Proceedings of the Twentieth ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 671–680. New York, NY, USA: Association for Computing Machinery (ACM).

Basu, S.; Banerjee, A.; and Mooney, R. J. 2004. Active Semi-supervision for Pairwise Constrained Clustering. In *Proceedings of the Fourth SIAM International Conference on Data Mining (SDM)*. Orlando, Florida, USA.

Basu, S.; Davidson, I.; and Wagstaff, K., eds. 2008. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. CRC Press.

Bera, S. K.; Das, S.; Galhotra, S.; and Kale, S. S. 2022. Fair k-Center Clustering in MapReduce and Streaming Settings. In *Proceedings of the Thirty-First ACM Web Conference 2022*, 1414–1422. Virtual Event, Lyon, France.

Brubach, B.; Chakrabarti, D.; Dickerson, J. P.; Srinivasan, A.; and Tsepenekas, L. 2021. Fairness, Semi-Supervised Learning, and More: A General Framework for Clustering with Stochastic Pairwise Constraints. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, 6822–6830. Vancouver, Canada: AAAI Press.

Charikar, M.; Guha, S.; Tardos, É.; and Shmoys, D. B. 1999. A Constant-Factor Approximation Algorithm for the k-Median Problem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC)*, 1–10. Atlanta, Georgia, USA.

Charikar, M.; Khuller, S.; Mount, D. M.; and Narasimhan, G. 2001. Algorithms for Facility Location Problems with Outliers. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms (SODA)*, 642–651. Washington, DC, USA.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair Clustering through Fairlets. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, 5036–5044. Montréal, Canada.

Coleman, T.; Saunderson, J.; and Wirth, A. 2008. Spectral Clustering with Inconsistent Advice. In *Proceedings of the Twenty-Fifth International Conference (ICML)*, 152–159. Helsinki, Finland.

Cygan, M.; Fomin, F. V.; Kowalik, Ł.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized algorithms*. Springer.

Davidson, I.; and Ravi, S. 2007. The Complexity of Non-hierarchical Clustering with Instance and Cluster Level Constraints. *Data Mining and Knowledge Discovery*, 14(1): 25–61.

Davidson, I.; and Ravi, S. 2009. Using Instance-Level Constraints in Agglomerative Hierarchical Clustering: Theoretical and Empirical Results. *Data Mining and Knowledge Discovery*, 18(2): 257–282.

Davidson, I.; Ravi, S.; and Shamis, L. 2010. A Sat-Based Framework for Efficient Constrained Clustering. In *Proceedings of the Tenth SIAM International Conference on Data Mining (SDM)*, 94. Society for Industrial and Applied Mathematics.

Davidson, I.; and Ravit, S. 2005. Clustering with Constraints: Feasibility Issues and the k-Means Algorithm. In *Proceedings of the Fifth SIAM International Conference on Data Mining (SDM)*, 138. Newport Beach, CA, USA.

Epasto, A.; Esfandiari, H.; and Mirrokni, V. 2019. On-Device Algorithms for Public-Private Data with Absolute Privacy. In *The World Wide Web Conference*, 405–416.

Ester, M.; Ge, R.; Gao, B. J.; Hu, Z.; and Ben-Moshe, B. 2006. Joint Cluster Analysis of Attribute Data and Relationship Data: The Connected k-Center Problem. In *Proceedings of the Sixth SIAM International Conference on Data Mining (SDM)*, 246–257. Bethesda, MD, USA: SIAM.

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

Group, M. W. 2020. MAWI Traffic Archive. https://mawi.wide.ad.jp/mawi/. Accessed 15 May 2023.

Hall, P. 1935. On Representatives of Subsets. *Journal of the London Mathematical Society*, 1(1): 26–30.

Hochbaum, D. S.; and Shmoys, D. B. 1985. A Best Possible Heuristic for the k-Center Problem. *Mathematics of Operations Research*, 10(2): 180–184.

Huang, J.; Feng, Q.; Huang, Z.; Xu, J.; and Wang, J. 2023. Fast Algorithms for Distributed k-Clustering with Outliers. In *Proceedings of the Fortieth International Conference on Machine Learning (ICML)*, 13845–13868. Honolulu, Hawaii, USA: PMLR.

IBM. 2022. v22.1: User's manual for CPLEX. https://www.ibm.com/docs/en/icos/22.1.0.

Jia, C.; Guo, L.; Liao, K.; and Lu, Z. 2023. Efficient Algorithm for the k-Means Problem with Must-Link and Cannot-Link Constraints. *Tsinghua Science and Technology*, 28(6): 1050–1062.

Khuller, S.; Pless, R.; and Sussmann, Y. J. 2000. Fault Tolerant k-Center Problems. *Theoretical Computer Science*, 242(1-2): 237–245.

Khuller, S.; and Sussmann, Y. J. 2000. The Capacitated k-center Problem. *SIAM Journal on Discrete Mathematics*, 13(3): 403–418.

Kleindessner, M.; Awasthi, P.; and Morgenstern, J. 2019. Fair k-Center Clustering for Data Summarization. In *Proceedings of the Thirty-Sixth International Conference on Machine Learning (ICML)*, 3448–3457.

Lim, A.; Rodrigues, B.; Wang, F.; and Xu, Z. 2005. k-Center Problems with Minimum Coverage. *Theoretical Computer Science*, 332(1-3): 1–17.

Lingam, G.; Rout, R. R.; and Das, S. K. 2020. Social Botnet Community Detection: A Novel Approach Based on Behavioral Similarity in Twitter Network Using Deep Learning. In *Proceedings of the Twenty-Seventh ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 708–718. Virtual Event, USA.

Liu, X.; Li, Q.; and Li, T. 2017. Private Classification with Limited Labeled Data. *Knowledge-Based Systems*, 133: 197–207.

Lloyd, S. 1982. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129–137.

Malkomes, G.; Kusner, M. J.; Chen, W.; Weinberger, K. Q.; and Moseley, B. 2015. Fast Distributed k-Center Clustering with Outliers on Massive Data. In *Advances in Neural Information Processing Systems (NeurIPS) 29*, 1063–1071. Montreal, Quebec, Canada.

Rand, W. M. 1971. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical association*, 66(336): 846–850.

Steinbach, M.; Ertóz, L.; and Kumar, V. 2013. The Challenges of Clustering High Dimensional Data. *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition*, 273.

Tavallaee, M.; Bagheri, E.; Lu, W.; and Ghorbani, A. A. 2009. A Detailed Analysis of the KDD CUP 99 Data Set. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 53–58. Ottawa, Canada.

Wagstaff, K.; and Cardie, C. 2000. Clustering with Instance-Level Constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 1103–1110. Stanford, CA, USA.

Wagstaff, K.; Cardie, C.; Rogers, S.; and Schrödl, S. 2001. Constrained k-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, 577–584. Williams College, Williamstown, MA, USA.

Wang, D.; Nie, F.; and Huang, H. 2014. Unsupervised Feature Selection via Unified Trace Ratio Formulation and k-Means Clustering. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014*, 306–321. Nancy, France: Springer.

Xing, E. P.; Ng, A. Y.; Jordan, M. I.; and Russell, S. 2002. Distance Metric Learning, with Application to Clustering with Side-Information. In *Advances in Neural Information Processing Systems (NeurIPS) 15*, 521–528. Vancouver, British Columbia, Canada.

Zhang, J.; Chen, C.; Xiang, Y.; Zhou, W.; and Vasilakos, A. V. 2013. An Effective Network Traffic Classification Method with Unknown Flow Detection. *IEEE Transactions on Network and Service Management*, 10(2): 133–147.