Clustering with Diversity

Jian Li[1], Ke Yi[2], and Qin Zhang[2]

[1] University of Maryland, College Park, MD, USA. E-mail: lijian@cs.umd.edu
[2] Hong Kong University of Science and Technology, Hong Kong, China
E-mail: {yike,qinzhang}@cse.ust.hk

Abstract. We consider the *clustering with diversity* problem: given a set of colored points in a metric space, partition them into clusters such that each cluster has at least ℓ points, all of which have distinct colors. We give a 2-approximation to this problem for any ℓ when the objective is to minimize the maximum radius of any cluster. We show that the approximation ratio is optimal unless $\mathbf{P} = \mathbf{NP}$, by providing a matching lower bound. Several extensions to our algorithm have also been developed for handling outliers. This problem is mainly motivated by applications in privacy-preserving data publication.

Key words: Approximation algorithm, k-center, k-anonymity, l-diversity

1 Introduction

Clustering is a fundamental problem with a long history and a rick collection of results. A general clustering problem can be formulated as follows. Given a set of points P in a metric space, partition P into a set of disjoint clusters such that a certain objective function is minimized, subject to some cluster-level and/or instance-level constraints. Typically, cluster-level constraints impose restrictions on the number of clusters or on the size of each cluster. The former corresponds to the classical k-center, k-median, k-means problems, while the latter has recently received much attention from various research communities [1, 15, 16]. On the other hand, instance-level constraints specify whether particular items are similar or dissimilar, usually based on some background knowledge [3, 26]. In this paper, we impose a natural instance-level constraint on a clustering problem, that the points are colored and all points partitioned into one cluster must have distinct colors. We call such a problem *clustering with diversity*. Note that the traditional clustering problem is a special case of ours where all points have unique colors.

As an illustrating example, consider the problem of choosing locations for a number of factories in an area where different resources are scattered. Each factory needs at least ℓ different resources allocated to it and the resource in one location can be sent to only one factory. This problem corresponds to our clustering problem where each kind of resource has a distinct color, and we have a lower bound ℓ on the the cluster size.

The main motivation to study clustering with diversity is privacy preservation for data publication, which has drawn tremendous attention in recent years in both the database community [7, 8, 21, 24, 28–30] and the theory community [1, 2, 11, 12, 22]. The goal of all the studies in privacy preservation is to prevent *linking attacks* [25]. Consider the table of patient records in Figure 1(a), usually called the *microdata*. There are three types of attributes in a microdata table. The *sensitive attribute* (SA), such as "Disease", is regarded as the individuals' privacy, and is the target of protection. The

T-1D(*Name*)	Age	Gender	Degree	Disease
1 (*Adam*)	29	M	M.Sc.	HIV
2 (*Bob*)	25	M	M.Sc.	HIV
3 (*Calvin*)	25	M	B.Sc.	pneumonia
4 (*Daisy*)	29	F	B.Sc.	bronchitis
5 (*Elam*)	40	M	B.Sc.	bronchitis
6 (*Frank*)	45	M	B.Sc.	bronchitis
7 (*George*)	35	M	B.Sc.	pneumonia
8 (*Henry*)	37	M	B.Sc.	pneumonia
9 (*Ivy*)	50	F	Ph.D.	dyspepsia
10 (*Jane*)	60	F	Ph.D.	pneumonia

(a)

QIs	Disease
(center, radius)	HIV
	HIV
(center, radius)	pneumonia
	bronchitis
(center, radius)	bronchitis
	bronchitis
(center, radius)	pneumonia
	pneumonia
(center, radius)	dyspepsia
	pneumonia

(b)

QIs	Disease
(center, radius)	HIV
	HIV
	pneumonia
	bronchitis
(center, radius)	bronchitis
	bronchitis
	pneumonia
	pneumonia
(center, radius)	dyspepsia
	pneumonia

(c)

Fig. 1. (a) The microdata; (b) A 2-anonymous table; (c) An 2-diverse table;

identifier, in this case "Name", uniquely identifies a record, hence must be ripped off before publishing the data. The rest of the attributes, such as "Age", "Gender", and "Education", should be published so that researchers can apply data mining techniques to study the correlation between these attributes and "Disease". However, since these attributes are public knowledge, they can often uniquely identify individuals when combined together. For example, if an attacker knows (i) the age (25), gender (M), and education level (Master) of Bob, and (ii) Bob has a record in the microdata, s/he easily finds out that Tuple 2 is Bob's record and hence, Bob contracted HIV. Therefore, these attributes are often referred to as the *quasi-identifiers* (QI). The solution is thus to make these QIs ambiguous before publishing the data so that it is difficult for an attacker to link an individual from the QIs to his/her SA, but at the same time we want to minimize the amount of information loss due to the ambiguity introduced to the QIs so that the interesting correlations between the QIs and the SA are still preserved.

The usual approach taken to prevent linking attacks is to partition the tuples into a number of *QI-groups*, namely clusters, and within each cluster all the tuples share the same (ambiguous) QIs. There are various ways to introduce ambiguity. A popular approach, as taken by [1], is to treat each tuple as a high-dimensional point in the QI-space, and then only publish the center, the radius, and the number of points of each cluster. To ensure a certain level of privacy, each cluster is required to have at least k points so that the attacker is not able to correctly identify an individual with confidence larger than $1/k$. This requirement is referred to as the k-*ANONYMITY* principle [1, 22]. The problem, translated to a clustering problem, can be phrased as follows: Cluster a set of points in a metric space, such that each cluster has at least r points. When the objective is to minimize the maximum radius of all clusters, the problem is called r-GATHERING and a 2-approximation is known [1].

However, the k-ANONYMITY principle suffers from the *homogeneity* problem: A cluster may have too many tuples with the same SA value. For example, in Figure 1(b), all tuples in QI-group 1, 3, and 4 respectively have the the same disease. Thus, the attacker can infer what disease all the people within a QI-group have contracted without identifying any individual record. The above problem has led to the development of many SA-aware principles. Among them, ℓ-*DIVERSITY* [21] is the most widely deployed [8, 13, 17, 21, 28, 29], due to its simplicity and good privacy guarantee. The principle demands that, in each cluster, at most $1/\ell$ of its tuples can have the same SA value.

Figure 1(c) shows a 2-diverse version of the microdata. In an ℓ-diverse table, an attacker can figure out the real SA value of the individual with confidence no more than $1/\ell$. Treating the SA values as colors, this problem then exactly corresponds to the clustering with diversity problem defined at the beginning, where we have a lower bound ℓ on the cluster size.

In contrast to the many theoretical results for r-GATHERING and k-ANONYMITY [1, 2, 22], no approximation algorithm with performance guarantees is known for ℓ-DIVERSITY, even though many heuristic solutions have been proposed [13, 19, 21].

Clustering with instance-level constraints and other related work. Clustering with instance-level constraints is a developing area and begins to find many interesting applications in various areas such as bioinformatics [5], machine learning [26, 27], data cleaning [4], etc. Wagstaff and Cardie in their seminal work [26] considered the following two types of instance-level hard constraints: A *must-link (ML)* constraint dictates that two particular points must be clustered together and a *cannot-link (CL)* constraint requires they must be separated. Many heuristics and variants have been developed subsequently, e.g. [27, 31], and some hardness results with respect to minimizing the number of clusters were also obtained [10]. However, to the best of our knowledge, no approximation algorithm with performance guarantee is known for any version of the problem. We note that an ℓ-diverse clustering can be seen as a special case where nodes with the same color must satisfy CL constraints.

As opposed to the hard constraints imposed on any clustering, the correlation clustering problem [6] considers soft and possibly conflicting constraints and aims at minimizing the violation of the given constraints. An instance of this problem can be represented by a complete graph with each edge labeled (+) or (-) for each pair of vertices, indicating that two vertices should be in the same or different clusters, respectively. The goal is to cluster the elements so as to minimize the number of disagreements, i.e., (-) edges within clusters and (+) edges crossing clusters. The best known approximations for various versions of the problem are due to Ailon et al. [3]. If the number of clusters is stipulated to be a small constant k, there is a polynomial time approximation scheme [14]. In the Dedupalog project, Arasu et al. [4] considered correlation clustering together with instance-level hard constraints, with the aim of de-duplicating entity references .

Approximation algorithms for clustering with outliers were first considered by Charikar et al. [9]. The best known approximation factor for r-GATHERING with outliers is 4 due to Aggrawal et al. [1].

Our results. In this paper, we give the first approximation algorithms to the clustering with diversity problem. We formally define the problem as follows.

Definition 1 (ℓ-DIVERSITY). *Given a set of n points in a metric space where each of them has a color, cluster them into a set $\mathcal{C}$ of clusters, such that each cluster has at least ℓ points, and all of its points have distinct colors. The goal is to minimize the maximum radius of any cluster.*

Our first result (Section 2) is a 2-approximation algorithm for ℓ-DIVERSITY. The algorithm follows a similar framework as in [1], but it is substantially more complicated.

The difficulty is mainly due to the requirement to resolve the conflicting colors in each cluster while maintaining its minimum size ℓ. To the best of our knowledge, this is first approximation algorithm for a clustering problem with instance-level hard constraints.

Next, we show that this approximation ratio is the best possible by presenting a matching lower bound (Section 3). A lower bound of 2 is also given in [1] for r-GATHERING. But to carry that result over to ℓ-DIVERSITY, all the points need to have unique colors. This severely limits to applicability of this hardness result. In Section 3 we give a construction showing that even with only 3 colors, the problem is NP-hard to approximate within any factor strictly less than 2. In fact, if there are only 2 colors, we show that the problem can be solved optimally in polynomial time via bipartite matching.

Unlike r-GATHERING, an instance to the ℓ-DIVERSITY problem may not have a feasible solution at all, depending on the color distribution. In particular, we can easily see that no feasible clustering exists when there is one color that has more than $\lfloor n/\ell \rfloor$ points. One way to get around this problem is to have some points not clustered (which corresponds to deleting a few records in the ℓ-DIVERSITY problem). Deleting records causes information loss in the published data, hence should be minimized. Ideally, we would like to delete points just enough such that the remaining points admit a feasible ℓ-diverse clustering. In Section 4, we consider the *ℓ-DIVERSITY-OUTLIERS* problem, where we compute an ℓ-diverse clustering after removing the least possible number of points. We give an $O(1)$-approximation algorithm to this problem.

Our techniques for dealing with diversity and cluster size constraints may be useful in developing approximation algorithms for clustering with more general instance-level constraints. Due to space constraints, we only provide complete details for our first result. We refer interested readers to the full version of the paper for all missing details and proofs [20].

2 A 2-Approximation for ℓ-DIVERSITY

In this section we assume that a feasible solution on a given input always exists. We first introduce a few notations. Given a set of n points in a metric space, we construct a weighted graph $G(V, E)$ where V is the set of points and each vertex $v \in V$ has a color $c(v)$. For each pair of vertices $u, v \in V$ with different colors, we have an edge $e = (u, v)$, and its weight $w(e)$ is just their distance in the metric space. For any $u, v \in V$, let $\texttt{dist}_G(u, v)$ be the shortest path distance of u, v in graph G. For any set $A \subseteq V$, let $N_G(A)$ be the set of neighbors of A in G. For a pair of sets $A \subseteq V, B \subseteq V$, let $E_G(A; B) = \{(a, b) \mid a \in A, b \in B, (a, b) \in E(G)\}$. The diameter of a cluster C of nodes is defined to be $d(C) = \max_{u,v \in C}(w(e(u, v)))$. Given a cluster C and its center v, the radius $r(C)$ of C is defined as maximum distance from any node of C to v, i.e., $r(C) = \max_{u \in C} w(u, v)$. By triangle inequality, it is obvious to see that $\frac{1}{2}d(C) \leq r(C) \leq d(C)$.

A *star forest* is a forest where each connected component is a star. A *spanning star forest* is a star forest spanning all vertices. The *cost* of a spanning forest $\mathcal{F}$ is the length of the longest edge in $\mathcal{F}$. We call a star forest *semi-valid* if each star component contains at least ℓ colors and *valid* if it is semi-valid and each star is *polychromatic*, i.e., each node in the star has a distinct color. Note that a spanning star forest with cost R

naturally defines a clustering with largest radius R. Denote the radius and the diameter of the optimal clustering by r^* and d^*, respectively.

We first briefly review the 2-approximation algorithm for the r-GATHERING problem [1], which is the special case of our problem when all the points have distinct colors. Let $e_1, e_2, \ldots$ be the edges of G in a non-decreasing order of their weights. The general idea of the r-GATHERING algorithm [1] is to first guess the optimal radius R by considering each graph G_i formed by the first i edges $E_i = \{e_1, \ldots, e_i\}$, as $i = 1, 2, \ldots$. It is easy to see that the cost of a spanning star forest of G_i is at most $w(e_i)$. For each G_i $(1 \leq i \leq m)$, the following condition is tested (rephrased to fit into our context):

(I) There exists a maximal independent set I such that there is a spanning star forest in G_i with the nodes in I being the star centers, and each star has at least r nodes.

It is proved [1] that the condition is met if the length of e_i is d^*. The condition implies the radius of our solution is at most d^* which is at most $2r^*$. Therefore, we get an 2-approximation. In fact, the independent set I can be chosen greedily and finding the spanning star forest can be done via a network flow computation.

Our 2-approximation for the ℓ-diversity problem follows the same framework, that is, we check each G_i in order and test the following condition:

(II) There exists a maximal independent set I such that there is a *valid* spanning star forest in G_i with the nodes in I being the star centers.

The additional challenge is of course that, while condition (I) only puts a constraint on the size of each star, condition (II) requires both the size of each star to be at least ℓ and all the nodes in a star have distinct colors. Below we first give a constructive algorithm that for a given G_i, tries to find an I such that condition (II) is met. Next we show that when $w(e_i) = d^*$, the algorithm is guaranteed to succeed. The approximation ratio of 2 then follows immediately.

To find an I to meet condition (II), the algorithm starts with an arbitrary maximal independent set I, and iteratively augments it until the condition is met, or fails otherwise. In each iteration, we maintain two tests. The first one, denoted by *flow test* F-TEST(G_i, I), checks if there exists a semi-valid spanning star forest in G_i with nodes in I being star centers. If I does not pass this test, the algorithm fails right away. Otherwise we go on to the second test, denoted by *matching test* M-TEST(G_i, I), which tries to find a valid spanning star forest. If this test succeeds, we are done; otherwise the failure of this test yields a way to augment I and we proceed to the next iteration. The algorithm is outlined in Algorithm 1.

We now elaborate on F-TEST and M-TEST. F-TEST(G_i, I) checks if there is a spanning star forest in G_i with I being the star centers such that each star contains at least ℓ colors. As the name suggests, we conduct the test using a network flow computation. We first create a source s and a sink t. For each node $v \in V$, we add an edge (s, v) with capacity 1, and for each node $o_j \in I (1 \leq j \leq |I|)$, we create a vertex o_j and add an outgoing edge (o_j, t) with capacity lower bound ℓ. For each node $o_j \in I$ and each color c, we create a vertex $p_{j,c}$ and an edge $(p_{j,c}, o_j)$ with capacity upper bound 1. For any $v \in V$ such that $(v, o_j) \in E_i$ or $v = o_j$, and v has color c, we add an edge from v to $p_{j,c}$ without capacity constraint. Finally, we add one new vertex o'_j for each $o_j \in I$, connect all $v \in V$ to o'_j without capacity constraint if $(v, o_j) \in E$ or

Algorithm 1: Algorithm to find an I in G_i to meet condition (II)

```
Let I be an arbitrary maximal independent set in G_i;
while F-TEST(G_i, I) is passed do
    (S, S') ← M-TEST(G_i, I)  /* S ⊂ V, S' ⊆ I */;
    if S = ∅ then
        Succeed;
    else
        I ← I − S' + S;
        Add nodes to I until it is a maximal independent set;
Fail;
```

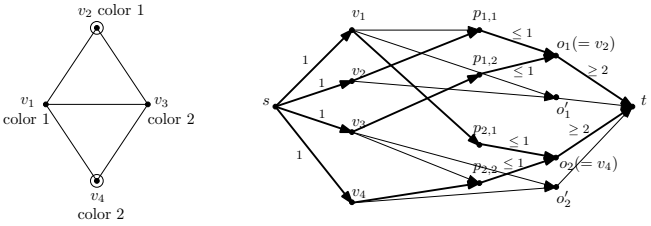

Fig. 2. The flow network construction. On the left is the original graph, $I = \{v_2, v_4\}$, $\ell = 2$. On the right is the corresponding flow network. Thick edges denote a feasible flow of value $|I|\ell = 4$.

$v = o_j$, and connect o'_j to t without capacity constraint. The capacity upper bound of $(p_{j,c}, o_j)$ forces at most one node with color c to be assigned to o_j. Therefore, all nodes assigned to o_j have distinct colors. The capacity lower bounds of (o_j, t)s require that each cluster has at least ℓ nodes. Nodes o'_js are used to absorb other unassigned nodes. It is not difficult to see that there exists a semi-valid spanning star forest with nodes in I being star centers in G_i if an n-units flow can be found. In this case we say that the F-TEST is passed. See Figure 2 for an example. Note that a network flow problem with both capacity lower bounds and upper bounds is usually referred to as the *circulation problem*, and is polynomially solvable [18].

Once G_i and I pass F-TEST, we try to redistribute those vertices that cause color conflicts. We do so by a bipartite matching test M-TEST(G_i, I) which returns two vertex sets S and S' that are useful later. Concretely, we test whether there exists a matching in the bipartite graph $B(I - C, C - I; E_{G_i}(I - C; C - I))$ for each color class C such that all vertices in $C - I$ are matched. If such matchings can be found for all the colors, we say that the M-TEST is passed. Note that all these matchings together give a spanning star forest such that each star is polychromatic. However, this does not guarantee that the cardinality constraint is preserved. The crucial fact here is that I passes *both* F-TEST and M-TEST. In Lemma 2 we formally prove that there exists a valid spanning star forest with nodes in I as star centers if and only if G_i and I pass both F-TEST and M-TEST. To actually find a valid spanning star forest, we can again use the network flow construction in F-TEST but without the o'_j nodes. If M-TEST fails, we know that for some color class C, there exists a subset $S \subseteq C - I$ such

that the size of its neighbor set $|N_B(S)|$ is less than $|S|$ by Hall's theorem [18]. In this case, M-TEST returns $(S, N_B(S))$; such a set S can be found by a maximum matching algorithm. Then we update the independent set $I \leftarrow I - N_B(S) + S$; we show that I is still an independent set in Lemma 1. Finally, we add nodes to I arbitrarily until it becomes a maximal independent set. Then, we start the next iteration with the new I. Since $|S| > |N_B(S)|$, we increase $|I|$ by at least one in each iteration. So the algorithm terminates in $\leq n$ iterations.

Before proving that Algorithm 1 is guaranteed to succeed when $w(e_i) = d^*$, we prove the two lemmas left in the description of our algorithm. The first lemma ensures that I is always an independent set.

Lemma 1. *The new set $I \leftarrow I - S' + S$ obtained in each update is still an independent set in G_i.*

Proof. Since all vertices in S have the same color, there is no edge among them. Therefore, we only need to prove that there is no edge between $I - S'$ and S, which is trivial since $S' = N_B(S)$. □

The second lemma guarantees that we find a feasible solution if both tests are passed.

Lemma 2. *Given $G_i = G(V, E_i)$ and I, a maximal independent set of G_i, both F-TEST(G_i,I) and M-TEST(G_i,I) are passed if and only if there exists a valid spanning star forest in G_i with nodes in I being star centers.*

Proof. The "if" part is trivial. We only prove the "only if" part. Suppose G_i and I pass both F-TEST(G_i,I) and M-TEST(G_i,I). Consider a semi-valid spanning star forest obtained in G_i after F-TEST. We delete a minimal set of leaves to make it a valid star (not necessarily spanning) forest F. Consider the bipartite graph $B(I - C; C - I, E_{G_i}(I - C; C - I))$ for each color class C. We can see $F \cap B$ is a matching in B. Since I passes M-TEST, we know that there exists a maximum matching such that all nodes in $C - I$ can be matched. If we use the Hungarian algorithm to compute a maximum matching with $F \cap B$ as the initial matching, the nodes in $I - C$ which are originally matched will still be matched in the maximum matching due to the property of the alternating path augmentation[3]. Therefore, the following invariants are maintained: each star is polychromatic and has at least ℓ colors. By applying the above maximum matching computation for each color class, we obtain a valid spanning star forest in G_i. □

Finally, we prove that Algorithm 1 is guaranteed to succeed on G_{i*} for the maximal index i^* such that $w(e_{i^*}) = d^*$, where d^* is the optimal cluster diameter of any valid spanning star forest of G.

Lemma 3. *Algorithm 1 will succeed on G_{i^*}.*

[3] Recall that an augmenting path P (with respect to matching M) is a path starting from unmatched node, alternating between unmatched and matched edges and ending also at an unmatched node (for example, see [23]). By taking the symmetric difference of P and M, which we call augmenting on P, we can obtain a new matching with one more edge.

Proof. Suppose $\mathcal{C}^* = \{C_1^*, \ldots, C_{k^*}^*\}$ is the set of clusters in the optimal clustering with cluster diameter d^*. Since G_{i*} include all edges of weights no more than d^*, each C_j^* induces a clique in G_{i^*} for all $1 \leq j \leq k^*$, thus it contains at most one node in any independent set. Therefore, any maximal independent set I in G_{i^*} can pass F-TEST(G_{i^*}, I), and we only need to argue that I will also pass M-TEST(G_{i^*}, I). Each update to the independent set I increases the size of I by at least 1 and the maximum size of I is k^*. When $|I| = k^*$, each C_j^* contains exactly one node in I and this I must be able to pass M-TEST(G_{i^*}, I). So Algorithm 1 must succeed in some iteration. □

By Lemma 3, the cost of the spanning star forest found by Algorithm 1 is at most d^*. Since the cost of the optimal spanning forest is at least $d^*/2$, we obtain a 2-approximation.

Theorem 1. *There is a polynomial-time 2-approximation for ℓ-DIVERSITY.*

3 The Lower Bound

We show that ℓ-DIVERSITY is NP-hard to approximate within a factor less than 2 even when there are only 3 colors. Note that if there are 2 colors, the problem can be solved in polynomial time by computing perfect matchings in the threshold graphs.

Theorem 2. *There is no polynomial-time approximation algorithm for ℓ-DIVERSITY that achieves an approximation factor less than 2 unless $P = NP$.*

4 Dealing with Unqualified Inputs

For the ℓ-DIVERSITY problem, a feasible solution may not exist depending on the input color distribution. The following simple lemma gives a necessary and sufficient condition for the existence of a feasible solution.

Lemma 4. *There exists a feasible solution for ℓ-DIVERSITY if and only if the number of nodes with the same color c is at most $\lfloor \frac{n}{\ell} \rfloor$ for each color c.*

To cluster an instance without a feasible solution, we must exclude some nodes as *outliers*. The following lemma characterizes the minimum number of outliers.

Lemma 5. *Let $C_1, C_2, \ldots, C_k$ be the color classes sorted in the non-increasing order of their sizes.*

1. *Let p be the maximum integer satisfying $\sum_{i=1}^{k} \min(p, |C_i|) \geq p\ell$. The minimum number of outliers is given by $q = \sum_{i=1}^{k} \max(0, |C_i| - p)$ and p is the number of clusters when we exclude q outliers.*
2. $p\ell \leq n - q < p(\ell + 1)$.

With lemma 5 at hand, it is natural to consider the following optimization problem: find an ℓ-DIVERSITY solution by clustering $n - q$ points such that the maximum cluster radius is minimized. We call this problem ℓ-DIVERSITY-OUTLIERS. From Lemma 5 we can see that p is independent on the metric and can be computed in advance. In addition, implicit from Lemma 5 is that the number of outliers of each color is also fixed, but we need to decide *which* points should be chosen as outliers.

In the fortunate case where we have a color class C with exactly p nodes, we know that there is exactly one node of C in each cluster of any feasible solution. By using a similar flow network construction used in F-TEST, we can easily get a 2-approximation using C as the cluster centers. However, the problem becomes much more difficult when the sizes of all color classes are different from p.

4.1 A Constant Approximation

We first define some notations. We call color classes of size larger than p *popular colors* and nodes having such colors *popular* nodes. Other color classes have at most p nodes, and these nodes are *unpopular*. We denote the set of popular nodes by $\mathcal{P}$ and the set of unpopular nodes by $\mathcal{N}$. Note that after removing the outliers, each popular color has exactly p nodes and each cluster will contain the same number of popular nodes. Let z be the number of popular nodes each cluster contains. We denote by G^d the power graph of G in which two vertices u, v are adjacent if there is path connecting u and v with at most d edges. The length of the edge (u, v) in G^d is set to be $\mathsf{dist}_G(u, v)$. Before describing the algorithm, we need the following simple lemma.

Lemma 6. *For any connected graph G, G^3 contains a Hamiltonian cycle which can be found in linear time.*

The algorithm still adopts the thresholding method, that is, we add edges one by one to get graphs $G_i = (V, E_i = \{e_1, e_2, \ldots, e_i\})$, for $i = 1, 2, \ldots$, and in each G_i, we try to find a valid star forest that spans G_i except q outliers. Let d^* be the diameter of the optimal solution that clusters $n - q$ points, and i^* be the maximum index such that $w(e_i) = d^*$. Let $G_i[\mathcal{N}]$ be the subgraph of G_i induced by all unpopular nodes. We define the ball of radius r around v to be $\mathsf{B}(v, r) = \{u \mid u \in \mathcal{N} \wedge \mathsf{dist}_G(v, u) \leq r\}$. For each G_i, we run the Algorithm: ℓ-DIVERSITY-OUTLIERS(G_i) (see below). We proceed to G_{i+1} when the algorithm claims failure.

The high level idea of the algorithm is as follows: Our goal is to show that the algorithm can find a valid star forest spanning $n - q$ nodes in $G_{i^*}^{28}$. It is not hard to see that this gives us an approximation algorithm with factor $28 \times 2 = 56$. First, we notice that F-TEST can be easily modified to work for the outlier version by excluding all o'_j nodes and testing whether there is a flow of value $n - q$. However, the network flow construction needs to know in advance the set of candidates of cluster centers. For this purpose, we attempt to attach p new nodes which we call *virtual centers* to G_i which serve as the candidates of cluster centers in F-TEST. In the ideal case, if these virtual centers can be distributed such that each of them is attached to a distinct optimal cluster, F-TEST can easily produce a 2-approximation. Since the optimal clustering is not known, this is very difficult in general. However, we show there is way to carefully distribute the virtual centers such that there is a perfect matching between these virtual centers and the optimal cluster centers and the longest matching edge is at most $27d^*$, which implies that our algorithm can find a valid spanning star forest in $G_{i^*}^{27+1} = G_{i^*}^{28}$

Algorithm*: ℓ-DIVERSITY-OUTLIERS(G_i).*

1. If $G_i[\mathcal{N}]$ has a connected component with $< \ell - z$ nodes, we declare failure.
2. Pick an arbitrary unpopular node v such that $|\mathsf{B}(v, w(e_i))| \geq \ell - z$ and delete all vertices in this ball; repeat until no such node exists. Then, pick an arbitrary unpopular node v and delete all vertices in $\mathsf{B}(v, w(e_i))$; repeat until no unpopular node is left. Let $\mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_k$ be the balls created during the process. If a ball contains at least $\ell - z$ unpopular nodes, we call it *big*. Otherwise, we call it *small*.
3. In $G_i[\mathcal{N}]$, shrink each B_j into a single node b_j. A node b_j is *big* if B_j is big and *small* otherwise. We define the *weight* of b_j to be $\mu(b_j) = \frac{|\mathsf{B}_j|}{\ell - z}$. Let the resulting graph with vertex set $\{b_j\}_{j=1}^k$ be D_i.

4. For each connected component C of D_i, do
   (a) Find a spanning tree T_C of C^3 such that all small nodes are leaves. If this is not possible, we declare failure.
   (b) Find (by Lemma 6) a Hamiltonian cycle $P = \{b_1, b_2, \ldots, b_h, b_{h+1} = b_1\}$ over all non-leaf nodes of C such that $\mathsf{dist}_{D_i}(b_j, b_{j+1}) \leq 9w(e_i)$.
5. We create a new color class U of p nodes which will serve as "virtual centers" of the p clusters. These virtual centers are placed in G_i "evenly" as follows. Consider each connected component C in D_i and the corresponding spanning tree T_C of C^3. For each non-leaf node b_j in T_C, let $L(b_j)$ be the set of leaves connected to b_j in T_C, and let $\eta(b_j) = \mu(b_j) + \sum_{b_x \in L(b_j)} \mu(b_x)$ and $\delta_j = \sum_{x=1}^{j} \eta(b_x)$. We attach $\lfloor \delta_i \rfloor - \lfloor \delta_{i-1} \rfloor$ virtual centers to the center of B_i by zero weight edges. If the total number of virtual centers used is not equal to p, we declare failure. Let H_i be the resulting graph (including all popular nodes, unpopular nodes and virtual centers).
6. Find a valid star forest in H_i^{28} using U as centers, which spans $n - q$ nodes (not including the nodes in U) by using F-TEST. If succeeds, we return the star forest found, otherwise we declare failure.

4.2 Analysis of the algorithm

We show that the algorithm succeeds on G_{i^*}. Since we perform F-TEST on $H_{i^*}^{28}$ in which each edge is of length $\leq 28d^*$, the radius of each cluster is at most $28d^*$. Therefore, the approximation ratio is 56.

Let H_{i^*} be the graph obtained by adding virtual centers to G_{i^*} as described above. Let $\mathcal{C}^* = \{C_1^*, \ldots, C_p^*\}$ be the optimal clustering. Let $I^* = \{\nu_1^*, \ldots, \nu_p^*\}$ be the set of cluster centers of $\mathcal{C}^*$ where ν_i^* is the center of C_i^*. We denote the balls grown in step 2 by $\mathsf{B}_1, \ldots, \mathsf{B}_k$. Let ν_i be the center of B_i.

The algorithm may possibly fail in step 1, step 4(a), step 5 and step 6. Obviously G_{i^*} can pass step 1. Therefore, we only check the other three cases.

Step 4(a) : We prove that the subgraph induced by all big nodes are connected in C^3. Indeed, we claim that each small node is adjacent to at least one big node in C from which the proof follows easily. Now we prove the claim. Suppose b_j is a small node and all its neighbors are small. We know that in $G_{i^*}[\mathcal{N}]$, ν_j has at least $\ell - z - 1$ neighbors because ν_j is an unpopular node and thus belongs to some optimal cluster. So we could form a big ball around ν_j, thus contradicting to the fact that ν_j is in a small ball. To find a spanning tree with all small nodes as leaves, we first assign each small node to one of its adjacent node arbitrarily and then compute a tree spanning all the big nodes.

Step 5 : We can see that in each connected component C (with big nodes $b_1, \ldots, b_h$) in D_{i^*}, the total number of virtual centers we have placed is $\sum_{i=1}^{h}(\lfloor \delta_i \rfloor - \lfloor \delta_{i-1} \rfloor) = \lfloor \delta_h \rfloor = \lfloor \sum_{x=1}^{h} \eta(b_x) \rfloor = \lfloor \sum_{b_j \in C} \mu(b_j) \rfloor = \left\lfloor \frac{|C|}{l-z} \right\rfloor$ where $|C| = \sum_{b_j \in C} |\mathsf{B}_j|$, the number of nodes in the connected component of $G_{i^*}[\mathcal{N}]$ corresponding to C. This is at least the number of clusters created for the component C in the optimal solution. Therefore, we can see the total number of virtual centers created is at least p. On the other hand, from Lemma 5(2), we can see that $p(\ell - z) \leq |\mathcal{N}| < (p+1)(\ell - z)$. Hence,

$p = \left\lfloor \frac{|\mathcal{N}|}{\ell - z} \right\rfloor = \left\lfloor \frac{\sum_C |C|}{\ell - z} \right\rfloor \geq \sum_{\mathsf{C}} \left\lfloor \frac{|C|}{\ell - z} \right\rfloor$. where the summation is over all connect components. So, we prove that exactly p virtual centers were placed in G_{i^*}.

Step 6 : We only need to show that there is a perfect matching M between U and the set of optimal centers I^* in $H_{i^*}^{27}$. We consider the bipartite subgraph $Q(U, I^*, E_{H_{i^*}^{27}}(U, I^*))$. From Hall's theorem, it suffices to show that $|N_Q(S)| \geq |S|$ for any $S \subseteq U$, which can be implied by the following lemma.

Lemma 7. *For any $S \subseteq U$, the union of the balls of radius $27d^*$ around the nodes of S, i.e, $\bigcup_{u \in S} \mathsf{B}(u, 27d^*)$, intersects at least $|S|$ optimal clusters in $\mathcal{C}^*$.*

Theorem 3. *There is a 56-approximation for ℓ-DIVERSITY-OUTLIERS.*

5 Further Directions

This work results in several open questions. First, as in [1], we could also try to minimize the sum of the radii of the clusters. However, this seems to be much more difficult, and we leave it as an interesting open problem. Another open problem is to design constant approximations for the problem with any fixed number of outliers, that is, for any given number k, find an optimal clustering if at most k outliers can be removed.

As mentioned in the introduction, our work can be seen as a stab at the more general problem of clustering under instance-level hard constraints. Although arbitrary CL (cannot-link) constraints seems hard to approximate with respect to minimizing the number of clusters due to the hardness of graph coloring [10], other objectives and special classes of constraints, e.g. diversity constraints, may still admit good approximations. Besides the basic ML and CL constraints, we could consider more complex constraints like the rules proposed in the Dedupalog project [4]. One example of such rules says that whenever we cluster two points a and b together, we must also cluster c and d. Much less is known for incorporating these types of constraints into traditional clustering problems and we expect it to be an interesting and rich further direction.

References

1. G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *PODS*, pages 153–162, 2006.
2. G. Aggarwal, T. Feder, K. Kenthapadi, R.Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT*, pages 246–258, 2005.
3. N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. In *J. ACM*, volume 55(5), pages 1–27, 2008.
4. A. Arasu, C. Ré, and D. Suciu. Large-scale deduplication with constraints using Dedupalog. In *ICDE*, pages 952–963, 2009.
5. A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence data bank and its supplement TrEMBL. *Nucleic acids research*, 25(1):31, 1997.
6. N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1):89–113, 2004.
7. A. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, pages 46–55, 2003.
8. R. C.-W.Wong, J. Li, A.-C. Fu, and K.Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *SIGKDD*, pages 754–759, 2006.

9. M. Charikar, S. Khuller, D. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *SODA*, pages 642–651, 2001.
10. I. Davidson and S. Ravi. Intractability and clustering with constraints. In *ICML*, pages 201–208, 2007.
11. C. Dwork, M. Naor, O. Reingold, G. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.
12. D. Feldman, A. Fiat, H. Kaplan, and K. Nissim. Private coresets. In *STOC*, pages 361–370, 2009.
13. G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss. In *VLDB*, pages 758–769, 2007.
14. I. Giotis and V. Guruswami. Correlation clustering with a fixed number of clusters. In *SODA*, pages 1176–1185, 2006.
15. F. Hoppner, F. Klawonn, R. Platz, and S. Str. Clustering with Size Constraints. *Computational Intelligence Paradigms: Innovative Applications*, 2008.
16. X. Ji. *Graph Partition Problems with Minimum Size Constraints*. PhD thesis, Rensselaer Polytechnic Institute, 2004.
17. D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *SIGMOD*, pages 217–228, 2006.
18. B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer-Verlag, 4th edition, 2007.
19. K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*, page 25, 2006.
20. J. Li, K. Yi, and Q. Zhang. Clustering with diversity. *http://arxiv.org/abs/1004.2968*, 2010.
21. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, page 24, 2006.
22. A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS*, pages 223–228, 2004.
23. M.H.Alsuwaiyel. *Algorithms: Design Techniques and Analysis*. World Scienfic, 1998.
24. H. Park and K. Shim. Approximate algorithms for k-anonymity. In *SIGMOD*, 2007.
25. P. Samarati. Protecting respondents' identities in microdata release. *TKDE*, 13(6):1010–1027, 2001.
26. K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML*, pages 1103–1110, 2000.
27. K. Wagstaff, C. Cardie, and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.
28. X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *VLDB*, pages 139–150, 2006.
29. X. Xiao and Y. Tao. m-invariance: Towards privacy preserving re-publication of dynamic datasets. In *SIGMOD*, pages 689–700, 2007.
30. X. Xiao, K. Yi, and Y. Tao. The hardness and approximation algorithms for l-diversity. *EDBT*, 2010.
31. E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, pages 505–512, 2003.