Composable Core-sets for Diversity and Coverage Maximization

[Extended Abstract]

Piotr Indyk
Massachusetts Institute of Technology
Cambridge, MA
indyk@mit.edu

Sepideh Mahabadi
Massachusetts Institute of Technology
Cambridge, MA
mahabadi@mit.edu

Mohammad Mahdian
Google, Inc.
Mountain View, CA
mahdian@alum.mit.edu

Vahab S. Mirrokni
Google, Inc.
New York, NY
mirrokni@gmail.com

ABSTRACT

In this paper we consider efficient construction of "composable core-sets" for basic diversity and coverage maximization problems. A core-set for a point-set in a metric space is a subset of the point-set with the property that an approximate solution to the whole point-set can be obtained given the core-set alone. A composable core-set has the property that for a collection of sets, the approximate solution to the union of the sets in the collection can be obtained given the union of the composable core-sets for the point sets in the collection. Using composable core-sets one can obtain efficient solutions to a wide variety of massive data processing applications, including nearest neighbor search, streaming algorithms and map-reduce computation.

Our main results are algorithms for constructing composable core-sets for several notions of "diversity objective functions", a topic that attracted a significant amount of research over the last few years. The composable core-sets we construct are small and accurate: their approximation factor almost matches that of the best "off-line" algorithms for the relevant optimization problems (up to a constant factor). Moreover, we also show applications of our results to diverse nearest neighbor search, streaming algorithms and map-reduce computation. Finally, we show that for an alternative notion of diversity maximization based on the maximum coverage problem small composable core-sets do not exist.

PODS'14, June 22–27, 2014, Snowbird, UT, USA.
ACM 978-1-4503-2375-8/14/06.
http://dx.doi.org/10.1145/2594538.2594560 .

Categories and Subject Descriptors

F.2.2 [**Nonnumerical Algorithms and Problems**]: Geometrical problems and computations; G.1.6 [**Optimization**]: Constrained optimization

Keywords

Core-set; Diversity; Streaming; Nearest Neighbor; Map-reduce;

1. INTRODUCTION

One of the most popular approaches to processing massive data is to first extract a compact representation (or synopsis) of the data and then perform further processing only on the representation itself. This approach significantly reduces the cost of processing, communicating and storing the data, as the representation size can be much smaller than the size of the original data set. Typically, the representation provides a smooth tradeoff between its size and the representation accuracy. Examples of this approach include techniques such as sampling, sketching, core-sets and mergeable summaries.

In this paper we focus on computing efficient representations for the purpose of *diversity-aware summarization and search*, a topic that has attracted significant attention over the last few years [20, 41, 7, 26, 40, 38, 18, 1, 32]. The goal of this line of research is to design efficient methods for searching and summarizing large data sets in a way that preserves the diversity of the data. In most formulations, the summary is a sub-set of the original data of some predefined size (say k) that maximizes a certain *diversity objective*. For example one could require that the minimum distance between any pair of points in the summary is as large as possible, i.e., the summary does not contain two "highly similar" items. Many other, more refined, diversity objectives have been studied, see Figure 1 for an overview.

In this paper we study specific diversity-aware representations called *composable core-sets*. An α-approximate *composable core-set* for a diversity objective is a mapping from a set S to a subset of S with the following property: for a collection of sets, the maximum diversity of the union of those sets is within an α factor of the maximum diversity

of the union of the corresponding core-sets. That is, one can construct an approximately optimal solution for a given data set by partitioning it into several (possibly overlapping) blocks, computing a core-set for each block, and then solving the problem for the union of the core-sets. Composable core-sets naturally lead to divide-and-conquer solutions to a collection of massive data processing problems. In particular, they have been used for the following tasks:

- Streaming computation: In the data stream model, a sequence of n data elements needs to be processed "on-the-fly" while using only limited storage. Such an algorithm can be easily obtained using composable core-sets [22, 5][1]. Specifically, if a composable core-set for a given problem has size k, we start by dividing the stream of data into $\sqrt{n/k}$ blocks of size $s = \sqrt{nk}$. The algorithm then proceeds block by block. Each block is read and stored in the main memory, its core-set is computed and stored, and the block is deleted. At the end, the algorithm solves the problem for the union of the core-sets. The whole algorithm takes only $O(\sqrt{kn})$ space. The storage can be reduced further by utilizing more than one level of compression, at the cost of increasing the approximation factor.

- Distributed data processing: composable core-sets can be also used to process data in a distributed system, where each machine holds a block of the data. The algorithm is virtually identical to the one for streaming data: for each block, a composable core-set is computed and sent to the central server, where the computation is completed. As an example, this idea is directly applicable in the map-reduce framework [17] and gives an approximation algorithm in one round of map-reduce: Using $\sqrt{n/k}$ mappers, each mapper gets $\sqrt{kn}$ points as input and computes a composable core-set of size k for this set. These sets will be passed to a single reducer. The input of this reducer is the union of the core-sets, which is of size at most $k\sqrt{n/k} = \sqrt{kn}$. It computes and outputs a solution on this union, which by the definition of core-sets is a good approximation to the original problem. Recently, variants of this technique have been applied for optimization under map-reduce framework [28, 31, 8].

- Similarity search: recently, composable core-sets have been used to construct efficient near neighbor search algorithms that maximize the diversity of the answers, both in theory [2] and in practice [1]. This is done by observing that several similarity search algorithms (notably those based on the *Locality-Sensitive Hashing* technique) proceed by hashing each point into multiple buckets. Each query is then answered by retrieving the points stored in the buckets that the query is mapped into. Since the number of points stored in a bucket might be large (which is the case, e.g., when the data set contains one big cluster of close points), the query answering procedure might be slow. To improve the performance, the paper [1] proposed to replace the content of each bucket by its core-set. By collecting the core-sets stored in all relevant buckets and performing the computation over the their union, the algorithm reports a diverse summary of the points close to the query in time that depends on the number of buckets, not the size of the whole data set. This is discussed in more details in Appendix A.

[1]The paper [22] introduced this approach for the special case of k-median clustering. More general formulation of this method with other applications appeared in [5].

The broad applicability of composable core-sets motivates the study of efficient methods for constructing them. This is the task we undertake in this paper.

Our results. In this paper we present a thorough study of composable core-sets for several well-studied diversity maximization problems. Suppose that the set S of interest lives in some metric space $(\Delta, dist)$, and let $div(S)$ be any function that maps a set into a non-negative real number. The goal of the diversity maximization problem is to find a subset S' of S of size k that maximizes the diversity objective.

The specific diversity functions div considered in this paper are described in Figure 1, following the taxonomy of dispersion measures introduced in [13]. For each dispersion function we provide the approximation factor of the composable core-set that we obtain for that function. We note that for all core-sets the approximation factor matches that of the best "off-line" algorithm for the corresponding diversity maximization problem [13] (up to a constant factor). All core-sets are of size k.

The interpretation of the diversity measures is as follows. First, remote-edge [1] and remote-clique [20] correspond to the well-studied diversity notions where the objective is to ensure that no two pairs of points are too "close" to each other, or that an average pair distance of points is not too "low", respectively. Remote-pseudoforest falls in between the two notions, as its goal is to ensure that the *average* distance of a point to its *nearest* neighbor is not too "low". Remote-pseudoforest can be viewed as a diversity analog of the well-studied Chamfer distance [27]. Remote-tree and remote t-tree measure the diversity by the cost of clustering the data using the Single Link algorithm [37]. Similarly, remote-star measures the diversity by the cost of connecting the points to the best center[2]. Finally, remote-matching, remote-cycle and remote-bipartition are more "exotic" combinatorial variations of the aforementioned measures. We include them to complete the table of [13].

All aforementioned notions of diversity are "pairwise", i.e., they are a function of the pair-wise distances between the selected items. We also consider a basic "higher order" notion of diversity which has been previously discussed in the context of diversity maximization[3, 10]. Intuitively, the idea is to model diversity by considering a set of topics that each item covers, and exploring the diversity or the union of topics *covered* by a set of items. More specifically, we consider the scenario where the items are binary vectors of topics, and the diversity of a set of items is equal to their *coverage* over another set of topics, i.e., the weight of the coordinate-wise OR of the item vectors. As before, the goal is to choose a set of size k which maximizes the total coverage. This is directly related to the maximum k-coverage problem that admits a tight $1 - 1/e$-approximation algorithm [19]. We show in Section 4 that this problem does not support composable core-sets of size *polynomial* in k. In particular, for

[2]Note that the values of remote-clique and remote-star objectives are within a factor of $\Theta(k)$ from each other, and thus the core-sets for the two objectives are equivalent up to constant factors.

Problem	Diversity of the point set S	Approx. factor
Remote-edge	$\min_{p,q \in S} dist(p,q)$	$O(1)$
Remote-clique	$\sum_{p,q \in S} dist(p,q)$	$O(1)$
Remote-tree	$wt(MST(S))$, weight of the minimum spanning tree of S	$O(1)$
Remote-cycle	$\min_C wt(C)$ where C is a TSP tour on S	$O(1)$
Remote t-trees	$\min_{S=S_1\|\ldots\|S_t} \sum_{i=1}^{t} wt(MST(S_i))$	$O(1)$
Remote t-cycles	$\min_{S=S_1\|\ldots\|S_t} \sum_{i=1}^{t} wt(TSP(S_i))$	$O(1)$
Remote-star	$\min_{p \in S} \sum_{q \in S \setminus \{p\}} dist(p,q)$	$O(1)$
Remote-bipartition	$\min_B wt(B)$, where B is a bipartition (i.e., bisection) of S	$O(1)$
Remote-pseudoforest	$\sum_{p \in S} \min_{q \in S \setminus \{p\}} dist(p,q)$	$O(\log k)$
Remote-matching	$\min_M wt(M)$, where M is a perfect matching of S	$O(\log k)$
Max k-Coverage	$\sum_{i \le d} \max_{p \in S} p_i$, where p_i denotes the ith coordinate of p	no $\frac{\sqrt{k}}{\log k}$-approx. core-set of size k^β

Table 1: Notions of diversity considered in this paper. We use $S = S_1|...|S_t$ to denote that $S_1 \ldots S_t$ is a partition of S into t sets.

any $\alpha \le \frac{\sqrt{k}}{\log k}$ and any constant $\beta > 0$, there exists a set of instances for which no α-approximate core-set of size k^β exists. As an illustrative example of submodular maximization [33, 25], the maximum coverage problem has been recently studied from distributed computation perspective [16], e.g., in the map-reduce framework [15, 29]. Our negative result for existence of core-sets for this problem implies that one cannot use the simple core-set approach to solve this problem in distributed or streaming settings.

Our techniques. Our techniques for constructing composable core-sets rely on off-line algorithms that solve the corresponding diversity maximization problems. The three algorithms are given in Preliminaries. Our contribution is to show that the solutions produced by those algorithms satisfy the composable core-set properties. The basic idea is to show that, for each algorithm, one can construct a mapping from each element in the optimum solution (to the whole data set) and an element of the core-set. This correspondence is then used to bound the error incurred by the core-set. Note that for the remote-edge diversity measure, this analysis is analogous to the analysis in [2] (although that analysis was focused on the k-center clustering as opposed to the diversity maximization).

Related Work.

Composable core-sets: The notion of core-sets has been introduced in [5]. Informally, a core-set for an optimization problem is a subset of the data with the property that solving the underlying problem on the subset gives an approximate solution for the original data. The notion is somewhat generic, and many variations of core-sets exist. The notion of composable core-sets used in this paper has been implicit in earlier works that used core-sets for streaming applications. For example, the paper [5] (Section 5) specifies composability properties of ϵ-kernels (a variant of core-sets) that are very similar to ours. To avoid confusion, in this paper the term "core-set" always means "composable core-set" according to the definition in the introduction.

The notion of composable core-sets is related to the notion of merge-able summaries introduced in [4]. The main difference between the two notions is that aggregating merge-able summaries does *not* increase the approximation error, while in our case the error amplifies (similarly to [22]). In particular, every merge-able summary that is obtained by taking a sub-set of the data is also a composable core-set, but the opposite does not hold.

Diversity Maximization: The diversity maximization problem studied in this paper generalizes the maximum dispersion problem [24, 20, 9]. This problem has been explored in the context of diversity maximization for recommender systems [20], and commerce search [9]. A 2-approximation greedy algorithm has been developed for the unconstrained variant of this problem [24], and the variant with knapsack constraints [9]. More recently, local search algorithms have been developed to get a 2-approximation algorithm for the maximum dispersion problem under matroid constraints [3, 10].

Diversity in Recommender Systems and Web Search: Ranking and relevance maximization along with diversification have been extensively studied in recommender systems, web search, and database systems. In the context of web search, maximizing diversity has been explored as a post-processing step [11, 39]. Other papers explore ranking while taking into account diversity by a query reformulation for re-ranking the top searches [34] or by sampling the search results by reducing homogeneity [6]. Other methods are based on clustering results into groups of related topics [30], or expectation maximization for estimating the model parameters and reaching an equilibrium [35]. Moreover, in the context of recommender systems, diversification has been explored in various recent papers [41, 40]. For example, topical diversity maximization is discussed in [41], and explanation-based diversity maximization is explored in [40]. Finally, this topic has been also explored in database systems for example by presenting decision trees to users [14].

2. PRELIMINARIES

We start by formalizing the notion of diversity used in this paper.

Definition 1. For a given set $S \subset \Delta$, its **k-diversity** is defined as $div_k(S) = \max_{S' \subset S, |S'|=k} div(S')$. We also refer to the maximizing subset S' as the **optimal k-subset** of S. Note that k-diversity is not defined in the case where $|S| < k$.

Definition 2. Let div be a diversity function defined for subsets of Δ. A function $c(S)$ that maps a set $S \subset \Delta$ into

one of its subsets is called an *α-composable core-set ($\alpha \geq 1$)* for *div*, if for any collection of sets $S_1 \dots S_L \subset \Delta$ with $|S_i| \geq k$, we have

$$div_k(c(S_1) \cup \dots \cup c(S_L)) \geq \frac{1}{\alpha} \cdot div_k(S_1 \cup \dots \cup S_L)$$

The core-set is of size k' if for every S, $|c(S)| \leq k'$. Note that in general k' does not need to be the same as k. For example, in all applications mentioned in the previous section, a core-set of size k^2 would work as well when k is a constant. However, as it turns out, all our positive results give core-sets of size k.

Our algorithms for constructing core-sets are based on existing off-line approximation algorithms for the corresponding diversity maximization problems. In the rest of this section we review three such algorithms: GMM, Local Search and Prefix.

2.1 GMM Algorithm

In this paper we use the following slight variation of the "GMM" algorithm introduced in [21, 36]. The algorithm receives a set of points S, and the parameter k as the input. Initially, it chooses some arbitrary point $a \in S$. Then it repeatedly adds the next point to the output set until there are k points. More precisely, in each step, it greedily adds the point whose minimum distance to the currently chosen points is maximized. This algorithm was also utilized in [13] to find approximation algorithms for several dispersion problems.

Algorithm 1 GMM

Input S: a set of points, k: size of the subset
Output S': a subset of S of size k.

1: $S' \leftarrow$ An arbitrary point a
2: **for** $i = 2, \dots, k$ **do**
3: find $p \in S \setminus S'$ which maximizes $\min_{x \in S'} dist(p, x)$
4: $S' \leftarrow S' \cup \{p\}$
5: **end for**
6: **return** S'

It is easy to see that the running time of the algorithm is $O(nk)$. Also, observe that if we define the radius value $r = \min_{p,q \in S'} dist(p, q)$ as the minimum pairwise distance in the set S', it is easy to see that the following two properties hold:

- $\forall p \in S' : dist(p, S' \setminus \{p\}) \geq r$
- $\forall p \in S : dist(p, S') \leq r$

Such sets S' are said to have the *anticover* property.

2.2 Local Search Algorithm

Algorithm 2 shows the local search algorithm. This was used in [3] to find a subset with approximate maximum diversity under matroid constraints for the case of Remote Clique. The algorithm iteratively improves the current solution by a factor of $(1 + \epsilon/n)$ and finds a more diverse set of k points. Since the initial set contains the two farthest points, the total number of iterations needed is at most $\log_{1+\epsilon/n}(k^2) = O(\frac{n}{\epsilon} \log k)$.

Algorithm 2 Local Search Algorithm

Input S: a set of points, k: size of the subset
Output S': a subset of S of size k.

1: $S' \leftarrow$ An arbitrary set of k points which contains the two farthest points
2: **while** there exists $p \in S \setminus S'$ and $p' \in S'$ such that $div(S' \setminus \{p'\} \cup \{p\}) \geq div(S')(1 + \frac{\epsilon}{n})$ **do**
3: $S' \leftarrow S' \setminus \{p'\} \cup \{p\}$
4: **end while**
5: **return** S'

2.3 Prefix Algorithm

The Prefix algorithm was introduced in [13] which is used to solve the approximate maximum dispersion problem in the case of Remote Pseudo-forest and Remote Matching. Note that the algorithm works only in the case when $k \leq n/2$.

Algorithm 3 PREFIX Algorithm

Input S: a set of points, k: size of the subset
Output S': a subset of S of size k.

1: Run GMM obtaining a set $Y = \{y_1, \cdots, y_k\}$ with corresponding radii $r_1, \cdots, r_k$.
2: $q \leftarrow$ the value from the set $\{1, \cdots, k-1\}$ which maximizes $q \cdot r_q$.
3: $Y_{q+1} \leftarrow$ the prefix subsequence of Y of length $q+1$
4: $Q_i \leftarrow$ vertices of distance at most $r_q/2$ from y_i for $i = 1, \cdots, q+1$.
5: $z \leftarrow \lfloor (q+1)/2 \rfloor$.
6: $\{Q_{i_1}, \cdots, Q_{i_z}\} \leftarrow$ the z sparsest spheres.
7: $S' \leftarrow$ the centers of $\{Q_{i_1}, \cdots, Q_{i_z}\}$
8: Add any set of $k - z$ vertices from $S \setminus \bigcup_{j=1}^{z} Q_{i_j}$ to S'
9: **return** S'

3. COMPOSABLE CORE-SETS FOR DIVERSITY MAXIMIZATION

This section provides algorithms for finding composable core-sets for different notions of diversity defined in Table 1. That is, we run one of the algorithms defined in Preliminaries to get k points in each of the instances of the problem and prove their union is an approximate core-set for the union of the instances.

In all of the following cases, we let $S_1, \cdots, S_L \subset \Delta$ be the subsets of Δ that correspond to the instances of the problem and let $S = \bigcup_{i=1}^{L} S_i$ denote their union. For each such instance S_i, we find a core-set T_i and we let $T = \bigcup_{i=1}^{L} T_i$ denote the union of the core-sets. Also we let $O = \{o_1, \cdots, o_k\}$ be the optimal k-subset of S, that is the subset of k points which maximizes the diversity. Moreover, we define $O_i = \{o \in O \cap S_i | \forall j < i : o \notin S_j\}$ to be the set of points from the optimal set in each of the instances (we impose extra condition in order to make O_i's a partition of O).

Next, for each notion of diversity, we describe how to choose T_i and compare k-diversity of T with that of S, which is equal to the diversity of O.

3.1 Remote Edge

LEMMA 1. *The GMM algorithm computes a 3-approximate composable core-set for the Remote Edge problem.*

PROOF. We run the GMM algorithm on each of the sets S_i and let $T_i = GMM(S_i)$ be the point set returned by the GMM and we let r_i denote the radius of T_i. Let $T = \bigcup_{i=1}^{L} T_i$ denote the union of the core-sets, and set $r = \max_i r_i$ to be the maximum radius over the instances. The goal is to prove that $div_k(T) \geq div_k(S)/3$.

Define a mapping $f : O_i \to T_i$ which maps each point $o \in O_i$ to one of its closest points in the set T_i, i.e., $dist(o, f(o)) = dist(o, T_i)$. By the anticover property of GMM we have $dist(o, f(o)) \leq r_i \leq r$. Note that since O_i's form a partition of O, for any $o \in O$, we can define $f(o) = f_i(o)$ if $o \in O_i$.

It is easy to see that for any i, since T is a superset of T_i, then $div_k(T) \geq div(T_i) = r_i$ and thus $div_k(T) \geq r$. Next, note that if for two points $o_1, o_2 \in O$, we have $f(o_1) = f(o_2)$, then

$$\begin{aligned} div(O) &\leq dist(o_1, o_2) \leq dist(o_1, f(o_1)) + dist(o_2, f(o_2)) \\ &\leq 2r \leq 2div_k(T) \end{aligned}$$

and the lemma is proved. Otherwise f is a 1-to-1 mapping. Now if $div(O) \leq 3r \leq 3div_k(T)$ then in this case the lemma is proved as well. Otherwise, we can assume that for any pair of points $o_1, o_2 \in O$, $dist(o_1, o_2) \geq 3r$ and thus $div(O) \geq 3r$. Hence, by triangle inequality

$$\begin{aligned} &dist(f(o_1), f(o_2)) \\ &\geq dist(o_1, o_2) - dist(o_1, f(o_1)) - dist(o_2, f(o_2)) \\ &\geq div(O) - 2r \\ &\geq div(O) - 2div(O)/3 \\ &\geq div(O)/3 \end{aligned}$$

since this holds for any pair o_1, o_2, the set $\{f(o_1), \cdots, f(o_k)\}$ has diversity at least $div(O)/3$ and thus $div_k(T) \geq div(O)/3$ and the lemma is proved. □

3.2 Remote Clique, Remote Star and Remote Bipartition

In this section, we show that the local search algorithm gives a constant-factor approximation for the following diversity notions: Remote Clique, Remote Star and Remote Bipartition.

LEMMA 2. *The local search algorithm computes a constant-factor approximate composable core-set for the remote-clique problem.*

PROOF. We run the Local Search algorithm on each of the sets S_i and let $T_i = LS(S_i)$ be the point set returned by the Local Search and let r_i represent the normalized diversity of the corresponding sets T_i, i.e., $r_i = \frac{1}{\binom{k}{2}} div(T_i)$ and set $r = \max_i r_i$.

Claim 1. There exists a 1-to-1 mapping $f : O \to T$ such that $dist(o, f(o)) \leq 25r$ for any $o \in O$

PROOF. Build an unweighted bipartite graph $G_x = (V_O, U_T, E_x)$ with vertices of one side corresponding to O and vertices of the other side corresponding to T as follows. For any $o \in O$ and $s \in T$, we connect $v_o \in V_O$ to $u_s \in U_T$ iff $dist(o, s) \leq x \times r$. Now, take any $o \in O$ and suppose that $o \in O_i \setminus T_i$, that is, o is in the ith instance but has not been selected by *LocalSearch* algorithm. However, since no more improvement on the set T_i could be made, we have

$$\sum_{s \in T_i} dist(o, s) \leq (k-1)(1 + \frac{\epsilon}{n}) r_i \leq kr$$

Note that since $|T_i| = k$, thus for at least $(1 - 1/x)$ fraction of the values s in the above equation, we have $dist(o, s) \leq xr$ and therefore the corresponding edges in the graph G_x exist. Thus the degree of each vertex v_o corresponding to $o \in O \setminus T$ is at least $k(1 - 1/x)$.

First, take the graph G_3. If G_3 has a matching which saturates the vertices of V_O, then the claim is proved. Otherwise, let M be a maximal matching in G_3 such that for any point $o \in O \cap T$, the corresponding vertices v_o and u_o are matched together. This means that the points corresponding to the set of unmatched vertices in U_T (which we denote by $T \setminus M$) is disjoint from O, and also $O \setminus M$ is disjoint from T. Let $A = O \setminus M$ be the set of points which corresponds to the unmatched vertices. Then for any point $a \in A$, since $a \notin T$, the degree of v_a is at least $2k/3$, and since M is a maximal matching, all the neighbors of v_a should be matched in M. Therefore there are at least $2k/3$ points $o \in O \setminus \{a\}$ such that $dist(o, a) \leq 6r$.

Now take the graph G_{25}. If all vertices in $V_A = V_O \setminus M$ are neighbors to all vertices in $U_T \setminus M$, then clearly G_{25} has a saturating matching for O and thus the claim is proved. Otherwise there exists a point $a \in A$ and $s \in T \setminus M$ such that $dist(a, s) > 25r$.

Let $B \subset O$ be the set of points whose distance is at most $6r$ from a. Then as we proved earlier $|B| > 2k/3$. Hence, if we replace the point a in the set O with the point s to get the set O' (note that since $T \setminus M$ is disjoint from O, we have $s \notin O$), the diversity will increase as follows.

$$\begin{aligned} div(O') - div(O) &= \sum_{o \in O \setminus \{a\}} dist(s, o) - dist(a, o) \\ &= \sum_{o \in B \setminus \{a\}} dist(s, o) - dist(a, o) \\ &+ \sum_{o \in O \setminus B} dist(s, o) - dist(a, o) \\ &\geq \sum_{o \in B \setminus \{a\}} dist(a, s) - 2dist(a, o) \\ &- \sum_{o \in O \setminus B} dist(a, s) \\ &\geq \frac{2k}{3} \times (dist(a, s) - 12r) - \frac{k}{3} \times dist(a, s) \\ &= \frac{k}{3}(dist(a, s) - 24r) \geq kr/3 \end{aligned}$$

which contradicts the fact that O has the optimal diversity. Therefore the claim holds. □

As claim 1 suggests, there is a 1-to-1 mapping between the vertices of O and the vertices of T such that for each $o \in O$ we have $dist(o, f(o)) \leq 25r$. First of all note that if $\binom{k}{2} \times r \geq div(O)/51$ the theorem is proved since for one of

the T_i we have $div(T_i) = \binom{k}{2} \times r$ and thus

$$div_k(T) \geq div(T_i) = \binom{k}{2} \times r \geq div(O)/51$$

Otherwise, we have that

$$\begin{aligned} div_k(T) &\geq \sum_{o_1,o_2 \in O} dist(f(o_1), f(o_2)) \\ &\geq \sum_{o_1,o_2 \in O} dist(o_1,o_2) - dist(o_1, f(o_1)) - dist(o_2, f(o_2)) \\ &\geq div(O) - \binom{k}{2} \times 50r \\ &\geq div(O)(1 - 50/51) = div(O)/51 \end{aligned}$$

So the lemma is proved and the algorithm computes a 51-approximate core-set of size k.

COROLLARY 1. *Local Search algorithm computes a constant factor core-set for the minimum star and minimum bipartition problems as well.*

PROOF. First note that for a set of k points Q, a star is the tree achieved by connecting one point to all the others, and its weight is sum of the weights of its edges. Also a bipartition of Q is a bipartite graph which divides the vertices of Q into two parts of cardinality $k/2$ and its weight is the sum of all the edges between the two parts. It can easily be seen that

- by symmetry $wt(\text{minimum star}(Q)) \leq 2wt(\text{ clique}(Q))/k$
- by triangle inequality

$$wt(\text{ clique}(Q)) \leq k \times wt(\text{minimum star}(Q))$$

and that

- $wt(\text{minimum bipartition}(Q)) \leq wt(\text{ clique}(Q))$
- by triangle inequality

$$wt(\text{ clique}(Q)) \leq 5 \times wt(\text{minimum bipartition}(Q))$$

Therefore the same algorithm computes a constant factor core-set for these two problems as well. □

3.3 Remote tree, Remote Cycle, Remote t-trees and Remote t-cycles

LEMMA 3. *The GMM algorithm computes a 6-approximate core-set for the remote-tree problem.*

PROOF. We run the GMM algorithm on each of the sets S_i and let $T_i = GMM(S_i)$ be the point set returned by the GMM and we let r_i denote the radius of T_i. Let $T = \bigcup_{i=1}^{L} T_i$ denote the union of the core-sets, and set $r = \max_i r_i$ to be the maximum radius over the instances. Now define a mapping (this time not a 1-to-1) $f : O_i \to T_i$ which maps each point $o \in O_i$ to one of its closest points in the set T_i, i.e., $dist(o, f(o)) = dist(o, T_i)$. By anticover property of GMM we have $dist(o, f(o)) \leq r_i \leq r$.

It is easy to see that for any i, $div_k(T) \geq div(T_i) \geq (k-1)r_i$ (since the minimum pairwise distance in T_i is r_i), and thus $div_k(T) \geq (k-1)r$. Now if $div(O) \leq 3(k-1)r \leq 3div_k(T)$, then the lemma is proved. Otherwise let $F = range(f) = \{f(o) | o \in O\}$ (note that F is a subset of T), and let $F^+ \subset T$ be an arbitrary superset of F of size k. Then by triangle inequality and shortcutting

$$\begin{aligned} div(O) = wt(MST(O)) &\leq wt(MST(F)) + kr \\ &\leq wt(MST(F)) + 2(k-1)r \end{aligned}$$

which uses the fact that $k > 1$, otherwise any one point is a solution. Next, note that given the $MST(F^+)$, we can double the edges and traverse them using DFS and remove the vertices not in F by shortcutting. Hence, by triangle inequality, we find a Hamiltonian cycle of length at most $2wt(MST(F^+))$ on the set F, therefore we have $wt(MST(F)) \leq 2wt(MST(F^+))$ and thus

$$\begin{aligned} div_k(T) &\geq wt(MST(F^+)) \\ &\geq wt(MST(F))/2 \\ &\geq \frac{1}{2}[div(O) - 2(k-1)r] \\ &\geq \frac{div(O)}{2} - \frac{div(O)}{3} \\ &\geq \frac{div(O)}{6} \end{aligned}$$

□

LEMMA 4. *The same algorithm computes a 6 core-set for the Remote-t-tree. The proof is very similar to that for the Remote tree and hence moved to Appendix B.*

COROLLARY 2. *Note that since the minimum TSP tour is within a factor 2 of the MST, the above algorithm also computes a constant factor core-set for the remote-cycle problem and remote t-cycle problem.*

3.4 Remote Pseudoforest and Remote Matching

LEMMA 5. *The GMM algorithm computes a $O(\log k)$ core-set for the remote-pseudoforest problem.*

PROOF. We run the GMM algorithm on each of the sets S_i and let $T_i = GMM(S_i)$ be the point set returned by the GMM. Let $T = \bigcup_{i=1}^{L} T_i$ denote the union of the core-sets.

It is shown in page 11 of the paper [13] that when we run the Prefix algorithm on an input set A, the diversity achieved by this algorithm is at least $q \cdot r_q/4$ and that $q \cdot r_q/4 \geq div_k(A)/O(\log k)$. Next, we compare running the PREFIX algorithm on the set S and on the set T. Let $r_1^S, \cdots, r_k^S$ be the radii defined in line 1 of Algorithm 2.3 , and let q^S be the index chosen in line 2, when we run it on the set S. Similarly, let us define $r_1^T, \cdot, r_k^T$ and q^T, when we run the algorithm on the set T.

However by Lemma 1, GMM algorithm computes a core-set for minimum pairwise distances. Together with the fact that running GMM in the Prefix algorithm on the sets S and T preserves the radii upto a constant factor, we get that $r_i^T \geq r_i^S/c$, for any value of $i \leq k$ and some constant c. The diversity achieved by the prefix algorithm is therefore $div_k(T) \geq q^T \cdot r_{q^T}^T/4 \geq q^S \cdot r_{q^S}^T/4 \geq q^S \cdot r_{q^S}^S/(4c) \geq \frac{div_k(S)}{O(\log k)}$ □

For the same reason the GMM algorithm computes a $O(\log k)$ core-set for the remote-matching problem as well

with the only difference that the value of the matching achieved by the prefix algorithm when we run in on the input set A is at least $qr_q/8$ instead of $qr_q/4$.

4. NON-EXISTENCE OF CORESET FOR THE MAX k-COVERAGE

An instance of the max k-coverage problem is a collection of sets. The objective is to find k sets in this collection whose union has the maximum size.

THEOREM 1. *For any $\alpha < \frac{\sqrt{k}}{\log k}$ and any constant $\beta > 1$, there is no α-approximate core-set of size k^β for the max k-coverage problem.*

PROOF. Let $\mathcal{U} = \{1, \ldots, N\}$ for a large N and $k = m^2$. We construct a number of instances of the problem as follows: For every subset $S \subset \mathcal{U}$ of size m^2, we have an instance I_S consisting of all m-subsets of S. Assume, for contradiction, that there is an α-approximate core-set, and let C_S denote the core-set on the instance I_S.

Now, fix a m^2-set S, and let R be a random m-subset of S. For each fixed $A \in C_S$, the random variable $|A \cap R|$ is distributed according to the binomial distribution $Bin(m, \frac{1}{m})$. The probability that the value of this variable is at least t is at most $\binom{m}{t}\frac{1}{m^t} < \frac{1}{t!}$. With $t = \log m$, this probability is at most $O(m^{-c})$ for every constant c. Using the union bound and the fact that $|C_S| \leq m^{2\beta}$, we get: $\Pr[\exists A \in C_S : |A \cap R| > \log m] < O(m^{2\beta - c})$ for every constant c. We say that R is an easy subset of S if $\exists A \in C_S : |A \cap R| > \log m$. Therefore for every S, at most a $O(m^{-\gamma})$ fraction of the m-subsets of S are easy, for every γ.

We construct a graph whose vertex set is the set of all m^2 subsets of $\mathcal{U}$. Two m^2-sets S_1 and S_2 in this graph are adjacent if $|S_1 \setminus S_2| = m$. We say that S_1 marks a neighbor S_2 as *bad* if $S_1 \setminus S_2$ is an easy subset of S_1. By the above argument, each vertex S_1 marks at most an $O(m^{-\gamma})$ fraction of its neighbors as bad. Since the total indegree of nodes in a graph is equal to the total outdegree, there must be a vertex S_1 in this graph such that at most an $O(m^{-\gamma})$ fraction of its neighbors have marked S_1 as bad. Therefore, at most an $O(m^{-\gamma})$ fraction of the neighbors of S_1 have either marked S_1 as bad or S_1 has marked them as bad. We call these neighbors the *bad* neighbors of S_1, and the remaining neighbors the *good* ones.

We now pick a collection of m^2-sets $S_1, S_2, \ldots$ as follows: S_1 is the vertex defined above. S_2 is an arbitrary good neighbor of S_1. S_{i+1} is a good neighbor of S_1 such that $S_{i+1} \setminus S_1$ does not intersect any of the sets $S_j \setminus S_1$ for $j \leq i$. We argue that for any $i < m^2$, there is a set S_{i+1} with the above properties that we can pick. This is because for any $x \in \mathcal{U} \setminus S_1$, the fraction of neighbors of S_1 that contain x is precisely $\frac{m}{N - m^2}$. Therefore, by the union bound, the fraction of neighbors of S_1 that contain any of the elements of $S_j \setminus S_1$ for $j \leq i$ is at most $\frac{m^2 i}{N - m^2}$, which is less than $1/2$ for $N > 3m^4$. This means that at most a $\frac{1}{2} + O(m^{-\gamma}) < 1$ fraction of neighbors of S_1 either have intersection with some $S_j \setminus S_1$ for $j \leq i$ or are bad. Thus, we can find S_{i+1} with the desired properties, for $i < m^2$.

Now, consider the union of the instances $I_{S_1}, I_{S_2}, \ldots, I_{S_{m^2 - m}}$. This instance has a perfect k-coverage solution: pick m non-overlapping m-subsets of S_1 to cover S_1, and for every $i > 1$, pick the m-subset $S_i \setminus S_1$. The total number of subsets picked is $m + m^2 - m = k$, and all of the $O(m^3)$ elements in $S_1 \cup \cdots \cup S_{m^2 - m}$ are covered. On the other hand, by our construction, we know that for every $i > 1$, $S_i \setminus S_1$ is not an easy subset of S_i. Therefore, any set in C_{S_i} covers at most $\log m$ elements of $S_i \setminus S_1$, and for $j \neq i$, C_{S_j} does not cover any element of $S_i \setminus S_1$. Thus, a collection of k sets in $\bigcup_i C_{S_i}$ can cover at most $k \log m$ elements in $\bigcup_i (S_i \setminus S_1)$ plus m^2 elements of S_1. This means that the ratio of the best solution on the union of these instances and the solution that is limited to the union of the core-sets is at most $\frac{m^2 \log m + m^2}{m^3} < \frac{\log k}{\sqrt{k}}$. □

5. CONCLUSIONS AND OPEN PROBLEMS

In this paper we presented constructions of composable core-sets for a wide range of diversity measures. As described in the introduction and Appendix A, our core-sets can be directly used to obtain constant-factor approximation algorithms (for the respective diversity measures) in the context of data stream computation, distributed data processing and diverse nearest neighbor search. Some of those implications are essentially known (in particular, the streaming and distributed algorithms for the remote-edge and the remote-star measures [22, 12, 23] and the nearest neighbor search algorithms for the remote-edge measure [2]). Other results and implications are, to the best of our knowledge, new.

Our work raises several interesting open questions. Are there any other applications of composable core-sets, in addition to the ones listed in this paper ? Is there a general characterization of diversity measures for which small composable core-sets exist ? Is it possible to obtain better approximation factors ?

Acknowledgments.

This work was supported in part by grants from MADALGO Center and NSF.

6. REFERENCES

[1] S. Abbar, S. Amer-Yahia, P. Indyk, and S. Mahabadi. Real-time recommendation of diverse related articles. In *WWW*, pages 1–12, 2013.

[2] S. Abbar, S. Amer-Yahia, P. Indyk, S. Mahabadi, and K. R. Varadarajan. Diverse near neighbor problem. In *SoCG*, pages 207–214, 2013.

[3] Z. Abbassi, V. S. Mirrokni, and M. Thakur. Diversity maximization under matroid constraints. In *KDD*, pages 32–40, 2013.

[4] P. K. Agarwal, G. Cormode, Z. Huang, J. Phillips, Z. Wei, and K. Yi. Mergeable summaries. In *Proceedings of the 31st symposium on Principles of Database Systems*, pages 23–34. ACM, 2012.

[5] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *Journal of the ACM (JACM)*, 51(4):606–635, 2004.

[6] A. Anagnostopoulos, A. Z. Broder, and D. Carmel. Sampling Search-Engine Results. In *WWW*, 2006.

[7] A. Angel and N. Koudas. Efficient diversity-aware search. In *Proceedings of the 2011 international conference on Management of data*, SIGMOD '11, pages 781–792, New York, NY, USA, 2011. ACM.

[8] M.-F. Balcan, S. Ehrlich, and Y. Liang. Distributed clustering on graphs. In *NIPS*, page to appear, 2013.
[9] S. Bhattacharya, S. Gollapudi, and K. Munagala. Consideration set generation in commerce search. In *WWW*, pages 317–326, 2011.
[10] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS*, pages 155–166, 2012.
[11] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.
[12] , M. Charikar, C. Chekuri, T. Feder and R. Motwani. Incremental Clustering and Dynamic Information Retrieval. SIAM J. Comput. 33(6), 2004.
[13] B. Chandra and M. M. Halldórsson. Approximation algorithms for dispersion problems. *Journal of algorithms*, 38(2):438–465, 2001.
[14] Z. Chen and T. Li. Addressing Diverse User Preferences in SQL-Query-Result Navigation. In *SIGMOD*, 2007.
[15] F. Chierichetti, R. Kumar, and A. Tomkins. Max-cover in map-reduce. In *WWW*, pages 231–240, 2010.
[16] G. Cormode, H. J. Karloff, and A. Wirth. Set cover algorithms for very large datasets. In *CIKM*, pages 479–488, 2010.
[17] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI*, pages 137–150, 2004.
[18] M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Record*, pages 41–47, 2010.
[19] U. Feige. A threshold of ln for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
[20] S. Gollapudi and A. Sharma. An axiomatic framework for result diversification. In *WWW*, pages 381–390, 2009.
[21] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, pages 293–306, 1985.
[22] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. *STOC*, 2001.
[23] S. Guha, Tight results for clustering and summarizing data streams. *ICDT*, 2009.
[24] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, 1997.
[25] R. K. Iyer and J. A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, pages 2436–2444, 2013.
[26] A. Jain, P. Sarda, and J. R. Haritsa. Providing diversity in k-nearest neighbor query results. In *PAKDD*, pages 404–413, 2004.
[27] M. W. Jones, J. A. Baerentzen, and M. Sramek. 3d distance fields: A survey of techniques and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):581–599, 2006.
[28] H. J. Karloff, S. Suri, and S. Vassilvitskii. A model of computation for mapreduce. In *SODA*, pages 938–948, 2010.
[29] R. Kumar, B. Moseley, S. Vassilvitskii, and A. Vattani. Fast greedy algorithms in mapreduce and streaming. In *SPAA*, pages 1–10, 2013.
[30] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram. A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. In *WWW*, 2004.
[31] S. Lattanzi, B. Moseley, S. Suri, and S. Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In *SPAA*, pages 85–94, 2011.
[32] H. Lin, J. Bilmes, and S. Xie. Graph-based submodular selection for extractive summarization.
[33] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions, 1978.
[34] F. Radlinski and S. T. Dumais. Improving Personalized Web Search using Result Diversification. In *SIGIR*, 2006.
[35] D. Rafiei, K. Bharat, and A. Shukla. Diversifying web search results. In *WWW*, pages 781–790, 2010.
[36] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Facility dispersion problems: Heuristics and special cases. *Algorithms and Data Structures*, pages 355–366, 1991.
[37] R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.
[38] M. J. Welch, J. Cho, and C. Olston. Search result diversity for informational queries. In *WWW*, pages 237–246, 2011.
[39] D. Xin, H. Cheng, X. Yan, and J. Han. Extracting Redundancy-Aware Top-k Patterns. In *SIGKDD 2006*, 2006.
[40] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia. Recommendation diversification using explanations. In *ICDE*, pages 1299–1302, 2009.
[41] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.

APPENDIX

A. APPLICATION TO APPROXIMATE NEAREST NEIGHBOR

In this section, we briefly describe how we can apply the aforementioned core-sets to solve k-diverse near neighbor problem. The problem is defined as follows. Given a query point $q \in \Delta$, the goal is to report the maximum diversity set S of k points in the ball of radius r around q. The points in the set S are chosen from a dataset of points $P \subset \Delta$ of size n which is given to the algorithm at the preprocessing time. We would like to answer queries in sublinear time which necessitates solving the approximate problem. The approximate k-diverse Near Neighbor is defined as follows. For some approximation factors $c > 1$ and $\alpha > 1$, we allow the points of the reported set S to be within distance cr of the query point, i.e., $S \subset P \cap B(q, cr)$. Moreover, we require that the diversity of the set S is within an α factor of the k-diversity of the optimal set, i.e., $div(S) \geq \frac{1}{\alpha} div_k(P \cap B(q, r))$.

The definitions and algorithm mentioned here are from [1, 2] and are only included for completeness. Please see the

original papers for the detailed theoretical [2] or experimental [1] analysis of its performance. The algorithm uses the techniques of *locality-sensitive hashing*. Its basic idea is to hash the data and query points in a way that the probability of collision is much higher for points that are close to each other, than for those which are far apart. Formally, we require the following.

Definition 3. A family $\mathcal{H} = h : \Delta \to U$ is (r_1, r_2, p_1, p_2)-sensitive for $(\Delta, dist)$, if for any $p, q \in \Delta$, we have

- if $dist(p, q) \leq r_1$, then $Pr_{\mathcal{H}}[h(q) = h(p)] \geq p_1$
- if $dist(p, q) \leq r_2$, then $Pr_{\mathcal{H}}[h(q) = h(p)] \leq p_2$

In order for a locality sensitive family to be useful, it has to satisfy inequalities $p_1 > p_2$ and $r_1 < r_2$.

Given an LSH family, the algorithm creates L hash functions $g_1, g_2, \cdots, g_L$, as well as the corresponding hash arrays $A_1, A_2, \cdots, A_L$. Each hash function is of the form $g_i =< h_{i,1}, \cdots, h_{i,K} >$, where $h_{i,j}$ is chosen uniformly at random from $\mathcal{H}$. Then each point p is stored in bucket $g_i(p)$ of A_i for all $1 \leq i \leq L$. In order to answer a query q, we then search points in $A_1[g_1(q)] \cup \cdots \cup A_L[g_L(q)]$. That is, in each array, we only retrieve points from the single bucket which corresponds to the query point q.

The aforementioned algorithm does not limit the number of points stored in a bucket, and hence its running time is unbounded. To avoid this problem we proceed as follows. During the preprocessing stage, for each of the buckets in all arrays A_i, we replace the bucket content by its core-set, using the algorithms presented in this paper. Then, given a query point q, we collect the core-set points from the corresponding buckets of q, i.e, $T = \bigcup_i c(A_i[g_i(q)])$. Since the core-sets has polynomial size in k, and the total number of hash functions L is sublinear in n, then the total number of points we collect in T is sublinear in n. By properties of core-sets, the k-diversity of the set T is comparable to k-diversity of the set $S = \bigcup_i A_i[g_i(q)]$. Moreover, one can set the parameters of LSH (i.e., L and K) such that with high probability the two following conditions hold:

- $P \cap B(q, r) \subset S$, every point in the r-neighborhood of q is included in the set S.
- $S \subset B(q, cr)$, any retrieved point is in the cr-neighborhood of q, i.e., there are no outliers.

Thus, if β shows the approximation factor of the core-set, then the value of $div_k(T)$ is within β factor of the value $div_k(S)$. Since S is a superset of $(P \cap B(q, r))$, we get that $div_k(T)$ is within β-factor of $div_k(P \cap B(q, r))$. Therefore we can run the "offline" algorithm on the set T to get an approximate k-diverse subset of T whose diversity approximates the diversity of the optimal set.

More specifically, if β' shows the best approximation factor for the "offline" version of diversity approximation, with running time of $T(m)$ on m points, we can get final bounds as follows. We can achieve approximation factor $\alpha = \beta\beta'$, with query time of

$$O(T(k(\log k)^{\frac{c}{c-1}} n^{\frac{1}{c-1}}) + \frac{d}{r}(\log k)^{\frac{c}{c-1}} n^{\frac{1}{c-1}} \log n)$$

and data structure space equal to $O((n \log k)^{1+\frac{1}{c-1}} + nd)$.

B. PROOF OF LEMMA 4

Let $wt(MST_t(A))$ of a set of points A, denote the minimum sum of the weights of spanning trees achieved by dividing A into t sets, i.e., $\min_{A=A_1|\ldots|A_t} \sum_{i=1}^{t} wt(MST(A_i))$, where $A = A_1|\ldots|A_t$ is a partition of A into t sets.

We run the GMM algorithm on each of the sets S_i and let $T_i = GMM(S_i)$ be the point set returned by the GMM and we let r_i denote the radius of T_i. Let $T = \bigcup_{i=1}^{L} T_i$ denote the union of the core-sets, and set $r = \max_i r_i$ to be the maximum radius over the instances. Now define a mapping (not necessarily a 1-to-1) $f : O_i \to T_i$ which maps each point $o \in O_i$ to one of its closest points in the set T_i, i.e., $dist(o, f(o)) = dist(o, T_i)$. By properties of GMM we have $dist(o, f(o)) \leq r_i \leq r$.

First of all, note that it only makes sense if $t \leq k/2$ otherwise in any optimum solution, at least $2t-k$ of the partitions include exactly one of the k points and therefore incur no cost. So instead we could consider the problem of choosing $k' = k-t$ points and having $t' = 2(k-t)$ partitions in which $t' \leq k'/2$.

It is easy to see that for any i, $div_k(T) \geq div(T_i) \geq (k-t)r_i$ (since the minimum pairwise distance in T_i is r_i), and thus $div_k(T) \geq (k-t)r$. Now if $div(O) \leq 3(k-t)r \leq 3div_k(T)$, then the lemma is proved. Otherwise let $F = range(f) = \{f(o) | o \in O\}$ (note that F is a subset of T), and let $F^+ \subset T$ be an arbitrary superset of F of size k. Then by triangle inequality and shortcutting

$$\begin{aligned} div(O) = wt(MST_t(O)) &\leq wt(MST_t(F)) + kr \\ &\leq wt(MST_t(F)) + 2(k-t)r \end{aligned}$$

which uses the fact that $t \leq k/2$. Next, note that given the $MST_t(F^+)$, we can double the edges and traverse them using DFS and remove the vertices not in F by shortcutting. Hence, by triangle inequality, we find a Hamiltonian cycle in each part of the partition with total length at most $2wt(MST_t(F^+))$ on the set F, therefore we have $wt(MST_t(F)) \leq 2wt(MST_t(F^+))$ and thus

$$\begin{aligned} div_k(T) &\geq wt(MST_t(F^+)) \geq wt(MST_t(F))/2 \\ &\geq \frac{1}{2}[div(O) - 2(k-t)r] \\ &\geq div(O)/2 - div(O)/3 \\ &\geq div(O)/6 \end{aligned}$$