Diversity Maximization in the Presence of Outliers

Daichi Amagata

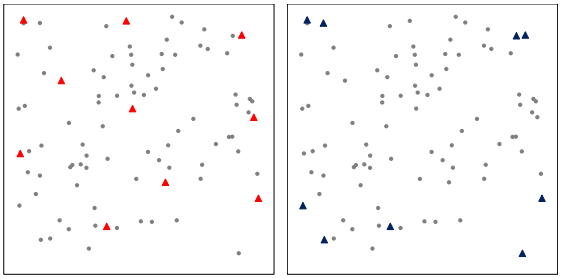
Osaka University
amagata.daichi@ist.osaka-u.ac.jp

Abstract

Given a set X of n points in a metric space, the problem of diversity maximization is to extract a set S of k points from X so that the diversity of S is maximized. This problem is essential in AI-related fields, such as web search, databases, recommender systems, and data mining. Although there have been extensive studies of this problem, these studies assume that X is clean. This usually does not hold, because real-world datasets usually contain outliers. The state-of-the-art algorithm for the diversity maximization problem is based on furthest point retrieval, which is too sensitive to outliers. We therefore address the problem of diversity maximization with outliers and propose two algorithms with performance guarantee. The first algorithm runs in $O((k+z)n)$ time, guarantees $\frac{1}{2}$-approximation, and returns no outliers, where z is the number of outliers. The second algorithm runs in $O(kz)$ time (which is independent of n), guarantees $\frac{1}{6(1+\epsilon)}$-approximation, and returns no outliers with constant probability. We conduct experiments on real datasets to demonstrate the effectiveness and efficiency of our algorithms.

1 Introduction

Given a set X of n points in a metric space, the problem of diversity maximization is to extract a set S of k points from X so that the diversity of S (or dissimilarity between the k points in S) is maximized. This is an important problem in AI-related fields, such as web search (Ceccarello, Pietracaprina, and Pucci 2018), databases (Agarwal, Sintos, and Steiger 2020), recommender systems (Hirata et al. 2022), and data mining (Bauckhage, Sifa, and Wrobel 2020). In the above applications, the sizes of datasets are growing, as we have many sources that generate data. Because of this, analysis of these large datasets and/or building machine-learning models on them often face a challenge of efficiency. Extracting a summary, i.e., a set of *representative* points, from a given dataset is a promising approach to overcoming this challenge, and the diversity maximization problem can output such a summary (Ceccarello, Pietracaprina, and Pucci 2020; Moumoulidou, McGregor, and Meliou 2021; Zadeh et al. 2017). This is because it can control the summary size, i.e., k, and the summary preserves the diversity of (or the information on) a given dataset as much as possible.

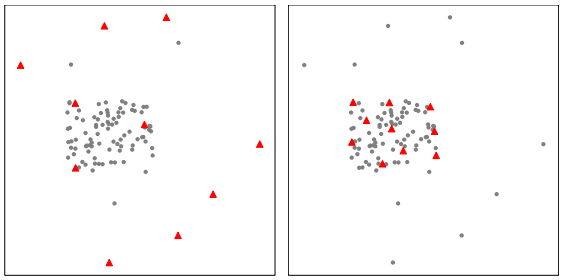


(a) Max-Min diversification (b) Max-Sum diversification

Figure 1: Difference between Max-Min and Max-Sum diversification ($k = 10$). Triangles are selected as solutions.

Given a set S of k points in X, the diversity of S is usually evaluated by an objective function. The most frequently used objective functions are Max-Min and Max-Sum. Given X and k, the Max-Min diversification problem is to select k points in X so that the minimum distance between any two distinct points in a solution set S is maximized. The Max-Sum diversification problem is to select k points in X so that the sum of the distances between all two distinct points in S is maximized. Figure 1 compares the $k = 10$ points (colored triangles) selected by the Max-Min and Max-Sum diversification problems. Figure 1(b) illustrates that the Max-Sum diversification problem returns S having similar points. On the other hand, Figure 1(a) illustrates that the result set obtained by the Max-Min diversification problem is distributed uniformly in the data space. This result is better as a summary of a given dataset, so this paper considers Max-Min as objective function.

Due to the effectiveness of the Max-Min diversification problem, there exist extensive works on this problem (and its variants) (Addanki et al. 2022; Aghamolaei, Farhadi, and Zarrabi-Zadeh 2015; Borassi et al. 2019; Drosou and Pitoura 2014; Erkut, Ülküsal, and Yenicerioğlu 1994; Indyk et al. 2014; Moumoulidou, McGregor, and Meliou 2021; Ravi, Rosenkrantz, and Tayi 1994; Wang, Fabbri, and Mathioudakis 2022). Because this problem is NP-hard, these works devised error-bounded approximation algorithms. The state-of-the-art algorithm for the Max-Min diversification problem is GMM. (Section 2.2 introduces this



(a) GMM (b) GREEDY

Figure 2: Result sets obtained by GMM (existing algorithm) and GREEDY (one of our algorithms) when $k = 10$

algorithm in detail.) Given a set X of n points in a metric space, GMM returns a $\frac{1}{2}$-approximation result in $O(kn)$ time. It is known that, unless P = NP, this bound is tight, i.e., cannot be improved in polynomial time.

One main issue of the above existing works is their assumption: a given dataset X is clean, i.e., X contains no outliers. This usually does not hold, because real-world datasets usually contain outliers that exist far from the other points (Bhaskara, Vadgama, and Xu 2019; Dahiya et al. 2021; Im et al. 2020; Wang, Guo, and Ding 2021). Unfortunately, the existing algorithms for the Max-Min diversification problem are sensitive to outliers. For example, GMM is based on furthest point retrieval, and the furthest point is usually an outlier. Figure 2(a) illustrates a result set S (consisting of red triangles) obtained by GMM. The points located in the center are the same as those in Figure 1, and the other points are outliers. GMM is clearly sensitive to the presence of outliers, as the points in S are dominated by outliers. Another state-of-the-art algorithm (Borassi et al. 2019) also has a similar case, which is presented in Section 5. This demonstrates that simply running an existing algorithm on X having outliers does not function. In addition, for example, when training a machine learning model through S, which is obtained by one of the above algorithms and contains outliers, we face "garbage in, garbage out."

Our Contributions. Motivated by the above observations, we address the problem of Max-Min diversification with outliers. If we can identify outliers, it is possible to remove them from X and then GMM is done on a set of the remaining points in X. A straightforward approach to identifying the outliers requires the evaluation of each point in X. This approach, however, does not scale to n, as it incurs $O(n^2)$ time, as shown in Section 2.3. Therefore, to scale well to n, an algorithm, which runs in *at most* linear time to n with approximation guarantee, is required. Designing such an algorithm is a non-trivial task and is challenging.

We overcome this challenge and propose two novel algorithms. Although they are simple, analysing their performances is not trivial. The main contributions of this paper are as follows:

- We tackle the problem of Max-Min diversification with outliers, for the first time.
- We propose GREEDY (cf. Theorem 1), an $O((k+z)n)$ time algorithm that guarantees a $\frac{1}{2}$-approximation and returns no outliers (z is the number of outliers), as Figure 2(b) illustrates.
- We propose CORESET (cf. Theorem 2), an $O(kz)$ time algorithm that guarantees a $\frac{1}{6(1+\epsilon)}$-approximation and returns no outliers with a constant probability (under a reasonable assumption), where $\epsilon < 1$ is a small constant.
- We conduct experiments using real datasets and demonstrate that our algorithms are much faster than a baseline one while preserving a competitive diversity. For example, CORESET *is three to four orders of magnitude faster than the baseline algorithm.*

2 Preliminary

2.1 Problem Definition

Let X be a set of n points in a metric space. We use $dist(x, x')$ to denote the distance between x and x'. We assume that $dist(\cdot, \cdot)$ satisfies the identity of indiscernibles, symmetry, and triangle inequality. Furthermore, we assume that $dist(\cdot, \cdot)$ can be evaluated in $O(1)$ time. We define $dist(x, X')$, i.e., the distance between a point x and a set X', as $\min_{x' \in X'} dist(x, x')$.

For ease of presentation, let us first consider that X contains only inliers (non-outliers). The problem of Max-Min diversification is to select k points in X so that the minimum distance between the k points is maximized. Formally,

DEFINITION 1 (MAX-MIN DIVERSIFICATION WITHOUT OUTLIERS PROBLEM). *Given a set X of points and an integer $k \geq 2$, this problem is to compute S^* such that*

$$S^* = \underset{S \subseteq X, |S|=k}{\arg\max} \min_{x,x' \in S} dist(x, x'). \tag{1}$$

We use $div(S)$ to denote $\min_{x,x' \in S} dist(x, x')$. It has been proven that (i) this problem is NP-hard and (ii) no polynomial time algorithms can return a solution with an approximation factor better than $\frac{1}{2}$ unless P = NP (Ravi, Rosenkrantz, and Tayi 1994).

Now consider that X contains outliers, so $X = X_{in} \cup X_{out}$, where X_{in} (X_{out}) is a set of inliers (outliers) in X. Our problem is to obtain S^* on $X \backslash X_{out}$.

DEFINITION 2 (MAX-MIN DIVERSIFICATION WITH OUTLIERS PROBLEM). *Given a set X of points and an integer $k \geq 2$, this problem is to compute S^* such that*

$$S^* = \underset{S \subseteq X \backslash X_{out}, |S|=k}{\arg\max} \min_{x,x' \in S} dist(x, x'). \tag{2}$$

We use l^* to denote $div(S^*)$. This problem is also NP-hard trivially, so this paper considers approximation algorithms.

To solve this problem, this paper puts the following assumptions.

ASSUMPTION 1. *We have $|X_{out}| = z$.*

ASSUMPTION 2. *For each $x \in X_{out}$, we have (i) $dist(x, X \backslash \{x\}) > dist(x', X \backslash \{x'\})$ for every $x' \in X_{in}$ and (ii) $dist(x, X \backslash \{x\}) > \alpha l^*$, where $\alpha \geq 1$ is a sufficiently large constant.*

Assumption 1 is the same as that in works of k-clustering with outliers (Bhaskara, Vadgama, and Xu 2019; Ceccarello, Pietracaprina, and Pucci 2019; Ding, Yu, and Wang 2019; Im et al. 2020). The first condition in Assumption 2 is derived from the problem of distance- or nearest neighbor-based outlier detection (Amagata, Onizuka, and Hara 2021, 2022), which has empirically good performance (Campos et al. 2016; Gu, Akoglu, and Rinaldo 2019). This assumption is also essentially similar to that held by the problem of k-clustering with outliers. The second condition in Assumption 2 is natural for the problem of Max-Min diversification with outliers. This is because, if $dist(x, X\backslash\{x\}) \leq l^*$, including an outlier x in S cannot be seen as unusual, and even the optimal solution S^* can contain x, which contradicts Definition 2. This therefore justifies the validity of the second condition in Assumption 2. Our theoretical analyses use the above assumptions.

2.2 GMM

We introduce GMM (Ravi, Rosenkrantz, and Tayi 1994), a state-of-the-art algorithm for the problem of Max-Min diversification *without* outliers (Definition 1), because this is a building block for our techniques. GMM initializes a solution set S by a random point in X. Then, it computes the furthest point from S, denoted by x^*, i.e.,

$$x^* = \underset{x \in X\backslash S}{\arg\max}\, dist(x, S), \tag{3}$$

and x^* is added into S. This is repeated until $|S| = k$. Algorithm 1 summarizes GMM and has the following facts (Ravi, Rosenkrantz, and Tayi 1994).

FACT 1. *Algorithm 1 runs in $O(kn)$ time and returns a $\frac{1}{2}$-approximate result for the problem of Max-Min diversification without outliers, i.e., $div(S) \geq \frac{div(S^*)}{2}$.*

FACT 2 (ANTICOVER PROPERTY). $S \leftarrow \text{GMM}(X, k)$ *has the following properties: (i) $\forall x \in S$, $dist(x, S\backslash\{x\}) \geq div(S)$ and (ii) $\forall x \in X$, $dist(x, S) \leq div(S)$.*

2.3 Baseline Algorithm

Since this is the first work on the problem of Max-Min diversification with outliers, we first consider how to solve this problem by employing existing techniques. Assumption 2 suggests that, if we run a nearest neighbor search for each point in X, we can identify the z outliers. After removing these outliers from X in this way, we have X_{in}. Therefore, by running $\text{GMM}(X_{in}, k)$, we can obtain a $\frac{1}{2}$-approximate result for the problem of Max-Min diversification with outliers. Algorithm 2 summarizes this baseline algorithm.

Although this baseline has a theoretical approximation guarantee, its worst-case running time is $O(n^2)$, since it runs a nearest neighbor search for *every* point in X. The practical time of this algorithm can be alleviated by using some data structure for nearest neighbor search in metric space, but this is still slow for large n.

2.4 Related Work

The diversity maximization problem has been extensively studied since the 1990s, as it outputs a succinct and effective subset of a given dataset. Particularly recently, diversity maximization under some constraint has been considered to satisfy observations or requirements in the real world (Addanki et al. 2022; Ceccarello, Pietracaprina, and Pucci 2018, 2020; Moumoulidou, McGregor, and Meliou 2021; Wang, Fabbri, and Mathioudakis 2022). The presence of outliers, however, has not been considered for the diversity maximization problem.

Algorithm 1: $\text{GMM}(X, k)$

```
S ← a random point in X
while |S| < k do
    x* = arg max_{x ∈ X\S} dist(x, S)
    S ← S ∪ {x*}
return S
```

Algorithm 2: BASELINE(X, k, z)

```
X_out ← z points with the largest distance to their nearest neighbor in X
return S ← GMM(X\X_out, k)
```

Max-Sum Diversification. This problem is also NP-hard, and efficient algorithms with bounded error guarantee were developed. A $\frac{1}{2}$-approximation algorithm was presented in (Borodin et al. 2017). A MapReduce algorithm was developed in (Ceccarello et al. 2017); it needs to assume a bounded doubling dimension. There are some works that consider constraints. For example, matroid constraint was considered in (Ceccarello, Pietracaprina, and Pucci 2018, 2020), whereas the work in (Zhang and Gionis 2020) considers clustered data.

Max-Min Diversification. Some other works, e.g., (Amagata and Hara 2019), also employ this objective function, and GMM provides a good solution for them. There are works (Amagata and Hara 2016; Drosou and Pitoura 2014) that consider how to deal with dynamic X. Fairness constraint has recently been considered in (Addanki et al. 2022; Moumoulidou, McGregor, and Meliou 2021; Wang, Fabbri, and Mathioudakis 2022). The algorithms proposed in these works were extended from the algorithms for Max-Min diversification with no constraint (Ravi, Rosenkrantz, and Tayi 1994; Borassi et al. 2019).

3 Outlier-aware Greedy Algorithm

The main drawback of Algorithm 2 is its quadratic time to n, which is not scalable for a large n. Recall that this time is derived from running a nearest neighbor search for every point in X. To improve the efficiency, we need to theoretically reduce the number of candidates for outliers, but this is not a trivial challenge. We overcome this challenge and prove that we can identify the z outliers *without running a nearest neighbor search for every point in X*. Our idea here is to *leverage the sensitivity of* GMM *to outliers.*

LEMMA 1. *Let S' be the output of* $\text{GMM}(X, k+z)$*, and S' contains the z outliers.*

Algorithm 3: GREEDY(X, k, z)

$S' \leftarrow \text{GMM}(X, k+z)$
$Z \leftarrow z$ points with the largest distance to their nearest neighbor in S'
return $S \leftarrow \text{GMM}(X \backslash Z, k)$

PROOF. Recall that GMM iteratively computes x^*, see Equation (3). Let x_i^* be x^* at the i-th iteration. Also, let S_i be S before x_i^* is inserted. It is important to notice that

$$dist(x_{i+1}^*, S_{i+1}) \leq dist(x_i^*, S_i) \tag{4}$$

This means that, as the size of S grows, $div(S)$ decreases.

Now assume that S' contains only $z' \leq z-1$ outliers. We have $l^* = div(S^*) \geq div(S' \backslash X_{out})$, because we have $k+1 \leq |S' \backslash X_{out}| \leq k+z$ and Equation (4). Notice that $X \backslash S'$ has $z - z'$ outliers, and each of these outliers, say x, has $dist(x, X \backslash \{x\}) > \alpha l^*$. This contradicts Fact 2, so S' must have z outliers. □

From this lemma, we can reduce the number of candidates for the outliers from n points to only $k+z$ points. This enables to design a *linear time algorithm* for the problem of Max-Min diversification with outliers.

Algorithm 3 describes our first algorithm GREEDY. It first runs $\text{GMM}(X, k+z)$ to obtain a set S' of $k+z$ candidate points for the outliers. Then, it computes the nearest neighbor for each $x \in S'$ to identify the z outliers. After that, it runs $\text{GMM}(X \backslash Z, k)$, where Z is a set of the z points. We introduce the main result of this section below.

THEOREM 1. *Algorithm 3 runs in $O((k+z)n)$ time and returns a $\frac{1}{2}$-approximate result, which has no outliers, for the problem of Max-Min diversification with outliers.*

PROOF. From Fact 1, $S' \leftarrow \text{GMM}(X, k+z)$ runs in $O((k+z)n)$ time. Identifying the z outliers from S' needs $O((k+z)n)$ time, as $|S'| = k+z$. It is straightforward to see that $\text{GMM}(X \backslash Z, k)$ runs in $O(k(n-z))$ time. Therefore, Algorithm 3 runs in $O((k+z)n)$ time.

Lemma 1 shows that the z outliers are included in S', so $X \backslash Z$ contains no outliers. From this observation and Fact 1, we have $div(S) \geq \frac{div(S^*)}{2}$. □

4 Coreset-based Algorithm with Probable Success Guarantee

This section proves that there exists an algorithm which does not have a factor of n as its time complexity with sacrifice in a success probability (the probability that S contains no outliers) *a bit*. This algorithm is based on a *coreset*, a good summary of X informally (its definition is introduced later, see Definition 3). Note that the coreset is constructed *offline*.

For ease of presentation, we first devise an outlier-robust online algorithm in Section 4.1. Then, Section 4.2 explains how to construct a coreset. After that, Section 4.3 introduces our main algorithm in this section.

4.1 Online Algorithm

For now, this section assumes that $X' \subseteq X$ is given for an online algorithm. We prove that our online algorithm in this section is linear only to k and $|X'|$. The main idea of making this algorithm robust to outliers is to select a result point that is not near S but not too far from S, which is different from the idea of GMM. To implement this idea, we use a guess of $l^* = div(S^*)$, denoted by $\hat{l}$.

Guessing l^*. Let S' be the set of $k+z$ points obtained by $\text{GMM}(X, k+z)$. Below, we show that $div(S') \geq \frac{div(S^*)}{2}$, where S^*, such that $|S^*| = k$, is the optimal solution for the problem of Max-Min diversification with outliers. This is not a trivial result, because (i) S' contains the z outliers (see Theorem 1), (ii) $|S'| = k+z \neq |S^*|$, and $S' \backslash X_{out}$ is *not* guaranteed to be the same as the output of $\text{GMM}(X_{in}, k)$.

COROLLARY 1. *Given $S' \leftarrow \text{GMM}(X, k+z)$, we have $div(S') \geq \frac{div(S^*)}{2}$.*

PROOF. To prove this corollary, it is sufficient to demonstrate that x^* in Equation (3) has $dist(x^*, S') \geq \frac{l^*}{2}$ — ($\star$). (Recall that S does not contain x^* at the corresponding iteration.) When x^* is an outlier, $dist(x^*, S') > l^*$, so ($\star$) holds. When x^* is an inlier, we show that ($\star$) holds by extending the proof of Theorem 2 in (Ravi, Rosenkrantz, and Tayi 1994).

Assume that $S^* = \{s_1^*, ..., s_k^*\}$. Let $B_i^* = \{x \in X \mid dist(s_i^*, x) < \frac{l^*}{2}\}$, and notice that B_i^* contains at least s_i^*. In addition, the proof of Theorem 2 in (Ravi, Rosenkrantz, and Tayi 1994) demonstrates that $B_i^* \cap B_j^* = \varnothing$ for $i \neq j$. It is also important to notice that all outliers in X do not belong to $\bigcup_k B_i^*$. Now consider the j-th iteration of $\text{GMM}(X, k+z)$. In this iteration, S' contains at most $k-1$ inliers. Hence, for some $i \in [1, k]$, we have $S' \cap B_i^* = \varnothing$. The definition of B_i^* derives that $dist(s_i^*, S') \geq \frac{l^*}{2}$. That is, there exists at least one inlier $x \in X \backslash S'$ such that $dist(x, S') \geq \frac{l^*}{2}$. From this, when x^* is an inlier, ($\star$) still holds. □

Consequently, we have $l^* \in [div(S'), 2div(S')]$. By setting $\hat{l} = (1+\epsilon)^i div(S')$ for $i \in [0, \log 2]$, where $\epsilon < 1$ is a small constant, we obtain $\hat{l} = \frac{l^*}{1+\epsilon}$. Recall that this guessing is done *offline*[1], and we later show that $div(S')$ is obtained as a side product of coreset construction, see Remark 2.

Algorithm Description. Algorithm 4 shows the online algorithm. As with GMM, it first adds a random point in the input set $X' \subseteq X$ into a temporary solution set S_{temp}. Then, given $\hat{l}$ (a guess of l^*), it scans the input set X'. During this, if a given point $x \in X'$ has $\frac{\hat{l}}{2} \leq dist(x, S_{temp}) \leq \hat{l}$, x is added into S_{temp}. This algorithm stops the scan when $|S_{temp}| = k$. This is repeated for each $\hat{l}$, and this algorithm finally returns the solution set with the best diversity.

LEMMA 2. *Algorithm 4 runs in $O(k|X'|)$ time. In addition, it returns no outliers and guarantees $\frac{1}{2(1+\epsilon)}$-approximation with probability at least $1 - \frac{z}{|X'|}$ for the problem of Max-Min diversification with outliers, if $\alpha \geq 2$.*

[1]This is common in (Addanki et al. 2022; Bhaskara, Vadgama, and Xu 2019; Ceccarello et al. 2017; Ding, Yu, and Wang 2019; Im et al. 2020; Moumoulidou, McGregor, and Meliou 2021).

Algorithm 4: STREAMING(X', k)

$S \leftarrow \varnothing$
for *each* $\hat{l}$ *(a guess of* l^**)* $\in L$ **do**
  $S_{temp} \leftarrow$ a random point in X'
  for *each* $x \in X'$ *s.t.* $\frac{\hat{l}}{2} \leq dist(x, S_{temp}) \leq \hat{l}$ **do**
    $S_{temp} \leftarrow S_{temp} \cup \{x\}$
    if $|S_{temp}| = k$ **then**
      break
  if $(|S_{temp}| = k) \wedge (div(S) < div(S_{temp}))$ **then**
    $S \leftarrow S_{temp}$
return S

PROOF. Given $\hat{l}$, we have $|S_{temp}| \leq k$ and the number of accessed points in X' is at most $|X'|$, as Algorithm 4 scans X' once. Since the number of guesses is at most $\log 2 = O(1)$, the time complexity of Algorithm 4 is $O(k|X'|)$.

Recall that Algorithm 4 selects a point $x \in X'$ such that $\frac{\hat{l}}{2} \leq dist(x, S_{temp})$. For $\hat{l} = \frac{l^*}{1+\epsilon}$, it is straightforward to see that Algorithm 4 returns S such that $div(S) \geq \frac{l^*}{2(1+\epsilon)}$.

Assume that $x_1 \in X'$ is firstly added into S, and x_1 is an inlier with probability at least $\frac{|X'|-z}{|X'|}$. Next, let $B(x, \hat{l})$ be a ball centered at $x \in S$ with radius $\hat{l}$. If $B(x, \hat{l})$ contains no outliers, *only inliers can be added into* S. When $l_{greedy} = l^*$, $\hat{l}$ is at most $2l^*$. Since each outlier $x' \in X_{out}$ has $dist(x', X\backslash\{x'\}) > \alpha l^*$, $B(x, 2l^*)$ contains no outliers if $\alpha \geq 2$. To summarize, as long as the first point in S is an inlier, S certainly contains only inliers if $\alpha \geq 2$. Now we complete the proof of Lemma 2. □

REMARK 1. *Recall that* α *is sufficiently large (see Section 2.1): outliers are significantly different to the others usually. Assuming* $\alpha \geq 2$ *is therefore still reasonable.*

4.2 Coreset Construction: Offline Processing

To start with, we formally define coreset below.

DEFINITION 3 (CORESET). *A set* $C \subseteq X$ *is a* β*-coreset, if we have* $S \subseteq C$ *such that* $div(S) \geq \frac{div(S^*)}{\beta}$*, where* $|S| = |S^*| = k$.

In (Indyk et al. 2014), the following fact is demonstrated (see its Lemma 1).

FACT 3. *When* X *contains no outliers,* GMM *yields a 3-coreset for the problem of Max-Min diversification.*

Note that *this bound is shown to be tight* in (Aghamolaei, Farhadi, and Zarrabi-Zadeh 2015).

Importantly, the existing work (Indyk et al. 2014) proves that, *if* $Y \subseteq X$ *satisfies the anticover property (see Fact 2),* Y *is a 3-coreset for the problem of Max-Min diversification.* We use this observation to prove the following.

LEMMA 3. GMM$(X, k+z)$ *returns a 3-coreset for the problem of Max-Min diversification with outliers.*

PROOF. From the proof of Lemma 1, $C \leftarrow$ GMM$(X, k+z)$ contains the z outliers. To prove Lemma 3, we show that $C\backslash X_{out}$ has the anticover property for any inlier in X.

For any inlier $x \in C$, we trivially have $dist(x, C\backslash\{X_{out} \cup \{x\}\}) \geq div(C\backslash X_{out})$. Also, for any inlier $x \in C$, we trivially have $dist(x, C\backslash X_{out}) = 0 \leq div(C\backslash X_{out})$. We therefore focus on each inlier $x' \in X\backslash C$ and consider whether x' has $dist(x', C\backslash X_{out}) \leq div(C\backslash X_{out})$. Assume that $\arg\min_{x \in C} dist(x, x')$ is an outlier. Let this outlier be x_{out}, and $dist(x', x_{out}) > \alpha l^*$. This contradicts Corollary 1, so $\arg\min_{x \in C} dist(x, x')$ must be an inlier. This means that $dist(x', C\backslash X_{out}) = dist(x', C)$. From Fact 2, $dist(x', C) \leq div(C)$. Now notice that $div(C) = div(C\backslash X_{out})$. These observations derive the fact that $dist(x', C\backslash X_{out}) \leq div(C\backslash X_{out})$ for any inlier $x' \in X\backslash C$. □

REMARK 2. *From (Aghamolaei, Farhadi, and Zarrabi-Zadeh 2015), this bound is also tight. In addition, as* GMM$(X, k+z)$ *is used to construct a coreset* C*, we have* $\frac{l^*}{2} \leq div(C) \leq l^*$ *from Corollary 1. Recall that Algorithm 4 requires a guess of* l^**, and* $div(C)$ *is used for guessing.*

4.3 Putting It All Together

Now we are ready to introduce our final algorithm CORESET, which is described in Algorithm 5. It constructs a coreset C offline. When computing a solution set S, it runs STREAMING(C, k).

Algorithm 5: CORESET(X, k, z)

/* Offline processing */
$C \leftarrow$ GMM(c) where $c = O(z)$ and $c \geq k + z$
/* Online processing */
return $S \leftarrow$ STREAMING(C, k)

We below introduce the main result of this section: *the time complexity of* STREAMING(C, k) *is independent of* n while guaranteeing an error bound and success probability.

THEOREM 2. *Given a coreset* C *built by* GMM(X, c) *where* $c = O(z)$ *and* $c \geq k + z$*,* STREAMING(C, k) *runs in* $O(kz)$ *time. In addition, it returns no outliers and guarantees* $\frac{1}{6(1+\epsilon)}$*-approximation for the problem of Max-Min diversification with outliers, with at least a constant probability, if* $\alpha \geq 2$.

PROOF. From Lemma 2, it is trivial to see that STREAMING(C, k) runs in $O(kz)$ time for $|C| = O(z)$. Also, $C \leftarrow$ GMM$(X, k+z)$ derives the $\frac{1}{6(1+\epsilon)}$-approximation bound, which is seen from Lemmas 2 and 3. Given a fixed (constant) success probability p, we have

$$1 - \frac{z}{|C|} = p \Leftrightarrow |C| = \frac{z}{1-p} = O(z). \tag{5}$$

The above discussions complete the proof. □

REMARK 3. *The above theorem assumes that* $\frac{z}{1-p} > k + z$*. This holds when* p *is sufficiently large (e.g.,* $p \geq 0.9$*) and* z *is not too small (i.e., a standard setting). If we do not have this case,* $|C| = O(k+z)$ *and* CORESET *needs* $O(k(k+z))$ *time (while the probable approximation guarantee still holds).*

REMARK 4. *A coreset C is available for any k such that $|C| \geq k + z$. Therefore, the offline processing can be done once for such k, i.e., this offline processing is not unique for a specific value of k.*

5 Experiment

All experiments were conducted on a Ubuntu 20.04 LTS machine equipped with Xeon Platinum 8268 CPU@2.90GHz and 768GB RAM.

Dataset. We used the following real datasets[2].

- FCT: a set of 10-dimensional cartographic variables for forest cover type, and $n = 580,812$.
- Household: a set of 7-dimensional sensor readings, and $n = 2,049,280$.
- KDD99: a set of 16-dimensional packet records, and $n = 311,029$.
- Mirai: a set of 115-dimensional Mirai malware infected network capture data, and $n = 764,137$.

We normalized each dataset so that its domain of each dimension was $[0, 100]$ to have the same scale. After this, we injected z outliers into a given dataset, as with (Bhaskara, Vadgama, and Xu 2019; Ceccarello, Pietracaprina, and Pucci 2019; Ding, Yu, and Wang 2019; Im et al. 2020). We used Euclidean distance for these datasets.

Algorithm. We evaluated the following algorithms.

- GMM (Ravi, Rosenkrantz, and Tayi 1994): a $\frac{1}{2}$-approximation algorithm for Max-Min diversification *without* outliers.
- PODS19 (Borassi et al. 2019): a $(\frac{1}{5} - \epsilon)$-approximation algorithm for Max-Min diversification *without* outliers.
- BASELINE: the $\frac{1}{2}$-approximation algorithm for Max-Min diversification with outliers (Algorithm 2).
- GREEDY: our $\frac{1}{2}$-approximation algorithm for Max-Min diversification with outliers (Algorithm 3).
- STREAMING: our $\frac{1}{2(1+\epsilon)}$-approximation algorithm for Max-Min diversification with outliers (Algorithm 4 with X as its input).
- CORESET: our $\frac{1}{6(1+\epsilon)}$-approximation algorithm for Max-Min diversification with outliers (Algorithm 5).

We set $\epsilon = 0.01$. For BASELINE and GREEDY, we employed a VP-tree (Yianilos 1993) to retrieve the nearest neighbor point, because it is one of the most efficient data structure for metric spaces (Chen et al. 2017). For CORESET, we set the coreset size so that the success probability was 0.95. All algorithms were implemented in C++, compiled by g++ 9.4.0 with -O3 flag, and single threaded. Source codes of our algorithms are available[3].

Parameter Setting. We set $k = 100$ and $z = 200$ by default. This setting of z is similar to those in the evaluation paper (Campos et al. 2016) and in the experiments using large datasets (Ceccarello, Pietracaprina, and Pucci 2019; Gupta et al. 2017). When studying the impact of k (resp. z), the value of z (resp. k) was fixed. We ran each algorithm 20 times and report the average result.

[2]https://archive.ics.uci.edu/ml/datasets.php

[3]https://github.com/amgt-d1/Max-Min-w-Outliers

Algorithm	FCT		Household	
	$div(S)$	Time	$div(S)$	Time
BASELINE	51.514	312.489	38.999	391.429
GREEDY	51.514	2.348	38.999	6.962
STREAMING	49.614	1.874	37.374	5.165
CORESET	50.158	0.005	38.369	0.006

Table 1: Average $div(S)$ and running time [sec] ($k = 100$ and $z = 200$) on FCT and Household

Algorithm	KDD99		Mirai	
	$div(S)$	Time	$div(S)$	Time
BASELINE	80.281	360.946	113.460	485.57
GREEDY	80.281	2.135	113.460	31.046
STREAMING	79.996	1.574	95.439	20.955
CORESET	77.064	0.009	106.352	0.098

Table 2: Average $div(S)$ and running time [sec] ($k = 100$ and $z = 200$) on KDD99 and Mirai

Algorithm	FCT	Household	KDD99	Mirai
BASELINE	311.955	390.577	360.585	477.996
GREEDY	1.814	5.256	1.774	23.474

Table 3: Average time to identify z outliers [sec]

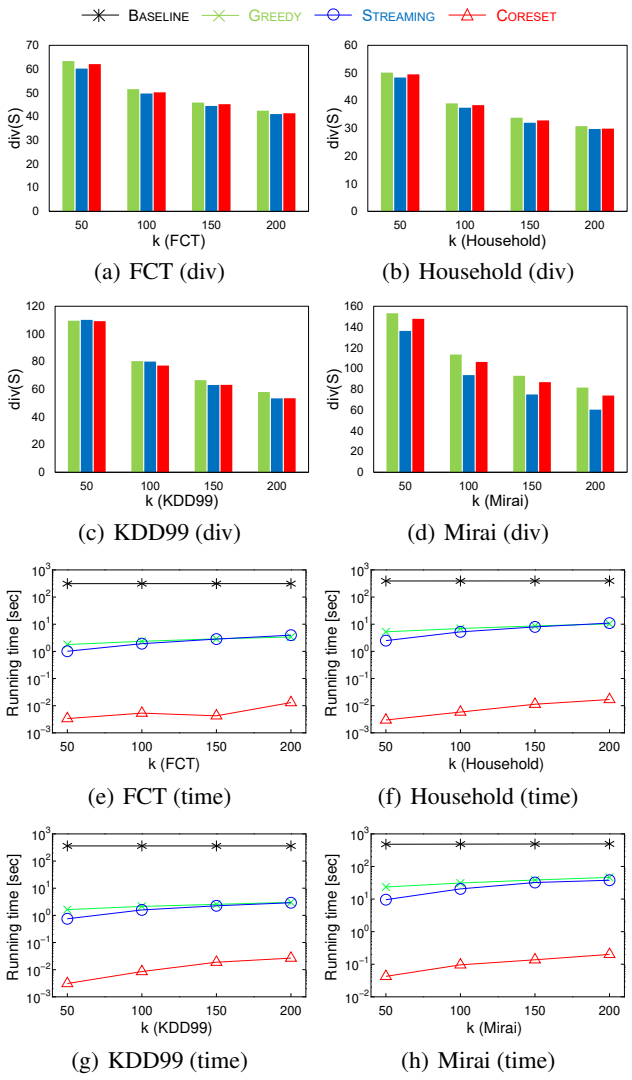
GMM and PODS19 are not appropriate. When S contains outliers, $div(S)$ tends to be large, which is trivial from Assumption 2. However, such S is meaningless, as demonstrated in Figure 2(a). We hence investigated how many outliers were included in S.

We found that 99% (at least 84%) points in S returned by GMM (PODS19) are outliers, suggesting that they do not yield a meaningful result. We therefore did not consider GMM and PODS19 in the subsequent experiments. Note that *the other algorithms did not include any outliers in S*.

Comparison with BASELINE. We compare our algorithms with BASELINE by using the default parameter setting. Tables 1 and 2 show their $div(S)$ and running time.

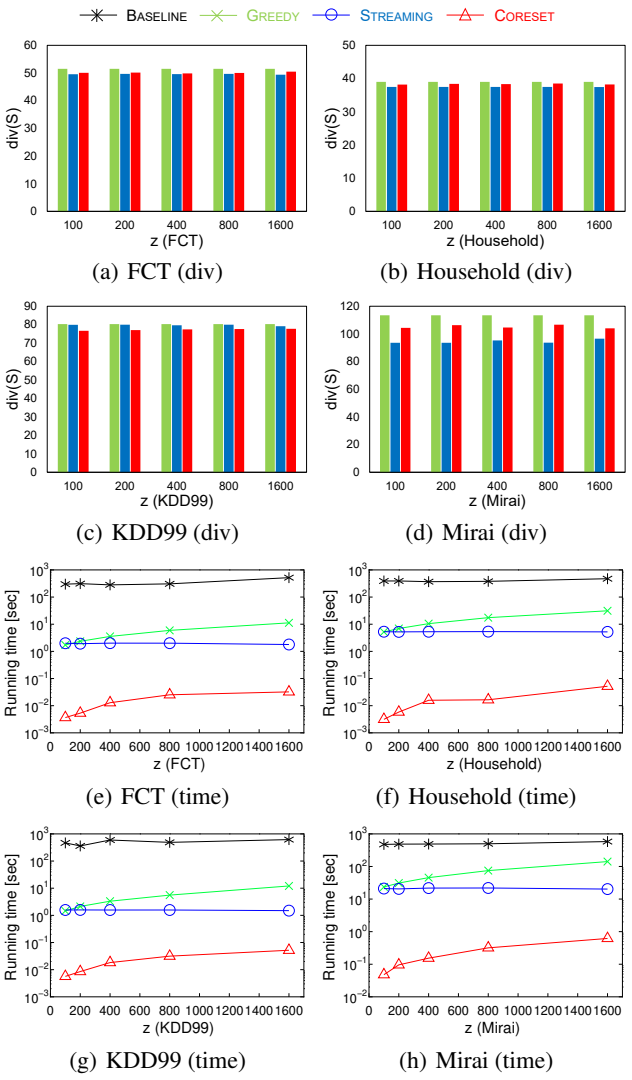
As BASELINE and GREEDY run GMM on X_{in}, they return the same S, so their $div(S)$ is the same. However, their running times are totally different, and GREEDY is at least one order of magnitude faster than BASELINE. Table 3 clarifies why we have this result and the efficacy of the outlier identification approach of GREEDY.

STREAMING and CORESET yield a diverse set competitive with that of GREEDY. In addition, CORESET is significantly faster than the other algorithms. For example, CORESET is up to 67,000 times faster than BASELINE, showing the efficacy of coreset even in the presence of outliers.



(a) FCT (div) (b) Household (div)
(c) KDD99 (div) (d) Mirai (div)
(e) FCT (time) (f) Household (time)
(g) KDD99 (time) (h) Mirai (time)

Figure 3: Impact of k (best viewed in color)



(a) FCT (div) (b) Household (div)
(c) KDD99 (div) (d) Mirai (div)
(e) FCT (time) (f) Household (time)
(g) KDD99 (time) (h) Mirai (time)

Figure 4: Impact of z (best viewed in color)

Impact of k. From the problem definition, it is trivial that $div(S)$ decreases as k increases. Figures 3(a)–3(d) illustrate this result and show that the relationship between the algorithms does not change for different k w.r.t. $div(S)$. Figures 3(e)–3(h) show the running times of the four algorithms. As BASELINE needs $O(n^2)$ time, its running time is stable. On the other hand, the running times of the other algorithms are linear to k, so the times increase as k increases. Since CORESET outperforms the other algorithms with a large margin, it is easy to imagine that CORESET can compute a solution much faster than them even when k is a larger scale.

Impact of z. Figures 4(a)–4(d) observe that $div(S)$ is robust against z, as they do not include the z outliers in S. Figures 4(e)–4(h) show that the running times of BASELINE and STREAMING are generally stable, whereas those of GREEDY and CORESET are linear to z. This result is consistent with their time complexities.

6 Conclusion & Future Work

This paper addressed the problem of Max-Min diversification with outliers for the first time, motivated by (i) the usefulness of the Max-Min diversification problem in many applications and (ii) the fact that real-world datasets usually contain outliers. Existing algorithms for Max-Min diversification without outliers cannot be effective when outliers exist. We hence proposed two effective and efficient algorithms with theoretical performance guarantee for the problem of Max-Min diversification with outliers. Our experimental results demonstrate their effectiveness and efficiency.

This paper has an assumption for outliers, and how to extend our algorithms for different assumptions is one of future works. Moreover, when points in X have demographic groups, fairness constraint is often considered, as introduced in Section 2.4. Addressing a fair case of our problem remains an open issue.

Acknowledgments

This research is partially supported by JST PRESTO Grant Number JPMJPR1931, JSPS Grant-in-Aid for Scientific Research (A) Grant Number 18H04095, and JST CREST Grant Number JPMJCR21F2.

References

Addanki, R.; McGregor, A.; Meliou, A.; and Moumoulidou, Z. 2022. Improved Approximation and Scalability for Fair Max-Min Diversification. In *ICDT*, 7:1–7:21.

Agarwal, P. K.; Sintos, S.; and Steiger, A. 2020. Efficient Indexes for Diverse Top-k Range Queries. In *PODS*, 213–227.

Aghamolaei, S.; Farhadi, M.; and Zarrabi-Zadeh, H. 2015. Diversity Maximization via Composable Coresets. In *CCCG*.

Amagata, D.; and Hara, T. 2016. Diversified Set Monitoring over Distributed Data Streams. In *DEBS*, 1–12.

Amagata, D.; and Hara, T. 2019. Correlation Set Discovery on Time-Series Data. In *DEXA*, 275–290.

Amagata, D.; Onizuka, M.; and Hara, T. 2021. Fast and exact outlier detection in metric spaces: a proximity graph-based approach. In *SIGMOD*, 36–48.

Amagata, D.; Onizuka, M.; and Hara, T. 2022. Fast, exact, and parallel-friendly outlier detection algorithms with proximity graph in metric spaces. *The VLDB Journal*, 31: 797–821.

Bauckhage, C.; Sifa, R.; and Wrobel, S. 2020. Adiabatic Quantum Computing for Max-Sum Diversification. In *SDM*, 343–351.

Bhaskara, A.; Vadgama, S.; and Xu, H. 2019. Greedy Sampling for Approximate Clustering in the Presence of Outliers. *NeruIPS*, 11148–11157.

Borassi, M.; Epasto, A.; Lattanzi, S.; Vassilvitskii, S.; and Zadimoghaddam, M. 2019. Better Sliding Window Algorithms to Maximize Subadditive and Diversity Objectives. In *PODS*, 254–268.

Borodin, A.; Jain, A.; Lee, H. C.; and Ye, Y. 2017. Max-Sum Diversification, Monotone Submodular Functions, and Dynamic Updates. *ACM Transactions on Algorithms*, 13(3): 1–25.

Campos, G. O.; Zimek, A.; Sander, J.; Campello, R. J.; Micenková, B.; Schubert, E.; Assent, I.; and Houle, M. E. 2016. On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical Study. *Data Mining and Knowledge Discovery*, 30(4): 891–927.

Ceccarello, M.; Pietracaprina, A.; and Pucci, G. 2018. Fast Coreset-based Diversity Maximization under Matroid Constraints. In *WSDM*, 81–89.

Ceccarello, M.; Pietracaprina, A.; and Pucci, G. 2019. Solving k-center Clustering (with Outliers) in MapReduce and Streaming, almost as Accurately as Sequentially. *PVLDB*, 12(7): 766–778.

Ceccarello, M.; Pietracaprina, A.; and Pucci, G. 2020. A General Coreset-based Approach to Diversity Maximization under Matroid Constraints. *ACM Transactions on Knowledge Discovery from Data*, 14(5): 1–27.

Ceccarello, M.; Pietracaprina, A.; Pucci, G.; and Upfal, E. 2017. MapReduce and Streaming Algorithms for Diversity Maximization in Metric Spaces of Bounded Doubling Dimension. *PVLDB*, 10(5): 469–480.

Chen, L.; Gao, Y.; Zheng, B.; Jensen, C. S.; Yang, H.; and Yang, K. 2017. Pivot-based Metric Indexing. *PVLDB*, 10(10): 1058–1069.

Dahiya, Y.; Fomin, F.; Panolan, F.; and Simonov, K. 2021. Fixed-Parameter and Approximation Algorithms for PCA with Outliers. In *ICML*, 2341–2351.

Ding, H.; Yu, H.; and Wang, Z. 2019. Greedy Strategy Works for k-Center Clustering with Outliers and Coreset Construction. In *ESA*, volume 144, 40:1–40:16.

Drosou, M.; and Pitoura, E. 2014. Diverse Set Selection over Dynamic Data. *IEEE Transactions on Knowledge and Data Engineering*, 26(5): 1102–1116.

Erkut, E.; Ülküsal, Y.; and Yenicerioğlu, O. 1994. A Comparison of p-dispersion Heuristics. *Computers & operations research*, 21(10): 1103–1113.

Gu, X.; Akoglu, L.; and Rinaldo, A. 2019. Statistical Analysis of Nearest Neighbor Methods for Anomaly Detection. In *NeurIPS*, 10923–10933.

Gupta, S.; Kumar, R.; Lu, K.; Moseley, B.; and Vassilvitskii, S. 2017. Local Search Methods for k-means with Outliers. *PVLDB*, 10(7): 757–768.

Hirata, K.; Amagata, D.; Fujita, S.; and Hara, T. 2022. Solving Diversity-Aware Maximum Inner Product Search Efficiently and Effectively. In *RecSys*, 198–207.

Im, S.; Qaem, M. M.; Moseley, B.; Sun, X.; and Zhou, R. 2020. Fast Noise Removal for k-means Clustering. In *AISTATS*, 456–466.

Indyk, P.; Mahabadi, S.; Mahdian, M.; and Mirrokni, V. S. 2014. Composable Core-sets for Diversity and Coverage Maximization. In *PODS*, 100–108.

Moumoulidou, Z.; McGregor, A.; and Meliou, A. 2021. Diverse Data Selection under Fairness Constraints. In *ICDT*, 13:1–13:25.

Ravi, S. S.; Rosenkrantz, D. J.; and Tayi, G. K. 1994. Heuristic and Special Case Algorithms for Dispersion Problems. *Operations Research*, 42(2): 299–310.

Wang, Y.; Fabbri, F.; and Mathioudakis, M. 2022. Streaming Algorithms for Diversity Maximization with Fairness Constraints. In *ICDE*, 41–53.

Wang, Z.; Guo, Y.; and Ding, H. 2021. Robust and Fully-Dynamic Coreset for Continuous-and-Bounded Learning (With Outliers) Problems. In *NeurIPS*, 14319–14331.

Yianilos, P. N. 1993. Data Structures and Algorithms for Nearest Neighbor. In *SODA*, volume 66, 311.

Zadeh, S. A.; Ghadiri, M.; Mirrokni, V.; and Zadimoghaddam, M. 2017. Scalable Feature Selection via Distributed Diversity Maximization. In *AAAI*, 2876–2883.

Zhang, G.; and Gionis, A. 2020. Maximizing Diversity over Clustered Data. In *SDM*, 649–657.