An Efficient Sum Query Algorithm for Distance-based Locally Dominating Functions*

Ziyun Huang[1] **and Jinhui Xu**[2]

1 **Department of Computer Science and Engineering, State University of New York at Buffalo, USA**
ziyunhua@buffalo.edu

2 **Department of Computer Science and Engineering, State University of New York at Buffalo, USA**
jinhui@buffalo.edu

Abstract

In this paper, we consider the following sum query problem: Given a point set P in $\mathbb{R}^d$, and a distance-based function $f(p,q)$ (*i.e.,* a function of the distance between p and q) satisfying some general properties, the goal is to develop a data structure and a query algorithm for efficiently computing a $(1+\epsilon)$-approximate solution to the sum $\sum_{p\in P} f(p,q)$ for any query point $q \in \mathbb{R}^d$ and any small constant $\epsilon > 0$. Existing techniques for this problem are mainly based on some core-set techniques which often have difficulties to deal with functions with local domination property. Based on several new insights to this problem, we develop in this paper a novel technique to overcome these encountered difficulties. Our algorithm is capable of answering queries with high success probability in time no more than $\tilde{O}_{\epsilon,d}(n^{0.5+c})$, and the underlying data structure can be constructed in $\tilde{O}_{\epsilon,d}(n^{1+c})$ time for any $c > 0$, where the hidden constant has only polynomial dependence on $1/\epsilon$ and d. Our technique is simple and can be easily implemented for practical purpose.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Sum Query, Distance-based Function, Local Domination, High Dimensions, Data Structure

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.47

1 Introduction

In this paper, we consider the following *sum query* problem: Given a set P of points in $\mathbb{R}^d$ (where the dimensionality d could be very high) and a function $f(,)$, the sum query problem is to build a data structure for P so that the sum of $\sum_{p\in P} f(p,q)$ can be efficiently computed or approximated for any query point q in $\mathbb{R}^d$, where $f(p,q)$ is a non-negative *distance-based* function. We say that $f(p,q)$ is *distance-based* if the value of $f(p,q)$ depends only on the distance between p and q. In other words, $f(p,q)$ can be written as $F(\|p,q\|)$ for some non-negative real function $F(\cdot)$.

The distance-based sum query problem are frequently encountered in many applications. A good example is the well known 1-median problem: given a point set P in $\mathbb{R}^d$, find a point q such that the objective value $C(q) = \sum_{p\in P}\|q-p\|$ is minimized. $C(q)$ is clearly an example of the distance-based sum query problem (with respect to the to-be-determined median

* This research was supported in part by NSF through Grants IIS-1422591, CCF-1422324, CNS-1547167, and CCF-1716400.

28th International Symposium on Algorithms and Computation (ISAAC 2017).
Editors: Yoshio Okamoto and Takeshi Tokuyama; Article No. 47; pp. 47:1–47:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

point q), where each term of the summation is trivially the Euclidean distance $\|q-p\|$ of p and q. The sum query problem also appears in many real world applications. For example, the problem of computing the illumination intensity of a given point can be viewed as a sum query problem. In such an application, the intensity of the query point may jointly be determined by the total amount of light received from multiple light sources. The light contributed by each source is inversely proportional to its squared distance to the given point (*i.e.* obeying the inverse squared distance law in physics). Note that in this case the distance-based functions may be different for each light source, depending on its base intensity. However, if we view a light source with base intensity w as a collection of w light sources with "unit" intensity located at the same place, we may still treat the intensity as a purely distance-based function.

Several previous results are closely related to some versions of the problem considered in this paper. They are mainly based on some *core-set* techniques [2, 7, 10]. In the 1-median problem, for example, a core-set of a point set P in $\mathbb{R}^d$ is a small-size (weighted) subset of P such that for any $q \in \mathbb{R}^d$, the sum $\sum_{p\in P}\|p-q\|$ can be approximated by just inspecting the distances between q and points in the core-set. In general, a core set of P with respect to a function $f(p,q)$ is a small subset of P such that for any q, $\sum_{p\in P} f(p,q)$ can be estimated by using only the information of the points in the core-set. For functions $f(p,q)$ satisfying certain properties, it is possible to construct a core-set for any point set P efficiently [8].

In this paper, we aim to develop an efficient algorithm for supporting distance-based functions that have *local domination property*[6], which means that $f(p,q)$ can be very large when $\|p-q\|$ is small. For example, a distance-based function obeying the inverse squared distance law (*i.e.* $f(p,q) = w/\|p-q\|^2$ for some constant w), is a function having such a property. While the aforementioned core-set method is useful for a large family of functions $f(p,q)$, it does not directly apply to functions which have local domination property. This is because the $\sum_{p\in P} f(p,q)$ could become infinitely large when q approaches any one of the points in P, which means that any "traditional" core-set of P will fail if the core-set is a proper subset of P.

The local domination property imposes additional challenges to the sum query problem. Particularly, it requires the query algorithm to be able to detect points that are close to the query points. This means that the algorithm should have certain ability for proximity search. However, in high dimensional space, highly accurate nearest neighbor search cannot be done very efficiently. Well-known techniques for high dimensional nearest neighbor search, such as the Locality Sensitive Hashing (LSH) [9], require almost linear time to achieve a c-approximate nearest neighbor when c is close to 1 [3]. Thus, for the sum query problem, we are required to develop an estimation algorithm with high accuracy, but not allowed to use the high accuracy proximity search techniques.

To deal with the additional challenge caused by the local domination property, we first assume that the distance function F satisfies the following local domination implied properties.

1. $F(\cdot)$ is positive and $F(0)$ could be infinite.
2. $F(\cdot)$ is monotonously decreasing. [1]
3. For any constant $\lambda \geq 1$, there exists a constant $\Delta(\lambda) \geq 1$, such that $F(x) \leq \Delta(\lambda)F(x\lambda)$ for any $x \geq 0$.

[1] Indeed this restriction can be greatly soften. Our scheme applies as long as $F(\cdot)$ is "not increasing rapidly", *i.e.*, $F(x_1) \leq CF(x_2)$ for some constant C when $x_1 > x_2$. The listed restriction is mainly for ease of presentation.

It is worth noting that although our technique is designed for functions with local domination property, it actually works for any distance-based non-negative functions. Particularly, our approach is capable of solving the “inverse" version of the problem, where $F(\cdot)$ is a monotonically increasing function satisfying some accordingly changed conditions. Since other types of distance-based functions have already been studied in [8], we focus our investigation on locally dominating functions in this paper.

Our Result: Our main result for the sum query problem is a novel scheme based on some sampling and searching techniques, and is capable of reporting a $(1+\epsilon)$-approximation for each sum query ($\sum_{p\in P} f(p,q) = F(\|p-q\|)$) in $\tilde{O}_{\epsilon,d}(n^{0.5+c})$ time with success probability at least $1-1/n$ for any $c > 0$. The query algorithm makes use of a soft boundary range reporting data structure to determine a number of points that are among the closest to the query point q. The soft boundary range reporting data structure can be computed within $\tilde{O}_{\epsilon,d}(n^{1+c})$ time for any $c > 0$. The hidden constants in the time complexities depend only polynomially on d and $1/\epsilon$. The error factor ϵ can be very small and is assumed to be within the range of $[8/\sqrt{n}, 1)$. One major advantage of our scheme is that the query algorithm runs much faster than the best existing $(1+\epsilon)$-approximate nearest neighbor search technique (which takes almost linear time) in high dimensional space for small enough ϵ .

Our Technique: Our query algorithm consists of 2 main steps. In the first step, we identify a number of points P_Ω that are among the closest to the query point q, and compute directly their contributions to the sum $\sum_{p\in P_\Omega} f(p,q)$. In the second step, we sample, from the rest of the points in P, a small subset of points to estimate their contributions to the sum. Intuitively speaking, since we have already identified a number of points that have the largest contribution to the sum before sampling, the error incurred by sampling the rest of points is relatively small and thus controllable. We combine the results from the 2 steps to obtain an approximate final solution. We use a soft boundary range reporting data structure to identify points that are among the closest to q. With properly chosen parameters, we are able to show that it suffices to use a relatively low quality approximate range search procedure to obtain an accurate solution.

Related Work: As mentioned earlier, the sum query problems can be solved by using core-sets for distance functions satisfying some “nice" properties. Our work can be viewed as a complement to those core-set results as it addresses a rather general case that is hard to use core-sets.

Our scheme makes use of some ideas from range search and top-k indexing. There are a number of previous results on both problems [13, 12, 4, 1]. Many of them are not the best fit, especially in high dimensional space, as they cannot be directly applied to our problem. The special property of our problem enable us to develop a range search scheme with better performance.

2 Query Algorithm by Searching and Sampling

In this section, we present our algorithm for the sum query problem. We start our discussion with a high level description of our ideas.

2.1 Starting Point: Estimation by Sampling

Answering a distance-based sum query for a given query point q is essentially estimating the sum (or equivalently, the mean) of a set of numbers: $\{f(p,q) \mid p \in P\}$. A common practice for efficiently estimating the mean of a set of numbers is using sampling. It is well-known that even for a large set of numbers, it suffices to take only a small sample from the set and calculate the mean of the sampled set. The calculated sum is very likely to be a high quality estimation of the mean value of the whole set. The following lemma is one of the known results on concentration of sample mean.

▶ **Lemma 1** (Hoeffding's Inequality [11]). *Let $X = \{x_1, \ldots, x_n\}$ be a multi-set of n real numbers, and $x'_1, \ldots, x'_m$ be a random sample drawn without replacement from X. Let $a = \min_{1 \le i \le n} x_i$ and $b = \max_{1 \le i \le n} x_i$. Then, for any $\epsilon > 0$,*

$$\mathbb{P}\Big(|\frac{1}{m}\sum_{i=1}^{n} x'_i - \mu| \ge \epsilon\Big) \le \exp\Big(-\frac{2m\epsilon^2}{(b-a)^2}\Big),$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean of X.

Note that the error bound in the above estimation depends on the spread (*i.e.* the difference between the largest and smallest elements) of the original number set. This implies that a straightforward application of the sampling technique may not be sufficient to achieve highly accurate solution (*i.e.* $(1+\epsilon)$-approximation) of the sum query problem. The error bound ensured by Lemma 1 could be small in terms of the *spread* (with high probability, by setting ϵ to be $\Theta((b-a))$ and $m = \Theta(\log n)$, for example), but still might be large compared to the *mean*. In the distance-based sum query problem where we are essentially estimating the mean of all the addictive terms, the error is evaluated with respect to the mean value $\sum_{p \in P} f(p,q)/n$. If the spread is very large compared to the mean (which could happen if, for example, the query point q is very close to one of the data point), the error (in terms of the mean value) will also be large.

Intuitively, since all additive terms ($f(p,q)$ for all $p \in P$) are nonnegative in the distance-based sum query problem, the largest terms in the sum tends to contribute more to the error incurred by sampling. This leads us to the idea of identifying a few of the largest terms in the sum and considering them separately. To implement this idea for distance-based sum query, we partition the input point set P into 2 subsets, P_O and P_Ω, based on $f(p,q)$ and q, where P_Ω contains the k points in P corresponding to the k largest terms of $\{f(p,q) \mid p \in P\}$ and $k \ll n$ is a factor to be determined later. We then estimate the contributions S_Ω and S_O of P_Ω and P_O, respectively. S_Ω can be computed directly from P_Ω, and S_O is determined from P_O by using standard sampling technique. Thus we can obtain the solution from $S = S_O + S_\Omega$. The estimation process is efficient if $k \ll n$ is sufficiently small. By intuition, this method could achieve better accuracy, since excluding P_Ω from the sampling process avoids the situation that a few very large additive terms exist in the sum, making the sampling technique not applicable.

2.2 Identifying Close Points: Soft Boundary Range Search

Clearly, the aforementioned approach requires that given any query point q, the set P_Ω has to be determined efficiently. Recall the assumption that $f(p,q)$ is a monotonously decreasing function with respect to $\|p-q\|$. This means that the set P_Ω is indeed the subset of P which consists of the k closest points in P to q. To perform this task efficiently, we need to build an k-nearest neighbor data structure for P, which is capable of reporting the k nearest neighbors of q in P for any query point q.

The k-nearest neighbor (kNN) problem in $\mathbb{R}^d$ is known to be hard when d is large due to the curse of dimensionality. If approximation is allowed, there are several techniques, for example the well known Locality Sensitive Hashing (LSH), that are applicable to kNN in arbitrary dimensions. Nonetheless these techniques do not directly provide a solution to our searching problem with the desired performance. When the approximation ratio is small, the nearest neighbor query using LSH takes near linear time in high dimensional space. It seems that it would also be the case for the distance-based sum query problem that the query would be inefficient for small ϵ.

To overcome this obstacle, we make use of a bi-criteria approximation scheme to report P_Ω. For a predefined parameter k and a controlling constant factor $\lambda > 1$, instead of reporting the k approximate nearest neighbors of the query point q, we try to report all points that lie in $B(q, r_O)$, where $B(q, x)$ denotes the closed ball centered at q and with radius x, and $r_O > 0$ satisfies the condition that $|B(q, \lambda r_O) \cap P| = O(k)$. In other words, we report the near neighbors of q in P that lie in a soft boundary that is based on the $O(k)$ nearest neighbor of q. When λ is not very close to 1, the reporting can be performed efficiently using known proximity search techniques (the technical details of the kNN soft boundary range search algorithm will be presented in later sections).

Note that in the above soft boundary range reporting scheme, the controlling factor λ does not depend on ϵ. This avoids the potential issue that it may take near linear time to answer a query when ϵ is small. Later we will show that λ does not need to be close to 1 (*i.e.* the accuracy of the soft boundary search does not need to be high) when ϵ is small. The reason is the follows. If k is small (*e.g.*, $k = O(\sqrt{n})$), the k-nearest neighbors of q in P is only a very small fraction of points in P. Therefore, we are able to afford large error from estimating these points, while still keeping the error of the final solution within the $(1 + \epsilon)$-approximation range.

The remaining problem of this scheme is how to determine the value of r_O efficiently when answering a query. This can be achieved by sampling. Suppose that we sample m points from P where m is a sufficiently large integer. Let P'_s be the sampled point set, and let p_α be the $\lceil mk/n \rceil$-th closest point to q in P'_s. Intuitively, by performing a “scaling" argument, p_α should be approximately the k-th (by $(mk/n) * (n/m) = k$) closest point to q in P. Later we will show that this intuition is correct. We then set $r_O = \|p_\alpha - q\|/\lambda$.

2.3 Algorithm for Sum Query

We summarize the above discussion with the following explanation of the query procedure. Suppose that the controlling factor λ is given, and k is set to be $\lceil \sqrt{n} \rceil$. Note that k is just for analysis purpose and the algorithm does not really depend on it. Let m be the size of the sample and assume that its value has already been provided. To answer a distance-based sum query for a query point q, we first sample a subset P'_s from P with size m. Let p_α be the $\lceil m/\sqrt{n} \rceil$-th closest point to q in P'_s. Then p_α is approximately the $\sqrt{n}$-th closest point to q in P. We choose r_O to be $\|p_\alpha - q\|/\lambda$, and use range search technique to determine the point set $P'_\Omega = B(q, r_O) \cap P$. P'_Ω contains points that are the closest to q. We use sampling to estimate the mean value of $f(p, q)$ for all point $p \in P \setminus P'_\Omega$ without incurring large error. This mean value gives us an estimation of the value $S_O = \sum_{p \in P \setminus P'_\Omega} f(p, q)$. The value $S_\Omega = \sum_{p \in P'_\Omega} f(p, q)$ can be directly computed. The sum of S_Ω and S_O is then an accurate estimation of the distance-based sum $\sum_{p \in P} f(p, q)$.

Below are the main steps of query algorithm, where the approximation factor ϵ satisfies the condition of $4/\sqrt{n} \le \epsilon < 1/2$ and $n \ge 100$. We assume the existence of a soft boundary range reporting data structure (details of the data structure will be discussed in later section)

Algorithm 1 ComputeSum(q, λ, ϵ)

```
Input: Query point $q$, controlling factor $\lambda > 1$, approximation ratio $\epsilon$
Output: A value $\bar{S}$ which is an approximate value of $S = \sum_{p \in P} f(p, q)$.
Set $\gamma = 262\Delta^2\epsilon^{-2}$. Randomly sample $m = \lceil \gamma\sqrt{n} \ln 4n \rceil$ points from $P$ without replacement. Let $P_s'$ denote the sampled point set.
Let $p_\alpha$ be the $\lceil m/\sqrt{n} \rceil$-th closest point to $q$ in $P_s'$. Let $r_\alpha$ denotes $\|p_\alpha - q\|$. Let $r_O = r_\alpha/\lambda$.
Report points lying inside $B(q, r_O)$ by using the $\lambda$-approximate soft boundary range search data structure. Let $P_\Omega'$ denote the set of reported points.
Compute $S_\Omega' = \sum_{p \in P_\Omega'} f(p)$.
Let $P_O' = P_s' \setminus B(q, r_O)$. Compute $S_O' = \sum_{p \in P_O'} f(p)$
Output $\bar{S} = S_\Omega + nS_O'/|P_O'|$ as the result
```

which can answer the range reporting query made by the algorithm. $\lambda > 1$ is a factor for controlling the accuracy of the soft boundary range reporting data structure. We let Δ denote the constant such that $F(x) \leq \Delta F(x\lambda)$ for any $x \geq 0$, where $F(\cdot)$ is the distance-based function for $f(p, q)$ (*i.e*, $f(p, q) = F(\|p - q\|)$). Since the query point q is given, we write $f(p, q)$ as $f(p)$ for convenience.

2.4 Algorithm Analysis

In this section we prove the correctness of the algorithm and analyze its performance.

For ease of our presentation, we assume that there is no more than one point with exactly the same distance to q. This assumption is actually not needed for our algorithm. Our argument still holds using any tie-break mechanism if multiple points have the same distance to q. For example, we may assign a unique integer label to every point in P and use it as a tie break.

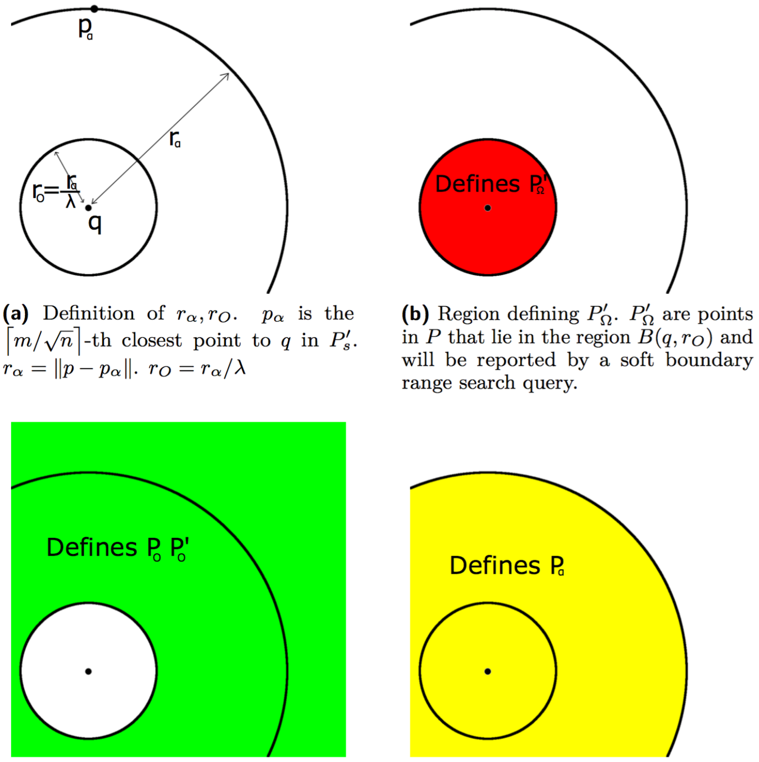
We first present some lemmas that will be used for later analysis. The following is a useful bound for random sample without replacement.

▶ **Lemma 2** (Bernstein's Inequality [5]). *Let $X = \{x_1, \ldots, x_n\}$ be a multi-set of n real numbers, and $x_1', \ldots, x_m'$ be a random sample drawn without replacement from X. Let $a = \min_{1 \leq i \leq n} x_i$ and $b = \max_{1 \leq i \leq n} x_i$. Let $\sigma = \frac{1}{n}\sum_{x \in X}(x - \mu)^2$ be the variance of X. For any $\epsilon > 0$,*

$$\mathbb{P}\Big(|\frac{1}{m}\sum_{i=1}^{n} x_i' - \mu| \geq \epsilon\Big) \leq \exp\Big(-\frac{m\epsilon^2}{2\sigma^2 + (2/3)(b - a)\epsilon}\Big).$$

▶ **Lemma 3.** *Let X be a set of $n \geq 1$ real numbers, $K \geq 1$, such that for each $x \in X$, $0 \leq x \leq K\sqrt{n}$. Let $\mu = \sum_{x \in X} x/n$ be the mean of X, and $\sigma^2 = \sum_{x \in X}(x - \mu)^2/n$ be the variance. Suppose $\mu \geq 1$. Then $\sigma^2/\mu^2 \leq K\sqrt{n}$.*

Proof. Fixing the value of $\mu \geq 1$, we consider how to construct X so that σ^2 is maximized, subject to the constraint that for each $x \in X$, $0 \leq x \leq K\sqrt{n}$. It is clear that σ^2 is maximized when X is in its most "uneven" state, *i.e.*, with the exception of at most 1 element in X, all other elements are either 0 or $K\sqrt{n}$. In fact, σ^2 can be written as $\sum_{x \in X} x^2/n - \mu^2$. If we can find 2 elements x_1 and x_2 in X, such that $0 < x_1 \leq x_2 < K\sqrt{n}$, increasing x_2 and decreasing x_1 by a same small number will increase the value of $\sum_{x \in X} x^2$ (since $f(x) = x^2$ is convex), while the mean μ of X is unchanged, which means that σ^2 is increased. This proves that when σ^2 is maximized, all elements in X are either 0 or $K\sqrt{n}$ with only 1 exception.



(a) Definition of r_α, r_O. p_α is the $\lceil m/\sqrt{n}\rceil$-th closest point to q in P'_s. $r_\alpha = \|p - p_\alpha\|$. $r_O = r_\alpha/\lambda$

(b) Region defining P'_Ω. P'_Ω are points in P that lie in the region $B(q, r_O)$ and will be reported by a soft boundary range search query.

(c) Region defining $P_O(P'_O)$. $P_O(P'_O)$ are points in $P(P'_s)$ that lie in the region $\mathbb{R}^d \setminus B(q, r_O)$.

(d) Region defining P_α. P_α are points in P that lie in the region $B(q, r_\alpha)$

Figure 1 Illustrations of $r_\alpha, r_O, P'_\Omega, P_O, P'_O, P_\alpha$.

We can easily list elements in such a set X. There exist integer $a \geq 0$ and real number $K\sqrt{n} > b \geq 0$, such that $n\mu = aK\sqrt{n} + b$. Therefore, X contains a elements of value $K\sqrt{n}$, $n - 1 - a$ elements of 0, and the rest of the elements take value b. The term $\sum_{x \in X} x^2$ can be easily computed as $aK^2n + b^2$. Then, we have

$$\sigma^2 = \sum_{x \in X} x^2/n - \mu^2 = (aK^2n + b^2)/n - \mu^2. \tag{1}$$

Note that $aK^2n + b^2 \leq aK^2n + bK\sqrt{n} = \sqrt{n}K(aK\sqrt{n} + b) = \sqrt{n}Kn\mu$. Combine this with the above inequality, we have

$$\sigma^2 = (aK^2n + b^2)/n - \mu^2 \leq \sqrt{n}K\mu - \mu^2. \tag{2}$$

Therefore $\sigma^2/\mu^2 \leq (\sqrt{n}K/\mu) - 1$. Since $\mu \geq 1$, $\sigma^2/\mu^2 \leq (\sqrt{n}K/\mu) - 1 \leq \sqrt{n}K/\mu \leq K\sqrt{n}$. ◀

In the following, we let $P_O = P \setminus B(q, r_O)$. Define $P_\alpha = P \cap B(q, r_\alpha)$. (Figure 1 gives a simple illustration of $r_\alpha, r_O, P'_\Omega, P_O, P'_O, P_\alpha$ for easy understanding of later analysis.)

▶ **Definition 4.** We say that the *Good Sample Condition* is satisfied in a query procedure of Algorithm 1, if all of the following conditions hold.

1. $\sqrt{n}/2 \leq |P_\alpha| \leq 2\sqrt{n}$.
2. $||P'_O|n/m - |P_O|| \leq \epsilon |P_O|$

▶ **Lemma 5.** *With probability at least* $1 - 1/2n$*, the good sample condition satisfies.*

Proof. We first show that $|P_\alpha| \le 2\sqrt{n}$ happens with probability at least $1 - 1/2n$.

Let P_β denote the set of $\lceil 2\sqrt{n} \rceil$ points in P that are the closest to q. For every $p \in P$, we define $x(p)$ as follows. $x(p) = 1$ if $p \in P_\beta$, and $x(p) = 0$ otherwise. The mean value of $x(p)$ for all $p \in P$ can be easily computed as $\mu = \lceil 2\sqrt{n} \rceil / n$. Let μ' be the mean value of $x(p)$ in the sampled set P'_s. Clearly $\mu' = |P'_s \cap P_\beta|/m$.

Recall that, from the definition of P_α, we know that P_α contains the $\lceil m/\sqrt{n} \rceil$ closest points in P'_s to q but does not include any point in P'_s farther than the $\lceil m/\sqrt{n} \rceil$ closest points. Consider the event that $|P_\alpha| > 2\sqrt{n}$. If it happens that $|P_\alpha| > 2\sqrt{n}$, it implies that the closest $\lceil 2\sqrt{n} \rceil$ points in P to q have no more than $\lceil m/\sqrt{n} \rceil$ points in P'_s, which means that $|P'_s \cap P_\beta| \le \lceil m/\sqrt{n} \rceil \le m/\sqrt{n} + 1 \le 1.01m/\sqrt{n}$ (where the last inequality comes from simple calculation $m/\sqrt{n} = \lceil \gamma\sqrt{n}\ln 4n \rceil/\sqrt{n} \ge \gamma \ln 4n - 1 \ge 262 - 1 = 261$). Therefore we have $\mu' = |P'_s \cap P_\beta|/m \le 1.01/\sqrt{n}$. From $\mu = \lceil 2\sqrt{n} \rceil / n \ge 2\sqrt{n}/n = 2/\sqrt{n}$, we get $|\mu - \mu'| \ge 0.99/\sqrt{n}$.

Now we bound the probability of the event $|\mu - \mu'| \ge 0.99/\sqrt{n}$ using Lemma 2. Applying Lemma 2 to sample P'_s of P about value $x(p)$, we have

$$\mathbb{P}\Big(|\mu' - \mu| \ge 0.99/\sqrt{n}\Big) \le \exp\Big(-\frac{m(0.99/\sqrt{n})^2}{2\sigma^2 + (2/3)(0.99/\sqrt{n})}\Big),$$

where $\sigma^2 = \sum_{p \in P}(x(p) - \mu)^2/n$. It is straightforward to calculate $\sigma^2 = 2(\lceil 2\sqrt{n} \rceil/n)(1 - \lceil 2\sqrt{n} \rceil/n)$. Thus, we have $\sigma^2 \le 2(\lceil 2\sqrt{n} \rceil/n) \le 5\sqrt{n}/n$ (estimation from the assumption that $n \ge 100$). Therefore, we know that

$$\exp\Big(-\frac{m(0.99/\sqrt{n})^2}{2\sigma^2 + (2/3)(0.99/\sqrt{n})}\Big) \le \exp\Big(-\frac{m(0.99/\sqrt{n})^2}{10/\sqrt{n} + (2/3)(0.99/\sqrt{n})}\Big).$$

The right hand side becomes $\exp(-(m/\sqrt{n})(0.99)^2/10.66)$. Note that $m = \lceil 262\Delta^2\epsilon^{-2}\sqrt{n}\ln 4n \rceil \ge 262\sqrt{n}\ln 4n$. By simple calculation, we have $(m/\sqrt{n})(0.99)^2/10.66 \ge \ln 4n$. As a result, we know that

$$\mathbb{P}\Big(|\mu' - \mu| \ge 0.99/\sqrt{n}\Big) \le e^{-\ln 4n} = 1/4n.$$

Since we have already shown that $|P_\alpha| > 2\sqrt{n}$ implies $|\mu - \mu'| \ge 0.99/\sqrt{n}$, we know that $|P_\alpha| > 2\sqrt{n}$ may also happen with probability at most $1/4n$.

Using the same argument we can also prove that the event $|P_\alpha| < \sqrt{n}/2$ happens with probability at most $1/4n$. We omit the proof for this case due to similarity with the above case. To summarize, we have proved that Condition 1 of the lemma, $\sqrt{n}/2 \le |P_\alpha| \le 2\sqrt{n}$, holds with probability at least $1 - 1/2n$.

For the second condition, *i.e.* $||P'_O|n/m - |P_O|| \le \epsilon|P_O|$, we will show that it follows from Condition 1.

From definition, we know that $P'_O \supseteq P'_s \setminus B(q, r_\alpha)$. Thus, $|P'_O| \ge |P'_s \setminus B(q, r_\alpha)| = m - \lceil m/\sqrt{n} \rceil$. Clearly we also have $|P'_O| \le m$. Therefore, we get $n - n\lceil m/\sqrt{n} \rceil/m \le |P'_O|n/m \le n$. It is easy to have an estimation $\lceil m/\sqrt{n} \rceil \le m/\sqrt{n} + 1 \le 1.01m/\sqrt{n}$. Thus we obtain $n - 1.01\sqrt{n} \le |P'_O|n/m \le n$

By Condition 1 and $P_O \supseteq P \setminus P_\alpha$, we have $|P_O| \ge n - |P_\alpha| \ge n - 2\sqrt{n}$. Note that we also clearly have $|P_O| \le n$. As a result, it follows that $||P'_O|n/m - |P_O|| \le (1.01 + 2)\sqrt{n} = 3.01\sqrt{n}$.

Now we need to prove that $3.01\sqrt{n} \le \epsilon|P_O|$. From Condition 1, we know that $|P_O| \ge n - 2\sqrt{n}$. It suffices to show that $3.01\sqrt{n} \le \epsilon(n - 2\sqrt{n})$. Indeed, this trivially follows from the assumption that $\epsilon \ge 4/\sqrt{n}$ and $n \ge 100$. ◀

Below is an important lemma which shows that our sampling scheme gives a good approximation of the mean value of $f(p)$ for all $p \in P_O$.

▶ **Lemma 6.** *Let μ'_O be the mean of $f(p)$ for all $p \in P'_O$. Let μ_O be the mean of $f(p)$ for all $p \in P_O$. Assume that the good sample condition holds. With probability at least $1 - 1/4n$, $|\mu'_O - \mu_O| \leq \epsilon S/n$*

Proof. Denote $f_*(p) = (\sqrt{n} f(p) + F(r_O))/F(r_O)$ for all $p \in P_O$. Let μ'_* be the mean of $f_*(p)$ for all $p \in P'_O$, and μ_* be the mean of $f_*(p)$ for all $p \in P_O$. Below we first show that

$$\mathbb{P}\Big(|\mu'_* - \mu_*| \geq \epsilon\mu_*/4\Delta\Big) \leq 1/4n. \tag{3}$$

We apply Lemma 2 to bound the probability of the event $|\mu'_*$ - $\mu_*| \geq \epsilon\mu_*/4\Delta$ as follows. The set P'_O can be viewed as a random sample without replacement of size $|P'_O|$ from set P_O, since for a fixed r_O, every $|P'_O|$-subset of P_O has equal probability to be the first $|P'_O|$ points in P'_s, sorted by decreasing order of distances to q. (Note that this fact is true regardless whether the sample satisfies the good sample condition.) Note that for any $p \in P_O$, it is easy to see that $f_*(p) \geq 1$ and $f_*(p) \leq \sqrt{n} + 1$ (since, by $\|q - p\| \leq r_O$, we have $F(r_O) \geq F(\|q - p\|)$). Let $\sigma^2 = (\sum_{p \in P_O}(f_*(p) - \mu_*)^2)/|P_O|$. From Lemma 2, we have

$$\mathbb{P}\Big(|\mu'_* - \mu_*| \geq \epsilon\mu_*/4\Delta\Big) \leq \exp\Big(-\frac{(1/16)|P'_O|\epsilon^2\Delta^{-2}\mu_*^2}{2\sigma^2 + (2/3)\sqrt{n}(\epsilon\mu_*/4\Delta)}\Big). \tag{4}$$

Let

$$\xi = \frac{(1/16)\mu_*^2}{2\sigma^2 + (2/3)\sqrt{n}(\epsilon\mu_*/4\Delta)}.$$

The right hand side of the above inequality (4) becomes $e^{-|P'_O|\epsilon^2\Delta^{-2}\xi}$.

To estimate ξ, we first bound σ^2/μ_*^2. From the good sample condition, we know that $||P'_O|n/m - |P_O|| \leq \epsilon|P_O|$. Thus, we have $|P_O| \geq |P'_O|n/(1+\epsilon)m$. Also, by the definition of P'_O, we know that $P'_s \setminus P_\alpha \subseteq P'_O$. Thus, we get $|P'_O| \geq |P'_s \setminus P_\alpha| \geq m - m/\sqrt{n} - 1$. Therefore, we obtain $|P_O| \geq n\frac{m - m/\sqrt{n} - 1}{(1+\epsilon)m} = n\frac{1 - 1/\sqrt{n} - 1/m}{(1+\epsilon)}$. Since $\epsilon < 1/2$, $m \geq 100$, and $n \geq 100$, we have a rough estimation of $|P_O| \geq n/4$. Also, we know that for any $p \in P_O$, $f_*(p) \leq \sqrt{n} + 1 \leq 2\sqrt{n}$. Consequently, we have $f_*(p) \leq 4\sqrt{|P_O|}$ for every $p \in P_O$. Applying Lemma 3, we know that $\sigma^2/\mu_*^2 \leq 4\sqrt{|P_O|}$. Thus, we get $\sigma^2/\mu_*^2 \leq 4\sqrt{n}$.

Next we show a lower bound for $|P'_O|$. In fact, we know that $|P'_O| = \gamma\sqrt{n}\ln 4n - |P'_s \cap B(q, r_O)| \geq \gamma\sqrt{n}\ln 4n - |P'_s \cap B(q, r_\alpha)| \geq \gamma\sqrt{n}\ln 4n - \gamma\ln 4n - 1 \geq (\gamma\sqrt{n}\ln 4n)/2$ (the last inequality can be easily obtained from the assumption of $n \geq 100$).

Now we estimate $\xi = (32\sigma^2/\mu_*^2 + (8/3)\sqrt{n}\epsilon/(\mu_*\Delta))^{-1}$. By $\sigma^2/\mu_*^2 \leq 4\sqrt{n}$, $\Delta \geq 1$ and $\mu_* \geq 1$, we have $\xi \geq (128\sqrt{n} + (8/3)\sqrt{n})^{-1} \geq (131\sqrt{n})^{-1}$. Then we immediately have $e^{-|P'_O|\epsilon^2\Delta^{-2}\xi} \leq e^{-\gamma\sqrt{n}\epsilon^2\Delta^{-2}\xi/2} \leq e^{-262\epsilon^{-2}\Delta^2\sqrt{n}\epsilon^2\Delta^{-2}(131\sqrt{n})^{-1}/2} \leq e^{-\ln 4n} = 1/4n$. Inequality (3) then follows from this and inequality (4).

Below we show that, $|\mu'_O - \mu_O| > \epsilon S/n$ implies that $|\mu'_* - \mu_*| \geq \epsilon\mu_*/4\Delta$. If this is the case, by inequality (3), we will know that the latter event happens with probability no more than $1/4n$, which also implies that the former event happens with probability no more than $1/4n$, and thus the lemma follows. We will prove the claim by showing that $|\mu'_* - \mu_*| < \epsilon\mu_*/4\Delta$ implies that $|\mu'_O - \mu_O| \leq \epsilon S/n$.

Assume that $|\mu'_* - \mu_*| < \epsilon\mu_*/4\Delta$. Multiplying each side of this inequality by a factor of $F(r_O)/\sqrt{n}$, we have

$$|(\sum_{p\in P'_O}(f(p)+F(r_O)/\sqrt{n})/|P'_O|) - (\sum_{p\in P_O}(f(p)+F(r_O)/\sqrt{n})/|P_O|)| \le \epsilon((\sum_{p\in P_O} f(p))/|P_O| + F(r_O)/\sqrt{n})/4\Delta. \tag{5}$$

Rearranging the terms gives us

$$|\mu'_O - \mu_O| \le \epsilon(\mu_O + F(r_O)/\sqrt{n})/4\Delta. \tag{6}$$

In the following, we will obtain an upper bound on $(\mu_O + F(r_O)/\sqrt{n})/4\Delta$ in terms of S.

We first consider $\mu_O/4\Delta$. For each $p \in P \setminus P_O$, since $\|p-q\| \le r_O$, we have $f(p) \ge f(p')$ for any $p' \in P_O$. Therefore we have $\sum_{p\in P\setminus P_O} f(p)/|P\setminus P_O| \ge \sum_{p'\in P_O} f(p')/|P_O| = \mu_O$. Note that S/n is the mean value of $f(p)$ for all $p \in P$. Thus we have $S/n \ge \min(\mu_O, \sum_{p\in P\setminus P_O} f(p)/|P\setminus P_O|)$. Hence we get $S/2n \ge \mu_O/2 \ge \mu_O/4\Delta$.

Now we bound $F(r_O)/4\sqrt{n}\Delta$ in terms of S. By the fact that $r_O = r_\alpha/\lambda$, we have $F(r_O) \le F(r_\alpha)\Delta$. Thus we know that $F(r_O)/4\sqrt{n}\Delta \le F(r_\alpha)/4\sqrt{n}$. From the good sample condition, we have $|P_\alpha| \ge \sqrt{n}/2$. For each $p \in P_\alpha = P \cap B(q, r_\alpha)$, since $\|p-q\| \le r_\alpha$, it follows that $f(p) \ge F(r_\alpha)$. Therefore we get $S = \sum_{p\in P} f(p) \ge \sum_{p\in P_\alpha} f(p) \ge F(r_\alpha)\sqrt{n}/2$. It then follows that $F(r_O)/4\sqrt{n}\Delta \le F(r_\alpha)/4\sqrt{n} \le S/2n$.

Combining the above results and recall (6), we obtain $|\mu'_O - \mu_O| \le \epsilon S/n$.

To summarize, we have showed that $|\mu'_* - \mu_*| < \epsilon\mu_*/4\Delta$ implies $|\mu'_O - \mu_O| \le \epsilon S/n$, which means that $|\mu'_O - \mu_O| > \epsilon S/n$ implies $|\mu'_* - \mu_*| \ge \epsilon\mu_*/4\Delta$. This completes the proof. ◀

Below is a result for soft boundary range search. The details of the method will be discussed in next section.

▶ **Lemma 7.** *For any $\lambda > 1$ and $\tau > 0$, there exists a λ-approximate soft boundary range search data structure that can be built in time $O(dn^{1+1/2\lambda+\tau})$. Each query, provided that the good sample condition is satisfied,*

1. *reports all the points in $P_\Omega = P \cap B(q, r_\alpha/\lambda)$ in Algorithm 1 (*i.e. *$P'_\Omega = P_\Omega$) with probability at least $1 - 1/4n$;*
2. *takes time $O(dn^{1/2\lambda+1/2+\tau})$.*

Finally, we have the main theorem for our algorithm.

▶ **Theorem 8.** *With probability at least $1 - 1/n$, $\bar{S}$ produced by Algorithm 1 satisfies the inequality $|\bar{S} - S| \le 2\epsilon S$.*

Proof. In the following argument, we assume that the good sample condition is satisfied.

P can be partitioned into 2 subsets according to their distances to q: $P = P_O \cup P_\Omega$, where $P_O = P \setminus B(q, r_O)$ (as defined before), and $P_\Omega = P \cap B(q, r_\alpha/\lambda) = P \setminus P_O$.

First, we show that the value $S'_O = \mu'_O|P'_O|n/m$ is a good approximation of $\sum_{p\in P_O} f(p)$. By Lemma 6, we know that with probability at least $1 - 1/4n$, $|\mu'_O - \mu_O| \le \epsilon S/n$. Thus, we get $|\mu'_O|P'_O|n/m - \mu_O|P'_O|n/m| \le \epsilon S|P'_O|/m \le \epsilon S$. By the good sample condition, we have $||P'_O|n/m - |P_O|| \le \epsilon|P_O|$. Since $P_O \subseteq P$, clearly we know that $|P_O|\mu_O = \sum_{p\in P_O} f(p) \le S$. Thus, we have $|\mu_O|P'_O|n/m - \mu_O|P_O|| \le \epsilon\mu_O|P_O| \le \epsilon S$.

$$
\begin{aligned}
|S'_O - \sum_{p \in P_O} f(p)| &= |\mu'_O|P'_O|n/m - \mu_O|P_O|| & (7)\\
&= |\mu'_O|P'_O|n/m - \mu_O|P'_O|n/m + \mu_O|P'_O|n/m - \mu_O|P_O|| & (8)\\
&\leq |\mu'_O|P'_O|n/m - \mu_O|P'_O|n/m| + |\mu_O|P'_O|n/m - \mu_O|P_O|| & (9)\\
&\leq \epsilon S + \epsilon S = 2\epsilon S. & (10)
\end{aligned}
$$

For P_Ω, by Lemma 7, we know that this set is identical to P_Ω with probability at least $1 - 1/4n$. Thus we have $S'_\Omega = \sum_{p \in P_\Omega} f(p)$ with probability at least $1 - 1/4n$.

From the above results, we immediately know that when the good sample condition is satified, $S - \bar{S} \leq 2\epsilon S$ with probability at least $1 - 1/2n$. Since the good sample condition holds with probability at least $1 - 1/2n$, the theorem then follows. ◀

3 Soft Boundary Range Reporting using Approximate Nearest Neighbor Search

In this section we present a method to report points in $P_\Omega = P \cap B(q, r_\alpha/\lambda)$. We assume that $\sqrt{n}/2 \leq |P_\alpha| \leq 2\sqrt{n}$, which is a part of the good sample condition.

We reduce the range search query to a number of nearest neighbor queries. Observe that, since $\sqrt{n}/2 \leq |P_\alpha| \leq 2\sqrt{n}$, if we take a sample Q of $\lceil \sqrt{n}/2 \rceil$ points from P uniformly and independently, with at least constant probability, Q and P_α share exactly 1 common point.

▶ **Lemma 9.** *For $n \geq 100$, the probability of the event that $|P_\alpha \cap Q| = 1$ happens with probability at least $\rho = 1/60$*

Proof. The probability that the event happens can be computed as follows.

$$\mathbb{P}(|P_\alpha \cap Q| = 1) = \lceil \sqrt{n}/2 \rceil \frac{|P_\alpha|}{n}(1 - \frac{P_\alpha}{n})^{\lceil \sqrt{n}/2 \rceil - 1}.$$

Since $|P_\alpha| \geq \sqrt{n}/2$, we have $\lceil \sqrt{n}/2 \rceil \frac{|P_\alpha|}{n} (\geq \sqrt{n}/3) \frac{|P_\alpha|}{n} \geq 1/6$.

Also, from $|P_\alpha| \leq 2\sqrt{n}$, we get $(1 - \frac{P_\alpha}{n})^{\lceil \sqrt{n}/2 \rceil - 1} \geq (1 - \frac{P_\alpha}{n})^{\sqrt{n}} \geq (1 - 2/\sqrt{n})^{\sqrt{n}}$. It is well known that $f(x) = (1 - 1/x)^x$ is monotonously increasing for $x > 1$. Thus $(1 - 2/\sqrt{n})^{\sqrt{n}} \geq (1 - 1/5)^{10} > 1/10$. Combining this with the above inequality, the lemma follows. ◀

If Q and P_α share exactly 1 common point, clearly every point in P_α have the same probability to be the common point. There are at most $2\sqrt{n}$ points in P_α; therefore every point has a probability at least $\rho/2\sqrt{n}$ to be the only common point of Q and P_α. Similar observations are used in some other range search techniques[4].

Now, if we are allowed to perform a λ-approximate nearest neighbor search on Q, and if a point $p \in P_\Omega \subseteq P_\alpha$ happens to be the only common point of P_α and Q, then the approximate nearest neighbor search will output p. This is because any other point in Q must be in $P \setminus P_\alpha$, and thus their distance to q must be larger than $r_\alpha = \lambda r_O \geq \|p - q\|$, which means that p is the only λ-approximate nearest neighbor of q in Q.

Therefore, if we sample a point set Q as stated above, and build a nearest neighbor data structure that is able to output a λ-approximate nearest neighbor Q for any q, with at least constant success probability $\delta > 0$(*e.g.* using technique in [9]), then for any $p \in P_\Omega$, this data structure will be able to discover p with probability at least $(\delta\rho/2\sqrt{n})$.

This suggest us to build a range search data structure in the following way. For some number t, independently create t samples $Q_1, \ldots, Q_t$, each one has size $\lceil\sqrt{n}/2\rceil$, by sampling uniformly and independently from P. Then we build an λ-approximate nearest neighbor data structure with success query probability δ for each of $Q_1, \ldots, Q_t$. For the reporting query, given q and r_α, we perform an λ-approximate nearest neighbor.

We set t so that $(1 - \delta\rho/2\sqrt{n})^t$ (*i.e.* the probability that a certain point $p \in P_\Omega$ is not reported) is less than $1/8\sqrt{n}n$. By simple calculation, we know that it is possible to find such a t that satisfies the condition $t = O(\sqrt{n}\log n)$. Since P_Ω contains no more than $2\sqrt{n}$ points, it means that if we perform a nearest neighbor search for all $Q_1, \ldots, Q_t$, with probability at least $1 - 1/4n$, we are able to output all points in P_Ω.

Note that using this scheme, for each range reporting query, we are required to perform $t = O(\sqrt{n}\log n)$ times nearest neighbor search queries. For any $\tau > 0$, it is possible to build a nearest neighbor data structure that answer each nearest neighbor search query in $O(d(\sqrt{n})^{1/\lambda+\tau})$ time[9]. Therefore the time required for a reporting process is $O(td(\sqrt{n})^{1/\lambda+\tau})$. By $t = O(\sqrt{n}\log n)$, we know that for any $\tau' > 0$, it is possible to perform the range reporting operation in time $O(d(\sqrt{n})^{1+1/\lambda+\tau'})$. For the construction time, each nearest neighbor data structure is built on a $O(\sqrt{n})$ point set, which can be built within time $O(d(\sqrt{n})^{1+1/\lambda+\tau})$ for any $\tau > 0$, Therefore, the total construction complexity is $(dn^{1+1/2\lambda+\tau'})$ for any $\tau' > 0$. The bounds for Lemma 7 are proved.

References

1 Peyman Afshani and Timothy M. Chan. Optimal halfspace range reporting in three dimensions. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 180–186. SIAM, 2009.

2 Pankaj K. Agarwal, Sariel Har-peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. In *Combinatorial and Computational Geometry, MSRI*, pages 1–30. University Press, 2005.

3 Alexandr Andoni, Piotr Indyk, Huy L Nguyễn, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1018–1028. SIAM, 2014.

4 Boris Aronov and Sariel Har-Peled. On approximating the depth and related problems. *SIAM Journal on Computing*, 38(3):899–921, 2008.

5 Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

6 D. Z. Chen, Z. Huang, Y. Liu, and J. Xu. On clustering induced voronoi diagrams. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 390–399, 2013.

7 Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

8 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578. ACM, 2011.

9 Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8(1):321–350, 2012.

10 Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300. ACM, 2004.

11 Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

12 Saladi Rahul and Yufei Tao. Efficient top-k indexing via general reductions. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 277–288. ACM, 2016.

13 Cheng Sheng and Yufei Tao. Dynamic top-k range reporting in external memory. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, pages 121–130. ACM, 2012.