





Sublinear Data Structures for Nearest Neighbor in Ultra High Dimensions

Martin G. Herold  

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Danupon Nanongkai  

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Joachim Spoerhase  

University of Liverpool, UK

Nithin Varma  

University of Cologne, Germany

Zihang Wu  

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Abstract

Geometric data structures have been extensively studied in the regime where the dimension is much smaller than the number of input points. But in many scenarios in Machine Learning, the dimension can be much higher than the number of points and can be so high that the data structure might be unable to read and store all coordinates of the input and query points.

Inspired by these scenarios and related studies in feature selection and explainable clustering, we initiate the study of geometric data structures in this ultra-high dimensional regime. Our focus is the *approximate nearest neighbor* problem.

In this problem, we are given a set of n points $C \subseteq \mathbb{R}^d$ and have to produce a *small* data structure that can *quickly* answer the following query: given $q \in \mathbb{R}^d$, return a point $c \in C$ that is approximately nearest to q , where the distance is under ℓ_1 , ℓ_2 , or other norms. Many groundbreaking $(1 + \epsilon)$ -approximation algorithms have recently been discovered for ℓ_1 - and ℓ_2 -norm distances in the regime where $d \ll n$.

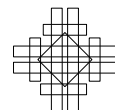
The main question in this paper is: *Is there a data structure with sublinear ($o(nd)$) space and sublinear ($o(d)$) query time when $d \gg n$?* This question can be partially answered from the machine-learning literature: ¹

- For ℓ_1 -norm distances, an $\tilde{O}(\log(n))$ -approximation data structure with $\tilde{O}(n \log d)$ space and $O(n)$ query time can be obtained from explainable clustering techniques [Dasgupta et al. ICML'20; Makarychev and Shan ICML'21; Esfandiari, Mirrokni, and Narayanan SODA'22; Gamlath et al. NeurIPS'21; Charikar and Hu SODA'22].
- For ℓ_2 -norm distances, a $(\sqrt{3} + \epsilon)$ -approximation data structure with $\tilde{O}(n \log(d)/\text{poly}(\epsilon))$ space and $\tilde{O}(n/\text{poly}(\epsilon))$ query time can be obtained from feature selection techniques [Boutsidis, Drineas, and Mahoney NeurIPS'09; Boutsidis et al. IEEE Trans. Inf. Theory'15; Cohen et al. STOC'15].
- For ℓ_p -norm distances, a $O(n^{p-1} \log^2(n))$ -approximation data structure with $O(n \log(n) + n \log(d))$ space and $O(n)$ query time can be obtained from the explainable clustering algorithms of [Gamlath et al. NeurIPS'21].

An important open problem is whether a $(1 + \epsilon)$ -approximation data structure exists. This is not known for any norm, even with higher (e.g. $\text{poly}(n) \cdot o(d)$) space and query time.

In this paper, we answer this question affirmatively. We present $(1 + \epsilon)$ -approximation data structures with the following guarantees.

¹ All data structures discussed here are randomized and answer each query correctly with $\Omega(1)$ probability. The space complexity refers to the number of words and the time is in the word-RAM model. $\tilde{O}(f(n))$ hides $\text{polylog}(f(n))$ factors



- For ℓ_1 - and ℓ_2 -norm distances: $\tilde{O}(n \log(d)/\text{poly}(\epsilon))$ space and $\tilde{O}(n/\text{poly}(\epsilon))$ query time. We show that these space and time bounds are tight up to $\text{poly}(\log n/\epsilon)$ factors.
- For ℓ_p -norm distances: $\tilde{O}(n^2 \log(d)(\log \log(n)/\epsilon)^p)$ space and $\tilde{O}(n(\log \log(n)/\epsilon)^p)$ query time.

Via simple reductions, our data structures imply sublinear-in- d data structures for some other geometric problems; e.g. approximate orthogonal range search (in the style of [Arya and Mount SoCG'95]), furthest neighbor, and give rise to a sublinear $O(1)$ -approximate representation of k -median and k -means clustering. We hope that this paper inspires future work on sublinear geometric data structures.

2012 ACM Subject Classification Theory of computation \rightarrow Nearest neighbor algorithms; Theory of computation \rightarrow Data structures design and analysis; Theory of computation \rightarrow Computational geometry; Mathematics of computing \rightarrow Probabilistic algorithms

Keywords and phrases sublinear data structure, approximate nearest neighbor

Digital Object Identifier 10.4230/LIPIcs.SoCG.2025.56

Related Version Full Version: <https://arxiv.org/pdf/2503.03079>

Funding Martin G. Herold: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 399223600.

Joachim Spoerhase: Part of this work was done when the author was a researcher at Max Planck Institute for Informatics & Saarland Informatics Campus, Saarbrücken, Germany, and at Aalto University, Finland. Partially supported by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 759557).

Nithin Varma: Part of this work was done when the author was a researcher at Max Planck Institute for Informatics & Saarland Informatics Campus, Saarbrücken, Germany.

1 Introduction

Nearest Neighbor. The *nearest neighbor* problem in geometric spaces is defined as follows: We are given a set of n points $C \subseteq \mathbb{R}^d$ to preprocess and then have to produce a data structure that can answer the following queries: given $q \in \mathbb{R}^d$, return the point c^* in C whose distance to q is smallest, where the distance can be induced by ℓ_1 -, ℓ_2 -, and other norms.

In the so-called *low-dimensional regime* [4] – when $d = o(\log n)$ and thus the data structure's space and time complexities can be *exponential* in d – classical data structures can solve this problem efficiently (e.g. [15, 25, 26, 8]). Recently, novel techniques such as locality-sensitive hashing and sketching (e.g. [22, 2, 3, 5, 23]) have led to a new wave of data structures in the *high-dimensional regime* that solve the problem approximately: for any $\alpha \geq 1$, a α -approximation data structure is the one that returns a point $c \in C$ whose distance to q is at most α times the distance between c^* and q . These recent data structures typically require space and time complexities that are linear or super linear in d , which is acceptable in most applications. In fact, many papers start with an assumption that $d = O(\log n)$ which can be assumed due to Johnson-Lindenstrauss lemma [24] at the cost of additional $\tilde{O}(d)$ space and query time.

Ultra-High Dimensions. But what if we cannot read all coordinates of the query q and cannot store all coordinates of all n points in C ? There can be various reasons for such a situation to arise. For example, we may be unable to access the whole q due to the high-cost concern of obtaining the data (e.g. high-cost medical tests or genome sequencing) and privacy concerns (e.g. providing only the necessary medical or genetic information while protecting

unrelated sensitive data). More crucially, the size of the data, i.e., d might be prohibitively large to query all of its attributes (as is usually the case for genetic data) and, yet, n can be much smaller than d . For example, each point in C may be a center of a cluster among some small number of clusters obtained from prior training and we aim to classify a new data point $q \in \mathbb{R}^d$ by assigning it to the nearest representative. In this case, to accurately classify q without accessing all attributes of q , we need to exploit the fact that there are typically a small number of clusters. Additionally, classifying q based on a few attributes is crucial for *explainability* – when it is important for human users to know which factors play a role in the classifying decision.

Related Work and Existing Results. Scenarios like the above are a motivation behind at least two lines of research in machine learning. (i) In *Feature Selection*, the goal is to select a subset of relevant features of the input dataset. For example, Boutsidis, Drineas, and Mahoney in NeurIPS’09 [9] (also see [10]) show that given a set $S \subseteq \mathbb{R}^d$ and a parameter $k \ll d$, we can select roughly k coordinates such that we are required to only access these coordinates when computing k -means clustering, at the expense of a $(3 + \epsilon)$ multiplicative approximation factor. The approximation factor was subsequently improved to $(1 + \epsilon)$ in the STOC’15 paper by Cohen et al. [16]. (ii) In *Explainable Clustering*, given data that are already classified into k clusters via cluster centers, the goal is to construct a decision tree with some desired properties that can classify the input data points into clusters without too much loss in the clustering cost compared to the original clustering. For example, Makarychev and Shan [27] show that given a k -median clustering on ℓ_1 distance, one can construct such a decision tree that returns a clustering whose cost is $\tilde{O}(\log k)$ times the original cost. Since the number of accessed coordinates intuitively plays an important role in explainability, it is not surprising that one of the desired properties of the decision tree is that it never accesses more than $O(k)$ coordinates.

The feature selection and explainable clustering problems are *static problems*; i.e., the algorithms for these problems receive all of their inputs before returning the output. This is in contrast to the nearest neighbor problem which is a *data structure problem*. Nevertheless, given the connections between clustering and nearest neighbor problems, techniques from some of the previous works on these static problems can be extended to the nearest neighbor problem. (All data structures discussed here are randomized and answer each query correctly with $\Omega(1)$ probability. The space complexity refers to the number of words and the time is in the word-RAM model. $\tilde{O}(f(n))$ hides $\text{polylog}(f(n))$ factors.)

- For ℓ_1 -norm distances, an $O(\log(n))$ -approximation data structure with $O(n \log(n) + n \log d)$ space and $O(n)$ query time can be obtained by adapting the explainable clustering algorithms of [20, 29]. This is because these algorithms only require the center of each cluster, and not the remaining input data points. Weaker results can also be obtained from [18, 27, 31, 19].
- For ℓ_2 -norm distances, a $(\sqrt{3} + \epsilon)$ -approximation data structure with $\tilde{O}(n \log(d)/\epsilon^2)$ space and $O(n \log(n)/\epsilon^2)$ query time can be obtained by reducing to feature selection algorithms [9, 10, 16] (see Appendix F).² Weaker results for the case of ℓ_2 -norm distances can also be obtained from explainable clustering algorithms of [28, 27, 18, 19, 11, 31].

² In Appendix F we also argue why a natural reduction to feature selection [16] fails to give a $(1 + \epsilon)$ -approximate nearest neighbor data structure.

- For ℓ_p -norm distances, an $O(n^{p-1} \log^2(n))$ -approximation data structure with $O(n \log(n) + n \log(d))$ space and $O(n)$ query time can be obtained from the explainable clustering algorithms of [19].

In contrast to other regimes, where many efficient $(1 + \epsilon)$ -approximation data structures are known, no $(1 + \epsilon)$ -approximation data structure is known for the ultra-high dimensional regime for any norm, and even when we allow higher space and query time (e.g. $\text{poly}(n) \cdot o(d)$ space and query time).

Our Results. We first observe that every $(1 + \epsilon)$ -approximate nearest neighbor data structure requires $\Omega(n)$ space and must make $\Omega(n)$ queries (see Theorem G.1). Our main contributions are data structures with nearly matching space and query bounds in the ultra-high dimensional regime. Our data structures require polynomial preprocessing time, answer each query correctly with $1 - \delta$ probability, and guarantee the following bounds.

- For ℓ_1 -norm distances: $O(n \log^2(n/\delta) \log(d)/(\epsilon^5 \delta^4))$ space and $O(n \log(n/\delta)/(\epsilon^3 \delta^2))$ query time. See Theorem 4.2.
- For ℓ_2 -norm distances: $O(n \log^2(n/\delta) \log(d)/(\epsilon^6 \delta^4))$ space and $O(n \log(n/\delta)/(\epsilon^3 \delta^2))$ query time. See Theorem B.3.
- For ℓ_p -norm distances: $O(n^2 \log(n/\delta) \log(d) (\log \log n)^p / \epsilon^{p+2})$ space and $O(n \log(n/\delta) (\log \log n)^p / \epsilon^{p+2})$ query time. See Theorem D.1.

Our data structure for ℓ_1 - and ℓ_2 -norm distances uses essentially the same space and query time as the previously best result and provides a stronger approximation guarantee. For other norms, our data structure needs higher (but sublinear) space and gives strong approximation guarantees.

Connection to Explainable Clustering. A recent line of research investigates explainable clustering [18, 27, 31, 19, 20, 29]. The input is a set C of k cluster centers. The output is a (random) decision tree assigning any query point to a cluster represented by a label in $[k]$. The current best result [20] for explainable k -median clustering under ℓ_1 -distances guarantees that for *any* set P of data points the clustering generated by the decision tree is in expectation within a factor $O(\log k)$ of the optimal assignment of P to the centers in C , that is, of the nearest-center assignment. The decision tree has k leaves. Therefore, it can be interpreted as $O(\log k)$ -approximate representation of the clustering C with sublinear $\tilde{O}(k)$ space and query complexity. There are two aspects why this representation is considered explainable: (i) the decision tree is easy to interpret for humans (ii) it is compact as it has only sublinear $\tilde{O}(k)$ space and query complexity. When applied to a single query point these data structures constitute a $O(\log k)$ -approximate nearest neighbor data structure in expectation.

Our techniques have implications in the converse direction. We argue (see Section C) that our techniques give a sublinear $\tilde{O}(k)$ space and query complexity data structure for an $O(1)$ -approximate nearest neighbor in *expectation*. This implies a sublinear $O(1)$ -approximate representation for k -median clustering improving upon the $O(\log k)$ -approximation obtained via decision trees. Notice that while our data structure does not produce decision trees it does constitute a compact representation of the clustering. Our techniques also imply a sublinear $O(1)$ -approximate representation of k -means clustering in Euclidean space improving upon the $O(\log^2 k)$ -bound implied by explainable k -means clustering [28].

Perspective: Sublinear Data Structures. Geometric data structures are fundamental computational objects that have been extensively studied and used for decades. To the best of our knowledge, there was no prior result on geometric data structure problems in the ultra-high dimensional regime. Showing that it is possible to design sublinear geometric data structures is the main conceptual contribution of our paper.

As another example of where our techniques are applicable, consider the orthogonal range search problem. Here, we are given a set $C \subseteq \mathbb{R}^d$ to preprocess and need to answer the following query: Given q_1 and q_2 in \mathbb{R}^d , return all points $p \in C$ such that the i^{th} attribute of p has value between the i^{th} attributes of q_1 and q_2 , for every i . This problem can be solved using range trees and k - d trees [7, 30, 12, 13] in the low-dimensional regime. For higher dimensions, some approximation algorithms have been proposed where some other points can be returned as long as they are not “far” from the query [6, 14, 17], i.e., the distance between the point returned and the range is within an ϵ fraction of the distance between q_1 and q_2 – we call such a range search as $(1 + \epsilon)$ -approximate range search. We approach this problem, as before, from a data-structure point of view. Using similar techniques, we can design a data structure of size $\tilde{O}(n)$ that processes n centers such that given any “valid” orthogonal range (q_1, q_2) (i.e., a range which contains at least one center point), queries only $\tilde{O}(n/\epsilon^2)$ coordinates of q_1 and q_2 to report all centers that are within the $(1 + \epsilon)$ -approximate range. The details of this can be found in Appendix E.

Besides orthogonal range search, simple reductions also lead to data structures for other proximity problems such as the furthest neighbor problem, where we can guarantee the same bounds as the nearest neighbor problem.

We hope that this paper inspires future work on sublinear geometric data structures.

Organization. Section 2 gives an overview of the main technical contributions of the paper. Section 4 describes our $(1 + \epsilon)$ -approximate nearest neighbor data structures with $\tilde{O}(n)$ space and query complexity for the ℓ_1 - and ℓ_2 -norm.

Appendix A contains the outline of a simple data structure with space $\tilde{O}(n^2)$ and query time $\tilde{O}(n)$ for ℓ_1 -norm, with some simplifying assumptions as warmup. Appendix B contains missing proofs of Section 4 and extension to ℓ_2 case. Appendix C describes our expected constant-approximate nearest neighbor data structure, and shows the connection to explainable clustering. Appendix D describes our $(1 + \epsilon)$ -approximate nearest neighbor data structure with space $\tilde{O}(n^2)$ and query time $\tilde{O}(n)$ that works for general ℓ_p -norms. Appendix E shows the implications of our results for Range Search problem. Appendix F shows the connection between approximate nearest neighbor and feature selection for clustering. Appendix G shows $\Omega(n)$ space and query tight lower bound.

2 Technical Overview

Warmup: Quadratic Space via Pairwise Sampling. Let c be an input point, let q be an unknown query point, and let $\|c - q\|_1 = \sum_b |c^{(b)} - q^{(b)}|$ denote their ℓ_1 distance. First, observe that approximating this distance requires memorizing *all* coordinates of c because c and q may differ in only a single coordinate. Our strategy is to *compare* distances to multiple input points without knowing the (approximate) distances. For intuition, consider when we have two input points c_1 and c_2 . Consider an extreme case when c_1 and c_2 differ only in one coordinate b . In this case, keeping only this coordinate suffices to tell for any query q if it is

closer to c_1 or c_2 . To extend this intuition to when c_1 and c_2 differ in many coordinates, a natural idea is to sample each coordinate b independently with probability proportional to the difference $|c_1^{(b)} - c_2^{(b)}|$; more precisely, the probability that we sample coordinate b is³

$$p^{(b)} = \frac{|c_1^{(b)} - c_2^{(b)}|}{\|c_1 - c_2\|_1}.$$

We further scale the sample entry by a factor of $1/p^{(b)}$. The expected number of sampled coordinates is then $\sum_b p^{(b)} = \sum_b |c_1^{(b)} - c_2^{(b)}| / \|c_1 - c_2\|_1 = 1$. Let $\epsilon \in (0, 1)$ be the approximation parameter, and let $\delta \in (0, 1)$ be a bound on the error probability. We repeat this process to store a set I of $O(\log(1/\delta)/\epsilon^2)$ coordinates in total. For $x \in \mathbb{R}^d$, we denote by $x^{(I)}$ the restriction of x to coordinates in I . We denote by $S = \text{diag}((1/p^{(b)})_{b \in I})$ denote the diagonal matrix representing the scaling factors. Then $Sx^{(I)}$ is the vector obtained from applying the above sampling process to x .

Under what we call *bounding box assumption* (which we will remove later), where $\min(c_1^{(b)}, c_2^{(b)}) \leq q^{(b)} \leq \max(c_1^{(b)}, c_2^{(b)})$ for each coordinate b , we can find a $(1 + \epsilon)$ -approximate nearest neighbor between c_1 and c_2 . This is due to the following property. (Details in Section A.)

Comparator Property: Let $\epsilon, \delta \in (0, 1)$ and assume we sample a set I of $O(\log(1/\delta)/\epsilon^2)$ coordinates as described above. If $\|c_2 - q\|_1 > (1 + \epsilon)\|c_1 - q\|_1$ then $\|Sc_2^{(I)} - Sq^{(I)}\|_1 > (1 + \epsilon/2)\|Sc_2^{(I)} - Sq^{(I)}\|_1$ with probability $1 - \delta$.

With this idea, we can construct an $\tilde{O}(n^2)$ -space data structure by repeating the above for every pair (c_i, c_j) of input points. It is not hard to find the approximately nearest neighbor to q with $n - 1$ comparisons: this is very simple if the comparator is exact, and the same idea can be extended for our approximate comparator as well; see Section A for details (also see [1] for previous work on imprecise comparisons).

Towards Linear Space via Global Sampling. The drawback of the previous algorithm is its $\tilde{O}(n^2)$ space complexity. To avoid this, a natural idea is to reduce the number of pairs for which we perform the above sampling process. However, it is unclear to us if this is possible at all. Another idea is to do the above process *more globally* instead of pairwise. Here is a simple way to do this: Let $C = \{c_1, \dots, c_n\}$ be the set of input points. We (independently) sample each coordinate b with probability $p^{(b)} = \max_{i,j} p_{ij}^{(b)}$ where $p_{ij}^{(b)} = |c_i^{(b)} - c_j^{(b)}| / \|c_i - c_j\|_1$ denotes the probability of sampling b under the pairwise sampling process for $c_i, c_j \in C$.

At least two questions arise from the idea above: (i) For any pair c_i and c_j and query q in the bounding box can we still tell (approximately) if c_i or c_j is nearer to q as in the pairwise process (in particular, are we creating noise by sampling some coordinates with higher probability than $p_{ij}^{(b)}$)? This is not so hard to address: Despite sampling each coordinate with probability higher than $p_{ij}^{(b)}$, the expectation does not change due to rescaling by $1/p^{(b)}$, and the concentration behavior is only better. More precisely, the comparator property can still be proven.

A more unclear point is: (ii) How many coordinates do we sample in total? For example, why is it not possible that in the worst case every pair c_i and c_j needs different coordinates and thus we are forced to select $\Omega(n^2)$ coordinates in total. We show that this bad situation

³ This can be viewed as an application of so-called importance sampling – a technique for variance reduction. The independent sampling process described here differs slightly from the one in Section A.

will not happen. To give an idea, we consider the special case $C \subseteq \{0, 1\}^d$. The overall expected number of coordinates sampled is $S(C) = \sum_b p^{(b)} = \sum_b \max_{ij} p_{ij}^{(b)}$. (We ignore sampling repetition, which increases our bound only by $O(\log n)$.) Assume w.l.o.g. that (c_1, c_2) is the closest pair in C . Notice that $p^{(b)} = p_{12}^{(b)}$ for every coordinate b in the set D of coordinates where c_1 and c_2 differ; they minimize $\|c_i - c_j\|_1$ and hence maximize $p_{ij}^{(b)}$. Therefore $S(C)$ drops by at most $\sum_b p_{12}^{(b)} = \sum_b |c_1^{(b)} - c_2^{(b)}| / \|c_1 - c_2\|_1 = 1$ if we remove the coordinates in D from all input points. On the other hand, we can identify c_1 and c_2 after removing D obtaining a set C' with at most $n - 1$ points. Thus $S(C) \leq S(C') + 1$. Using induction we can upper bound $S(C)$ by n . This inductive proof strategy can be extended to the general ℓ_1 metric although the proof is notably more technical. Even though ℓ_1 and ℓ_2 metrics are equivalent on the Boolean cube, we do not know how to extend the above proof strategy to the general ℓ_2 metric. To obtain a linear bound for ℓ_2 we rely on an entirely different, linear-algebraic approach leveraging a latent connection between our sampling probabilities and the singular-value decomposition. Our proof is based on a connection to the feature selection approach by Boustidis et al. [10] for k -Means clustering. In fact, our sampling probabilities can be upper bounded by theirs, which are based on SVD. Notice, however, that we achieve a $(1 + \epsilon)$ -approximation for nearest neighbor as compared to their $(3 + \epsilon)$ -approximation for k -Means.⁴

Let I denote the set of coordinates sampled by our global scheme. Memorizing these coordinates for all input points would still require $n|I| = \tilde{O}(n^2)$ space. To obtain $\tilde{O}(n)$ space we apply standard dimension reduction tools such as the Johnson-Lindenstrauss transform for ℓ_2 or the technique by Indyk [21] for ℓ_1 . Specifically, we sample a suitable random $m \times |I|$ matrix M , where $m = O(\log n)$, and store the set $C' = \{MSc_i^{(I)} \mid c_i \in C\}$ requiring $mn = \tilde{O}(n)$ space. Storing M itself requires $m|I| = \tilde{O}(n)$ space, too, as does storing I itself. Upon receiving a query q , we compute $MSq^{(I)}$ and output the nearest neighbor in C' , which takes $\tilde{O}(n)$ coordinate queries. Here we use that the comparator property guarantees $\|Sq^{(I)} - Sc_j^{(I)}\|_1 > (1 + \epsilon/2)\|Sq^{(I)} - Sc_i^{(I)}\|_1$ if $\|q - c_j\|_1 > (1 + \epsilon)\|q - c_i\|_1$. Therefore, if we bound the distance distortion of the dimension reduction by, say, $1 + \epsilon/10$ then $F(MSq^{(I)}, MSc_j^{(I)}) > (1 + \epsilon/5)F(MSq^{(I)}, MSc_i^{(I)})$ for the function F from the ℓ_1 dimension reduction by Indyk [21]. This is sufficient for approximately comparing the distances.

Removing the Bounding Box Assumption. If a query point q does not satisfy the bounding box property for two points $c_i, c_j \in C$ the comparator property may fail with too high probability. Specifically, it may happen that $\|q - c_j\|_1 > (1 + \epsilon)\|q - c_i\|_1$ and yet $\|Sq^{(I)} - Sc_j^{(I)}\|_1 = (1 + \xi)\|Sq^{(I)} - Sc_i^{(I)}\|_1$ for some positive $\xi \ll \epsilon$. In such a scenario, applying dimension reduction via matrix M may result in $F(MSq^{(I)}, MSc_j^{(I)}) < F(MSq^{(I)}, MSc_i^{(I)})$ because it distorts distances by factor $1 + \Theta(\epsilon)$.

A natural attempt to resolve this issue is to “truncate” the query point so it comes sufficiently close to the bounding box. This approach is described in Section D and can in fact be applied to general ℓ_p metrics. However, it requires explicitly storing the projected points and therefore $\tilde{O}(n^2)$ space.

To retain our near-linear space bound, we do not actually change the sampling or the query procedure. Rather, we relax in our analysis on the requirement that the comparator property needs to hold for all point pairs. Specifically, we argue (using Markov) that with

⁴ In Section F, we provide a formal reduction to feature selection by Boustidis et al. [10]. There is follow-up work achieving even guarantee $1 + \epsilon$ for feature selection in clustering [16]. However, we do not know how to apply their approach to our setting. In particular, the above reduction is not applicable.

sufficiently high probability our distance estimate to the (unknown) nearest neighbor o is within a bounded factor (*). We argue that under this condition all comparisons with o do satisfy the comparator property, which is enough to satisfy correctness. Towards this, we distinguish for each $c \in C$ between *good* coordinates b where $q^{(b)}$ is relatively close to the interval $[c^{(b)}, o^{(b)}]$. For the set of good coordinates, strong concentration properties hold – similar to those under the bounding box assumption. For the remaining (bad) coordinates, we use two properties: First, the query point is indifferent between c and o on the set of bad coordinates as it is sufficiently distant and thus indifferent between o and c on these coordinates. This and property (*) automatically implies that the total contribution of the bad coordinates cannot be too high. Together, these properties imply that (a relaxation of) the comparator property holds for all comparisons with o thereby implying the $(1 + \epsilon)$ -approximation guarantee also after removing the bounding box assumption. For details, we refer to the full technical sections below.

3 Preliminaries

In this section, we list some basic terminology, definitions and notation that we use throughout.

► **Definition 3.1** (Approximate Notation). *Given real numbers $a, b > 0$ and $0 < \epsilon < 1$, we say that $a \approx_\epsilon b$ if $(1 - \epsilon)b \leq a \leq (1 + \epsilon)b$.*

► **Definition 3.2** (Coordinate Index Notation). *For a point $a \in \mathbb{R}^d$, we use $a^{(i)}$ to represent the i -th coordinate of a . For a set or multiset $I = \{i_1, i_2, \dots, i_m\} \subseteq [d]$, we denote $a^{(I)}$ to be $(a^{(i_1)}, a^{(i_2)}, \dots, a^{(i_m)}) \in \mathbb{R}^m$.*

In this paper, we look into the approximate nearest neighbor problem, defined as follows.

► **Definition 3.3** (α -Approximate Nearest Neighbor Problem). *Given a set $C = \{c_1, c_2, \dots, c_n\}$ of n points in a metric space (X, dist) , build a data structure that given any query point $q \in X$, returns index $i \in [n]$, such that $\text{dist}(q, c_i) \leq \alpha \cdot \min_{j \in [n]} \text{dist}(q, c_j)$.*

When C only has 2 points, we use the shorthand *2-point comparator* to denote such an approximate nearest neighbor data structure.

► **Definition 3.4** (2-point Comparator). *Given a set $C = \{c_1, c_2\}$ of 2 points in a metric space (X, dist) and parameters $0 \leq \epsilon, \delta < 1$, we call a data structure as a 2-point comparator, if for any query point $q \in X$, with probability $1 - \delta$, it returns index $i \in \{1, 2\}$, such that $\text{dist}(q, c_i) \leq (1 + \epsilon) \cdot \min_{j \in \{1, 2\}} \text{dist}(q, c_j)$.*

We use the following Chernoff bounds in this paper.

► **Theorem 3.5** (Chernoff Bound). *Let X_1, X_2, \dots, X_n be independent random variables, and $X_i \in [0, c]$. Define $X = \sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[X]$. Then*

$$\mathbb{P}[X \geq (1 + \delta)\mu] \leq \exp(-\delta^2\mu/(3c)), \text{ for } 0 < \delta \leq 1$$

$$\mathbb{P}[X \geq (1 + \delta)\mu] \leq \exp(-\delta\mu/(3c)), \text{ for } \delta \geq 1$$

$$\mathbb{P}[X \leq (1 - \delta)\mu] \leq \exp(-\delta^2\mu/(2c)), \text{ for } \delta \geq 0.$$

4 Near-linear Space Data Structures for ℓ_1 and ℓ_2 Metrics

In this section, we design a data structure with near-linear space and query complexity for approximate nearest neighbor under ℓ_1 and ℓ_2 metrics. We prove the following theorem.

► **Theorem 4.1.** Consider a set C of n points $c_1, \dots, c_n \in \mathbb{R}^d$ equipped with ℓ_1 or ℓ_2 metric. Given $0 < \epsilon < 1/4$ and $0 < \delta < 1$, we can efficiently construct a randomized data structure such that for any query point $q \in \mathbb{R}^d$ the following conditions hold with probability $1 - \delta$. (i) The data structure reads $O(n \log(n)/(\text{poly}(\epsilon, \delta)))$ coordinates of q . (ii) It returns $i \in [n]$ where c_i is a $(1 + \epsilon)$ -approximate nearest neighbor of q in C . (iii) The data structure has a size of $O(n \log^2(n) \log(d)/(\text{poly}(\epsilon, \delta)))$ words.

The proof of Theorem 4.1 for the case of ℓ_1 -metric is presented in Section 4.1 with missing proofs deferred to Appendix B.1, where as the proof for the case of ℓ_2 -metric is in Appendix B.2. Both proofs follow the same structure although they differ in details.

4.1 Data Structure for ℓ_1 -metric with Near-linear Space and Query Time

The preprocessing and querying procedures of our data structure are described in Section 4.1. The algorithm receives a set C of n points and parameters $\epsilon, \delta \in (0, 1)$ as inputs for its preprocessing phase. During preprocessing, it iteratively samples coordinates from $[d]$ to form a multiset I .⁵ Let R denote the set C of n points restricted to I . An ℓ_1 dimension reduction is then applied to R to further reduce the space. In the query phase, the algorithm receives a query $q \in \mathbb{R}^d$. It probes q on all coordinates in I and applies the same ℓ_1 dimension reduction to the coordinate-selected query. The output, identifying which point in C is the nearest neighbor of q , is determined by which point, after coordinate selection and dimension reduction, is closest to the query.

■ **Algorithm 1** An approximate nearest neighbor data structure for n points in \mathbb{R}^d for ℓ_1 metric.

```

1 Preprocessing ( $C, \epsilon, \delta$ ) // Inputs:  $C = \{c_1, \dots, c_n\} \subseteq \mathbb{R}^d, \epsilon \in (0, \frac{1}{4}), \delta \in (0, 1)$ 
2   Let  $I \leftarrow \emptyset$  be a multiset and  $T \leftarrow O(\log(n/\delta)/(\epsilon^3 \delta^2))$ ;
3   for  $t \in [T]$  do
4     for  $b \in [d]$  do
5       Add  $b$  to  $I$  with probability  $p^{(b)} \triangleq \max_{(i,j) \in \binom{[n]}{2}} \frac{|c_i^{(b)} - c_j^{(b)}|}{\|c_i - c_j\|_1}$ ;
6   Let  $R = \{r_1, r_2, \dots, r_n\} \subseteq \mathbb{R}^I$ , where for  $i \in [n], b \in I$ , we have  $r_i^{(b)} = c_i^{(b)}/p^{(b)}$ ;
7   Let  $m \leftarrow O(\log(n/\delta)/(\epsilon^2 \delta^2))$ ;
8   Sample  $M \in \mathbb{R}^{[m] \times I}$ , where each entry of  $M$  follows the Cauchy distribution,
   whose density function is  $c(x) = \frac{1}{\pi(1+x^2)}$ . Notice that  $M(\cdot): \mathbb{R}^I \rightarrow \mathbb{R}^m$  is an
   oblivious linear mapping [21];
9   store  $I, M(R) = \{Mr_i | i \in [n]\}, \{p^{(b)} | b \in I\}, M$ 
10 Query ( $q$ ) // Inputs:  $q \in \mathbb{R}^d$ 
11   Query  $q^{(b)}, \forall b \in I$ ;
12   Let  $u \in \mathbb{R}^I$ , where  $\forall b \in I, u^{(b)} = q^{(b)}/p^{(b)}$ ;
13   Let  $F((x_1, \dots, x_m), (y_1, \dots, y_m)) := \text{median}(|x_1 - y_1|, \dots, |x_m - y_m|)$ ;
14   Let  $\hat{i} = \arg \min_{i \in [n]} F(Mr_i, Mu)$ ;
15   return  $\hat{i}$ 

```

⁵ Multiple coordinates may be added to I in the same iteration, and I may contain repeated coordinates.

► **Theorem 4.2.** *Consider a set C of n points $c_1, \dots, c_n \in \mathbb{R}^d$ equipped with ℓ_1 metric. Given $0 < \epsilon < 1/4$ and $0 < \delta < 1$, with probability $(1 - \delta)$, we can efficiently construct a randomized data structure (See Section 4.1) such that for any query point $q \in \mathbb{R}^d$ the following conditions hold with probability $1 - \delta$. (i) The data structure reads $O(n \log(n/\delta)/(\epsilon^3 \delta^2))$ coordinates q . (ii) It returns $i \in [d]$ where c_i is a $(1 + \epsilon)$ -approximate nearest neighbor of q in C . (iii) The data structure has a size of $O(n \log^2(n/\delta) \log(d)/(\epsilon^5 \delta^4))$ words.*

To prove Theorem 4.2, we consider the following four events and prove that each of them holds with probability $1 - \delta'$, where $\delta' = \delta/4$. We establish the correctness of the data structure by conditioning on the first three events. By conditioning on the last event, we demonstrate the space and query complexity.

1. The first event is that the distance between a query and its nearest neighbor is overestimated at most by the factor $4/\delta$.
2. The second event is that for a query point the estimated distance to its nearest neighbor is significantly smaller than the estimated distance to any center that is not an approximate nearest neighbor.
3. The third event is that the dimension reduction preserves the distances between the coordinate-selected points and the query.
4. The fourth event is that at most near-linear many coordinates are sampled in the preprocessing and accessed in the query phase.

Finally, by applying a union bound, we show that all four events hold with probability $1 - \delta$, which implies that the requirements of Theorem 4.1 are met. We show the proof in the following subsections in line with the structure described above.

4.1.1 Upper Bound for Estimate of Distance between Query and its Nearest Neighbor

We first show that the distance between a query point and its nearest neighbor is overestimated at most by a factor $1/\delta'$ with probability $1 - \delta'$.

▷ **Claim 4.3.** Let $q \in \mathbb{R}^n$ be a query and let $i^* \in [n]$ be the index of its nearest neighbor. It holds that $\frac{1}{T} \|r_{i^*} - u\|_1 \leq \frac{1}{\delta'} \|c_{i^*} - q\|_1$, with probability $1 - \delta'$.

4.1.2 Coordinate Selection Preserves Distance Ratio

In this section, we show that, the following holds: for any query point $q \in \mathbb{R}^n$, whose nearest neighbor is c_{i^*} , with probability $1 - 2\delta'$, for all $c_i \in C$ such that $\frac{\|c_i - q\|_1}{\|c_{i^*} - q\|_1} \geq 1 + \epsilon$, the estimated ratio is significantly larger than one, that is $\frac{\|r_i - u\|_1}{\|r_{i^*} - u\|_1} \geq 1 + \epsilon\delta'/4$. This statement is formalized in Lemma 4.8.

To prove the statement, we first partition the coordinates in $[d]$ into good and bad regions for each pair of points. In the good region are the coordinates on which the query is not too far away from one of the points. The remaining coordinates are in the bad region. The intuition is that, restricted to the coordinates in the good region, the distance between one of the points and the query is bounded. This implies that we can estimate these distances using a Chernoff bound. For the bad coordinates the intuition is that the distances from the query to the 2 points are so large that the ratio between these distances is close to one. Additionally these distances are not overestimated because of the previous event in Section 4.1.1.

► **Definition 4.4** (Good region and bad region). Consider two points $c_1, c_2 \in \mathbb{R}^d$, and a query point $q \in \mathbb{R}^d$. We define the good region $\text{Good}(c_1, c_2, q)$ w.r.t. (c_1, c_2, q) as

$$\{z \in [d] \mid \min\{c_1^{(z)}, c_2^{(z)}\} - \frac{8}{\epsilon\delta'} |c_1^{(z)} - c_2^{(z)}| \leq q^{(z)} \leq \max\{c_1^{(z)}, c_2^{(z)}\} + \frac{8}{\epsilon\delta'} |c_1^{(z)} - c_2^{(z)}|\}.$$

We define the bad region w.r.t. (c_1, c_2, q) as $\text{Bad}(c_1, c_2, q) = [d] \setminus \text{Good}(c_1, c_2, q)$. When there is no ambiguity, we use Good for $\text{Good}(c_1, c_2, q)$, and Bad for $\text{Bad}(c_1, c_2, q)$.

Additionally, we define the distance between points restricted to subsets of coordinates.

► **Definition 4.5** (Restricted distance). Given $x, y \in \mathbb{R}^d$, we define the distance between x and y restricted to region $J \subseteq [d]$ as $\text{dist}_J(x, y) = \sum_{z \in J} |x^{(z)} - y^{(z)}|$. Furthermore, for $u, v \in \mathbb{R}^I$, where I is a multiset, whose elements are from $[d]$, we define $\text{dist}_J(u, v) = \sum_{z \in I} \mathbb{1}_{z \in J} \cdot |x^{(z)} - y^{(z)}|$.

The following lemma shows that for two points, whose distances to the query differ significantly, then the majority of the distance between the two points is present in the good region. The intuition of the lemma is that if the bad region contributes a lot to the distance $\|b - a\|_1$, it implies that the bad region contributes much to $\|b - q\|_1$ and $\|a - q\|_1$, then the ratio between the two distances should be very close to 1.

► **Lemma 4.6.** Consider three points $a, b, q \in \mathbb{R}^d$. If $\frac{\|b - q\|_1}{\|a - q\|_1} \geq 1 + \epsilon$, then $\frac{\text{dist}_{\text{Good}(a, b, q)}(b, a)}{\|b - a\|_1} \geq \frac{13}{16}$.

The following lemma shows that for two points c_i, c_j and a query q such that q is significantly closer to c_j , we can estimate the distances, restricted to the good region, between one of the points and the query.

► **Lemma 4.7.** Let c_i, c_j where $i, j \in [n]$ be such that for the query point $q \in \mathbb{R}^d$ it holds that $\|c_i - q\|_1 \geq (1 + \epsilon)\|c_j - q\|_1$. Let Good be shorthand for $\text{Good}(c_i, c_j, q)$ and Bad for $\text{Bad}(c_i, c_j, q)$. The following statements are true:

1. $\text{dist}_{\text{Good}}(c_i, q) \geq (1 + \epsilon)\text{dist}_{\text{Good}}(c_j, q)$.
2. If $\text{dist}_{\text{Good}}(c_j, q) \geq \frac{1}{8}\text{dist}_{\text{Good}}(c_i, c_j)$, then with probability $1 - \delta'/n$,

$$\begin{aligned} \text{dist}_{\text{Good}}(r_i, u) &\approx \frac{\epsilon}{16} T \cdot \text{dist}_{\text{Good}}(c_i, q), \\ \text{dist}_{\text{Good}}(r_j, u) &\approx \frac{\epsilon}{16} T \cdot \text{dist}_{\text{Good}}(c_j, q). \end{aligned}$$

3. If $\text{dist}_{\text{Good}}(c_j, q) < \frac{1}{8}\text{dist}_{\text{Good}}(c_i, c_j)$, then $\frac{\text{dist}_{\text{Good}}(c_i, q)}{\text{dist}_{\text{Good}}(c_j, q)} > 7$. With probability $1 - \delta'/n$,

$$\begin{aligned} \text{dist}_{\text{Good}}(r_i, u) &\approx \frac{\epsilon}{16} T \cdot \text{dist}_{\text{Good}}(c_i, q), \\ \frac{\text{dist}_{\text{Good}}(r_i, u)}{\text{dist}_{\text{Good}}(r_j, u)} &\geq 3. \end{aligned}$$

In the following lemma we state that our bounds on the distance estimates are good enough. Concretely, for a point that is not an approximate nearest neighbor, the estimated distance between this point and the query is significantly larger than the estimated distance between the nearest neighbor and the query.

► **Lemma 4.8.** For any query point $q \in \mathbb{R}^d$, with probability $1 - 2\delta'$, it holds for all $i \in [n]$, if $\frac{\|c_i - q\|_1}{\|c_{i^*} - q\|_1} \geq 1 + \epsilon$, then $\frac{\|r_i - u\|_1}{\|r_{i^*} - u\|_1} \geq 1 + \epsilon\delta'/4$.

4.1.3 ℓ_1 Dimension Reduction Preserves the Distance Ratio

► **Theorem 4.9** (Theorem 1 in [21]). Consider 2 points $a, b \in \mathbb{R}^d$ and parameters $0 < \epsilon, \delta < 1$. We sample a random matrix $M \in \mathbb{R}^{m \times d}$, where $m = O(\log(1/\delta)/(\epsilon^2))$ and each entry of M is sampled from Cauchy distribution, whose density function is $c(x) = \frac{1}{\pi(1+x^2)}$. Then with probability $1 - \delta$, $F(Ma, Mb) \approx_\epsilon \|a - b\|_1$, where $F((x_1, \dots, x_m), (y_1, \dots, y_m)) := \text{median}(|x_1 - y_1|, \dots, |x_m - y_m|)$.

▷ **Claim 4.10.** Let $q \in \mathbb{R}^n$ be a query. Furthermore, let $i^* \in [n]$ be the index of its nearest neighbor. Section 4.1 outputs an index i such that

$$\frac{\|c_i - q\|_1}{\|c_{i^*} - q\|_1} \leq 1 + \epsilon$$

with probability $1 - 3\delta'$.

Proof. Applying Theorem 4.9 with $m = O(\log(n/\delta')/(\epsilon^2\delta'^2))$, given any $v \in R$, it holds with probability $1 - \delta'/(n+1)$, that $F(Mv, Mu) \in [1 - \epsilon\delta'/100, 1 + \epsilon\delta'/100]\|v - u\|_1$. By union bound, with probability $1 - \delta'$, it holds for all $v \in R$. By Lemma 4.8 it holds with probability $1 - 2\delta'$ that for all $i \in [n]$, if $\frac{\|c_i - q\|_1}{\|c_{i^*} - q\|_1} \geq 1 + \epsilon$, then $\frac{\|r_i - u\|_1}{\|r_{i^*} - u\|_1} \geq 1 + \epsilon\delta'/4$, thus $\frac{F(Mr_i, Mu)}{F(Mr_{i^*}, Mu)} > 1$ with probability $1 - 3\delta'$, i.e. the algorithm will not output i as the index of the approximate nearest neighbor. \triangleleft

4.1.4 Space and Query Complexity

Next, we argue the space and query complexity. The main result in this section is that for any set C of centers the sum of probabilities $\sum_{b \in [d]} p^{(b)}$ is upper bounded by the number of centers.

► **Lemma 4.11.** $1 \leq \sum_{b \in [d]} p^{(b)} \leq n$.

Proof. The lower bound for $\sum_{b \in [d]} p^{(b)}$ can be seen as follows: Fix arbitrary $i_1, j_1 \in [n]$ s.t.

$\|c_{i_1} - c_{j_1}\|_1 \neq 0$. We have, $\sum_{b \in [d]} p^{(b)} \geq \sum_{b \in [d]} \frac{|c_{i_1}^{(b)} - c_{j_1}^{(b)}|}{\|c_{i_1} - c_{j_1}\|_1} = 1$.

Next we prove the upper bound for $\sum_{b \in [d]} p^{(b)}$. We first introduce an operation that we call “collapse”. For real numbers $\tau_1 \leq \tau_2$,

$$\text{collapse}_{\tau_1, \tau_2}(x) := \begin{cases} x, & x \leq \tau_1, \\ \tau_1, & \tau_1 < x \leq \tau_2, \\ x - (\tau_2 - \tau_1), & x > \tau_2. \end{cases}$$

For $\tau_1 > \tau_2$, $\text{collapse}_{\tau_1, \tau_2}(x) := \text{collapse}_{\tau_2, \tau_1}(x)$.

We also slightly abuse the notation, and define for $a, b, c \in \mathbb{R}^d$,

$$\text{collapse}_{a,b}(c) \triangleq (\text{collapse}_{a_1, b_1}(c_1), \text{collapse}_{a_2, b_2}(c_2), \dots, \text{collapse}_{a_d, b_d}(c_d)) \in \mathbb{R}^d.$$

▷ **Claim 4.12.** Let $x, y, \tau_1, \tau_2 \in \mathbb{R}$, where $x \geq y$. We have (i) $x - |\tau_1 - \tau_2| \leq \text{collapse}_{\tau_1, \tau_2}(x) \leq x$ and (ii) $x - y - |\tau_1 - \tau_2| \leq \text{collapse}_{\tau_1, \tau_2}(x) - \text{collapse}_{\tau_1, \tau_2}(y) \leq x - y$.

Proof. Denote $x' = \text{collapse}_{\tau_1, \tau_2}(x)$ and $y' = \text{collapse}_{\tau_1, \tau_2}(y)$. Assume w.l.o.g. that $\tau_1 \leq \tau_2$, otherwise we switch them.

For the first argument, $\forall x \in \mathbb{R}, x - |\tau_1 - \tau_2| \leq \text{collapse}_{\tau_1, \tau_2}(x) \leq x$, this is true because of the definition of collapse.

For the second argument, $\forall x \geq y \in \mathbb{R}, x - y - |\tau_1 - \tau_2| \leq \text{collapse}_{\tau_1, \tau_2}(x) - \text{collapse}_{\tau_1, \tau_2}(y) \leq x - y$, consider

$$f(x) = x - \text{collapse}_{\tau_1, \tau_2}(x) = \begin{cases} 0, & x \leq \tau_1, \\ x - \tau_1, & \tau_1 \leq x \leq \tau_2, \\ \tau_2 - \tau_1, & x > \tau_2. \end{cases}$$

Notice that $f(x)$ is a non-decreasing function, thus $f(x) \geq f(y)$, which proves $x - y \geq x' - y'$. Notice that $0 \leq f(x) \leq \tau_2 - \tau_1$, thus $f(x) \leq f(y) + (\tau_2 - \tau_1)$, which proves $x' - y' \geq x - y - (\tau_2 - \tau_1)$. \triangleleft

\triangleright **Claim 4.13.** Let c_{i_0}, c_{j_0} be the closest pair of points in a set $C = \{c_1, c_2, \dots, c_n\}$ of points, i.e., $(i_0, j_0) = \arg \min_{(i, j): i \neq j} \|c_i - c_j\|_1$. Define $P_n(C)$ to be $\sum_{b \in [d]} \max_{i \neq j} \frac{|c_i^{(b)} - c_j^{(b)}|}{\|c_i - c_j\|_1}$. The set $C' = \{c'_i = \text{collapse}_{c_{i_0}, c_{j_0}}(c_i) \mid i \in [n]\}$ is such that (i) it contains $n - 1$ points and (ii) $P_n(C) \leq P_{n-1}(C') + 1$.

Proof. Note, by the definition of the collapse operation, that $c'_{i_0} = c'_{j_0}$. Hence, C' has at most $n - 1$ points, proving part (i) of the claim.

To prove (ii), define $p_b = \max_{c_i, c_j \in C, i \neq j} \frac{|c_i^{(b)} - c_j^{(b)}|}{\|c_i - c_j\|_1}$ and $p'_b = \max_{c'_i, c'_j \in C', i \neq j} \frac{|c'_i - c'_j|}{\|c'_i - c'_j\|_1}$. We prove that $p_b - p'_b \leq \frac{|c_{i_0}^{(b)} - c_{j_0}^{(b)}|}{\|c_{i_0} - c_{j_0}\|_1}, \forall b \in [d]$. The claim follows, since $P_k(C) - P_{k-1}(C') = \sum_b (p_b - p'_b) \leq \sum_b \frac{|c_{i_0}^{(b)} - c_{j_0}^{(b)}|}{\|c_{i_0} - c_{j_0}\|_1} = 1$.

Fix some b , choose $(u, v) = \arg \max_{(i, j): c_i^{(b)} \neq c_j^{(b)}} \frac{|c_i^{(b)} - c_j^{(b)}|}{\|c_i - c_j\|_1}$, i.e. $p_b = \frac{|c_u^{(b)} - c_v^{(b)}|}{\|c_u - c_v\|_1}$.

If $c_u^{(b)} \neq c_v^{(b)}$, $p_b' \geq \frac{|c'_u - c'_v|}{\|c'_u - c'_v\|_1} \geq \frac{|c_u^{(b)} - c_v^{(b)}| - |c_{i_0}^{(b)} - c_{j_0}^{(b)}|}{\|c_u - c_v\|_1}$. The second inequality follows from Claim 4.12. Thus $p_b - p'_b \leq \frac{|c_{i_0}^{(b)} - c_{j_0}^{(b)}|}{\|c_u - c_v\|_1} \leq \frac{|c_{i_0}^{(b)} - c_{j_0}^{(b)}|}{\|c_{i_0} - c_{j_0}\|_1}$. If $c_u^{(b)} = c_v^{(b)}$, it means $|c_u^{(b)} - c_v^{(b)}| \leq |c_{i_0}^{(b)} - c_{j_0}^{(b)}|$, thus $p_b - p'_b \leq p_b = \frac{|c_u^{(b)} - c_v^{(b)}|}{\|c_u - c_v\|_1} \leq \frac{|c_{i_0}^{(b)} - c_{j_0}^{(b)}|}{\|c_{i_0} - c_{j_0}\|_1}$. \triangleleft

Solving the recurrence given by Claim 4.13 with the boundary condition that for a set C'' of two points, $P_2(C'') = 1$, we get that $P_n(C) \leq n$. Since $P_n(C)$, for the set of n points, is by definition, equal to $\sum_{b \in [d]} p^{(b)}$, the statement of the lemma follows. \blacktriangleleft

\triangleright **Claim 4.14.** With probability $1 - \delta'$, we have $|I| \leq 2Tn$.

Proof. We define $X_{t,z}$ for $t \in [T], z \in [d]$ to be the indicator random variable for the event that we add z to I at iteration t . Note that $\Pr[X_{t,z} = 1] = p^{(z)}$. We can see that $|I| = \sum_{t \in [T], z \in [d]} X_{t,z}$. Hence, $\mathbb{E}[|I|] = \sum_{t \in [T], z \in [d]} p^{(z)} \leq T \cdot \sum_{b \in [d]} p^{(b)} \leq T \cdot n$, where the last inequality follows from Lemma 4.11. Similarly, we can see that $\mathbb{E}[|I|] \geq T$. Using Chernoff bound, we have $\Pr[|I| - \mathbb{E}[|I|] \geq \mathbb{E}[|I|]] \leq \exp(-\mathbb{E}[|I|]/3) \leq \exp(-T/3) \leq \delta/4$. Thus, with probability $1 - \delta/4$, we have $|I| \leq 2\mathbb{E}[|I|] \leq 2Tn$. \triangleleft

► Lemma 4.15. With prob. $1 - \delta'$, the space complexity of Section 4.1 is $O(\frac{n \log^2(n/\delta) \log(d)}{\epsilon^5 \delta^4})$ wordsize, and the number of coordinates that we query is $O(\frac{n \log(n/\delta)}{\epsilon^3 \delta^2})$.

Proof. By Claim 4.14, we have that the number of coordinates of q that we query is $|I| = O(Tn) = O(\frac{n \log(n/\delta)}{\epsilon^3 \delta^2})$. Next we analyze the space complexity of the data structure. Matrix M is of wordsize $O(m|I|)$, $M(R)$ is of wordsize $O(mn)$, I is of wordsize $O(|I| \log(d))$. Thus the space complexity of the our data strcuture is $O(m|I| + mn + |I| \log(d)) = O(\frac{n \log^2(n/\delta) \log(d)}{\epsilon^5 \delta^4})$ wordsize. \blacktriangleleft

Using union bound, the success probability of the data structure is at least $1 - 4\delta' = 1 - \delta$.

References

- 1 Miklós Ajtai, Vitaly Feldman, Avinatan Hassidim, and Jelani Nelson. Sorting and selection with imprecise comparisons. *ACM Trans. Algorithms*, 12(2):19:1–19:19, 2016. doi:10.1145/2701427.
- 2 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008. doi:10.1145/1327452.1327494.
- 3 Alexandr Andoni, Piotr Indyk, Huy L. Nguyen, and Ilya P. Razenshteyn. Beyond locality-sensitive hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA’14)*, pages 1018–1028. SIAM, 2014. doi:10.1137/1.9781611973402.76.
- 4 Alexandr Andoni, Huy L. Nguyen, Aleksandar Nikolov, Ilya P. Razenshteyn, and Erik Waingarten. Approximate near neighbors for general symmetric norms. In *(STOC’17)*, pages 902–913. ACM, 2017. doi:10.1145/3055399.3055418.
- 5 Alexandr Andoni and Ilya P. Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *(STOC’15)*, pages 793–801. ACM, 2015. doi:10.1145/2746539.2746553.
- 6 Sunil Arya and David M. Mount. Approximate range searching. *Comput. Geom.*, 17(3-4):135–152, 2000. doi:10.1016/S0925-7721(00)00022-5.
- 7 Jon Louis Bentley. Multidimensional divide-and-conquer. *Commun. ACM*, 23(4):214–229, 1980. doi:10.1145/358841.358850.
- 8 Alina Beygelzimer, Sham M. Kakade, and John Langford. Cover trees for nearest neighbor. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML’06)*, volume 148 of *ACM International Conference Proceeding Series*, pages 97–104. ACM, 2006. doi:10.1145/1143844.1143857.
- 9 Christos Boutsidis, Petros Drineas, and Michael W Mahoney. Unsupervised feature selection for the k-means clustering problem. In *Advances in Neural Information Processing Systems (NeurIPS’09)*, volume 22. Curran Associates, Inc., 2009.
- 10 Christos Boutsidis, Anastasios Zouzias, Michael W. Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *IEEE Trans. Inf. Theory*, 61(2):1045–1062, 2015. doi:10.1109/TIT.2014.2375327.
- 11 Moses Charikar and Lunjia Hu. Near-optimal explainable k-means for all dimensions. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, (SODA’22)*, pages 2580–2606. SIAM, 2022. doi:10.1137/1.9781611977073.101.
- 12 Bernard Chazelle. Filtering search: A new approach to query-answering. *SIAM J. Comput.*, 15(3):703–724, 1986. doi:10.1137/0215051.
- 13 Bernard Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17(3):427–462, 1988. doi:10.1137/0217026.
- 14 Bernard Chazelle, Ding Liu, and Avner Magen. Approximate range searching in higher dimension. *Comput. Geom.*, 39(1):24–29, 2008. doi:10.1016/J.COMGEO.2007.05.008.
- 15 Kenneth L. Clarkson. Nearest neighbor queries in metric spaces. *Discret. Comput. Geom.*, 22(1):63–93, 1999. doi:10.1007/PL00009449.
- 16 Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, (STOC’15)*, pages 163–172. ACM, 2015. doi:10.1145/2746539.2746569.
- 17 Guilherme Dias da Fonseca and David M. Mount. Approximate range searching: The absolute model. *Comput. Geom.*, 43(4):434–444, 2010. doi:10.1016/J.COMGEO.2008.09.009.
- 18 Hossein Esfandiari, Vahab S. Mirrokni, and Shyam Narayanan. Almost tight approximation algorithms for explainable clustering. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, (SODA’22)*, pages 2641–2663. SIAM, 2022. doi:10.1137/1.9781611977073.103.

- 19 Buddhima Gamblath, Xinrui Jia, Adam Polak, and Ola Svensson. Nearly-tight and oblivious algorithms for explainable clustering. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, (NeurIPS'21)*, pages 28929–28939, 2021. URL: <https://proceedings.neurips.cc/paper/2021/hash/f24ad6f72d6cc4cb51464f2b29ab69d3-Abstract.html>.
- 20 Anupam Gupta, Madhusudhan Reddy Pittu, Ola Svensson, and Rachel Yuan. The price of explainability for clustering. In *64th IEEE Annual Symposium on Foundations of Computer Science, (FOCS'23)*, pages 1131–1148. IEEE, 2023. doi:10.1109/FOCS57990.2023.00067.
- 21 Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006. doi:10.1145/1147954.1147955.
- 22 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC'98)*, pages 604–613. ACM, 1998. doi:10.1145/276698.276876.
- 23 Piotr Indyk and Tal Wagner. Approximate nearest neighbors in limited space. In *Proc. Conference On Learning Theory (COLT'18)*, volume 75, pages 2012–2036, 2018. URL: <http://proceedings.mlr.press/v75/indyk18a.html>.
- 24 William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemp. Math.*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984.
- 25 David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pages 741–750. ACM, 2002. doi:10.1145/509907.510013.
- 26 Robert Krauthgamer and James R. Lee. Navigating nets: simple algorithms for proximity search. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA'04)*, pages 798–807. SIAM, 2004. URL: <http://dl.acm.org/citation.cfm?id=982792.982913>.
- 27 Konstantin Makarychev and Liren Shan. Near-optimal algorithms for explainable k-medians and k-means. In *(ICML'21)*, volume 139 of *Proceedings of Machine Learning Research*, pages 7358–7367. PMLR, 2021. URL: <http://proceedings.mlr.press/v139/makarychev21a.html>.
- 28 Konstantin Makarychev and Liren Shan. Explainable k -means: don't be greedy, plant bigger trees! In *Proc. 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC'22)*, pages 1629–1642, 2022. doi:10.1145/3519935.3520056.
- 29 Konstantin Makarychev and Liren Shan. Random cuts are optimal for explainable k-medians. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, (NeurIPS'23)*, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/d3408794e41dd23e34634344d662f5e9-Abstract-Conference.html.
- 30 Edward M. McCreight. Priority search trees. *SIAM J. Comput.*, 14(2):257–276, 1985. doi:10.1137/0214021.
- 31 Michal Moshkovitz, Sanjoy Dasgupta, Cyrus Rashtchian, and Nave Frost. Explainable k-means and k-medians clustering. In *(ICML'20)*, volume 119 of *Proceedings of Machine Learning Research*, pages 7055–7065. PMLR, 2020. URL: <http://proceedings.mlr.press/v119/moshkovitz20a.html>.