K-NEAREST NEIGHBOR APPROXIMATION VIA THE FRIEND-OF-A-FRIEND PRINCIPLE

JACOB D. BARON R.W.R. DARLING

National Security Agency, Fort George G. Meade, MD 20755-6844, USA

Abstract. Suppose V is an n-element set where for each $x \in V$, the elements of $V \setminus \{x\}$ are ranked by their similarity to x. The K-nearest neighbor graph is a directed graph including an arc from each x to the K points of $V \setminus \{x\}$ most similar to x. Constructive approximation to this graph using far fewer than n^2 comparisons is important for the analysis of large high-dimensional data sets. *K-Nearest Neighbor Descent* is a parameter-free heuristic where a sequence of graph approximations is constructed, in which second neighbors in one approximation are proposed as neighbors in the next. We provide a rigorous justification for $O(n \log n)$ complexity of a similar algorithm, using range queries, when applied to a homogeneous Poisson process in suitable dimension, but show that the basic algorithm fails to achieve subquadratic complexity on sets whose similarity rankings arise from a "generic" linear order on the $\binom{n}{2}$ inter-point distances in a metric space.

Keywords: similarity search, nearest neighbor, ranking system, linear order, ordinal data, random graph, proximity graph, expander graph
MSC class: Primary: 90C35; Secondary: 06A07

I get by with a little help from my friends.

— John Lennon and Paul McCartney, 1967

1. K-Nearest Neighbor Approximation

1.1. **Motivation.** Approximation to the K-nearest neighbor graph for a data set is central to data science and machine learning. Typically K is small (say at most 50) but n is in the millions or more, rendering exhaustive search impractical. Muja and Lowe [20] write: "the most computationally expensive part of many computer vision and machine learning algorithms consists of finding nearest neighbor matches to high dimensional vectors that represent the training data." Examples include the popular DBSCAN clustering algorithm [9], and nearest neighbor classifiers described in Devroye et al [7].

1.2. **Metric space versus ranking approaches.** Metric spaces provide a natural context for similarity rankings. In a finite metric space (X, ρ) where for each $x \in X$ the $|X| - 1$ distances from x to the other points are distinct, ρ induces a similarity ranking for each point in an obvious way: x is more similar to y than to z iff $\rho(x, y) < \rho(x, z)$. However

Date: August 23, 2019.

the K-nearest neighbor concept does not require a metric space. It suffices to have an oracle which determines, for distinct points x, y, z, whether y is more similar to x than z is.[1] In other words, for each x, the oracle knows a ranking of the other points by their similarity to x. When two different metrics may be applied to the same set of points, and their induced similarity rankings coincide, any method that proposes different approximate K-nearest neighbor graphs is *non-intrinsic*.[2]

Given a finite subset S of $\mathbf{R}^d$ with the L_p metric, for $1 \leq p \leq \infty$, the balanced box-decomposition tree of Arya et al. [2], a variation of the classical *k-d tree* [7], constructs an approximate K-nearest neighbor graph in $O(dKn \log n)$ steps, with a data structure of $O(dn)$ size. Muja and Lowe [20] propose improved tree-based algorithms. By using random projections, Indyk & Motwani's [14] *locality sensitive hashing* has $O(Kn^{1+\rho} \log n)$ query time and requires $O(dn + n^{1+\rho})$ space, for a problem-dependent constant $\rho > 0$ described in Datar et al. [6], and requires tuning to the specific problem.

Houle and Nett [13] describe a rank-based similarity search algorithm called the *rank cover tree*, which they claim outperforms in practice both the balanced box-decomposition tree and locality sensitive hashing. Haghiri et al [12] propose another comparison-tree-based nearest neighbor search. Tschopp et al [22] present a randomized rank-based algorithm using the "combinatorial disorder" parameter of Goyal et al [11].

Here we analyze a heuristic called **K-nearest neighbor descent**, not cited in the works above, requiring no parameter choices except the integer K.

1.3. K-nearest neighbor descent.

Dong, Charikar, and Li [8] propose and implement K-nearest neighbor descent (NND) for approximation to the K-nearest neighbor graph, and report $O(n^{1.14})$ complexity in their experiments. In the metaphor of social networks, the principle is:

(FOF) *A friend of a friend could likely become a friend.*[3]

We call this the ***friend-of-a-friend principle* (FOF)**.

NND has an appealing simplicity and generality:

- No parameter choices.
- No precomputed data structures, such as trees or hash tables.
- It is intrinsic, not even requiring a metric.
- Simple, concise code, needing only addition of a similarity oracle for a new application.
- Initialization is fully random, so repeated NND runs give alternative approximations.

Perhaps for these reasons, NND was the method of choice in the UMAP dimensionality reduction algorithm [4, 19] of McInnes, Healy, et al.

The main drawback of NND has been its lack of any theoretical justification. The present work seeks to elucidate this issue. Various software efficiency optimizations render the algorithm as presented in [8] opaque to rigorous analysis. Instead we analyze non-optimized versions and variations which preserve the essential concept, which is FOF.

---

[1] For ordinal data analysis in general, see Kleindessner & von Luxburg [17].

[2] Compare with differential geometry, where an operation is called non-intrinsic if it depends on the coordinate system.

[3] Here "likely" should be read as "more likely than the average person," not that the probability is necessarily greater than 50%.

1.4. **Outline of the paper.** We begin by describing NND and several related K-nearest neighbor approximation algorithms that take advantage of FOF (Section 2). Next we give examples that show how these algorithms behave in different contexts, succeeding in some and failing in others (Section 3).

We develop the ranking framework in detail, and use it to demonstrate the failure of FOF-based algorithms to achieve sub-quadratic complexity in metric spaces which arise from generic *concordant ranking systems* (Section 4).

Finally we use random graph methods to prove that one FOF-based algorithm succeeds for a homogeneous Poisson process on a compact metric space with only $O(n \log n)$ work—better than the $O(n^{1.14})$ claimed in [8] (Section 5).

2. K-Nearest Neighbor Approximation Algorithms Exploiting the Friend-of-a-Friend Principle

Let us first formalize two notions we have used already.

Definition 2.1. *A* ranking system $\mathcal{S} = (S, (r_x)_{x \in S})$ *is a finite set* S *together with, for each* $x \in S$*, a ranking* $r_x : S \setminus \{x\} \to [|S| - 1]$ *of the other points of* S*. We say that* x prefers y *to* z *iff* $r_x(y) < r_x(z)$ *(for distinct* $x, y, z \in S$*). We typically abbreviate* $(r_x)_{x \in S}$ *by* $\mathbf{r}$*.*

Definition 2.2. *For a ranking system* $\mathcal{S} = (S, \mathbf{r})$ *and a positive integer* $K \leq |S|$*, the* K-nearest neighbor graph *for* $\mathcal{S}$ *is the directed graph on* S *that contains, for each distinct* $x, y \in S$*, an arc from* x *to* y *iff* $r_x(y) \leq K$*.*

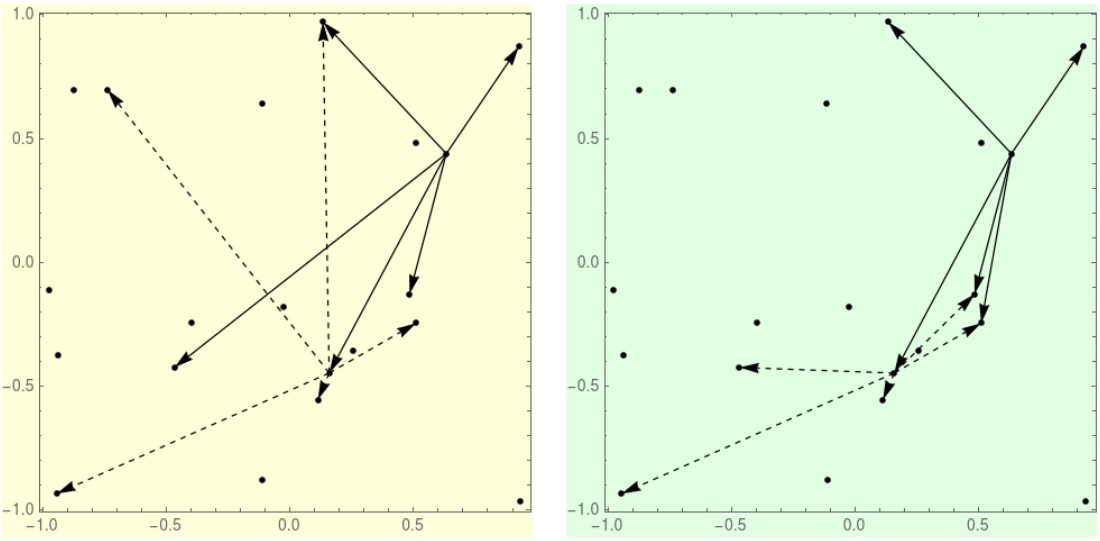
All the K-nearest neighbor approximation algorithms we will describe follow a common framework. They take as input a set S and an integer K. In the background is a vector $\mathbf{r}$ of rankings that makes S into a ranking system $\mathcal{S} = (S, \mathbf{r})$. We do not know $\mathbf{r}$ (if we did, we would be done), but we are allowed to query it, in the following sense. For any distinct $x, y, z \in S$, we may ask whether $r_x(y) < r_x(z)$ or vice-versa. We are never told the value of any $r_x(y)$.

We will always have in memory some directed graph on S which is our current best approximation to the true K-nearest neighbor graph for $\mathcal{S}$. Our initial approximation is a uniformly random K-out-regular digraph on S; that is, each point chooses K initial outneighbors uniformly, with the choices for different points independent. In successive iterations called *rounds*, we allow points to "share information" in some way that, we hope, leverages FOF.

At each round t, each point x will be the source of some arcs (typically K of them); we write $F_t(x) \subseteq S \setminus \{x\}$ for the set of targets of these arcs, which we call the ***friends*** of x. The points that view x as a friend are called the ***cofriends*** of x, denoted $C_t(x) := \{u \in S \setminus \{x\} : x \in F_t(u)\}$ (possibly $C_t(x) = \varnothing$). To leverage FOF, we perform some operation intended to "make introductions" between certain pairs of points which FOF suggests might "get along" (i.e. rank each other highly). This operation may consist of one or more *friend list requests* or *friend barters* (described next), or similar actions. As a result of these "introductions," one (or more) point(s) x updates her friend set if she has just "met" some new point y whom she prefers to some current friend z. After these updates, we move to round $t+1$ and repeat. When some condition is met, we stop and return our approximate K-nearest neighbor graph.

In a **friend list request**, a point x asks some other point y to provide x with $F_t(y)$, y's current list of friends. Typically $y \in F_t(x)$, which leads x to think y might have friends that x would like to meet. Then, via queries to r_x, x determines her K most-preferred elements of $F_t(x) \cup F_t(y) \setminus \{x\}$. Finally, x updates her friend set to consist of these K points.

FIGURE 1. **Friend barter:** *Out of twenty points in the left pane, we show two, each with five friends, indicated by dashed and solid arrows, respectively. The lower point is a friend of the upper point. The right pane shows their new friend sets after the friend barter. The lower gained two new friends (and discarded two), while the upper gained one (and discarded one).*



A **friend barter** is a pair of reciprocal friend list requests, shown in Figure 1. A point x makes a friend list request of another point y, and vice-versa, and each updates her respective friend set as appropriate. Typically $y \in F_t(x)$, but this does not mean $x \in F_t(y)$. Thus x is introduced to friends of a friend, while y is introduced to friends of a cofriend.

Within this framework of iterated "introductions" and updates, specifying an algorithm amounts to deciding which introductions occur in a given round and fixing a stopping condition. Shortly we will specify several algorithms in this way, but before doing so, it is worth reflecting on the initialization step. There is good reason to initialize randomly, quite apart from convenience.

2.1. **Consequences of random initialization: expander graph.** Let D be any K-out-regular directed graph on S, and let G be the corresponding undirected graph, with mean vertex degree about $2K$, obtained by ignoring the direction of arcs in D. Consider initializing a FOF-based algorithm with D. In any FOF-based algorithm, information travels, via introductions to friends of friends and/or friends of cofriends, by at most 1 step in G per round. Thus if $r_x(y) \leq K$ and y is at distance d from x in G, any FOF-based algorithm will need to run at least d rounds before knowledge of y reaches x.

The number of rounds needed for every point to have the possibility of learning about its true nearest neighbors is thus related to the *diameter* of G, i.e. the maximum over $x, y \in S$ of the length of the shortest path from x to y. We would like this diameter to be small.

On the other hand, since we initially have no knowledge of $\mathbf{r}$, we should not bias our algorithm by arranging to include, say, an $(n-1)$-leaf star in G. We would like to keep degrees in G roughly constant.

Taking D uniformly random strikes a good balance. Both diameter and degrees in G are then small. Each point has only a Binomial$(n-1, \frac{K}{n-1})$ number of cofriends. In Lemma A.1 of the Appendix, we will prove that with probability tending to 1 as $n \to \infty$, G is an *expander graph*, implying that its diameter is $O(\log n)$.

2.2. **Batchwise K-nearest neighbor descent.** A batchwise algorithm for K-nearest neighbor descent is proposed and implemented in [8]. Each round consists of a "batch" of friend barters, one for each arc of the present graph. At round t, each point x scans the friends of all the points in $F_t(x) \cup C_t(x)$ (roughly $2K^2$ points on average), and chooses her favorite K from among these friends-of-friends and friends-of-cofriends (along with $F_t(x)$, and distinct from x) as her new friend set $F_{t+1}(x)$. If x needs to scan m candidates, then x's update takes at most $m\lceil \log_2 K \rceil$ queries to r_x: x maintains a ranked list of her top K choices and for each candidate performs a binary search for the appropriate position for that candidate.

Stop when all friend sets have ceased to change.

Five well-studied data sets were used in [8] for experiment, with $n < 10^6$. The metrics were ℓ_1, ℓ_2, earth mover distance, and Jaccard and cosine similarity.

Each round of batchwise NND requires $O(nK^2 \log K)$ queries to $\mathbf{r}$, accounting for the scanning of a list of candidates at each point, where the sum of the sizes of the n lists is at most $2nK^2$. (Each arc xy contributes at most K elements to the lists at x and y.) Lemma A.1 suggests at least $O(\log n/(\log K))$ rounds. Together these suggest batchwise NND requires at least $O(nK^2 \log n)$ work.

2.3. **Randomized K-nearest neighbor descent.** A conceptually simpler NND goes as follows. At round t, perform a friend barter between a randomly chosen point x and a randomly chosen $y \in F_t(x)$. Stop after the (nK)th friend barter that yields no change, or perhaps after the bth for some other pre-chosen integer b. Or perhaps, instead, stop after some some pre-chosen number of rounds, say $\lceil nK(\log n)/(\log K) \rceil$. In any case, each round involves scanning only two lists, each of size at most K, taking $O(K \log K)$ work.

Alternatively, we could replace friend barters in the above description with friend list requests, for a factor of 2 speedup per round. If we believe it is better to meet friends of friends than friends of cofriends, then the speedup may be worth the possibly reduced spreading of information.

2.4. **Scheduled K-nearest neighbor descent.** Fix an arbitrary cyclic permutation π of S, and $x_1 \in S$. As in randomized NND, each round consists of a single friend barter between some point and a randomly chosen friend of that point. But instead of choosing the friend-barter-initiator randomly, use x_1 in round 1 and thereafter proceed in the order indicated by π. Work per round and choice of stopping condition are the same as for randomized NND, as is the option to replace friend barters by friend list requests.

2.5. **Pointwise-batchwise variants.** In each round, one point initiates a friend barter (alternatively, friend list request) with *each* of her friends, and she and each friend updates as appropriate. The initiator for successive rounds could be chosen either randomly or according to a fixed permutation. Choose a stopping condition as before. Work per round is $O(K^2 \log K)$.

2.6. **Parallel stream implementation of NND.** We indicate how to implement a version of NND in the Java `parallelStream()` paradigm. Let $G := (V, A)$ denote the state of the K-NN approximation at a given round, and let (V, E) be the undirected graph obtained by disregarding direction of arcs in A. Here are the steps to build the next approximation $G' := (V, A')$.

(1) Create a parallel stream of vertices in V.
(2) For each $v \in V$, iterate over pairs u, u' of neighbors of v in the undirected graph (V, E). There are about $4K^2$ such pairs on average.
(3) If u' is more similar to u than is u's K-th nearest neighbor in A, then arc $u \to u'$ is added to a list of *candidate arcs* for u. Likewise exchanging roles of u and u'.
(4) Collect all these candidate arcs, grouped by the source vertex.
(5) Stream a second time through vertices, in which for each v the best K out of arcs $v \to v' \in A$ and candidate arcs $v \to v''$ are collected to form the set A'.
(6) Clear the graph G from memory, and instantiate the directed graph $G' := (V, A')$.

2.7. **Second neighbor range query.** Suppose we run batchwise NND on a set S whose elements are distributed uniformly in a compact metric space (X, ρ). Intuitively, given the index t of the current round, we should be able to predict the rough distance r from a typical point to its least-preferred friend. The *second neighbor range query algorithm* (2NRQ) harnesses this intuition. The principle of 2NRQ, made precise in Section 5.2, may be summarized as:

Each vertex allows each pair of its neighbors to discover if they are close to each other.

The basic unit of work in 2NRQ is not a friend barter but a *range query*. A range query is similar to one pass through steps (1)–(6) of the parallel stream NND implementation above, with changes to the steps that determine membership in A'. In a range query, we fix $r > 0$ and we retain some but not all arcs satisfying $\rho(u, u') \le r$, according to a certain acceptance sampling policy. We discard arcs where $\rho(u, u') > r$. The out-degree of (S, A) is no longer exactly K; the idea is to choose r so that the expected out-degree is K. A single range query is illustrated in Figure 2.

The 2NRQ algorithm is not proposed for practical use; it is merely the closest algorithm to NND among those whose behavior we can analyze rigorously, at least for the case where the set S is a realization of a homogeneous Poisson process. The only case in which explicit parameters for 2NRQ will be derived is where the metric space is the d-dimensional torus with the ℓ_∞ norm.

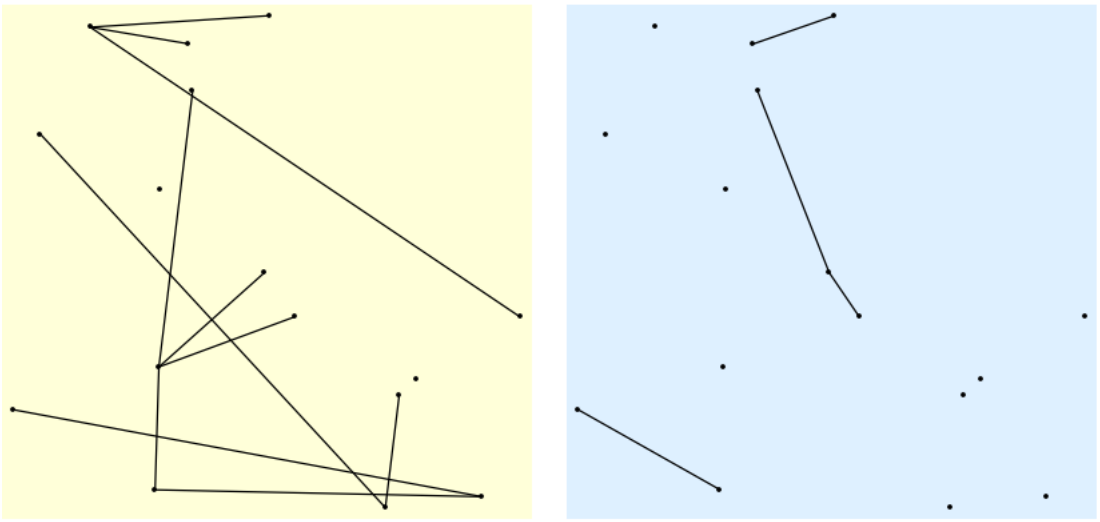
We defer a full description of 2NRQ to Section 5, but emphasize here its key differences with NND:

- 2NRQ requires a metric space, whereas NND does not.
- 2NRQ determines friends based on distance, whereas NND queries a ranking oracle.
- 2NRQ allows vertex degrees to vary, whereas in NND they are fixed at K.
- 2NRQ uses undirected edges, whereas NND uses directed arcs.
- 2NRQ is a theoretical tool for investigating complexity, whereas NND is proposed for practical deployment.

3. Examples

This section considers how FOF-based algorithms fare against various example ranking systems. We mostly restrict attention to ranking systems induced by distance in a metric space $(\mathcal{X}, \rho)$, as in Section 1.2. In each such example, the point set of the ranking system will

FIGURE 2. **Left:** *A random graph with 11 edges among 16 vertices embedded as i.i.d. uniform random points in* $[-1,1]^2$. **Right:** *The result of a* ***second neighbor range query*** *operation: distinct vertices with a common neighbor in the left graph become adjacent if their separation does not exceed* 0.75. *The 2NRQ update (10) is slightly more complicated, because not all proposed edges which meet the separation criterion are accepted.*



be called $X \subseteq \mathcal{X}$, and $|X| =: n$. For any triples $x, y, z \in X$ with $y \neq z$ but $\rho(x,y) = \rho(x,z)$, we can choose arbitrarily whether $r_x(y) < r_x(z)$ or vice-versa.

3.1. **Paris metric.** Let $\mathcal{X}$ be ℓ_1^n, meaning the n-dimensional real Banach space where

$$\|(a_1, \ldots, a_n)\| = \sum_1^n |a_i|.$$

Let e_i denote the i-th basis vector. Take X to consist of n points of the form

$$\{x_i := \eta_i e_i\}_{1 \leq i \leq n},$$

where $0 < \eta_1 < \cdots < \eta_n$ are real numbers. Then

$$\rho(x_i, x_j) = \|x_i - x_j\| = \eta_i + \eta_j, \quad i \neq j,$$

which has been called the *Paris metric*. For each $j > K$, the K-nearest neighbor set of x_j is the set

$$\{x_1, x_2, \ldots, x_K\}.$$

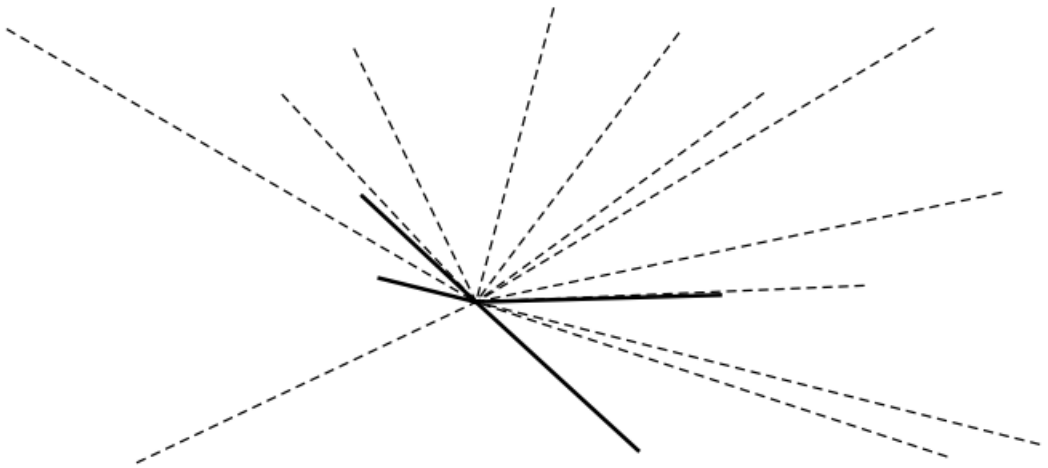
For $j \leq K$, it is

$$\{x_1, x_2, \ldots, x_{K+1}\} \setminus \{x_j\}.$$

It is natural to view x_i as a leaf node of an edge-weighted star, centered at $x_0 \notin X$, where edge $x_0 x_i$ carries weight η_i. The Paris metric is the path metric on this star; see Figure 3.

NND works well in this example because every pair of vertices are in perfect agreement about how they rank the other $n-2$ vertices. Thus x_i's information is always useful—or at least trustworthy—to x_j, and vice-versa. Each tells the other about the vertices she knows that are closest to x_0.

FIGURE 3. **Paris metric:** Let X be the leaf nodes of the star graph shown embedded in the plane. The distance between two leaves is the sum of their distances from the central node. Here $K = 4$, and 12 of the vertices (dashed arcs) have the same set of K nearest neighbors, which are the K nodes (solid arcs) closest to the center.



Suppose our goal is to discover the K-nearest neighbor graph exactly, within some specified number of rounds. Proposition A.1 suggests that using batchwise NND, if $K \geq 3$,

$$r^* := \lceil (\log n)/(\log (K-1)) \rceil$$

rounds suffice. Suppose $i \leq K$, and consider all paths of length r^* starting at x_i in the initial undirected graph. By Proposition A.1, with probability tending to 1 with n, every point is on one of these paths. As the algorithm progresses, every point along each of these paths will retain x_i in its friend set. Hence every point will have x_i in its friend set after r^* rounds.

3.2. **Random points on a circle.** Suppose $\mathcal{X}$ is the circle S^1, and the distance between two points is the angle subtended in radians. Let X consist of n points sampled independently according to Lebesgue measure on the circle. Section 5 provides rigorous analysis consistent with the following intuition: there is some constant $\gamma > 1$ such that each friend barter involving $x \in X$ reduces the expected angular span of its friend set by at least a factor γ. If this is true, then after m friend barters at x, elements of the friend set $F_t(x)$ typically lie within a span of $2\pi\gamma^{-m}$ radians. The K nearest neighbors of v typically lie within a span of $2\pi K/n$ radians. This implies that

$$m := \lceil (\log (n/K))/\log \gamma \rceil$$

friend barters should suffice at each vertex. This suggests that batchwise NND should yield a good approximate K-nearest neighbor graph after $O(\log n)$ rounds (here ignoring factors involving K and γ), for a total of $O(n \log n)$ work.

3.3. **Random ranking system.** In this example there is no metric. Given a set X of size n, simply choose a ranking system on X uniformly at random. Do this by choosing for each $x \in X$ a uniformly random ranking r_x of $X \setminus \{x\}$, with all choices independent.

There is no hope for decent performance from any FOF-based algorithm in this context, because FOF is simply not true here. Since r_x, r_y are independent for distinct x, y, a point

that y ranks highly is no more or less likely to be ranked highly by x than is any other point, regardless of how x and y rank each other.

Consider applying scheduled pointwise-batchwise NND (Section 2.5) using only friend list requests. (The analysis would be similar for any other NND variant, if a little less clean.) Consider the progress of the algorithm from the point of view of some fixed x. How long until x's friend set contains, say, $\lceil K/2 \rceil$ of her true best friends $B := \{y \in X \setminus \{x\} \ : \ r_x(y) \leq K\}$? There is an update at x only every nth round, and in these rounds x meets at most K^2 new points, probably fewer. Each new point z is equally likely to be ranked in any position by r_x. Thus we expect x to have to meet about $n/2$ points, over the course of at least $n/(2K^2)$ relevant rounds, before she finds $\lceil K/2 \rceil$ points of B. Running through $n/(2K^2)$ relevant rounds means running $n^2/(2K^2)$ rounds total, taking $O(n^2 \log K)$ work. Essentially, x is performing an inefficient version of exhaustive search during her rounds. Since x was arbitrary, all other points have the same experience.

It is fair to characterize this performance as worse than exhaustive search.

3.4. **Longest common substring.** Take a finite or countable alphabet $\mathcal{A}$, and a large positive integer m. Consider the metric space $(\mathcal{X}, \rho)$ consisting of strings $\mathbf{x} := x_1 x_2 \cdots x_m$ of length m, with characters $x_i \in \mathcal{A}$, under the metric based on the *longest common substring*. To be precise, the length $M(\mathbf{x}, \mathbf{x}')$ of the longest common substring between two strings is given by

$$M(\mathbf{x}, \mathbf{x}') := \max\{s : \exists\, i, j \text{ such that } x_{i+1} x_{i+2} \cdots x_{i+s} = x'_{j+1} x'_{j+2} \cdots x'_{j+s}\},$$

and the metric itself is defined by

$$\rho(\mathbf{x}, \mathbf{x}') := 1 - \frac{M(\mathbf{x}, \mathbf{x}')}{m}.$$

Symmetry $\rho(\mathbf{x}, \mathbf{x}') = \rho(\mathbf{x}', \mathbf{x})$ is immediate, as is $\rho(\mathbf{x}, \mathbf{x}) = 0$, while the triangle inequality is verified in a note of Bakkelund [3].

Let μ be a non-trivial probability measure on $\mathcal{A}$. Define

$$p := \sum_{a \in \mathcal{A}} \mu(\{a\})^2 \in (0, 1).$$

Let $X \subset \mathcal{X}$ consist of n strings of the form $\mathbf{x} := X_1 X_2 \cdots X_m$, where each X_i is sampled independently from $\mathcal{A}$ according to the distribution μ, with distinct $\mathbf{x}, \mathbf{y} \in X$ independent.

Given three random strings, ties such as $M(\mathbf{x}, \mathbf{y}) = M(\mathbf{x}, \mathbf{z}) = q$ will be fairly common. We break ties by ranking $r_{\mathbf{x}}(\mathbf{y}) < r_{\mathbf{x}}(\mathbf{z})$ if the product of probabilities of the characters in the longest substring that $\mathbf{y}$ shares with $\mathbf{x}$ is less than the corresponding product for $\mathbf{z}$ and $\mathbf{x}$. If there are still ties, break these arbitrarily.

Given an integer K much smaller than n, there is an integer q_K (also depending on m and μ) such

$$\Pr[M(\mathbf{x}, \mathbf{y}) \geq q_K + 1] < \frac{K}{n} \leq \Pr[M(\mathbf{x}, \mathbf{y}) \geq q_K].$$

The importance of q_K is that the K nearest neighbors of $\mathbf{x}$ in X are typically a subset of

$$R(\mathbf{x}) := \{\mathbf{y} \in X : M(\mathbf{x}, \mathbf{y}) \geq q_K\}.$$

Arratia, Gordon & Waterman [1, Theorem 2] provide an asymptotic estimate of the value of q_K. We will not state their theorem precisely, and we omit some intermediate calculations,

but the upshot is this. The K nearest neighbors of a specific string $\mathbf{x}$ will typically be strings whose longest common substring with $\mathbf{x}$ has length at least

$$q_K = \log_{1/p}(m^2(1-p)) + \log_{1/p}(n/K) = \frac{2\log(m) + \log((1-p)n/K)}{-\log p}.$$

On the other hand, replacing K by 1, the nearest neighbor of $\mathbf{x}$ will typically share a common substring with $\mathbf{x}$ of length q_1, where

$$q_1 = \frac{2\log(m) + \log((1-p)n)}{-\log p}.$$

Notice that $q_K \approx q_1 \ll m/K$. Thus for any distinct $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$, even if $\mathbf{y}$ and $\mathbf{z}$ are ranked highly by $\mathbf{x}$, the longest substring that $\mathbf{x}$ shares with $\mathbf{y}$ will typically be disjoint from that shared between $\mathbf{x}$ and $\mathbf{z}$. In other words, knowing that $\max(r_{\mathbf{x}}(\mathbf{y}), r_{\mathbf{x}}(\mathbf{z})) \leq K$ makes it virtually no more likely that $r_{\mathbf{y}}(\mathbf{z}) \leq K$—i.e. $\mathbf{x}$'s information is virtually irrelevant to $\mathbf{y}$ even if $\mathbf{y}$ is a good friend of $\mathbf{x}$. Likewise $\mathbf{y}$'s information to $\mathbf{x}$.[4]

This is precisely the problem we encountered for a random ranking system, if a bit less extreme. The friend of a friend principle fails, and FOF-based algorithms stand little chance.

Numerical example: In "big data" applications, q_K is in double digits, while m is in the thousands. For example, if $m = 2^{16}$, $n = 2^{33}$, $K = 2^5$, and $p = 2^{-4}$, then $q_K = 15$ and $q_1 = 16$. In other words, the 32 nearest neighbors of some string $\mathbf{x}$ will typically share common substrings with $\mathbf{x}$ of length 15 to 17. Moreover these shared substrings will be different for different neighbors, covering at most a proportion $Kq_K/m \approx 2^{-7}$ of the characters in $\mathbf{x}$.

3.5. **Powers of 2.** Take $\mathcal{X} = \mathbf{R}$ with the usual distance, and let X consist of the first n nonnegative powers of 2. We remain agnostic as to whether NND should work in this example. It seems to lie somewhere between the opposite extremes occupied by Paris and longest common substring. For distinct x, y with $r_x(y) \leq K$, there are on average about $K/2$ points z for which $\max(r_x(z), r_y(z)) \leq K$. The corresponding quantity for Paris is $K - 1 \approx K$; for longest common substring it is $K/n \approx 0$.

4. Failure of NND

Our goal in this section is to explore what "went wrong" for NND in the longest common substring example. We understand vaguely: the rankings $r_{\mathbf{x}}$ and $r_{\mathbf{y}}$ for distinct $\mathbf{x}, \mathbf{y}$ are sufficiently "uncorrelated" or "independent" that $\mathbf{y}$'s information is basically useless to $\mathbf{x}$ even if $\mathbf{y}$ is a friend or cofriend of $\mathbf{x}$. The same problem arose in the case of a random ranking system in Section 3.3. But while it is natural that this would happen for a random raking system, it is much more surprising in the context of a metric space, because intuitively, the triangle inequality should cause FOF to be helpful.

We now build a framework to formalize these ideas. With framework in hand, will prove the striking result that not only is a metric space insufficient to cause FOF to be helpful, but in fact in "almost all" metric spaces it fails to help.

4.1. **Metrizability and concordancy.** Our first step is to nail down just which ranking systems arise from a metric space.

[4]The situation is not symmetric: $\mathbf{y}$ is a friend of $\mathbf{x}$ while $\mathbf{x}$ is a cofriend of $\mathbf{y}$.

Definition 4.1. *A ranking system $\mathcal{S} = (S, \mathbf{r})$ is called* metrizable *if there is a metric ρ on S that induces the rankings $\mathbf{r} =: (r_x)_{x\in S}$, i.e. such that for each distinct $x, y, z \in S$, $\rho(x, y) < \rho(x, z)$ iff $r_x(y) < r_x(z)$.*

Goyal et al. [11] also formalize ranking systems, but omit the concept of concordancy, which we now define. For a set X and an integer t, let $\binom{X}{t}$ denote the set of t-element subsets of X. As in graph theory we frequently omit brackets for sets of size 2, e.g. $ab := \{a, b\}$.

Definition 4.2. *Let $\mathcal{S} = (S, \mathbf{r})$ be a ranking system. For $x \in S$, let $\preccurlyeq_x$ be the linear order on $S_x := \{xy \ : \ y \in S \setminus \{x\}\}$ defined by*

$$\{x, r_x^{-1}(1)\} \preccurlyeq_x \{x, r_x^{-1}(2)\} \preccurlyeq_x \cdots \preccurlyeq_x \{x, r_x^{-1}(|S| - 1)\}.$$

Formally[5] *$\preccurlyeq_x$ is a set of ordered pairs of elements of S_x; for example $\preccurlyeq_x$ contains the element $(\{x, r_x^{-1}(2)\}, \{x, r_x^{-1}(5)\})$ (because $2 \leq 5$). Let $R_{\mathcal{S}}$ be the union of these sets:*

$$R_{\mathcal{S}} \ := \ \bigcup_{x\in S} \preccurlyeq_x ;$$

this is a set of ordered pairs of elements of $\cup_{x\in S} S_x = \binom{S}{2}$. We call $\mathcal{S}$ concordant *if $R_{\mathcal{S}}$ is a subset of some partial order on $\binom{S}{2}$. In this case, the* order type *of $\mathcal{S}$, denoted $\preccurlyeq_{\mathcal{S}}$, is the unique minimal partial order that contains $R_{\mathcal{S}}$ (i.e. the intersection of all partial orders containing $R_{\mathcal{S}}$).*

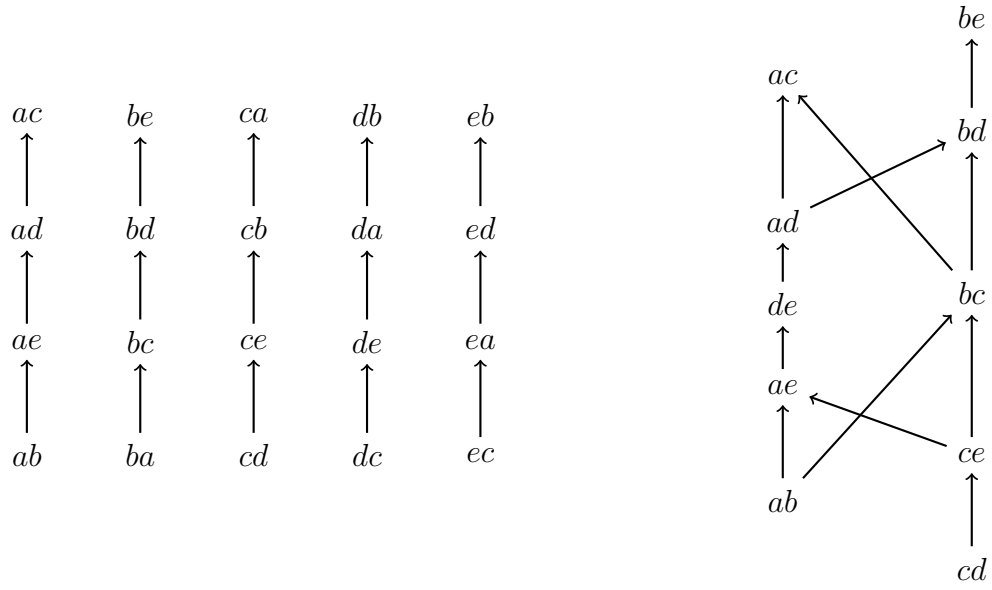
Henceforth we shorten "concordant ranking system" to **CRS**.

Example 4.3 (CRS)**.** *Let $\mathcal{S}$ be the ranking system on $S = \{a, b, c, d, e\}$ with rankings*

$$\begin{aligned}(1, 2, 3, 4) &= (r_a(b), r_a(e), r_a(d), r_a(c)) \\ &= (r_b(a), r_b(c), r_b(d), r_b(e)) \\ &= (r_c(d), r_c(e), r_c(b), r_c(a)) \\ &= (r_d(c), r_d(e), r_d(a), r_d(b)) \\ &= (r_e(c), r_e(a), r_e(d), r_e(b)).\end{aligned}$$

Diagrammed below left are, respectively, the linear orders $\preccurlyeq_a$ on S_a; $\preccurlyeq_b$ on S_b; $\preccurlyeq_c$ on S_c; $\preccurlyeq_d$ on S_d; $\preccurlyeq_e$ on S_e. (These are called Hasse diagrams*; an arrow appears iff the lower item is less than the upper item and there is no item in between.) Because there exists a partial order on $\binom{S}{2} = \cup_{x\in S} S_x$ that simultaneously extends all of these linear orders, $\mathcal{S}$ is concordant. The order type $\preccurlyeq_{\mathcal{S}}$ of $\mathcal{S}$ is the minimal such order, i.e. the one that contains no information beyond what is forced by $R_{\mathcal{S}}$. Its Hasse diagram is below right.*

[5]Recall that formally a partial order on X is a (reflexive, antisymmetric, transitive) binary relation on X, i.e. a subset of $X \times X$. For example on $\{1, 2, 3, 4\}$, the usual (linear) order $\leq$ *is* the set $\{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$. The statement "$x \leq y$" is nothing more than an abbreviation for "$(x, y) \in \leq$."



Example 4.4 (Non-concordant ranking system)**.** *The (unique up to isomorphism) smallest non-concordant ranking system $\mathcal{N}$ has $S = \{a, b, c\}$ and $(1, 2) = (r_a(b), r_a(c)) = (r_b(c), r_b(a)) = (r_c(a), r_c(b))$, and thus $ab \preccurlyeq_a ac$, $bc \preccurlyeq_b ba$, and $ca \preccurlyeq_c cb$. Any relation R containing $R_{\mathcal{N}}$ satisfies $ab\ R\ ac\ R\ bc\ R\ ab$, and thus cannot be both transitive and antisymmetric.*

"Almost no" ranking systems are concordant, in the sense that if a set S is made into a ranking system uniformly randomly as in Section 3.3, then the probability that it is concordant tends to 0 as $|S| \to \infty$. *Viz.,* we expect to see $\binom{|S|}{3}/4$ triples $\{x, y, z\} \subseteq S$ that are isomorphic to the $\mathcal{N}$ of Example 4.4, and even the lack of such a triple by no means implies concordance.

It turns out that the concordant ranking systems are precisely the metrizable ones.

Lemma 4.5. *For a ranking system $\mathcal{S} = (S, \mathbf{r})$, the following are equivalent:*

(A) *$\mathcal{S}$ is concordant;*
(B) *$\mathcal{S}$ is metrizable.*

Remark: Lemma 4.5 is an easier result than [16, Theorem 4] on *ordinal embedding*, for which a linear order on all $\binom{n}{2}$ inter-point distances is the input, and a Euclidean embedding is the output.

Proof. (B) $\Rightarrow$ (A). Given a metric ρ on S inducing $\mathbf{r}$, define a binary relation $\preccurlyeq$ on $\binom{S}{2}$ by $xy \preccurlyeq zw$ iff $\rho(x, y) \leq \rho(z, w)$ (well-defined by the symmetry of ρ). Evidently this is a partial order since it inherits reflexivity, antisymmetry and transitivity from the usual order on $[0, \infty)$, the codomain of ρ. Now for any $x \in S$ and $y, z \in S \setminus \{x\}$, we have

$$xy \preccurlyeq_x xz \quad \iff \quad r_x(y) \leq r_x(z) \quad \iff \quad \rho(x, y) \leq \rho(x, z) \quad \iff \quad xy \preccurlyeq xz,$$

using the fact that ρ induces $\mathbf{r}$ for the middle equivalence. Thus $\preccurlyeq$ extends each $\preccurlyeq_x$, so it extends $R_{\mathcal{S}}$.

(A) $\Rightarrow$ (B). Set $N := \binom{|S|}{2}$ and let ℓ_∞^N denote the N-dimensional real Banach space equipped with the sup (max) norm $\|\cdot\|_\infty$. Our goal is to exhibit an embedding $f : S \to \ell_\infty^N$ satisfying

(1) for each distinct $x, y, z \in S$, $\|f(x) - f(y)\|_\infty < \|f(x) - f(z)\|_\infty$ iff x prefers y to z.

Let $\preccurlyeq$ be any linear order on $\binom{S}{2}$ that extends $\preccurlyeq_{\mathcal{S}}$ (every partial order has a linear extension[6]). Define $\sigma : \binom{S}{2} \to [N]$ by

$$\sigma(xy) = \left|\left\{zw \in \tbinom{S}{2} \mid zw \preccurlyeq xy\right\}\right|$$

(σ enumerates the elements of $\binom{S}{2}$ from "bottom up" according to $\preccurlyeq$). Let $\preccurlyeq'$ be an arbitrary linear order on S (needed for technical convenience only). For $x, y, z \in S$ with $y \neq z$ and $y \preccurlyeq' z$, set the $\sigma(yz)$th coordiante of $f(x)$ as:

$$f(x)_{\sigma(yz)} = \begin{cases} 1 + \sigma(yz)/N & \text{if } x = y \\ -1 - \sigma(yz)/N & \text{if } x = z \\ 0 & \text{otherwise.} \end{cases}$$

(To see "what f does," it may be helpful to read Example 4.6 before continuing.)

We must check (1). Observe that for all $x \in S$ and $zw \in \binom{S}{2}$, we have $|f(x)_{\sigma(zw)}| \leq 2$. Moreover, for distinct $x, y \in S$, the only coordinate on which $f(x)$ and $f(y)$ are both nonzero is the $\sigma(xy)$th, and

$$|f(x)_{\sigma(xy)} - f(y)_{\sigma(xy)}| = 2 + 2\sigma(xy)/N,$$

which is strictly greater than 2. Thus $\|f(x) - f(y)\|_\infty = 2 + 2\sigma(xy)/N$. This completes the proof: $\preccurlyeq$ extends $\preccurlyeq_x$, so x prefers y to z iff $xy \preccurlyeq xz$ iff $\sigma(xy) < \sigma(xz)$ iff $\|f(x) - f(y)\|_\infty < \|f(x) - f(z)\|_\infty$. □

Example 4.6 (Embedding a CRS into ℓ_∞)**.** *Let $\mathcal{S}$ be as in Example 4.3, and suppose we want to embed S into ℓ_∞ as described in the proof of Lemma 4.5. We first pick a linear extension $\preccurlyeq$ of $\preccurlyeq_{\mathcal{S}}$, say the one shown in the top row of the following table. (The reader can verify by comparing to the Hasse diagram for $\preccurlyeq_{\mathcal{S}}$ that $\preccurlyeq$ is in fact such an extension.) Next we pick an arbitrary linear order $\preccurlyeq'$ on S; say $\preccurlyeq'$ orders S alphabetically. Then the function f constructed in the proof of Lemma 4.5 is given in the following table, i.e. the value at row x, column yz is $f(x)_{\sigma(yz)}$ (where a blank entry means 0).*

	ab	$\preccurlyeq cd$	$\preccurlyeq ce$	$\preccurlyeq ae$	$\preccurlyeq bc$	$\preccurlyeq de$	$\preccurlyeq ad$	$\preccurlyeq ac$	$\preccurlyeq bd$	$\preccurlyeq be$
a	1.1			1.4			1.7	1.8		
b	−1.1				1.5				1.9	2
c		1.2	1.3		−1.5			−1.8		
d		−1.2				1.6	−1.7		−1.9	
e			−1.3	−1.4		−1.6				−2

Note: each column xy has nonzero entries exactly at rows x, y; all column sums are 0; and the absolute value of every nonzero entry is in $(1, 2]$. All of this implies that the ℓ_∞-difference between rows x and y is realized uniquely at column xy. This, along with the fact that the absolute values of the nonzero entries increase from left to right (in $\preccurlyeq$ order), implies (1).

4.2. **Sampling from the CRSes.** We saw in Section 3.3 that FOF-based algorithms fail for generic ranking systems. We aim to show that FOF-based algorithms fail even for generic *concordant* ranking systems.

Given the concordancy restriction, the word "generic" presents a difficulty. In view of the comment after Example 4.4, a uniformly random ranking system will almost certainly fail to be concordant.

[6]Recall that a *linear extension* $\preccurlyeq_l$ of a partial order $\preccurlyeq_p$ is a linear order on the same ground set as $\preccurlyeq_p$ satisfying $x \preccurlyeq_l y$ whenever $x \preccurlyeq_p y$.

We know of no feasible method to sample uniformly from the CRSes on a set S. As there are $(|S|-1)!^{|S|}$ ranking systems, simply listing them and marking the concordant ones is impractical whenever $|S| > 6$ (observe $6!^7 \approx 10^{22}$).

Here is an alternative approach. Given a linear order $\preccurlyeq$ on $\binom{S}{2}$, define, for each distinct $x, y \in S$,

$$r_x(y) = \left|\{z \in S \setminus \{x\} \ : \ xz \preccurlyeq xy\}\right|. \tag{2}$$

Equivalently, define $\preccurlyeq_x$ as the restriction of $\preccurlyeq$ to S_x. This corresponds to reading row x in Example 4.6 and listing the column headings of its nonzero entries left to right.

Evidently the resulting ranking system $(S, (r_x)_{x\in S})$ is concordant, because there exists a partial order extending $\cup_{x\in S} \preccurlyeq_x$, namely $\preccurlyeq$.

Definition 4.7. *If $\preccurlyeq$ is a linear order on $\binom{S}{2}$ for some finite set S, let $\varphi(\preccurlyeq)$ denote the CRS on S defined by* (2)*:*

$$\{\text{linear orders on } \tbinom{S}{2}\} \quad \xrightarrow{\varphi} \quad \{\text{CRSes on } S\}. \tag{3}$$

Observe that the preimages under φ of a CRS $\mathcal{S}$ are precisely the linear extensions of $\preccurlyeq_{\mathcal{S}}$. (This is tautological: by construction $\preccurlyeq$ extends each $\preccurlyeq_x$ of $\varphi(\preccurlyeq)$, and by definition $\preccurlyeq_{\varphi(\preccurlyeq)}$ is the *minimal* partial order that does so, so $\preccurlyeq$ must extend $\preccurlyeq_{\varphi(\preccurlyeq)}$.) Since every partial order has a linear extension, φ is surjective. However it is not injective. For example the $\preccurlyeq_{\mathcal{S}}$ of Example 4.3 has many linear extensions, i.e. preimages. We explore this non-injectivity at length in Section 4.5.

Letting $\preccurlyeq$ be a linear order on $\binom{S}{2}$ chosen uniformly at random, we take $\varphi(\preccurlyeq)$ as our instantiation of a *generic* CRS on S. This corresponds to choosing a metric on S for which the inter-point distances are randomized.

4.3. **NND fails for a generic CRS.** As in Section 3.3 we analyze scheduled pointwise-batchwise NND with friend list requests (Section 2.5), choosing this variant only for convenience.

Proposition 4.8. *Let $\preccurlyeq$ be a uniformly random linear order on $\binom{S}{2}$. Running scheduled pointwise-batchwise NND on $\varphi(\preccurlyeq)$, for each $x \in S$ the expected number of rounds needed before x's friend set contains at least $\lceil K/2 \rceil$ elements of $B_x := \{y \in X \setminus \{x\} \ : \ r_x(y) \le K\}$ is at least about $n^2/(2K^2)$.*

Proposition 4.8 follows easily from the following elementary fact, whose proof we omit.

Lemma 4.9. *Let $\preccurlyeq$ be a uniformly chosen linear order on a finite set X, and let $A, B \subseteq X$ be disjoint. Then the random variables $\preccurlyeq|_A$ (that is, $\preccurlyeq$ restricted to A) and $\preccurlyeq|_B$ are independent.*

Proof of Proposition 4.8. Let $A = \binom{S}{2} \setminus S_x$ and $\mathcal{S}' = (S \setminus \{x\}, \mathbf{r}') = \varphi(\preccurlyeq|_A)$. Then Lemma 4.9 implies that r_x, that is x's ranking in $\varphi(\preccurlyeq)$, is independent of $\mathbf{r}'$, that is anyone else's ranking of any third party. In other words, the fact that y is at some time a friend of x does not make y's friend z any better for x than a uniformly selected vertex. The rest of the proof exactly follows Section 3.3: in each of x's friend list requests she discovers at most K^2 new points, which are from her perspective uniformly random. Thus x expects to have to meet about $n/2$ points, over $n/(2K^2)$ relevant rounds, to find $\lceil K/2 \rceil$ points of B_x. □

4.4. **CRS perspective on longest common substring.** Consider three distinct random strings $\mathbf{x}, \mathbf{y}, \mathbf{z}$ generated as in Section 3.4. Suppose $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are the longest common substrings between $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{y}$ and $\mathbf{z}$, and $\mathbf{z}$ and $\mathbf{x}$ respectively. The lengths of $\mathbf{a}, \mathbf{b}, \mathbf{c}$ will not exceed $q_1 \ll m$. Since $\mathbf{a}$ and $\mathbf{b}$ are so short, and likely to occur in different parts of $\mathbf{y}$, their existence does not help to supply a long $\mathbf{c}$. In other words, for $q = 1, 2, \ldots, q_1$

$$\Pr[M(\mathbf{x}, \mathbf{z}) \geq q \mid \min\{M(\mathbf{x}, \mathbf{y}), M(\mathbf{y}, \mathbf{z})\} \geq q] \approx \Pr[M(\mathbf{x}, \mathbf{z}) \geq q].$$

This is essentially the situation when we choose a generic CRS with random inter-point distances, which leads to quadratic run time for NND.

4.5. **Digression: properties of the map φ.** Here we begin to explore the many mysteries of the map φ. We believe this is fertile ground for future research.

Fix a set S of size n. Let L_n denote the set of $\binom{n}{2}!$ linear orders on $\binom{S}{2}$, and let R_n denote the set of CRSes on S. Our vantage point for insight into the map (3) $\varphi : L_n \to R_n$ is the following graphical model.

Consider the graph G on vertex set L_n, taking $\preccurlyeq$ and $\preccurlyeq'$ to be adjacent in G iff $\preccurlyeq'$ agrees with $\preccurlyeq$ after swapping the order of consecutive items xy and zw. Call G the **equivalent metrics graph**. Any linear order has $d := \binom{n}{2} - 1$ pairs of consecutive items, so G is d-regular. Any linear order can be rearranged into any other via some sequence of swaps of consecutive items, so G is connected.

Color the edges white or black as follows. An edge between $\preccurlyeq$ and $\preccurlyeq'$ is colored white if $\varphi(\preccurlyeq) = \varphi(\preccurlyeq')$, i.e. if these two linear orders map to the same ranking system, and colored black otherwise.

Proposition 4.10. *An edge of G joining $\preccurlyeq$ and $\preccurlyeq'$ is white iff the consecutive items xy, zw that are swapped between $\preccurlyeq$ and $\preccurlyeq'$ are disjoint.*

Proof. Simply observe that if $xy \cap zw = \varnothing$, then for any $u \neq v \in S$, the computation of $r_u(v)$ in (2) does not change when $\preccurlyeq$ is replaced by $\preccurlyeq'$. If on the other hand $x = w$ (say), then $r_x(y)$ and $r_x(z)$ swap when $\preccurlyeq$ is replaced by $\preccurlyeq'$. □

Proposition 4.11. *Let $\preccurlyeq$ and $\preccurlyeq'$ be linear orders on $\binom{S}{2}$. Then $\varphi(\preccurlyeq) = \varphi(\preccurlyeq')$ iff there is a path of white edges joining $\preccurlyeq$ and $\preccurlyeq'$ in G.*

Proof. One direction is immediate from Proposition 4.10. For the other, let $N = \binom{n}{2}$ and consider the "selection sort" algorithm to rearrange the list of pairs $L = (A_1 \preccurlyeq A_2 \preccurlyeq \cdots \preccurlyeq A_N)$ into the list $L' = (A'_1 \preccurlyeq' A'_2 \preccurlyeq' \cdots \preccurlyeq' A'_N)$:

- For each $i = 1, 2, \ldots, N$ do:
  - Find A'_i in L, say at position $j \geq i$
  - Move A'_i down to position i via $j - i$ swaps involving A'_i and the item just below.

We claim there are no "bad swaps" (swaps of nondisjoint items). To see this, observe that in the course of the algorithm, for each i, j, items A_i, A_j are swapped at most once. Thus if there were a bad swap, say with $A_i \cap A_j = \{x\}$, then we would have $\varphi(\preccurlyeq) \neq \varphi(\preccurlyeq')$ as witnessed by r_x. □

Corollary 4.12. *The subgraph of G consisting of the white edges, which we call the* **white graph***, decomposes into components, each of which consists of the inverse image under φ of a distinct CRS.*

The components of the white graph vary dramatically in size, from as small as 1 to as large as $(n/2)!^{n-1} \approx \exp(n^2 \log(n)/2)$ or larger, as the next examples show. To present these examples we adopt the following practical convention. For small n, we represent a linear order $\preccurlyeq$ as a matrix where rows are indexed by S and each column is the indicator of an element of $\binom{S}{2}$. We agree that the columns are arranged in $\preccurlyeq$-increasing order left to right (as in Example 4.6), and we leave blanks for zeros.

Example 4.13 (CRS with a unique φ-preimage)**.** *We return to the example of Section 3.5. Let $S = \{1, 2, 4, 8, 16, 32\}$ and let $\preccurlyeq$ order the pairs of points of S by their usual distance. If row i corresponds to 2^i, then $\preccurlyeq$ is represented by the matrix*

(4)
$$\begin{bmatrix}
1 & & 1 & & & 1 & & & & 1 & & & & & 1 \\
1 & 1 & & & 1 & & & & 1 & & & & & 1 & \\
 & 1 & 1 & 1 & & & & 1 & & & & & 1 & & \\
 & & & 1 & 1 & 1 & 1 & & & & & 1 & & & \\
 & & & & & & 1 & 1 & 1 & 1 & 1 & & & & \\
 & & & & & & & & & & 1 & 1 & 1 & 1 & 1
\end{bmatrix}.$$

In view of Proposition 4.10, since every pair of consecutive columns shares a nonzero row, every edge of G incident to $\preccurlyeq$ is black. Thus $\preccurlyeq$ is an isolated point of the white graph.

By examining the matrix (4) we can derive a lower bound on the number of isolated points of the white graph. Indeed, if we replace each dashed box in (4) by any permutation matrix of the same size, the resulting matrix represents another isolated point. This is because it retains from (4) the property that every pair of consecutive columns shares a nonzero row. Thus the white graph contains at least $\prod_{k=1}^{n-2} k! \approx \exp(n^2 \log(n)/2)$ isolated points.

For odd n, another source of isolated points in the white graph is Eulerian circuits[7] in the complete graph on S. Indeed, if $C = (e_1, e_2, \ldots, e_{\binom{n}{2}})$ is any such circuit, then the linear order $e_1 \preccurlyeq e_2 \preccurlyeq \cdots \preccurlyeq e_{\binom{n}{2}}$ is an isolated point. This is because since C is a walk, by definition $e_i \cap e_{i+1} \neq \varnothing$ for each i. McKay and Robinson [21] prove an asymptotic formula for the number of Eulerian circuits on K_n in which the dominant term is $\exp(n^2 \log(n)/2)$, roughly matching the number of isolated points arising from powers-of-2-type orders as above.

Example 4.14 (CRS with many φ-primages)**.** *For n even, let $(P_1, \ldots, P_{n-1})$ be a partition of $\binom{S}{2}$ into partitions of S.*[8] *Let $\preccurlyeq$ be any linear order on $\binom{S}{2}$ satisfying $xy \preccurlyeq zw$ whenever $xy \in P_i$, $zw \in P_j$, and $i < j$. Then $\varphi(\preccurlyeq)$ has at least $(n/2)!^{n-1}$ φ-preimages, because each P_i can be arbitrarily reordered (relative to $\preccurlyeq$) independently of the others, without changing the image under φ. The following matrix represents such a linear order $\preccurlyeq$ with $n = 6$.*

$$\begin{bmatrix}
1 & & & & 1 & & 1 & & 1 & & 1 & & \\
1 & & & 1 & & & & 1 & & 1 & & & 1 \\
 & 1 & & 1 & & & 1 & & 1 & & & 1 & \\
 & 1 & & & 1 & & & 1 & & & 1 & & 1 \\
 & & 1 & & 1 & & 1 & & & 1 & & 1 & \\
 & & 1 & & & 1 & & & 1 & & 1 & & 1
\end{bmatrix}$$

[7]Recall that an *Eulerian circuit* in a graph is a closed walk that traverses each edge exactly once.

[8]This is a partition of the complete graph on S into disjoint perfect matchings; by a theorem in elementary graph theory, such a partition exists for any even n.

The dashed vertical lines partition the columns into $n-1$ partitions of S, each of size $n/2$. The first three columns represent P_1, the next three P_2, etc. Notice that in this example even the consecutive columns that cross the boundaries of the partitions share no nonzero rows, meaning that every edge incident to $\preccurlyeq$ in the equivalent metrics graph is white.

We close this section with two easy calculations intended to give some (minimal) sense of the overall structure of the equivalent metrics graph.

Proposition 4.15. *The probability p that an edge of G selected uniformly at random is white is*

$$p := \frac{\binom{n-2}{2}}{\binom{n}{2} - 1} = 1 - \frac{4}{n+1}.$$

Proof. Pick a uniformly random edge by (a) picking a uniformly random vertex and then (b) picking a uniformly random edge incident to it. At step (b), out of d possible choices of a pair of consecutive elements xy, zw, on average $\binom{n-2}{2}$ of these pairs will be disjoint. □

Proposition 4.16 (Linear orders per CRS)**.** *Let H count the linear orders on $\binom{S}{2}$ which lie in the preimage under $\varphi : L_n \to R_n$ of a CRS on S chosen uniformly at random. Then*

$$\frac{\binom{n}{2}!}{((n-1)!)^n} \quad < \quad \mathbf{E}[H] = \frac{|L_n|}{|R_n|} \quad < \quad \frac{\binom{n}{2}!}{\prod_{k=1}^{n-2} k!}. \tag{5}$$

Remark: Applying Stirling's approximation for the factorials, the lower bound for $\log \mathbf{E}[H]$ has leading term $n^2(1-\log(2))/2 \approx 0.15n^2$, whereas the upper bound for the log has leading term $n^2 \log(n)/2$.

Proof. The equality in (5) follows from

$$|R_n|\mathbf{E}[H] = \sum_{r \in R_n} |\varphi^{-1}(r)| = |L_n|.$$

For the inequalities, let $\tilde{R}_n \supset R_n$ denote the set of all ranking systems (possibly non-concordant) on S, and let $R_n^1 \subset R_n$ denote the CRSes $r \in R_n$ for which $|\varphi^{-1}(r)| = 1$. It was proved above that

$$|R_n| > |R_n^1| \geq \prod_{k=1}^{n-2} k!.$$

It is immediate from the definitions that

$$|R_n| < |\tilde{R}_n| = ((n-1)!)^n.$$

Replacing $|R_n|$ in (5) by $|R_n^1|$ and $|\tilde{R}_n|$, the bounds follow. □

5. Second Neighbor Range Query on a Homogeneous Poisson Process

5.1. **Compact metric space with invariant measure.** Take a compact metric space $(\mathcal{X}, \rho)$, whose topology is second countable and Hausdorff, and which supports a positive finite Borel measure λ, with the invariance property that all balls of the same radius have the same measure. Without loss of generality, assume that the diameter $\sup\{\rho(x,y) : x, y \in \mathcal{X}\}$

is 1. Denote by $B_r(x)$ the closed ball of radius r, centered at $x \in \mathcal{X}$, and denote the ratio of the volume of such a ball to the volume of the space by

$$h(r) := \frac{\lambda(B_r(x))}{\lambda(\mathcal{X})}, \quad x \in \mathcal{X}. \tag{6}$$

Take $V \subset \mathcal{X}$ to be a Poisson process using the measure $n\lambda(\cdot)/\lambda(\mathcal{X})$, as defined in Kallenberg [15]. In particular, the size $|V|$ is a Poisson(n) random variable, and the number of points of V in any fixed ball of radius r is Poisson($nh(r)$).

5.2. **2NRQ graph process.** Construct a finite sequence $(V, E_t)_{0 \le t \le T}$ of undirected random graphs on V, with the aim of presenting (V, E_T) as an approximate K-nearest neighbor graph. Introduce a sequence of rates (θ_t), and a sequence of distances (r_t):

$$\frac{K}{n} = \theta_0 < \theta_1 < \cdots < \theta_T < 1; \quad 1 = r_0 > r_1 > \cdots > r_T > \frac{K}{n}. \tag{7}$$

These are coupled together using the identity

$$\theta_t = \frac{K}{nh(r_t)}. \tag{8}$$

Formulas for (r_t) will be given later. The graphs (E_t) will be constructed so as to exhibit a desired **sampling property**. For every $v \in V$:

The neighbors of v under E_t are a random sample at rate θ_t of the elements of

$$Q_t(v) := (V \setminus \{v\}) \cap B_{r_t}(v).$$

In particular, $\rho(v, v') > r_t$ implies $vv' \notin E_t$.

This property holds for $t = 0$ by construction, letting E_0 be uniformly sampled at rate K/n from the complete graph on V.

2NRQ graph update: We shall now describe how E_t is constructed from E_{t-1}, given r_t. Also refer to Figure 2. A vertex v'' of degree $d \ge 2$ in (V, E_{t-1}) has $\binom{d}{2}$ pairs of neighbors. Suppose (v, v') is such a pair. If $\rho(v, v') > r_t$, do nothing. If $\rho(v, v') \le r_t$, then edge vv' is added to E_t according to the success of a Bernoulli($f_t(v, v')$) trial, where the function f_t is chosen such that the neighbors of v under E_t are distributed uniformly on $Q_t(v)$, rather than biased towards points close to v. The formula is as follows. When $\rho(v, v') \le r$, denote by $\nu_r(v, v')$ the volume of the intersection $B_r(v) \cap B_r(v')$. Also define

$$g(r, s) := \min_{u,u' \in \mathcal{X}: \rho(u,u') \le s} \nu_r(u, u') > 0, \tag{9}$$

which is a positive function[9] when $0 < s < 2r$. Take the Bernoulli parameter

$$f_t(v, v') := \frac{g(r_t, r_{t-1})}{\nu_{r_{t-1}}(v, v')} \le 1. \tag{10}$$

The purpose of this acceptance sampling computation will be apparent in the proof of Proposition 5.2: whenever $\rho(v, v') \le r_t$, the pool of common neighbors v'' of both v and v' in E_{t-1} has a Poisson number of elements whose mean is

$$\frac{n\theta_{t-1}^2 \nu_{r_{t-1}}(v, v')}{\lambda(\mathcal{X})} \tag{11}$$

[9] We will compute $g(r, s)$ explicitly for the torus in (13).

which is reduced by acceptance sampling to

$$\frac{n\theta_{t-1}^2 g(r_t, r_{t-1})}{\lambda(\mathcal{X})} \tag{12}$$

regardless of the actual distance between v and v'. It is this non-dependence on distance which gives E_t the desired sampling property. After this update we shall compute θ_t as the mean rate at which elements of $Q_t(v)$ occur as neighbors of v under E_t.

Conclusion from final graph: Suppose this procedure can be shown to work in such a way that

$$\frac{\theta_t n \lambda(B_{r_t}(v))}{\lambda(\mathcal{X})} = K, \quad t = 0, 1, \ldots, T.$$

Then for any v, the neighbors of v in E_T, whose expected number is K, form a proportion θ_T of all the points of V within distance r_T. If θ_T is close enough to 1, the neighbors of v in E_T may be regarded as a satisfactory set of approximate nearest neighbors. A successful algorithm means that, at least for some K and d, a set (7) of parameters can be designed so that T is $O(\log n)$, giving $O(n \log n)$ total work. This we will do for the torus example in Theorem 5.3. If $O(n)$ steps were required, for example, the algorithm would appear to be useless.

5.3. **Metric embedding of V in d-dimensional torus.** Let T^d denote the d dimensional hypercube $[-1, 1]^d$ in which opposite faces are glued together to give the d-dimensional torus. The metric ρ on T^d is induced by the ℓ_∞^d norm:

$$\rho((x_1, \ldots, x_d), (y_1, \ldots, y_d)) := \max_i \{\min\{|x_i - y_i|, |x_i - y_i \pm 2|\}\}.$$

Henceforward arithmetic on $\mathbf{x} := (x_1, \ldots, x_d)$ will be interpreted so that, for each i, replacing x_i by $x_i \pm 2$ makes no difference. The compact metric space (T^d, ρ) has diameter 1. A closed ball centered at $\mathbf{x}$ with radius r is denoted $B_r(\mathbf{x}) \subset T^d$. When $r < 1$,

$$B_r(\mathbf{0}) := [-r, r]^d.$$

Let λ denote Lebesgue measure on the Borel sets of T^d, which has total measure 2^d. The volume ratio function (6) becomes

$$h(r) := \frac{(2r)^d}{2^d} = r^d, \quad r \le 1.$$

Our vertex set V will be a random subset of T^d given by a realization of a homogeneous Poisson point process associated with the measure $n2^{-d}\lambda$. Thus the cardinality $|V|$ is a Poisson(n) random variable.

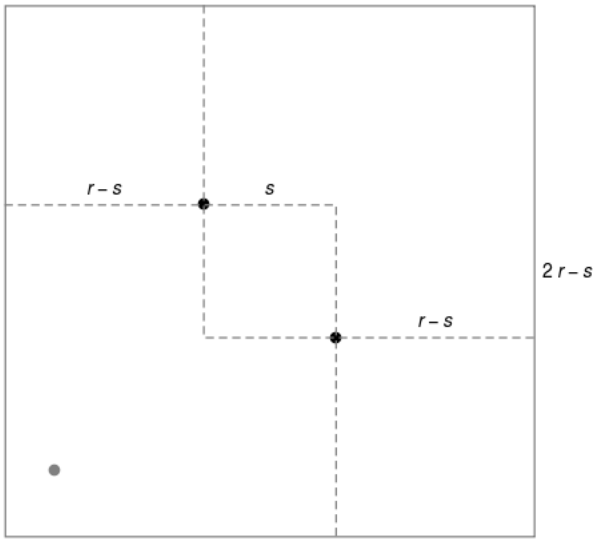
5.4. **Intersecting balls on the torus.** We shall not prove the following geometric fact, but instead ask the reader to inspect Figure 4.

Lemma 5.1. *Consider the metric space T^d with the ℓ_∞^d norm. Suppose*[10] *$0 < s < r < (1+s)/2$. When $\rho(v, v') \le r$, denote by $\nu_r(v, v')$ the volume of the intersection $B_r(v) \cap B_r(v')$, as in (9). Then*

$$g(s, r) := \min_{u, u' : \rho(u, u') \le s} \nu_r(u, u') = (2r - s)^d. \tag{13}$$

[10]The upper bound serves to avoid wrap-around effects on the torus.

FIGURE 4. *Suppose $0 < s < r$, and v, v' are two points in $\mathbf{R}^d$ with the ℓ_∞^d norm, separated by distance s. The volume of the set of points v'' within distance r of both v and v' is at least $(2r-s)^d$. The case $d = 2$ is illustrated.*



Remark: For another metric, the formula would change. For example, in the Euclidean metric on $\mathbf{R}^3$, integral calculus gives

$$\frac{\pi}{12}(4r+s)(2r-s)^2.$$

5.5. **General update formula for 2NRQ parameters.** We shall now establish the validity of the inductive step on which 2NRQ depends, and declare the update formulas for the parameters (7).

Proposition 5.2 (2NRQ parameter update)**.** *Suppose V is a homogeneous Poisson process, with mean cardinality n, on a compact separable metric space $(\mathcal{X}, \rho)$ as in Section 5.1. Suppose that, for some $t \geq 1$, and for each $v \in V$. the neighbors of v under E_{t-1} are a uniform random sample at rate θ_{t-1} of the elements of $V \setminus \{v\}$ in the ball of radius r_{t-1} centered at v, where θ_{t-1} and r_{t-1} are parameters connected by (8). If there exists $r_t < r_{t-1}$ satisfying the formula:*

$$\left(\frac{K}{nh(r_{t-1})}\right)^2 \frac{ng(r_t, r_{t-1})}{\lambda(\mathcal{X})} = -\log\left(1 - \frac{K}{nh(r_t)}\right), \tag{14}$$

use this new radius r_t to construct E_t according to the 2NRQ update, with acceptance sampling as described in (10). In that case, for each $v \in V$, the neighbors of v under E_t are a random sample at rate $\theta_t := K/(nh(r_t))$ of the elements of $V \setminus \{v\}$ in the ball of radius r_t centered at v.

Remark: We do not assert that for every K, n, and for every metric space, the equation (14) is solvable for r_t. See Theorem 5.3.

Proof. Assume that the random graph (V, E_{t-1}) has the properties described. In our probability calculations, we do not condition on any knowledge about the positions of points in V which might have been gathered in previous 2NRQ steps. Let r_t be as in (14). Suppose

v, v' are a pair of vertices with separation $\rho(v, v') \le r_t$. By the uniformity hypothesis, the number of vertices

$$v'' \in B_{r_{t-1}}(v) \cap B_{r_{t-1}}(v')$$

is a Poisson random variable with mean

$$\frac{n\nu_{r_{t-1}}(v, v')}{\lambda(\mathcal{X})}.$$

Any such vertex v'' has a probability of

$$\theta_{t-1}^2 = \left(\frac{K}{nh(r_{t-1})}\right)^2$$

of being adjacent to both v and v' under E_{t-1}. Thus the number of common neighbors of v and v' under E_{t-1} is Poisson with the mean given by (11), which is reduced by acceptance sampling to (12), which we shall abbreviate to μ. Note that the formula (12) does not depend on the value of $\rho(v, v')$, provided it is less than r_t. This is the reason why the neighbors of v under E_t are uniformly distributed in the ball of radius r_t about v.

The probability that no edge between v and v' exists in E_t is $e^{-\mu}$, which is the mass at zero of a Poisson(μ) random variable. Hence the new rate is

$$\theta_t := \frac{K}{nh(r_t)} = 1 - e^{-\mu}.$$

Rewrite as $\mu = -\log(1 - \theta_t)$. After the substitution (8), this is exactly the formula (14).

Thus E_t has the desired properties, and the relationship (8) between θ_t and r_t has been maintained. □

5.6. **2NRQ for uniform points on the torus completes in $O(\log n)$ steps.** Here is our positive result for 2NRQ on the d-dimensional torus. We believe that, with more effort, similar results could be obtained for the Euclidean metric on a bounded subset of $\mathbf{R}^d$, and for geodesic distance on a sphere.

Parameter estimates: Parameter computation for 2NRQ amounts to solving (14) for r_t in terms of r_{t-1}, for as many steps t as desired, to obtain the full parameter sequence (7), where $h(r) = r^d$ on the torus. Lemma 5.1 shows that, for the torus, (14) takes the simpler form

$$\left(\frac{K}{nr_{t-1}^d}\right)^2 \frac{n(2r_{t-1} - r_t)^d}{2^d} = -\log\left(1 - \frac{K}{nr_t^d}\right). \tag{15}$$

To prove Theorem 5.3, we shall bound the radii (r_t) between two geometric progressions, which allows us to show that the number of steps of 2NRQ is logarithmic in n. For this we prepare some parameter estimates.

The mean number K of neighbors must satisfy a lower bound $K > 2^d$ in terms of the dimension d. This allows us to set the main scaling parameter

$$\gamma := 1 - \sqrt{1 - \frac{2}{K^{1/d}}} \in (0, 1). \tag{16}$$

Select a nominal *success rate* $\alpha \in (0, 1)$ such that

$$\tau := \frac{\log(\alpha n/K)}{d \log(1/\gamma)} \tag{17}$$

is a positive integer, and also $\beta := -\alpha^{-1}\log(1-\alpha)$ satisfies $K/\beta > 2^d$. The integer $\tau \geq 1$ will be the number of steps of 2NRQ. This gives an implicit lower bound on the mean number n of points in terms of the parameters d, K, α:

$$n \geq \frac{K}{\alpha\gamma^d}. \tag{18}$$

A lower bound on the success rate will be $\alpha_* < \alpha$, given by

$$\frac{\log(\alpha_* n/K)}{\log(\alpha n/K)} := \frac{\log(1/\gamma_*)}{\log(1/\gamma)},$$

where $\gamma_* \in (\gamma, 1)$ depends on α and β:

$$\gamma_* := 1 - \sqrt{1 - 2(\beta/K)^{1/d}} \in (0,1).$$

An example of a valid set of parameters is:

n	K	d	τ	γ	γ_*	α	α_*
5,983,713	60	3	3	0.301	0.344	0.5	0.15

Observe the modest 15% lower bound on the success rate in the last column, albeit after a mere three rounds.

Theorem 5.3 (Homogeneous Poisson process on torus in d dimensions)**.** *Consider a homogeneous Poisson process associated with the measure $n2^{-d}\lambda$ on the d-dimensional torus, as described in Section 5.3. Assume $K > 2^d$, and set parameter γ as in (16). Assume α has been chosen so that $K/\beta > 2^d$ and $n \geq K/(\alpha\gamma^d)$, which allows us to run the 2NRQ algorithm for $\tau \geq 1$ steps as in (17). Then for each $v \in V$, the neighbors of v under E_τ are a random sample at rate α_* or more of the elements of $V \setminus \{v\}$ in the ball of radius r_τ centered at v, where $\gamma^\tau < r_\tau \leq \gamma_*^\tau$. The mean number of such neighbors is K.*

Remark: The important implication from (17) is that, as n increases, holding other parameters fixed, only $O(\log n)$ steps of 2NRQ are required before the neighbors of an arbitrary vertex v represent a uniform sample, at rate no less than α_*, of all vertices within a fixed radius of v. Also see Figure 5 and Corollary 5.4.

Proof. On the torus with ℓ_∞^d norm, we saw that $h(r) = r^d$. Lemma 5.1 shows that

$$g(r_t, r_{t-1}) = (2r_{t-1} - r_t)^d$$

which implies that distance update formula (14) takes the form (15) on the torus.

In order to apply Proposition 5.2, which will ensure validity of the 2NRQ algorithm, we must check the existence and uniqueness of the solution of (15) for r_t, in terms of r_{t-1}.

Given $\alpha \in (0,1)$, define $\beta := -\alpha^{-1}\log(1-\alpha) > 1$. Convexity of $x \mapsto -\log(1-x)$ guarantees that

$$x < -\log(1-x) \leq \beta x, \quad x \in (0, \alpha].$$

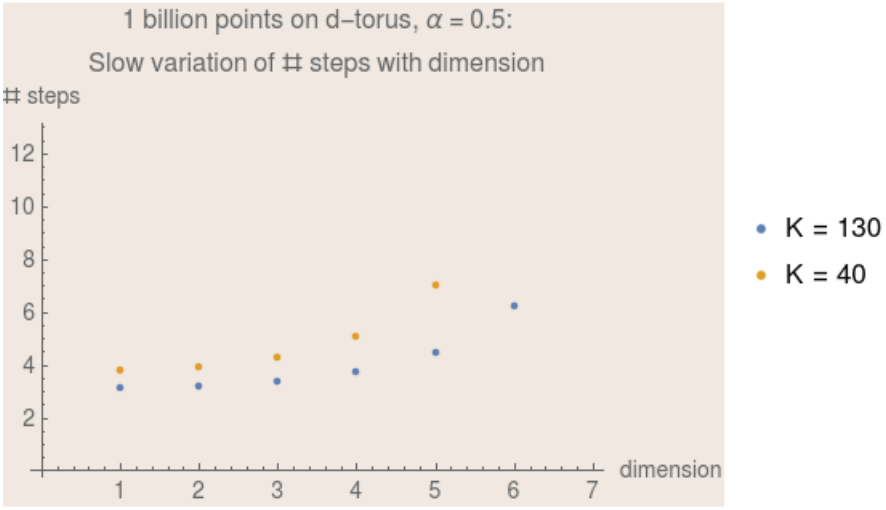
It follows that, so long as $\theta_t := K/(nr_t^d) \leq \alpha$,

$$\frac{K}{nr_t^d} < -\log\left(1 - \frac{K}{nr_t^d}\right) = \frac{K^2}{nr_{t-1}^d}\left(1 - \frac{r_t}{2r_{t-1}}\right)^d \leq \frac{\beta K}{nr_t^d}.$$

Cancel terms in the inequalities, and abbreviate $r_t/r_{t-1} < 1$ to y, to obtain:

$$1 < Ky^d(1 - y/2)^d \leq \beta.$$

FIGURE 5. *Number of steps of 2NRQ increases slowly as a function of dimension $d < \log_2 K$, according to (17), without rounding to an integer. Plots for $K = 60$ and $K = 130$ are shown, with $n = 10^9$ and $\alpha = 0.5$.*



Divide by K, take the $1/d$ power, subtract $1/2$, and complete the square:

$$\frac{1}{K^{1/d}} - \frac{1}{2} < -\frac{1}{2}(1-y)^2 \le \left(\frac{\beta}{K}\right)^{1/d} - \frac{1}{2}.$$

Double, change signs, and take the positive square root:

$$\sqrt{1 - 2(\beta/K)^{1/d}} \le 1 - y < \sqrt{1 - 2K^{-1/d}}.$$

In other words, there is a unique solution $y := r_t/r_{t-1} < 1$ to (14), and this solution satisfies:

$$\gamma < \frac{r_t}{r_{t-1}} \le \gamma_*.$$

Iterative multiplication, starting at $r_0 = 1$, shows that

$$\gamma^t < r_t \le \gamma_*^t, \quad t = 1, 2, \ldots,$$

so long as $\theta_t \le \alpha$, which we shall now check. By definition of θ_t, the previous pair of inequalities can be written

$$\frac{K}{n\gamma_*^{td}} \le \theta_t < \frac{K}{n\gamma^{td}} \le \frac{K}{n\gamma^{\tau d}}, \quad t = 1, 2, \ldots, \tau.$$

The term on the right is α, by the definition of τ in (17), so indeed it is true that

$$\theta_t \le \alpha, \quad t = 1, 2, \ldots, \tau.$$

Thus Proposition 5.2 applies, and the 2NRQ algorithm runs correctly for τ steps, at the end of which

$$\alpha_* := \frac{K}{n\gamma_*^{\tau d}} \le \theta_\tau < \alpha := \frac{K}{n\gamma^{\tau d}}$$

as claimed. □

Corollary 5.4 (2NRQ work estimate for Poisson process on torus in d dimensions)**.** *The mean number of distance evaluations to achieve a success rate at least α_* in the second neighbor range query, with a neighbor average of K, is bounded above by a quantity proportional to*

$$\frac{nK^2 \log(\alpha n/K)}{d\sqrt{1-2K^{-1/d}}}$$

in dimension $d < \log_2 K$, provided $n \geq K/(\alpha\gamma^d)$.

Remark: See Figure 5 for the dependence of the number of steps of 2NRQ on dimension. Each evaluation involves a distance computation which is $O(d)$ complexity, meaning that overall complexity, as a function of dimension, increases as $O((1-2K^{-1/d})^{-1/2})$, so long as $d < \log_2 K$. In other algorithms, such as tree partitioning [2], work increases linearly with d, or faster.

Proof. Use the inequality $-\log(1-x) > x$, for $x \in (0,1)$, to bound the denominator $d\log(1/\gamma)$ in (17). Hence the number of steps of the 2NRQ algorithm is

$$\tau := \frac{\log(\alpha n/K)}{d\log(1/\gamma)} < \frac{\log(\alpha n/K)}{d\sqrt{1-2K^{-1/d}}}$$

At each step, $O(\binom{K}{2})$ neighbor pairs must be checked, for each of n vertices. Hence the upper bound on the work. □

6. Conclusions and Future Work

We have seen that NND fails to achieve sub-quadratic complexity for generic concordant ranking systems. It would be interesting to know whether a simiar failure occurs for rank cover trees [13] and comparison trees [12].

The second neighbor range query, which exploits FOF but is not based on rankings, provably achieves $O(n\log n)$ complexity when applied to a class of homogeneous Poisson processes on compact separable metric spaces. The $O(\log n)$ diameter of the expander graph used to initialize NND suggests an $O(n\log n)$ lower bound on complexity.

Combinatorial disorder, introduced by Goyal et al. [11], defines approximate triangle inequalities on ranks. These authors define a disorder constant D such that, in the notation of Definition 2.1,

$$r_y(x) \leq D(r_z(x) + r_z(y)), \quad \forall x, y, z.$$

Possibly some condition on D, or a similar notion, would guarantee that NND finishes in $O(n\log n)$ work with a constant depending on D.

Can one say more about the structure of the equivalent metrics graph of Section 4.5? Further insight into its structure may shed light on how frequently NND can be expected to work well in practice. It is possible that our model of a generic CRS is biased in favor of those for which NND has quadratic complexity.

Acknowledgments: The authors thank Leland McInnes (TIMC) for drawing our attention to nearest neighbor descent, and Kenneth Berenhaut and Katherine More (WFU) for valuable discussions about data science based on rankings.

Appendix A. Bounding Diameter of the Initial Random Graph

Proposition A.1. *Suppose Γ_x is a uniform random K-subset of $S \setminus \{x\}$ for each $x \in S$, with $\{\Gamma_x, x \in S\}$ independent. For $|S| = n$, consider the directed graph $D = D_n^K$ on S with arcs $\bigcup_x \{(x, y), y \in \Gamma_x\}$, and the corresponding undirected graph $G = G_n^K$ on S with edges $\bigcup_x \{\{x, y\}, y \in \Gamma_x\}$. Let $B = B_n^K$ denote the diameter of G. For all $K \geq 3$ and $\epsilon > 0$,*

$$\lim_{n\to\infty} \Pr\left[B_n^K \leq (1+\epsilon)\frac{\log n}{\log(K-1)}\right] = 1.$$

Remark: Bollobás & de la Vega [5] prove that, for any $\epsilon > 0$, the diameter of a uniform K-regular random graph, with $K \geq 3$, almost surely does not exceed the smallest integer r such that

$$(K-1)^r \geq (2+\epsilon)Kn\log n.$$

Fernholz & Ramachandran [10, Theorem 5.1] estimate the diameter of graphs generated uniformly at random according to a given vertex degree sequence. When vertex degrees X_i are sampled independently with $X_i - K \sim \text{Binomial}(n-1, \frac{K}{n-1})$, [10, Theorem 5.1] implies a diameter estimate of $\log(n)/\log(2K) + o(\log n)$. This does not apply directly to our model, in which vertex degrees are negatively dependent, because they sum to exactly $2Kn$. Krivelevich [18, Lemma 8.2] shows that expander graphs in general have $O(\log n)$ diameter. We give below a proof from first principles.

A.1. **Proof of Proposition A.1.** Proposition A.1 is an immediate consequence of two estimates concerning D. For $X \subseteq S$, let $N(X) := \{y \in S \setminus X \,:\, \exists x \in X \text{ with } y \in \Gamma_x\}$.

Proposition A.2 (Vertex Expanders)**.** *With D as in Lemma A.1, for all $\varepsilon > 0$, there exists $\alpha > 0$ (depending also on K) such that the following holds with probability tending to 1 as $n \to \infty$. For all nonempty $X \subseteq S$ with $|X| < \alpha n/\log n$,*

$$|N(X)| > (K - 1 - \varepsilon)|X|. \tag{19}$$

Proof of Proposition A.2. We follow Vadhan [23, Theorem 4.4] with minor changes. For $l < \alpha n/\log n$ (with α to be defined below), let p_l be the probability that there is some $X \subseteq S$ of size exactly l that violates (19). We will show that $\sum_{l=1}^{\alpha n/\log n} p_l = o(1)$.

Fix l and $X = \{x_1, x_2, \ldots, x_l\} \subseteq S$, and imagine choosing one by one the elements of Γ_{x_1}, followed by the elements of Γ_{x_2}, ... followed by the elements of Γ_{x_l}. Call a choice *bad* if the vertex chosen is an element of X or is the same as some previously-chosen vertex. The probability that any particular choice is bad is at most $l(K+1)/n$, even conditioned on all prior choices. Failure of (19) for X requires at least $(1+\varepsilon)l$ bad choices, and thus has probability at most

$$\binom{Kl}{(1+\varepsilon)l}\left(\frac{l(K+1)}{n}\right)^{(1+\varepsilon)l}$$

(the binomial coefficient choosing which choices are bad). Therefore, summing over possible sets X (of size l), Stirling's approximation gives

$$\begin{aligned} p_l \leq \binom{n}{l}\binom{Kl}{(1+\varepsilon)l}\left(\frac{l(K+1)}{n}\right)^{(1+\varepsilon)l} &\leq \left(\frac{ne}{l}\right)^l \left(\frac{Kle}{(1+\varepsilon)l}\right)^{(1+\varepsilon)l}\left(\frac{l(K+1)}{n}\right)^{(1+\varepsilon)l} \\ &< \left(\frac{e^{2+\varepsilon}(K+1)^{2+2\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\left(\frac{l}{n}\right)^{\varepsilon}\right)^l. \end{aligned}$$

Now, fix $\alpha = [e^{2+\varepsilon}(K+1)^{2+2\varepsilon}]^{-1/\varepsilon}$ and (for convenience) $\gamma = (1+\varepsilon)^{-(1+\varepsilon)} < 1$. Then for all $l < \alpha n/\log n$, we have $p_l < \gamma^l/(\log n)^{l\varepsilon} \le \gamma^l/(\log n)^{\varepsilon}$, so

$$\sum_{l=1}^{\alpha n/\log n} p_l \;<\; (\log n)^{-\varepsilon} \sum_{l=1}^{\infty} \gamma^l \;=\; o(1). \qquad \square$$

Let $d_D(x,y)$ be the distance from x to y in D, and let $N^r(x) := \{y \in S \;:\; d_D(x,y) = r\}$ and $\overline{N}^r(x) := \cup_{i=0}^r N^i(x)$.

Proposition A.3 (Diameter Bound)**.** *With D as in Lemma A.1 and any $\varepsilon > 0$, the following holds with probability tending to 1 as $n \to \infty$. For all $x, y \in S$, letting $r = \lceil \log_{(K-1-\varepsilon)}(2n\log n)^{1/2} \rceil$, either*

$$\overline{N}^r(x) \cap \overline{N}^r(y) \neq \varnothing \tag{20}$$

or $\exists z \in N^r(x), w \in N^r(y)$ such that $w \in \Gamma_z$; in other words, the diameter of G does not exceed $2r+1$.

Proof of Proposition A.3. Fixing any $Y \subseteq S$ and $z \in S \setminus Y$,

$$\begin{aligned}\Pr(\Gamma_z \cap Y = \varnothing) &= \left(\frac{n-|Y|}{n-1}\right)\cdots\left(\frac{n-|Y|-K+1}{n-K}\right) < \left(\frac{n-|Y|+1}{n}\right)^K \\ &= \left(1 - \frac{|Y|-1}{n}\right)^K < \exp[-K(|Y|-1)/n].\end{aligned}$$

For any x, y, Proposition A.2 implies $|N^r(x)| > \sqrt{2n\log n}$, and likewise for y. Fix x, y, and condition on the event that $\overline{N}^r(x) \cap \overline{N}^r(y)$ is empty; the conditional probability that there exists no $z \in N^r(x), w \in N^r(y)$ for which $w \in \Gamma_z$ is at most $\exp[-K|N^r(x)||N^r(y)|/n] < \exp[-2K\log n] = n^{-2K}$. This bound is the same for all x, y. Summing over the n^2 choices of x, y yields a probability at most $n^{2-2K} = o(1)$ of the existence of a particular x, y for which the shortest connecting path exceeds length $2r+1$. $\square$

To complete the proof of Proposition A.1, the distance in G between any $x, y \in S$ satisfying one of the alternatives in Proposition A.3 is at most $2r+1$, with r as in Proposition A.3. Now

$$2r + 1 < (1 + o(1))\frac{\log n}{\log(K-1) + \log(1-\varepsilon)},$$

which we want to show to be at most

$$(1+\epsilon)\frac{\log n}{\log(K-1)},$$

for some fixed ϵ. Since Proposition A.3 holds for any ε, we can easily choose ε sufficiently small.

References

[1] Richard Arratia, Louis Gordon & Michael Waterman. An extreme value theory for sequence matching. Ann. Statist. Volume 14, Number 3 971-993, 1986

[2] S. Arya; D. M. Mount; N. S. Netanyahu; R. Silverman; A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM, Volume 45, pages 891–923, 1998

[3] D Bakkelund. An LCS-based string metric. Unpublished notes, University of Oslo. 2009

[4] Etienne Becht; Leland McInnes; John Healy; Charles-Antoine Dutertre; Immanuel W. H. Kwok; Lai Guan Ng; Florent Ginhoux; Evan W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. Nature Biotechnology volume 37, pages 38-44, 2019
[5] B. Bollobás & W. F. de la Vega. The diameter of random regular graphs. Combinatorica 2, 125-134, 1982
[6] Mayur Datar; Nicole Immorlica; Piotr Indyk; Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the twentieth annual symposium on computational geometry (SCG '04). ACM, New York, NY, USA, 253-262, 2004
[7] Luc Devroye; Laszlo Gyrfi; Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer Science & Business Media, 2013
[8] Dong, Wei; Moses, Charikar; Li, Kai. Efficient k-nearest neighbor graph construction for generic similarity measures. Proceedings of the 20th International Conference on World Wide Web, 577–586, 2011
[9] M. Ester; H.-P. Kriegel; J. Sander; X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. 2nd Int. Conf. Knowl. Discov. Data Mining, 226-231, 1996
[10] Daniel Fernholz & Vijaya Ramachandran. The diameter of sparse random graphs. Random Structures and Algorithms 31, 482-516, 2007
[11] N. Goyal; Y. Lifshits; H. Schutze. Disorder inequality: a combinatorial approach to nearest neighbor search. WSDM '08 Proc. Intern. Conf. Web Search Web Data Mining 25-32, 2008
[12] Siavash Haghiri; Debarghya Ghoshdastidar; Ulrike von Luxburg. Comparison based nearest neighbor search. arXiV: 1704.01460, 2017
[13] Michael E. Houle ; Michael Nett. Rank-based similarity search: reducing the dimensional dependence. IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 136-150, 2015
[14] P. Indyk & R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. Thirteenth *Symposium on Theory of Computing*, pages 604-613, 1998.
[15] Olav Kallenberg. *Foundations of Modern Probability*, 2nd ed. Springer, New York 2001.
[16] M. Kleindessner & U. von Luxburg. Uniqueness of ordinal embedding. JMLR: Workshop and Conference Proceedings vol 35:128, 2014
[17] M. Kleindessner & U. von Luxburg. Lens depth function and k-relative neighborhood graph: versatile tools for ordinal data analysis. Journal of Machine Learning Research 18, 1-52, 2017
[18] M Krivelevich. Expanders: How to find them, and what to find in them. Surveys in Combinatorics, ed. A. Lo, R. Mycroft, G. Perarnau, A. Treglown, Cambridge University Press 2019
[19] Leland McInnes; John Healy; James Melville. UMAP: uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426, 2018.
[20] Marius Muja; David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36, 2227–2240, 2014
[21] B. McKay and R. W. Robinson. Asymptotic enumeration of Eulerian circuits in the complete graph. Combinatorica: 10 (1995), no. 4, pages 367-377.
[22] D. Tschopp, S. Diggavi, P. Delgosha. Randomized Algorithms for Comparison-based Search. NIPS 2011
[23] Salil P. Vadhan. *Pseudorandomness*. Foundations and Trends in Theoretical Computer Science: Vol. 7 (2012): No. 1–3, pp 1–336.