Approximating Nearest Neighbor Distances

Michael B. Cohen[1], Brittany Terese Fasy[2], Gary L. Miller[3], Amir Nayyeri[4], Donald R. Sheehy[5](✉), and Ameya Velingker[3]

[1] Massachusetts Institute of Technology, Cambridge, USA
[2] Montana State University, Bozeman, USA
[3] Carnegie Mellon University, Pittsburgh, USA
[4] Oregon State University, Corvallis, USA
[5] University of Connecticut, Mansfield, USA
don.r.sheehy@gmail.com

Abstract. Several researchers proposed using non-Euclidean metrics on point sets in Euclidean space for clustering noisy data. Almost always, a distance function is desired that recognizes the closeness of the points in the same cluster, even if the Euclidean cluster diameter is large. Therefore, it is preferred to assign smaller costs to the paths that stay close to the input points.

In this paper, we consider a natural metric with this property, which we call the nearest neighbor metric. Given a point set P and a path γ, this metric is the integral of the distance to P along γ. We describe a $(3+\varepsilon)$-approximation algorithm and a more intricate $(1+\varepsilon)$-approximation algorithm to compute the nearest neighbor metric. Both approximation algorithms work in near-linear time. The former uses shortest paths on a sparse graph defined over the input points. The latter uses a sparse sample of the ambient space, to find good approximate geodesic paths.

1 Introduction

Many problems lie at the interface of computational geometry, machine learning, and data analysis, including, but not limited to: clustering, manifold learning, geometric inference, and nonlinear dimensionality reduction. Although the input to these problems is often a Euclidean point cloud, a different distance measure may be more *intrinsic* to the data, other than the metric inherited from the Euclidean space. In particular, we are interested in a distance that recognizes the closeness of two points in the same cluster, even if their Euclidean distance is large, and, conversely, recognizes a large distance between points in different clusters, even if the Euclidean distance is small. For example, in Figure 1, the distance between a and b must be larger than the distance between b and c.

There are at least two seemingly different approaches to define a non-Euclidean metric on a finite set of points in $\mathbb{R}^d$. The first approach is to form a graph metric on the point set. An example of a potential graph is the kth nearest neighbor graph, where an edge between two points exists if and only if they are

Partially supported by the NSF grant CCF-1065106.

DOI: 10.1007/978-3-319-21840-3_17

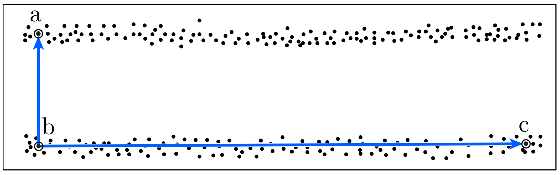

Fig. 1. The intrinsic density-based distance should recognize two points within the same cluster as cloesr than two points in different clusters, regardless of the actual Euclidean distance

both in the k nearest neighbor set of the other. In this graph, the edge weights may be a constant or the Euclidean distances. In this paper, we consider the complete graph, where the edge lengths are a power of their Euclidean lengths. We are particularly interested in the squared length, which we refer to as the edge-squared metric.

The second approach is to endow all of $\mathbb{R}^d$ with a new metric. We start with a cost function $c : \mathbb{R}^d \to \mathbb{R}$ that takes the point cloud into account. Then, the length of a path $\gamma : [0, 1] \to \mathbb{R}^d$ is the integral of the cost function along the path.

$$\ell_c(\gamma) = \int_\gamma c(s)ds = \int_0^1 c(\gamma(t)) \left| \frac{d\gamma}{dt}(t) \right| dt. \tag{1}$$

The distance between two points $x, y \in \mathbb{R}^d$ is then the length of the shortest path between them:

$$\mathbf{d}_{\mathrm{c}}(x, y) = \inf_\gamma \ell_c(\gamma), \tag{2}$$

where the infimum is over paths that start at x and end at y. Note that the constant function, $c(x) = 1$ for all $x \in \mathbb{R}^d$, gives the Euclidean metric; whereas, other functions allow space to be stretched in various ways.

In order to reinforce paths within clusters, one would like to assign smaller lengths to paths that stay close to the point cloud. Therefore, the simplest natural cost function on $\mathbb{R}^d$ is the distance to the point cloud. More precisely, given a finite point set P the cost $c(x)$ for $x \in \mathbb{R}^d$ is chosen to be $\mathrm{N}(x)$, the Euclidean distance from x to $\mathrm{NN}(x)$, where $\mathrm{NN}(x)$ denotes the nearest point to x in P. The *nearest neighbor length* (N-length) $\ell_{\mathrm{N}}(\gamma)$ of a curve is given by (1), where we set $c(x) = \mathrm{N}(x)$ for all points $x \in \mathcal{C}$. We refer to the corresponding metric given by (2) as the *nearest neighbor metric* or simply the N-distance.

In this paper, we investigate approximation algorithms for N-distance computation. We describe a $(3 + \varepsilon)$-approximation algorithm and a $(1 + \varepsilon)$-approximation algorithm. The former comes from comparing the nearest neighbor metric with the edge-squared metric. The latter is a tighter approximation that samples the ambient space to find good approximate geodesics.

1.1 Overview

In Section 3, we describe a constant factor approximation algorithm obtained via an elegant reduction into the edge-squared metric introduced by [BRS11] and [VB03]. This metric is defined between pairs of points in P by considering the graph distance on a complete weighted graph, where the weight of each edge is the square of its Euclidean length. We show that the N-distance and edge-squared metric are equivalent up to a factor of three (after a scaling by a factor of four). As a result, because spanners for the edge-squared metric can be computed in nearly linear time [LSV06], we obtain a $(3+\varepsilon)$-approximation algorithm for computing N-distance.

Theorem 1. *Let P be a set of points in $\mathbb{R}^d$, and let $x, y \in P$. The nearest neighbor distance between x and y can be approximated within a $(3+\varepsilon)$ factor in $O(n \log n + n\varepsilon^{-d})$ time, for any $0 < \varepsilon \leq 1$.*

In Section 4, we describe a $(1+\varepsilon)$-approximation algorithm for the N-distance that works in time $\varepsilon^{-O(d)} n \log n$. Our algorithm computes a discretization of the space for points that are sufficiently far from P. Nevertheless, the sub-paths that are close to P are computed exactly. We can adapt our algorithm to work for any Lipschitz cost function that is bounded away from zero; thus, the algorithm can be applied to many different scenarios.

Theorem 2. *For any finite set of points $P \subset \mathbb{R}^d$ and any fixed number $0 < \varepsilon < 1$, the shortest* N*-distance between any pair of points of the space can be $(1+\varepsilon)$-approximated in time $O(\varepsilon^{-O(d)} n \log n)$.*

1.2 Related Work

Computing the distance between a pair of points with respect to a cost function encompasses several significant problems that have been considered by different research communities for at least a few centuries. As early as 1696, Johann Bernoulli introduced the *brachistochrone* curve, the shortest path in the presence of gravity, as "an honest, challenging problem, whose possible solution will bestow fame and remain as a lasting monument" [Ber96]. With six solutions to his problem published just one year after it was posed, this event marked the birth of the field of *calculus of variations*.

Rowe and Ross [RR90] as well as Kime and Hespanha [KH03] consider the problem of computing anisotropic shortest paths on a terrain. An anisotropic path cost takes into account the (possibly weighted) length of the path and the direction of travel. Note that this problem can be translated into the problem of computing a shortest path between two compact subspaces of $\mathbb{R}^6$ under a certain cost function

When c is a piecewise constant function, the problem is known as the weighted region problem [MP91]. Mitchell and Papadimitriou [MP91] gave a linear-time approximation scheme in the plane and list the problem for more general cost functions as an open problem (See Section 10, problem number

(3)). Further work on this problem has led to fast approximations for shortest paths on terrains [AMS05].

Similar distances have been used in semi-supervised machine learning under the name density-based distance (DBDs) [SO05]. The goal here is to place points that can be connected through dense regions in the same cluster. Several approaches [VB03, BCH04] have been suggested that first estimate the density and then discretize space in a similar manner to that of Tsitsiklis [Tsi95], however, they do not provide any analysis on the complexity of the discretized space. Another approach is to search for shortest paths among a sample [BRS11] and this approach was shown to give good approximations to sufficiently long paths [HDI14]. The nearest neighbor metric can be viewed as a special case of density-based distance when the underlying density is the nearest neighbor density estimator.

2 Preliminaries

2.1 Metrics

In this paper, we consider three metrics. Each metric is defined by a length function on a set of paths between two points of the space. The distance between two points is the length of the shortest path between them.

Euclidean metric. This is the most natural metric defined by the Euclidean length. We use $\ell(\gamma)$ to denote the Euclidean length of a curve γ; $\ell(\gamma)$ can also be defined by setting $c(x) = 1$ for all $x \in \mathbb{R}^d$ in (1). We use $\mathbf{d}(x, y)$ to denote the distance between two points $x, y \in \mathbb{R}^d$ based on the Euclidean metric.

Nearest neighbor metric. As mentioned above, the nearest neighbor length of a curve with respect to a set of points P, is defined by setting $c(\cdot)$ to be $\mathrm{N}(\cdot)$ in (1). The nearest neighbor length of a curve γ is denoted by $\ell_{\mathrm{N}}(\gamma)$, and the distance between two points $x, y \in \mathbb{R}^d$ with respect to the nearest neighbor metric is denoted by $\mathbf{d}_{\mathrm{N}}(x, y)$.

Edge-squared metric. Finally, the edge-squared metric is defined as the shortest path metric on a complete graph on a point set P, where the length of each edge is its Euclidean length squared. The length of a path γ in this graph is naturally the total length of its edges and it is denoted by $\ell_{\mathrm{sq}}(\gamma)$. The edge-squared distance between two points $x, y \in P$ is the length of the shortest path and is denoted by $\mathbf{d}_{\mathrm{sq}}(x, y)$.

2.2 Voronoi Diagrams and Delaunay Triangulations

Let P be a finite set of points, called *sites*, in $\mathbb{R}^d$, for some $d \geq 1$. The Delaunay triangulation $\mathrm{Del}(P)$ is a decomposition of the convex closure of P into simplices such that for each simplex $\sigma \in \mathrm{Del}(P)$, the Delaunay empty circle property is satisfied; that is, there exists a sphere C such that the vertices of σ are on the boundary of C and $\text{int}(C) \cap P$ is empty. The Voronoi diagram, denoted $\text{Vor}(P)$, is the dual to $\text{Del}(P)$. We define the in-ball of a Voronoi cell with site p to be the maximal ball centered at p that is contained in the cell. The inradius of a Voronoi cell is the radius of its in-ball. We refer the reader to [DBVKOS00] for more details.

3 *N*-Distance Versus Edge-Squared Distance

In this section, we show that the nearest neighbor distance of two points $x, y \in P$ can be approximated within a factor of three by looking at their edge-squared distance. More precisely, $\mathbf{d}_{\text{sq}}(x, y)/4 \geq \mathbf{d}_{\text{N}}(x, y) \geq \mathbf{d}_{\text{sq}}(x, y)/12$ (see Lemma 1 and Lemma 3).

As a consequence, a constant factor approximation of the N-distance can be obtained via computing shortest paths on a weighted graph, in nearly-quadratic time. This approximation algorithm becomes more efficient, if the shortest paths are computed on a Euclidean spanner of the points, which is computable in nearly linear time [Hp11]. A result of Lukovszki et al. (Theorem 16(ii) of [LSV06]) confirms that a $(1+\varepsilon)$-Euclidean spanner is a $(1+\varepsilon)^2$-spanner for the edge squared metric. Therefore, we obtain Theorem 1.

3.1 The Upper Bound

We show that the edge-squared distance between any pair of points $x, y \in P$ (with respect to the point set P) is always larger than four times the N-distance between x and y (with respect to P). To this end, we consider any shortest path with respect to the edge-squared measure and observe that its N-length is an upper bound on the N-distance between its endpoints.

Lemma 1. *Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let $\mathbf{d}_{\text{N}}$ and $\mathbf{d}_{\text{sq}}$ be the associated nearest neighbor and edge-squared distances, respectively. Then, for any distinct points $x, y \in P$, we have that $\mathbf{d}_{\text{N}}(x, y) \leq \frac{1}{4}\mathbf{d}_{\text{sq}}(x, y)$.*

3.2 The Lower Bound

Next, we show that the edge-squared distance between any pair of points from P cannot be larger than twelve times their N-distance. To this end, we break a shortest path of the N-distance into segments in a certain manner, and shadow the endpoints of each segment into their closest point of P to obtain a short edge-squared path. The following definition formalizes our method of discretizing paths.

Definition 1. *Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let $x, y \in P$. Let $\gamma : [0, 1] \to \mathbb{R}^d$ be an (x, y)-path that is internally disjoint from P. A sequence $0 < t_0 \leq t_1 \leq \cdots \leq t_k < 1$ is a* proper breaking sequence *of γ if it has the following properties:*

1. *The nearest neighbors of $\gamma(t_0)$ and $\gamma(t_k)$ in P are x and y, respectively.*
2. *For all $1 \leq i \leq k$, we have $\ell(\gamma[t_{i-1}, t_i]) = \frac{1}{2}(\mathrm{N}(\gamma(t_{i-1})) + \mathrm{N}(\gamma(t_i)))$*

The following lemma guarantees the existence of breaking sequences.

Lemma 2. *Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let $x, y \in P$. Let γ be a path from x to y that is internally disjoint from P. There exists a proper breaking sequence of γ.*

Given a path γ that realizes the nearest neighbor distance between two points x and y, in the proof of the following lemma we show how to obtain another (x, y)-path with bounded edge-squared length. The proof heavily relies on the idea of breaking sequences.

Lemma 3. *Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let $\mathbf{d}_\mathrm{N}$ and $\mathbf{d}_\mathrm{sq}$ be the associated nearest neighbor and edge-squared distances, respectively. Then, for any distinct points $x, y \in P$, $\mathbf{d}_\mathrm{N}(x, y) \geq \frac{1}{12}\mathbf{d}_\mathrm{sq}(x, y)$.*

4 A $(1 + \varepsilon)$-Approximation of the N-Metric

In this section, we describe a polynomial time approximation scheme to compute the N-distance between a pair of points from a finite set $P \subset \mathbb{R}^d$. The running time of our algorithm is $\varepsilon^{-O(d)} n \log n$ for n points in d-dimensional space. We start with Section 4.1, which describes an exact algorithm for the simple case in which P consists of just one site. Section 4.2 describes how to obtain a piecewise linear path using infinitely many Steiner points, the technical details of which may be found in the full version [CFM+15]. Section 4.3 combines ideas from 4.2 and 4.1 to cut down the required Steiner points to a finite number. Finally, Section 4.4 describes how to generate the necessary Steiner points.

4.1 Nearest Neighbor Distance with One Site

We describe a method for computing $\mathbf{d}_\mathrm{N}$ for the special case that P is a single point using complex analysis. This case will be important since distances will go to zero at an input point and thus we must be more careful at input points. Far from input points, we use a piecewise constant approximation for the nearest neighbor function, and near input points, we use exact distances. More than likely this case has been solved by others since the solution is so elegant. We refer the interested reader to [Str] for more general methods to solve similar problems in the field of calculus of variations.

Suppose we want to compute $\mathbf{d}_\mathrm{N}(x, y)$ where $P = \{(0, 0)\}$. Writing $(x, y) \in \mathbb{C}$ in polar coordinates as $z = re^{i\theta}$, we define the *quadratic transformation* $f\colon \mathbb{C} \to \mathcal{R}$ by

$$f(z) = z^2/2 = (r^2/2)e^{i2\theta},$$

where $\mathcal{R}$ is the two-fold Riemann surface; see Figure 2. The important point here is that the image is a double covering of $\mathbb{C}$. For example, the points 1 and -1

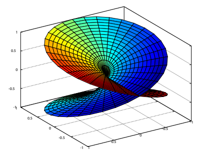

Fig. 2. To make the complex function one-to-one, one needs to extend the complex plane to the two-fold cover called the two-fold Riemann Surface

are mapped to different copies of 1/2. Therefore, on the Riemann surface, the distance between 1 and -1 is one and the shortest path goes through the origin. More generally, given any two nonzero points p and q on the surface, the minimum angle between them (measured with respect to the origin) will be between 0 and 2π. Moreover, if this angle is $\geq \pi$, then the shortest path between them will consist of the two straight lines $[p, 0]$ and $[0, q]$. Otherwise, the line $[p, q]$ will be a line on the surface and, thus, the geodesic from p to q.

Let $\mathbf{d}_{\mathcal{R}}$ denote the distance on the Riemann surface. We next show that for a single point, the nearest neighbor geodesic is identical to the geodesic on the Riemann surface.

Lemma 4. *Let $\gamma\colon [0,1] \to \mathbb{C}$ be a curve. Then, the image of γ under f, denoted by $f \circ \gamma$ satisfies the following property:*

$$\mathbf{d}_{\mathcal{R}}(f \circ \gamma) = \ell_{\mathrm{N}}(\gamma).$$

Proof. Suppose $\gamma\colon [0,1] \to \mathbb{C}$ is any piecewise differentiable curve, and let $\alpha := f \circ \gamma$. The N-length $\ell_{\mathrm{N}}(\gamma)$ of γ is the finite sum of the N-length of all differentiable pieces of γ. If the path γ goes through the origin, we further break the path at the origin so that α is also differentiable. Thus, it suffices to consider $(a, b) \subset [0, 1]$ so that $\gamma[a, b]$ is a differentiable piece of γ. Then, we have

$$\begin{aligned}
\ell_{\mathrm{N}}(\gamma[a,b]) &= \int_a^b |\gamma(t)||\gamma'(t)|\,dt && |\cdot| \text{ is modulus.}\\
&= \int_a^b |\gamma(t)\gamma'(t)|\,dt && \text{Modulus commutes with product.}\\
&= \int_a^b |\alpha'(t)|\,dt && \text{Chain rule.}\\
&= \ell_{\mathcal{R}}(\alpha[f(a), f(b)]).
\end{aligned}$$

Corollary 1 (Reduction to Euclidean Distances on a Riemann Surface). *Given three points x, y, and p in $\mathbb{R}^d$ such that $p = \mathrm{NN}(x) = \mathrm{NN}(y)$, the nearest neighbor geodesic G from x to y satisfies the following properties:*

1. *G is in the plane determined by x, y, p.*
2. *(a) If the angle formed by x, p, y is $\pi/2$ or more, then G consists of the two straight segments $\overline{xp}$ and $\overline{py}$.*

(b) Otherwise, G is the preimage of the straight line from $f(x)$ to $f(y)$, where f is the quadratic map in the plane given by x, y, p to the Riemann Surface.

4.2 Approximating with Steiner Points

Assume $P \subset \mathbb{R}^d$, $x, y \in P$, and let γ be an arbitrary (x, y)-path. We show how to approximate γ with a piecewise linear path through a collection of Steiner points in $\mathbb{R}^d$. To obtain an accurate estimation of γ, we require the Steiner points to be sufficiently dense. The following definition formalizes this density with a parameter δ.

Definition 2 (δ-sample). *Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let $D \subseteq \mathbb{R}^d$. For a real number $0 < \delta < 1$, a δ-*sample *is a (possibly infinite) set of points $T \subseteq D$ such that if $z \in D \setminus P$, then $\mathbf{d}(z, T) \leq \delta \cdot \mathrm{N}(z)$.*

The following lemma guarantees that an accurate estimation of γ can be computed using a δ-sample. Its proof may be found in the full paper [CFM+15].

Lemma 5. *Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let S be a δ-sample, and let $0 < \delta < 1/10$. Then, for any pair of points $x, y \in P$, there is a piecewise linear path $\eta = (x, s_1, \ldots, s_k, y)$, where $s_1, \ldots, s_k \in S$, such that:*

$$\ell_{\mathrm{N}}(\eta) \leq (1 + C_1 \delta^{2/3}) \mathbf{d}_{\mathrm{N}}(x, y),$$

and, for all $1 \leq i \leq k-1$,

$$\ell_{\mathrm{N}}((s_i, s_{i+1})) \leq C_2 \cdot \delta^{2/3} \cdot \mathrm{N}(s_i).$$

C_1 and C_2 are universal constants.

4.3 The Approximation Graph

So far we have shown that any shortest path can be approximated using a δ-sample that is composed of infinitely many points. In addition, we know how to compute the exact N-distance between any pair of points if they reside in the same Voronoi cell of $\mathrm{Vor}(P)$. Here, we combine these two ideas to be able to approximate any shortest path using only a finite number of Steiner points. The high-level idea is to use the Steiner point approximation while γ passes through regions that are far from P and switch to the exact distance computation as soon as γ is sufficiently close to one of the points in P.

Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of points in $\mathbb{R}^d$, and let B be any convex body that contains P. Fix $\delta \in (0, 1)$, and for any $1 \leq i \leq n$, let $r_i = r_P(p_i)$ be the inradius of the Voronoi cell with site p_i. Also, let $u_i = (1 - \delta^{2/3}) r_i$. Finally, let S be a δ-sample on the domain $B \setminus \bigcup_{1 \leq i \leq n} B(p_i, u_i)$.

Definition 3 (Approximation Graph). *The approximation graph $\mathcal{A} = \mathcal{A}(P, \{u_1, \ldots, u_n\}, S, \delta) = (V_{\mathcal{A}}, E_{\mathcal{A}})$ is a weighted undirected graph, with weight function $w : E_{\mathcal{A}} \to \mathbb{R}^+$. The vertices in $V_{\mathcal{A}}$ are in one to one correspondence with the points in $S \cup P$; for simplicity we use the same notation to refer to corresponding elements in $S \cup P$ and $V_{\mathcal{A}}$. The set $E_{\mathcal{A}}$ is composed of three types of edges:*

1. *If $s_1, s_2 \in S$ and $s_1, s_2 \in B(p_i, r_i)$ for any p_i, then $(s_1, s_2) \in E_{\mathcal{A}}$ and $w(s_1, s_2) = \mathbf{d}_{\mathrm{N}}(s_1, s_2)$. We compute this distance using Corollary 1.*
2. *Otherwise, if $s_1, s_2 \in S$ and $\ell(s_1, s_2) \leq C_2 \delta^{2/3} \max(\mathrm{N}(s_1), \mathrm{N}(s_2))$, where C_2 is the constant of Lemma 5, then $(s_1, s_2) \in E_{\mathcal{A}}$ and $w(s_1, s_2) = \max(\mathrm{N}(s_1), \mathrm{N}(s_2)) \cdot \ell(s_1, s_2)$.*
3. *If $s_1 \in S$ and $s_1 \in B(p_i, r_i)$ then $(p_i, s_1) \in E_{\mathcal{A}}$ and $w(p_i, s_1) = \mathbf{d}_{\mathrm{N}}(p_i, s_1) = (\mathbf{d}(p_i, s_1))^2/2$; see Corollary 1.*

For $x, y \in V_{\mathcal{A}}$ let $\mathbf{d}_{\mathcal{A}}(x, y)$ denote the length of the shortest path from x to y in the graph $\mathcal{A}$.

The following lemma guarantees that the shortest paths in the approximation graph are sufficiently accurate estimations. Its proof my be found in the full paper [CFM+15].

Lemma 6. *Let $\{u_1, \ldots, u_n\}$, S and δ be defined as above. Let $\mathcal{A}(P, \{u_1, \ldots, u_n\}, S, \delta)$ be the approximation graph for P. For any pair of points $x, y \in P$ we have:*

$$(1 - C_2 \delta^{2/3}) \cdot \mathbf{d}_{\mathrm{N}}(x, y) \leq \mathbf{d}_{\mathcal{A}}(x, y) \leq (1 + C_4 \delta^{2/3}) \cdot \mathbf{d}_{\mathrm{N}}(x, y),$$

where C_2 and C_4 are constants computable in $O(1)$ time.

4.4 Construction of Steiner Points

The only remaining piece that we need to obtain an approximation scheme is an algorithm for computing a δ-sample. For this section, given a point set T and $x \in T$, let $r_T(x)$ denote the inradius of the Voronoi cell of $\mathrm{Vor}(T)$ that contains x. Also, given a set T and an arbitrary point x (not necessarily in T), let $\mathbf{f}_T(x)$ denote the distance from x to its *second* nearest neighbor in T.

We can apply existing algorithms for generating meshes and well-spaced points to compute a δ-sample on $\mathcal{D} \setminus \bigcup_i B(p_i, u_i)$, where $\mathcal{D} \subseteq \mathbb{R}^d$ is a domain, and $u_i = (1 - \delta^{2/3}) r_P(p_i)$. The procedure consists of two steps:

1. Use the algorithm of [MSV13] to construct a well-spaced point set M (along with its associated approximate Delaunay graph) with aspect ratio τ in time $2^{O(d)}(n \log n + |M|)$.
2. Then over-refine M to S for the sizing function $g(x) = \frac{2\delta}{11\tau} \mathbf{f}_P(x)$ (while maintaining aspect ratio τ) in time $2^{O(d)}|S|$ by using the algorithm of Section 3.7 in [She11]. (see also [HOMS10] for an earlier use of this technique)

In the above algorithm, we will choose τ to be a fixed constant, say, $\tau = 6$. Both of the meshing algorithms listed above are chosen for their theoretical guarantees on running time. In practice, one could use any quality Delaunay meshing algorithm, popular choices include Triangle [She96] in $\mathbb{R}^2$ and Tetgen [Si11] or CGAL [ART+12] in $\mathbb{R}^3$.

From the guarantees in ([She11]), we know that

$$|S| = O\left(\int_{\mathcal{D}} \frac{dx}{g(x)^d}\right) = \delta^{-O(d)} n \log \varDelta, \tag{3}$$

where $\varDelta$ is the *spread* of P, i.e., the ratio of the largest distance between two points in P to the smallest distance between two points in P.

Now, it remains to show that the point set S is indeed a δ-sample on $\mathcal{D} \setminus \bigcup_i B(p_i, u_i)$. This is provided by the following lemma, whose proof may be found in the full paper [CFM+15].

Lemma 7. *S is a δ-sample on $\mathcal{D} \setminus \bigcup_i B(p_i, u_i)$.*

Now, we calculate the number of edges that will be present in the approximation graph defined in the previous section. For this, we require a few lemmas.

Lemma 8. *Let $A = B(p_i, r_P(p_i)) \setminus B(p_i, u_i)$ be an annulus around p_i. Then, $|A \cap S| = \delta^{-O(d)}$.*

Proof. By the meshing guarantees of [She11], we know that for any point $s \in A \cap S$, $B(s,t)$ does not contain a point from $S \setminus \{s\}$ for $t = \varOmega(r_S(s)) = \varOmega(\delta \cdot r_P(p))$. Thus, the desired result follows using a simple sphere packing argument.

Lemma 9. *If $s \in S$, then $|B(s, C_2\delta^{2/3}\mathrm{N}(s)) \cap S| = \delta^{-O(d)}$, where C_2 is the constant in Lemma 5.*

Proof. As in the previous lemma, meshing guarantees tell us that for any $s' \in B(s, C_2\delta^{2/3}\mathrm{N}(s))$, we have that $B(s',t)$ does not contain a point from $S \setminus \{s'\}$ for $t = \varOmega(\delta \cdot \mathrm{N}(s')) = \varOmega(\delta \cdot \mathrm{N}(s))$. Thus, we again obtain the desired result from a sphere packing argument.

From the above lemmas, we see that $\mathcal{A}$ is composed of $|S| = \delta^{-O(d)} n \log \varDelta$ vertices and $n\delta^{-O(d)} + |S| \cdot \delta^{-O(d)} = |S| \cdot \delta^{-O(d)}$ edges.

Remark. Note that the right hand side of (3) is in terms of the spread, a non-combinatorial quantity. Indeed, one can construct examples of P for which the integral in (3) is not bounded from above by any function of n. However, for many classes of inputs, one can obtain a tighter analysis. In particular, if P satisfies a property known as *well-paced*, one can show that the resulting set S will satisfy $|S| = 2^{O(d)}n$ (see [MPS08, She12]).

In a more general setting (without requiring that P is well-paced), one can modify the algorithms to produce output in the form of a *hierarchical* *mesh* [MPS11]. This then produces an output of size $2^{O(d)}n$, and $(1+\varepsilon)$-approximation algorithm for the nearest neighbor metric can be suitably modified so that the underlying approximation graph uses a hierarchical set of points instead of a full δ-sample. However, we ignore the details here for the sake of simplicity of exposition.

The above remark, along with the edge count of $\mathcal{A}$ and the running time guarantees from [MSV13], yields Theorem 2, the main theorem of this section.

5 Discussion

Motivated by estimating geodesic distances within subsets of $\mathbb{R}^n$, we consider two distance metrics in this paper: the N-distance and the edge-squared distance. The main focus of this paper is to find an approximation of the N-distance. One possible drawback of our $(1+\varepsilon)$-approximation algorithm is its exponential dependency on d. To alleviate this dependency a natural approach is using a Johnson-Lindenstrauss type projection. Thereby, we would like to ask which properties are preserved under random projections such as those in Johnson-Lindenstrauss transforms.

We are currently working on implementing the approximation algorithm presented in Section 3. We hope to show that this approximation is fast in practice as well as in theory.

Acknowledgement. The authors would like to thank Larry Wasserman for helpful discussions.

References

[AMS05] Aleksandrov, L., Maheshwari, A., Sack, J.-R.: Determining approximate shortest paths on weighted polyhedral surfaces. J. ACM **52**(1), 25–53 (2005)

[ART+12] Alliez, P., Rineau, L., Tayeb, S., Tournois, J., Yvinec, M.: 3D mesh generation. In: CGAL User and Reference Manual. CGAL Editorial Board, 4.1 edn. (2012)

[BCH04] Bousquet, O., Chapelle, O., Hein, M.: Measure based regularization. In: 16th NIPS (2004)

[Ber96] Bernoulli, J.: Branchistochrone problem. Acta Eruditorum, June 1696

[BRS11] Bijral, A.S., Ratliff, N.D., Srebro, N.: Semi-supervised learning with density based distances. In: Cozman, F.G., Pfeffer, A. (eds.) UAI, pp. 43–50. AUAI Press (2011)

[CFM+15] Cohen, M.B., Fasy, B.T., Miller, G.L., Nayyeri, A., Sheehy, D., Velingker, A.: Approximating nearest neighbor distances, 2015. CoRR, abs/1502.08048

[DBVKOS00] De Berg, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O.C.: Computational Geometry. Springer (2000)

[HDI14] Hwang, S.J., Damelin, S.B., Hero, A.O., III: Shortest path through random points (2014). arXiv/1202.0045v3

[HOMS10] Hudson, B., Oudot, S.Y., Miller, G.L., Sheehy, D.R.: Topological inference via meshing. In: SOCG: Proceedings of the 26th ACM Symposium on Computational Geometry (2010)

[Hp11] Har-peled, S.: Geometric Approximation Algorithms. American Mathematical Society, Boston (2011)

[KH03] Kim, J., Hespanha, J.P.: Discrete approximations to continuous shortest-path: Application to minimum-risk path planning for groups of uavs. In: 42nd IEEE ICDC, January 2003

[LSV06] Lukovszki, Tamás, Schindelhauer, Christian, Volbert, Klaus: Resource efficient maintenance of wireless network topologies. Journal of Universal Computer Science **12**(9), 1292–1311 (2006)

[MP91] Joseph, S.B.: Mitchell and Christos H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. J. ACM **38**(1), 18–73 (1991)

[MPS08] Miller, G.L., Phillips, T., Sheehy, D.R.: Linear-size meshes. In: CCCG: Canadian Conference in Computational Geometry (2008)

[MPS11] Miller, G.L., Phillips, T., Sheehy, D.R.: Beating the spread: Time-optimal point meshing. In: SOCG: Proceedings of the 27th ACM Symposium on Computational Geometry (2011)

[MSV13] Miller, G.L., Sheehy, D.R., Velingker, A.: A fast algorithm for well-spaced points and approximate delaunay graphs. In: 29th SOCG. SoCG 2013, pp. 289–298. ACM, New York (2013)

[RR90] Rowe, Neil, Ross, Ron: Optimal grid-free path planning across arbitrarily contoured terrain with anisotropic friction and gravity effects. IEEE Transactions on Robotics and Automation **6**(5), 540–553 (1990)

[She96] Shewchuk, J.R.: Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In: Lin, M.C., Manocha, Dinesh (eds.) FCRC-WS 1996 and WACG 1996. LNCS, vol. 1148, pp. 203–222. Springer, Heidelberg (1996)

[She11] Sheehy, D.: Mesh Generation and Geometric Persistent Homology. PhD thesis, Carnegie Mellon University, Pittsburgh, July 2011. CMU CS Tech Report CMU-CS-11-121

[She12] Sheehy, Donald R.: New Bounds on the Size of Optimal Meshes. Computer Graphics Forum **31**(5), 1627–1635 (2012)

[Si11] Si, H.: TetGen: A quality tetrahedral mesh generator and a 3D Delaunay triangulator, January 2011. http://tetgen.org/

[SO05] Sajama and Orlitsky, A.: Estimating and computing density based distance metrics. In: ICML 2005, pp. 760–767. ACM, New York (2005)

[Str] Strain, J.: Calculus of variation. http://math.berkeley.edu/strain/170.S13/cov.pdf

[Tsi95] Tsitsiklis, John N.: Efficient algorithms for globally optimal trajectories. IEEE Transactions on Automatic Control **40**, 1528–1538 (1995)

[VB03] Vincent, P., Bengio, Y.: Density sensitive metrics and kernels. In: Snowbird Workshop (2003)