Proximity search in high dimensions

Ioannis Psarros

Institute of Computer Science,
University of Bonn

November 5, 2020

Outline

1. Introduction
2. When the dimension is constant
3. When the dimension is high
4. When the data are trajectories

Outline

1. Introduction

2. When the dimension is constant

3. When the dimension is high

4. When the data are trajectories

The main problem

Definition (c-Approximate Nearest Neighbor problem)

Given a finite set $P \subset \mathcal{M}$, a distance function $\mathrm{d}(\cdot,\cdot)$, and an approximation factor $c > 1$, preprocess P into a data structure which supports the following type of queries:

$$\text{given } q \in \mathcal{M}, \text{ find } p^* \text{ such that } \forall p \in P: \ d_{\mathcal{M}}(q, p^*) \leq c \cdot \mathrm{d}_{\mathcal{M}}(q, p).$$

Our focus: $\mathcal{M}$ is $\mathbb{R}^d$ or $\{0,1\}^d$, $\mathrm{d}_{\mathcal{M}}$ is $\|\cdot\|_2$ or $\|\cdot\|_1$.

The main problem

Hopefully the following problem is easier.

Definition ((c, r)-Approximate Near Neighbor (ANN) problem)

Given a finite set $P \subset \mathbb{R}^d$, an approximation factor $c > 1$, and a range $r > 0$, preprocess P into a data structure which supports the following type of queries:

- if $\exists p^* \in P$ s.t. $\|p^* - q\| \leq r$, then it returns any point $p' \in \mathbb{R}^d$ s.t. $\|p' - q\| \leq c \cdot r$,
- if $\forall p \in P$, $\|p - q\| > c \cdot r$, then report "Fail".

The data structure returns either a point at distance $\leq c \cdot r$ or "Fail".

Outline

Computational model

Assumption

We assume that every hashing operation takes worst-case $\mathcal{O}(1)$ time.

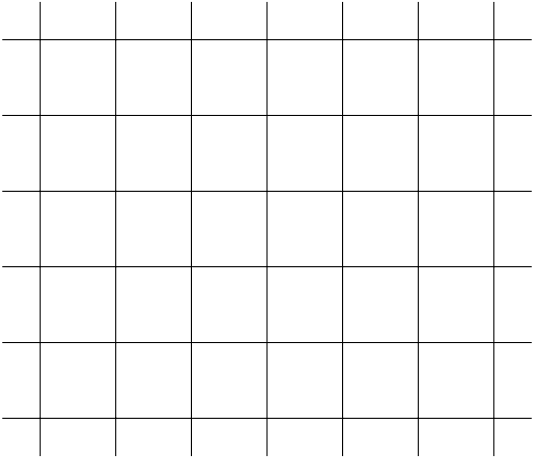
See e.g. Perfect Hashing in CLRS.

Assumption

We use the unit cost RAM model. Every operation on reals in $\mathcal{O}(1)$ time, including $\lfloor\cdot\rfloor$.

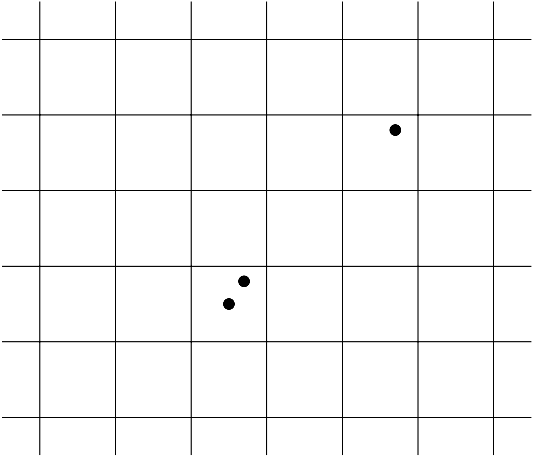
The grid

$\mathcal{G}_\delta$ is the grid of side-length δ.



Grid in $\mathbb{R}^2$.

The grid



Store points for point location.

The grid

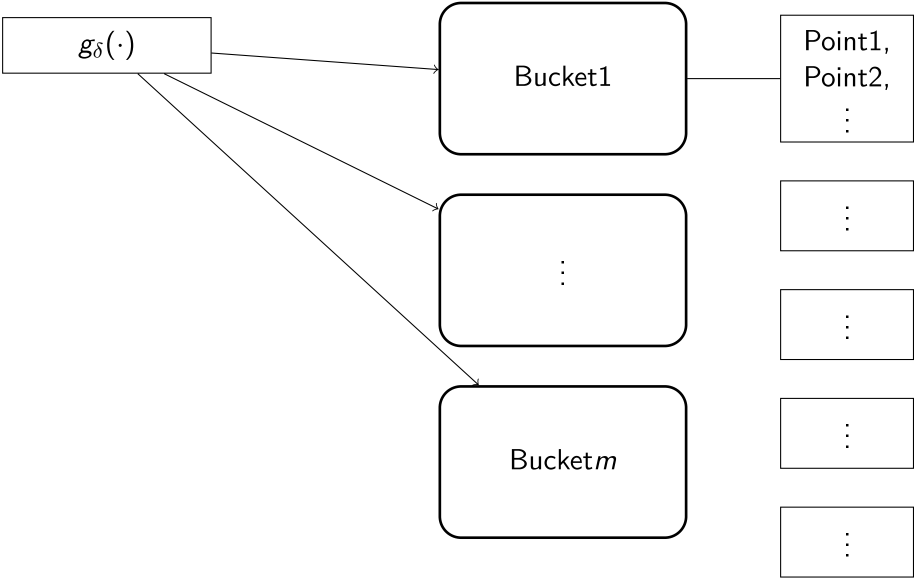
For any $x \in \mathbb{R}^d$, we define

$$g_\delta(x) = \left(\left\lfloor \frac{x_1}{\delta} \right\rfloor, \left\lfloor \frac{x_2}{\delta} \right\rfloor, \ldots, \left\lfloor \frac{x_d}{\delta} \right\rfloor \right).$$

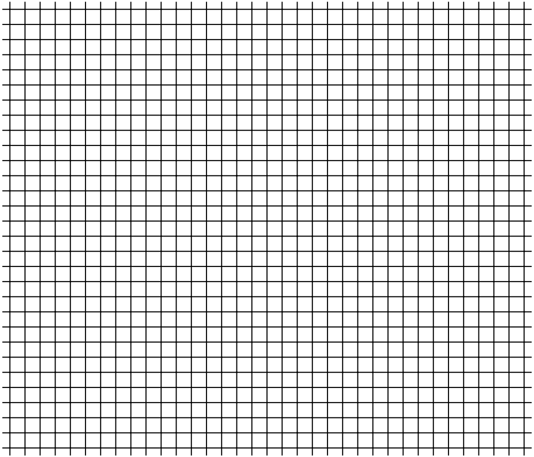
Idea: Use $g_\delta(\cdot)$ as a key; store cells in buckets. Each bucket contains a linked list of pointers to the points lying in the corresponding cell.

Store a set P of n points in a grid using $\mathcal{O}(dn)$ storage.
Queries of the form: "for $q \in \mathbb{R}^d$, return a pointer to the list of points of P which lie in the same cell" in $\mathcal{O}(d)$ time.

The grid

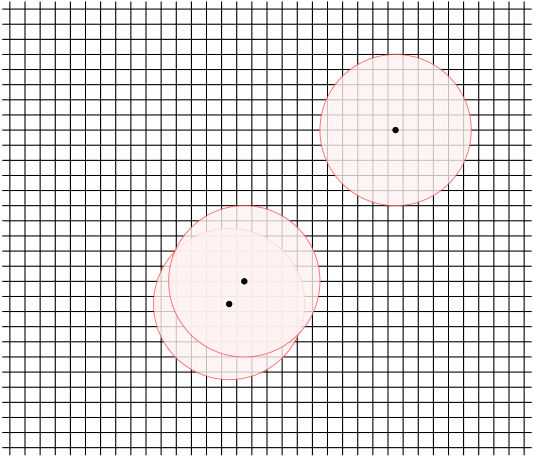


ANN data structure-fast query



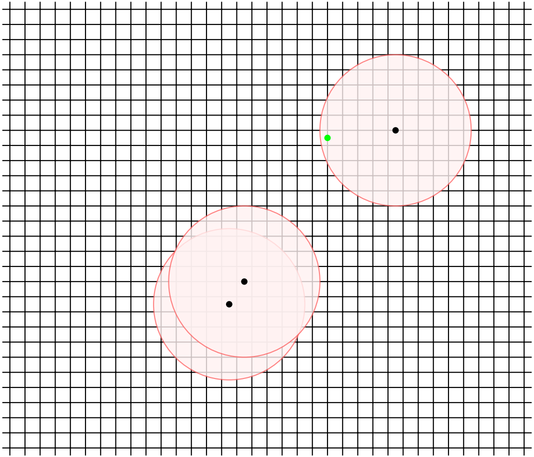
Improve resolution.

ANN data structure-fast query



For each $p \in P$, store a pointer to p in the cells intersecting the ball of radius 1 centered at p.

ANN data structure-fast query



To answer a query: compute $g_\delta(q)$, probe the hash-table.

ANN data structure-fast query

How many non-empty cells?

For a set P of n points, we have $\mathcal{O}(n \cdot N_\delta^d)$ non-empty cells.
In order to achieve $1+\varepsilon$ approximation, we set $\delta = \varepsilon/\sqrt{d}$.

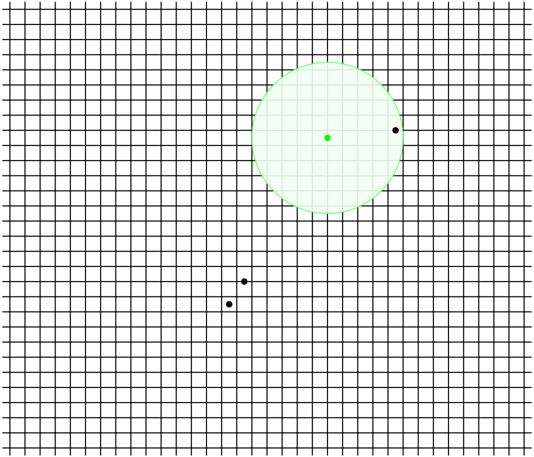
For $\delta = \varepsilon/\sqrt{d}$,

$$N_\delta^d = \mathcal{O}\left(\frac{1}{\varepsilon}\right)^d.$$

It suffices to bound the volume of a ball of radius $2/\delta$ in $\mathbb{R}^d$:

$$N_\delta^d \leq \frac{\mathrm{vol}(\bigcirc(2))}{\mathrm{vol}(\square(\delta))} = \frac{\mathrm{vol}(\bigcirc(2/\delta))}{\mathrm{vol}(\square(1))} = \frac{(2 \cdot \Gamma(1+1/2))^d}{\Gamma(1+d/2)} \cdot \left(\frac{2}{\delta}\right)^d = \mathcal{O}\left(\frac{1}{\varepsilon}\right)^d.$$

ANN data structure-efficient space



To answer a query: compute $g_\delta(q)$, make all necessary probes.

Outline

Random projections

Idea: Randomly project points to reduce the dimension.

2-stability property

For any vector g of d independent random variables following $N(0,1)$ and any vector $u \in \mathbb{R}^d$, we have $\langle g, u \rangle \sim N(0, \|u\|^2)$.

Moment generating function of X^2

Let $X \sim N(0,1)$. Then if $t < 1/2$,

$$\mathbb{E}\left[e^{tX^2}\right] = \frac{1}{\sqrt{1-2t}}.$$

Random projections

Let G be a matrix of size $k \times d$ with elements i.i.d. random variables following $N(0,1)$. Sample G and set $A := \frac{1}{\sqrt{k}} G$.

Lemma

For any $x \in \mathbb{R}^d$ and $\varepsilon < 1/2$,

$$\Pr\left[\|Ax\| \notin (1 \pm \varepsilon)\|x\|\right] \leq \frac{2}{\mathrm{e}^{\frac{\varepsilon^2 k}{8}}}$$

Proof

Let $\|x\| = 1$,

$$\Pr\left[\|Ax\|^2 \geq (1+\varepsilon)\right] \overset{t \geq 0}{=} \Pr\left[\mathrm{e}^{t\|Ax\|^2} \geq \mathrm{e}^{t(1+\varepsilon)}\right] \leq \frac{\mathbb{E}\left[\mathrm{e}^{t\|Ax\|^2}\right]}{\mathrm{e}^{t(1+\varepsilon)}}$$

Random projections

Proof (cont.)

But we can use the 2-stability property, to obtain:

$$\mathbb{E}\left[\mathrm{e}^{t\|Ax\|^2}\right] = \mathop{\mathbb{E}}_{X_i \sim N(0,1)}\left[\mathrm{e}^{t\sum_{i=1}^k X_i^2}\right] = \left(\mathop{\mathbb{E}}_{X \sim N(0,1)}\left[\mathrm{e}^{tX^2}\right]\right)^k = \left(\frac{1}{\sqrt{1-2t}}\right)^k$$

So, we have

$$\Pr\left[\|Ax\|^2 \geq (1+\varepsilon)\right] \leq \left(\frac{1}{\sqrt{1-2t}}\right)^k \cdot \mathrm{e}^{-t(1+\varepsilon)} \overset{t=\varepsilon/(2(1+\varepsilon))}{\leq} \mathrm{e}^{-\varepsilon^2 k/8}.$$

Bounding the probability of having large contraction is similar.

Johnson Lindenstrauss lemma

> Suppose that we have n points in $\mathbb{R}^d$. What target dimension k is needed so that all pairwise distances are approximately preserved?

Mapping A is linear. We have $\binom{n}{2}$ vectors, so the probability that there exists one which is arbitrarily distorted is:

$$\binom{n}{2} \cdot \Pr\left[\|Ax\| \notin (1 \pm \varepsilon)\|x\|\right] \leq \binom{n}{2} \cdot \frac{2}{\mathrm{e}^{\frac{\varepsilon^2 k}{8}}}.$$

So there exists $k = \mathcal{O}(\varepsilon^{-2} \log n)$ such that all distances are approximately preserved.

ANN data structure

Fast query time.

Randomly project points, then use the grid.

- Space: $n^{\mathcal{O}(1/\varepsilon^2)} + \mathcal{O}(dn)$
- Query: $\mathcal{O}(d)$

Efficient space.

- Space: $\mathcal{O}(dn)$
- Query: $n^{\mathcal{O}(1/\varepsilon^2))}$

ANN data structure

Fast query time.

Randomly project points, then use the grid.

- Space: $n^{\mathcal{O}(1/\varepsilon^2)} + \mathcal{O}(dn)$
- Query: $\mathcal{O}(d)$

Efficient space.

- Space: $\mathcal{O}(dn)$
- Query: $n^{\mathcal{O}(1/\varepsilon^2))}$

Random projections with false positives

Idea: Further reduce the dimension, check more candidate points.

When we project to dimension k, the expected number of *false positives*

$$n \cdot \frac{2}{e^{\frac{\varepsilon^2 k}{8}}}.$$

In the randomly projected space, check at most $m \approx n \cdot \frac{2}{e^{\frac{\varepsilon^2 k}{8}}}$ points.

Query time:

$$\mathcal{O}\left(\frac{1}{\varepsilon}\right)^k + n \cdot \frac{2}{e^{\frac{\varepsilon^2 k}{8}}} = n^{1-\mathcal{O}(\varepsilon^2/\log(1/\varepsilon))}.$$

Locality sensitive hashing

Definition

Let reals $r_1 < r_2$ and $p_1 > p_2 > 0$. We call a family F of hash functions (p_1, p_2, r_1, r_2)-sensitive for a metric space $\mathcal{M}$ if, for any $x, y \in \mathcal{M}$, and h distributed randomly in F, it holds:

- $\mathrm{d}_{\mathcal{M}}(x,y) \leq r_1 \implies Pr[h(x) = h(y)] \geq p_1$,
- $\mathrm{d}_{\mathcal{M}}(x,y) \geq r_2 \implies Pr[h(x) = h(y)] \leq p_2$.

We will now focus on the Hamming space $(\{0,1\}^d, \|\cdot\|_1)$.

Locality sensitive hashing

For any $x = (x_1, \ldots, x_d) \in \{0,1\}^d$, $h_i(x) = x_i$.

$$\mathcal{H} = \{h_i \mid \forall i \in [d]\}.$$

Pick uniformly at random $h \in \mathcal{H}$. Then

$$\Pr[h(x) = h(y)] = 1 - \frac{\|x - y\|_1}{d}.$$

The family $\mathcal{H}$ is $(r, cr, 1 - \frac{r}{d}, 1 - \frac{cr}{d})$-sensitive, where $r > 0$, $c > 1$.

However the probability of having a false positive is quite large.

Locality sensitive hashing

Define new family $G(\mathcal{H}) := \mathcal{H}^k$.

Preprocessing:

1. Pick uniformly at random L functions $g_1, \ldots, g_L \in G(\mathcal{H})$
2. For each $p \in P$, assign p in bucket with key $g_i(p)$

Query:

1. For each $i = 1, \ldots, L$:
   1. for each p in bucket $g_i(q)$:
      1. if number of retrieved points $> 3L$ then return "no"
      2. if $\|q - p\|_1 < cr$ then return p

Space usage: $\mathcal{O}(Ln + dn)$.
Query time: $\mathcal{O}(L(k + d))$.

Locality sensitive hashing

Let $p_1 = 1 - \frac{r}{d}$, $p_2 = 1 - \frac{cr}{d}$.
The probability of having a false positive:

$$\Pr[g_i(p) = g_i(q) \mid \|p - q\|_1 \geq cr] \leq \left(1 - \frac{cr}{d}\right)^k = \frac{1}{n}$$

for $k = \log_{1/p_2} n$. So the total number of expected false positives:

$$L \cdot n \cdot \frac{1}{n} = L,$$

And by Markov's inequality, the probability that the number of false positives exceeds $3L$ is at most $1/3$.

Locality sensitive hashing

The probability of finding a near neighbor in one hashtable is

$$\left(1 - \frac{r}{d}\right)^k = \frac{1}{n^{\frac{\log(1/p_1)}{\log(1/p_2)}}}.$$

So the probability of not finding it in the L hashtables:

$$\left(1 - \frac{1}{n^{\frac{\log(1/p_1)}{\log(1/p_2)}}}\right)^L = \frac{1}{\mathrm{e}},$$

for $L = n^{\frac{\log(1/p_1)}{\log(1/p_2)}} \leq n^{\frac{1}{1+\varepsilon}}$.

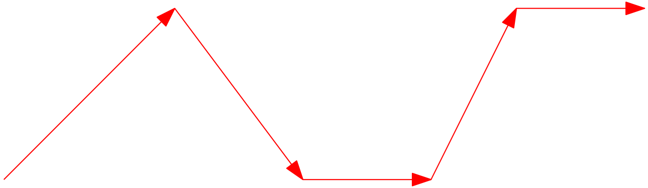
Outline

1. Introduction
2. When the dimension is constant
3. When the dimension is high
4. When the data are trajectories

Discrete Fréchet Distance

What is a polygonal curve?

A sequence of vertices $v_1, \ldots, v_m$ in $\mathbb{R}^d$, with edges $\overline{v_1 v_2}, \overline{v_2 v_3}, \ldots \overline{v_{m-1} v_m}$.



Why curves?

Trajectories, data from mobiles, GPS sensors, video analysis etc.

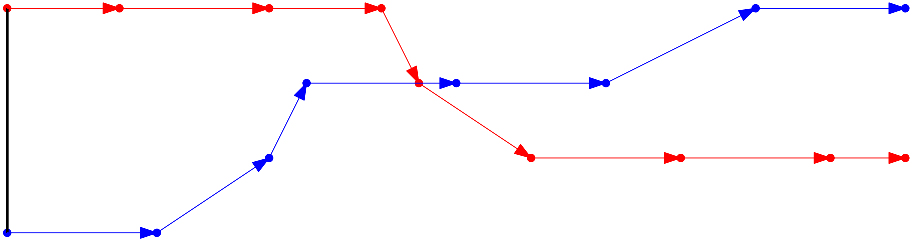
Discrete Fréchet Distance

Definition (Traversal)

Given polygonal curves $V = v_1, \ldots, v_{m_1}$, $U = u_1, \ldots, u_{m_2}$, a traversal $T = (i_1, j_1), \ldots, (i_t, j_t)$ is a sequence of pairs of indices s.t.:

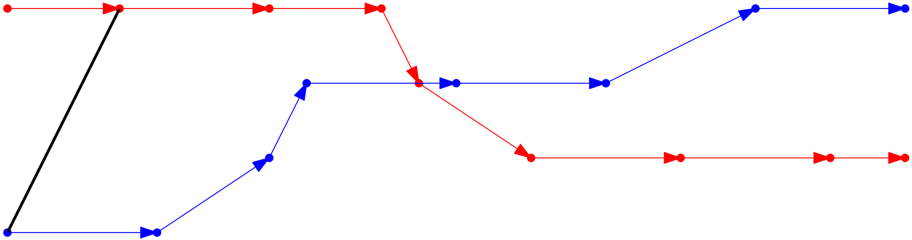
1. $i_1, j_1 = 1$, $i_t = m_1$, $j_t = m_2$.
2. $\forall (i_k, j_k) \in T : i_{k+1} - i_k \in \{0, 1\}$ and $j_{k+1} - j_k \in \{0, 1\}$.
3. $\forall (i_k, j_k) \in T : (i_{k+1} - i_k) + (j_{k+1} - j_k) \geq 1$.

Discrete Fréchet Distance



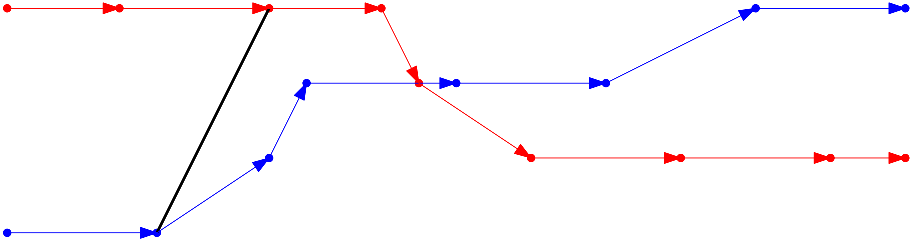
Starting position

Discrete Fréchet Distance



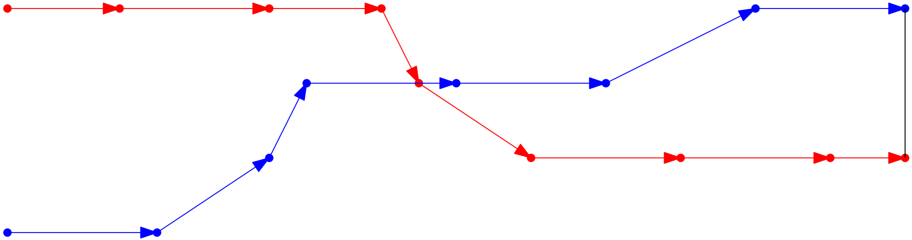
1st jump

Discrete Fréchet Distance



2nd jump

Discrete Fréchet Distance



Final position

Discrete Fréchet Distance

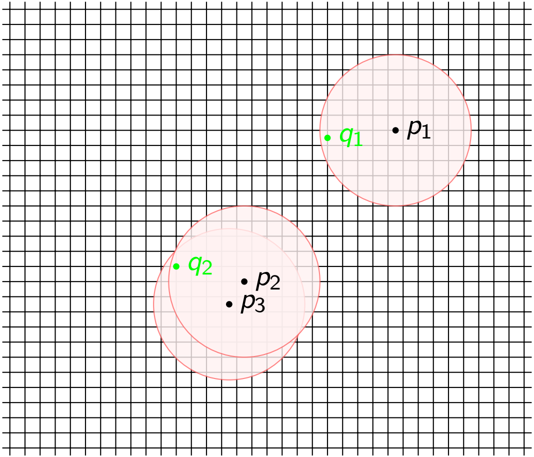
Definition (Discrete Fréchet Distance)

Given polygonal curves $V = v_1, \ldots, v_{m_1}$, $U = u_1, \ldots, u_{m_2}$, we define the discrete Fréchet distance between V and U as the following function:

$$d_{dF}(V, U) = \min_{T \in \mathcal{T}} \max_{(i_k, j_k) \in T} \|v_{i_k} - u_{j_k}\|,$$

where $\mathcal{T}$ denotes the set of all possible traversals for V and U.

Data structure



Enumerate candidate query sequences. How many?

Data structure

Each polygonal curve has at most m vertices.
Enumerate all possible (approximate) query sequences: use the $m \cdot N_\delta^d$ near points.
Naive upper bound:

$$m^m \cdot \mathcal{O}\left(\frac{1}{\varepsilon}\right)^{dm}$$

candidate curves.

Better bound possible if we take into account the ordering of the vertices.