Approximate Furthest Neighbor in High Dimensions

Rasmus Pagh, Francesco Silvestri, **Johan Sivertsen**, and Matthew Skala

October 13, 2015

IT UNIVERSITY OF COPENHAGEN

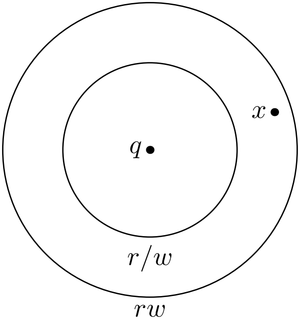
Agenda

- Annulus Query
- The Furthest Neighbor Problem
- Techniques and results
- Experiments
- Open problems

Annulus query

Definition

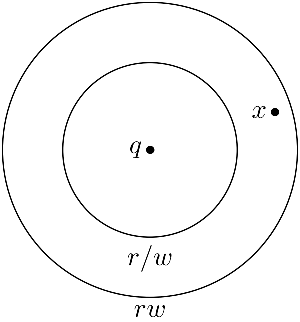
Given $S \subseteq \mathbb{R}^d$ a query point q and parameters $r, w > 1$ return x such that $\frac{r}{w} \leq ||x - q||_2 \leq wr$.



Annulus query

Definition

Given $S \subseteq \mathbb{R}^d$ a query point q and parameters $r, w > 1$ return x such that $\frac{r}{w} \leq ||x - q||_2 \leq wr$.

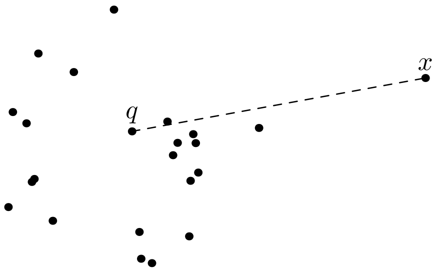


Applications in recommender systems.

The Furthest Neighbor Problem

Definition

Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$ find x with $\max ||x - q||_2$.



Sublinear Furthest Neighbor

- For $q \in \{0, 1\}^d$:

Sublinear Furthest Neighbor

- For $q \in \{0,1\}^d$:
- Furthest Neighbor $-q$ = Nearest Neighbor q [Goel et al.,'09]

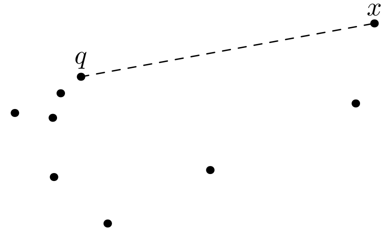
Sublinear Furthest Neighbor

- For $q \in \{0,1\}^d$:
- Furthest Neighbor $-q$ = Nearest Neighbor q [Goel et al.,'09]
- Sublinear time Nearest Neighbor breaks SETH [Williams,'04], [Alman & Williams,'15]

Approximate Furthest Neighbor

Definition

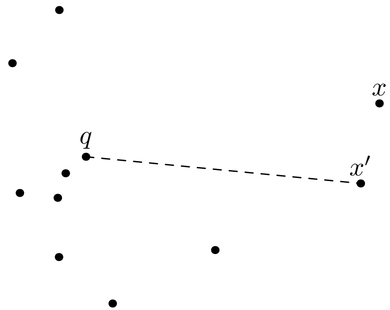
Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$, let x be the furthest neighbor. Return x' such than $||x' - q|| \geq \frac{||x-q||}{c}$. We call this c-FN.



Approximate Furthest Neighbor

Definition

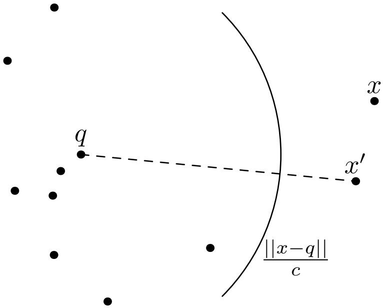
Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$, let x be the furthest neighbor. Return x' such than $||x' - q|| \geq \frac{||x-q||}{c}$. We call this c-FN.



Approximate Furthest Neighbor

Definition

Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$, let x be the furthest neighbor. Return x' such than $||x' - q|| \geq \frac{||x-q||}{c}$. We call this c-FN.



Related work

Though not nearly as popular as Nearest Neighbor there is notable prior work on Furthest neighbor.

Paper	**c**	**Query Time**
Bespamyatnikh '96	$c > 1$	$\mathcal{O}((1+\frac{1}{c-1})^{d-1})$
Goel et al. '01	$c > \sqrt{2}$	$\mathcal{O}(d^2)$
Indyk et al. '03	$c > 1$	$\mathcal{O}(n^{1/c^2} d \log^{(1-1/c)/2}(n) \log_{1+\delta}(d) \log\log_{1+\delta} d)$
This paper	$c > 1$	$\mathcal{O}(n^{1/c^2} \log^{\frac{c^2}{2}-\frac{1}{3}}(n)(d+\log n))$

Related work

Though not nearly as popular as Nearest Neighbor there is notable prior work on Furthest neighbor.

Paper	**c**	**Query Time**
Bespamyatnikh '96	$c > 1$	$\mathcal{O}((1 + \frac{1}{c-1})^{d-1})$
Goel et al. '01	$c > \sqrt{2}$	$\mathcal{O}(d^2)$
Indyk et al. '03	$c > 1$	$\mathcal{O}(n^{1/c^2} d \log^{(1-1/c)/2}(n) \log_{1+\delta}(d) \log\log_{1+\delta} d)$
This paper	$c > 1$	$\mathcal{O}(n^{1/c^2} \log^{\frac{c^2}{2}-\frac{1}{3}}(n)(d + \log n))$

- Split tree

Related work

Though not nearly as popular as Nearest Neighbor there is notable prior work on Furthest neighbor.

Paper	c	Query Time
Bespamyatnikh '96	$c > 1$	$\mathcal{O}((1 + \frac{1}{c-1})^{d-1})$
Goel et al. '01	$c > \sqrt{2}$	$\mathcal{O}(d^2)$
Indyk et al. '03	$c > 1$	$\mathcal{O}(n^{1/c^2} d \log^{(1-1/c)/2}(n) \log_{1+\delta}(d) \log\log_{1+\delta} d)$
This paper	$c > 1$	$\mathcal{O}(n^{1/c^2} \log^{\frac{c^2}{2}-\frac{1}{3}}(n)(d + \log n))$

- Split tree
- Minimum enclosing ball

Related work

Though not nearly as popular as Nearest Neighbor there is notable prior work on Furthest neighbor.

Paper	**c**	**Query Time**
Bespamyatnikh '96	$c > 1$	$\mathcal{O}((1 + \frac{1}{c-1})^{d-1})$
Goel et al. '01	$c > \sqrt{2}$	$\mathcal{O}(d^2)$
Indyk et al. '03	$c > 1$	$\mathcal{O}(n^{1/c^2} d \log^{(1-1/c)/2}(n) \log_{1+\delta}(d) \log\log_{1+\delta} d)$
This paper	$c > 1$	$\mathcal{O}(n^{1/c^2} \log^{\frac{c^2}{2}-\frac{1}{3}}(n)(d + \log n)$

- Split tree
- Minimum enclosing ball
- Random Projections, binary search

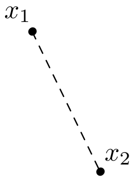
Related work

Though not nearly as popular as Nearest Neighbor there is notable prior work on Furthest neighbor.

Paper	**c**	**Query Time**
Bespamyatnikh '96	$c > 1$	$\mathcal{O}((1 + \frac{1}{c-1})^{d-1})$
Goel et al. '01	$c > \sqrt{2}$	$\mathcal{O}(d^2)$
Indyk et al. '03	$c > 1$	$\mathcal{O}(n^{1/c^2} d \log^{(1-1/c)/2}(n) \log_{1+\delta}(d) \log\log_{1+\delta} d)$
This paper	$c > 1$	$\mathcal{O}(n^{1/c^2} \log^{\frac{c^2}{2}-\frac{1}{3}}(n)(d + \log n)$

- Split tree
- Minimum enclosing ball
- Random Projections, binary search
- Random Projections, single query

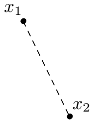
Random Projections



Random Projections

Lemma (Distance preservation)

$$a_i \cdot (x_1 - x_2) \sim \mathcal{N}(0,1)||x_1 - x_2||_2 \tag{1}$$

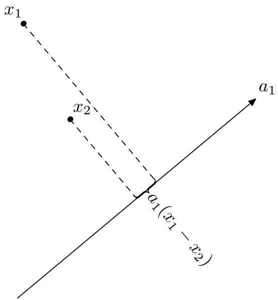


$$a_i = \{g_1, g_2, .., g_d\} \text{ where } g_j \sim \mathcal{N}(0,1)$$

Random Projections

Lemma (Distance preservation)

$$a_i \cdot (x_1 - x_2) \sim \mathcal{N}(0,1)||x_1 - x_2||_2 \quad (1)$$

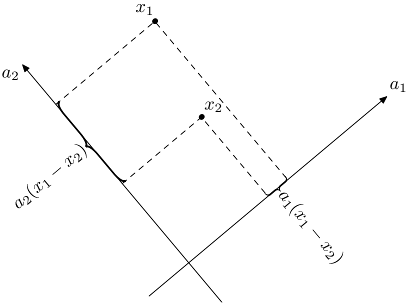


$$a_i = \{g_1, g_2, .., g_d\} \text{ where } g_j \sim \mathcal{N}(0,1)$$

Random Projections

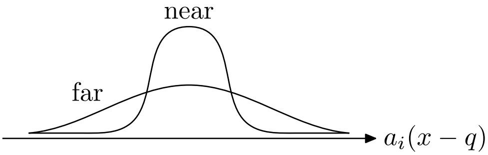
Lemma (Distance preservation)

$$a_i \cdot (x_1 - x_2) \sim \mathcal{N}(0,1)||x_1 - x_2||_2 \quad (1)$$



$$a_i = \{g_1, g_2, .., g_d\} \text{ where } g_j \sim \mathcal{N}(0,1)$$

Crossing the threshold



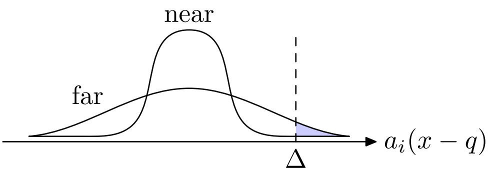
Crossing the threshold

Lemma (Threshold projection)

$\exists \Delta$:

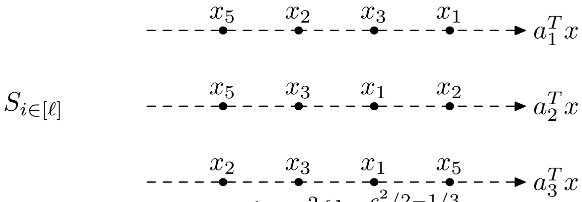
$$\Pr_a [a \cdot (x - q) \geq \Delta] \leq \frac{\log^{c^2/2-1/3} n}{n}, \textit{ for } ||x - q||_2 < r/c$$

$$\Pr_a [a \cdot (x - q) \geq \Delta] \geq (1 - o(1)) \frac{1}{n^{1/c^2}}, \textit{ for } ||x - 1||_2 \geq r/c$$



Data structure

Points are stored in their projection order. We use $\ell = 2n^{1/c^2}$ projections and store the top m points in each.

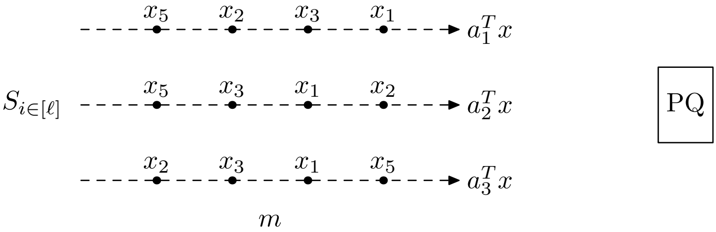


$$m = 1 + e^2 \ell \log^{c^2/2 - 1/3} n$$

$\mathcal{O}(\ell m d)$ space

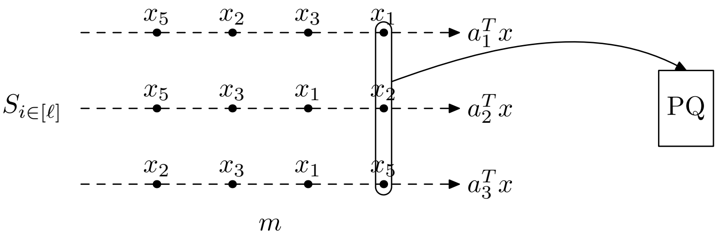
Query procedure

- Create an empty priority queue PQ.



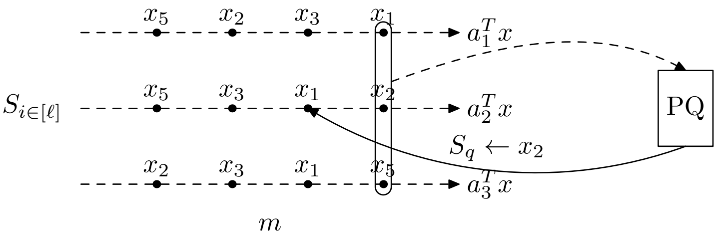
Query procedure

- Create an empty priority queue PQ.
- Add the $\ell = 2n^{1/c^2}$ points .
- Points are added with priority $a_i \cdot x - a_i \cdot q$.



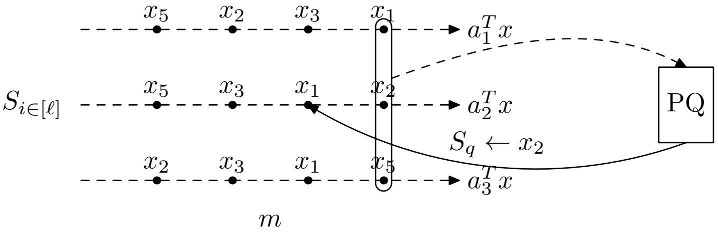
Query procedure

- Create an empty priority queue PQ.
- Add the $\ell = 2n^{1/c^2}$ points .
- Points are added with priority $a_i \cdot x - a_i \cdot q$.
- Take out the top priority element and examine its distance to q.
- If it is not far enough away take its neighbor on the projection it came from.



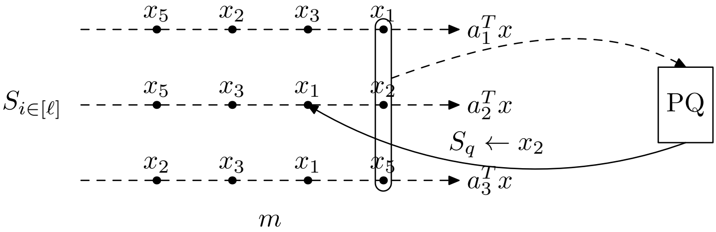
Query procedure

- Create an empty priority queue PQ.
- Add the $\ell = 2n^{1/c^2}$ points .
- Points are added with priority $a_i \cdot x - a_i \cdot q$.
- Take out the top priority element and examine its distance to q.
- If it is not far enough away take its neighbor on the projection it came from.
- We will look at at most $m = 1 + e^2 \ell \log n^{c^2/2 - 1/3}$ points.



Query procedure

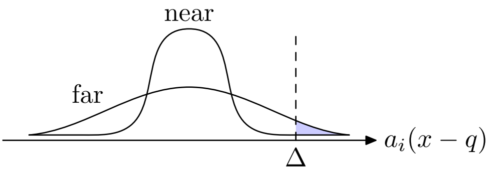
- Create an empty priority queue PQ.
- Add the $\ell = 2n^{1/c^2}$ points .
- Points are added with priority $a_i \cdot x - a_i \cdot q$.
- Take out the top priority element and examine its distance to q.
- If it is not far enough away take its neighbor on the projection it came from.
- We will look at at most $m = 1 + e^2 \ell \log n^{c^2/2 - 1/3}$ points.
- Time $\mathcal{O}(\ell + m(d + \log \ell))$



Theoretical results

Corollary (Failure probability)

Pr[Missing the far point] $\leq (1 - 1/n^{1/c^2})^{\ell} \leq 1/e^2$.
Pr[Too many close points] $\leq Pr[\ell \log^{c^2/2-1/3} n > m] \leq 1/e^2$.



$\ell = 2n^{1/c^2}$
$m = 1 + e^2\ell$

Theoritical results

Theorem (Approximate Furthest Neighbor)

There exists a datastructure for $c-FN$ over any set $S \in \mathbb{R}^d$ of at most n points, such that:

- *Queries take $\widetilde{\mathcal{O}}(n^{1/c^2}d)$ time.*
- *The data structure uses $\widetilde{\mathcal{O}}(n^{1+1/c^2}d)$ space.*

With probability of success at least $1 - 2/e^2 \geq 0.72$

Experimental results

NASA dataset $d = 128$

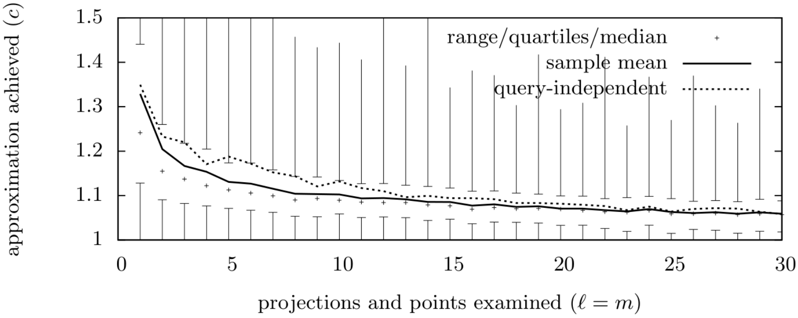


Fig. 3. Experimental results for SISAP nasa database

Experimental results

Normally distributed dataset $d = 10$

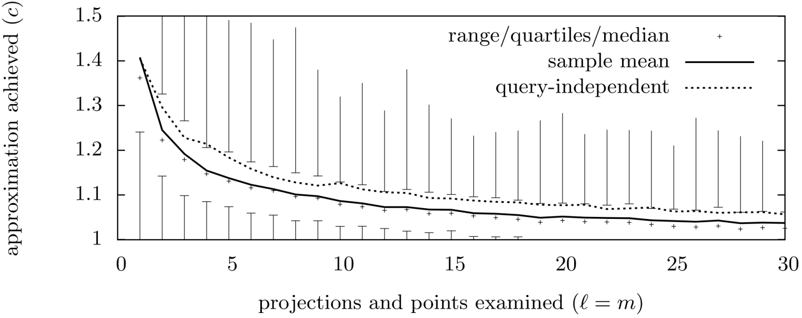


Fig. 2. Experimental results for 10-dimensional normal distribution

Query Independent: Use same size m set for all query points.
Relation to convex hull.

Combining these techniques with LSH gets:

Theorem (Annulus query)

There exists a data structure for (c, w, r)*-AAQ over any set* $S \in \mathbb{R}^d$ *of at most* n *points, such that:*

- *Queries can be answered in time* $\widetilde{\mathcal{O}}(n^{\rho+1/c^2})$
- *The data structure takes space* $\widetilde{\mathcal{O}}(n^{1+\rho+1/c^2})$ *in addition to storing* S*.*

The failure probability of the data structure is less than 0.98*.*

Open problems

- Expand the random projection technique to other spaces.
  General Metric, Hamming?

Open problems

- Expand the random projection technique to other spaces. General Metric, Hamming?
- Use furthest neighbor to improve LSH output sensitivity?

Open problems

- Expand the random projection technique to other spaces. General Metric, Hamming?
- Use furthest neighbor to improve LSH output sensitivity?
- Improve the space usage?

Thank you!

Thank you!

Questions?