Similarity Search in high Dimensions

Searching near and far

February 17, 2016

IT UNIVERSITY OF COPENHAGEN

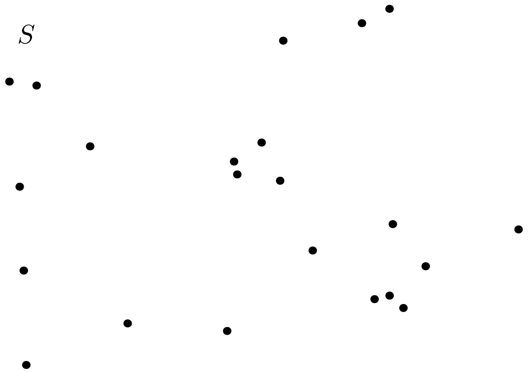
Parts

1. Locality Sensitive Hashing
2. Furthest Neighbor and the Annulus Query

Nearest Neighbor

Definition (Nearest Neighbor)

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



Nearest Neighbor

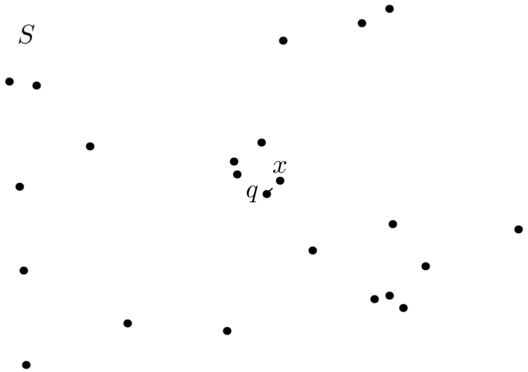
Definition (Nearest Neighbor)

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

Nearest Neighbor

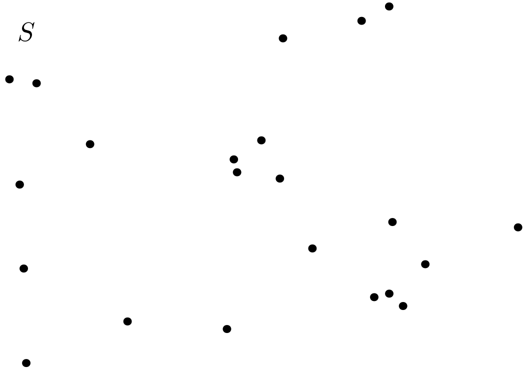
Definition (Nearest Neighbor)

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

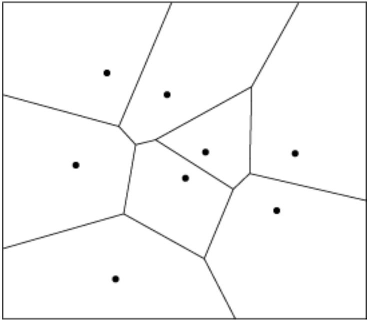


Low dimension (d=small constant)

Nearest Neighbor in low dimension



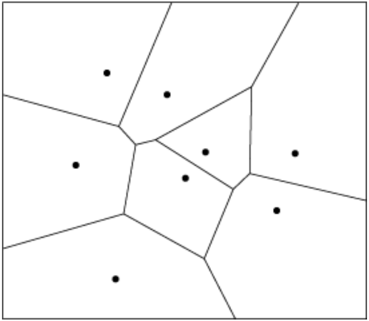
Preprocessing



Find a good partitioning H of the space.

- For $x \in S$ store $H(x)$

Preprocessing



Find a good partitioning H of the space.

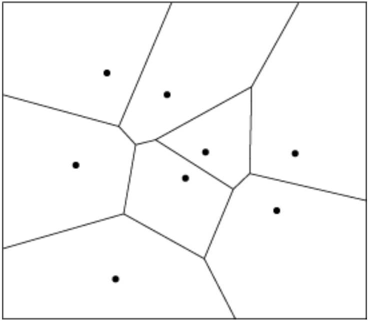
- For $x \in S$ store $H(x)$
- At query time look at $H(q)$

Perfect partitioning

Definition (Voronoi diagram)

Exact space partitioning. In each cell the cell center is the nearest neighbor.

Figure: Voronoi diagram

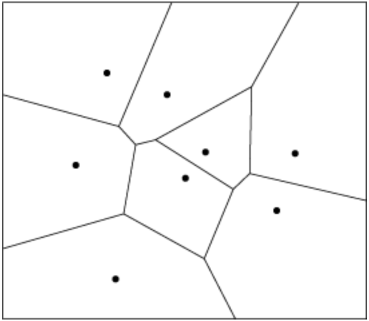


Perfect partitioning

Definition (Voronoi diagram)

Exact space partitioning. In each cell the cell center is the nearest neighbor. Space: $O(n^{\lceil \frac{1}{2} d \rceil})$

Figure: Voronoi diagram

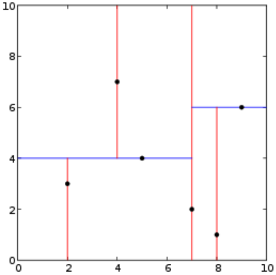


K-d Tree

Definition (K-d Tree)

[Bently '76] Partition the space, splitting each dimension down a tree. Nearest Neighbor Queries in $O(\log n)$.

Figure: 2d-Tree

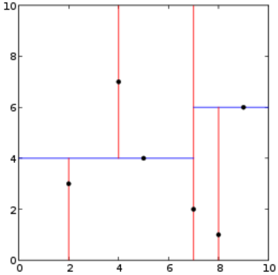


K-d Tree

Definition (K-d Tree)

[Bently '76] Partition the space, splitting each dimension down a tree. Nearest Neighbor Queries in $O(\log n)$. Query time degrades in k as $n^{1-1/d}$.

Figure: 2d-Tree



As d grows...

- Space exponential in d

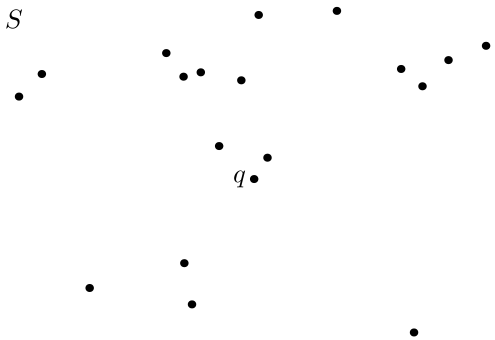
As d grows...

- Space exponential in d
- Query time linear in n

r-NN

Definition (r-Near Neighbor Decision Version)

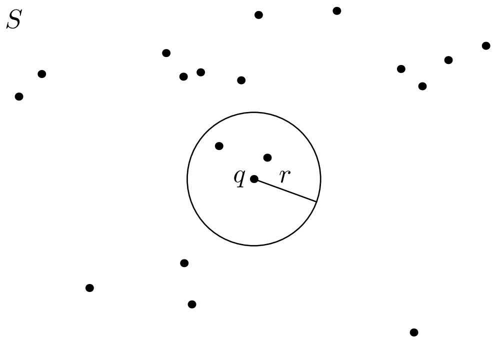
Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$ and $r > 0$ if $\exists x \in S$ such that $||x - q||_2 \leq r$ return yes, else return no.



r-NN

Definition (r-Near Neighbor Decision Version)

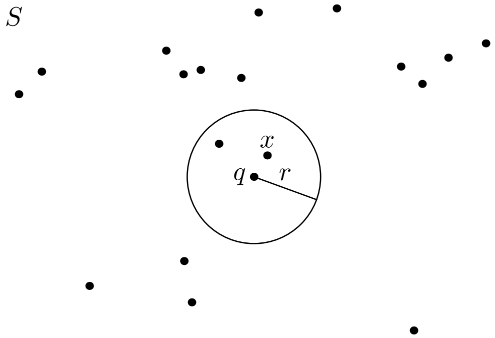
Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$ and $r > 0$ if $\exists x \in S$ such that $||x - q||_2 \leq r$ return yes, else return no.



r-NN

Definition (r-Near Neighbor Decision Version)

Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$ and $r > 0$ if $\exists x \in S$ such that $||x - q||_2 \leq r$ return yes, else return no.



SETH

Seems to be a good reason we have no sub-linear time solution, if you believe in SETH

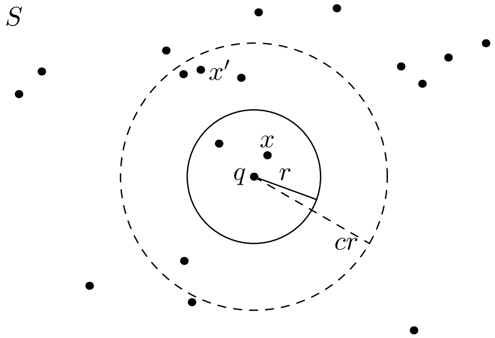
Theorem (Sublinear hamming r-NN breaks SETH)

[Williams '04],[Alam & Williams '15]
Decision r-NN in time $n^{0.99}2^{o(d)}$ implies k-SAT with n variables can be solved in time α^n where $\alpha < 2$.

Approximation version

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

Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$, $c > 1$:
If $\exists x \in S$ where $||x - q||_2 \leq r$ return some x' with $||x - q||_2 \leq cr$.



Locality Sensitive Hashing

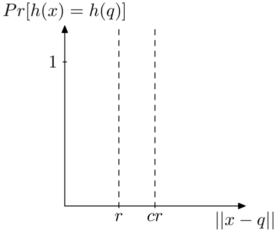
Definition (Indyk & Motwani 1998)

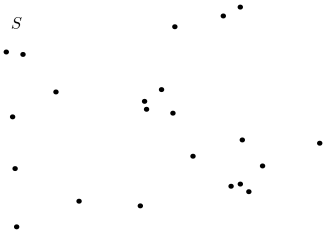
A hash-famliy $\mathcal{H}$ is called (r, cr, P_1, P_2)-sensitive LSH if

$$Pr_{\mathcal{H}}[h(q) = h(x)] \geq P_1 \text{ when } D(x, q) \leq r \tag{1}$$

$$Pr_{\mathcal{H}}[h(q) = h(x)] \leq P_2 \text{ when } D(x, q) \geq cr \tag{2}$$

And $P_1 > P_2$.





Locality Sensitive Hashing

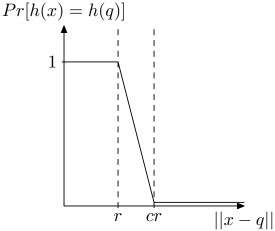
Definition (Indyk & Motwani 1998)

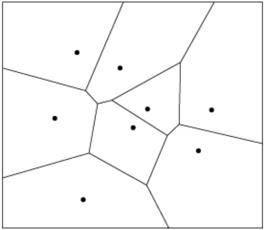
A hash-famliy $\mathcal{H}$ is called (r, cr, P_1, P_2)-sensitive LSH if

$$Pr_{\mathcal{H}}[h(q) = h(x)] \geq P_1 \text{ when } D(x,q) \leq r \tag{1}$$

$$Pr_{\mathcal{H}}[h(q) = h(x)] \leq P_2 \text{ when } D(x,q) \geq cr \tag{2}$$

And $P_1 > P_2$.





Locality Sensitive Hashing

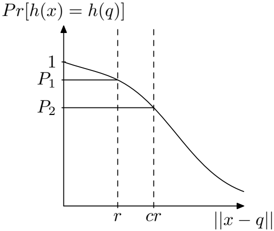
Definition (Indyk & Motwani 1998)

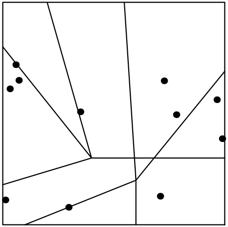
A hash-famliy $\mathcal{H}$ is called (r, cr, P_1, P_2)-sensitive LSH if

$$Pr_{\mathcal{H}}[h(q) = h(x)] \geq P_1 \text{ when } D(x, q) \leq r \tag{1}$$

$$Pr_{\mathcal{H}}[h(q) = h(x)] \leq P_2 \text{ when } D(x, q) \geq cr \tag{2}$$

And $P_1 > P_2$.





Example, Indyk, Motwani 1998

$S \subseteq \{0,1\}^d$
$S =$
010111100110101010100101010
010100010101011011101101011
111110110010101001010101010
010101010101011101100001010
000001110101011101010010101
110101001001101010101001010
010101010101011101001010100

Example, Indyk, Motwani 1998

$S =$
0101111001101010101001010100
0101000101010110111011010111
1111101100101010010101010100
0101010101010111011000010100
0000011101010111010100101011
1101010010011010101010010100
0101010101010111010010101000
$h_1(x) = x_4$, 2 buckets

Example, Indyk, Motwani 1998

$S =$
010111100110101010100101010
010100010101011011101101011
111110110010101001010101010
01010101010101011101100001010
000001110101011101010010101
110101001001101010101001010
010101010101011101001010100
$h_1(x) = x_4$, 2 buckets
$P_1 = 1 - r/d$
$P_2 = 1 - cr/d$

Example hash function

Definition (Indyk, Motwani 1998)

$P_1 = 1 - r/d$
$P_2 = 1 - cr/d$
For $c > 1$ we have $P_1 > P_2$
"Quality" measure $\rho = \frac{\log 1/P_1}{\log 1/P_2} = 1/c$.

Example, Indyk, Motwani 1998

$S =$
01011110011010101010100101010
01010001010101101110110101011
11111011001010100101010101010
01010101010101110110000101010
00000111010101110101001010101
11010100100110101010101001010
01010101010101110100101010100
$h_1(x) = x_4$, 2 buckets

Example, Indyk, Motwani 1998

$S =$
0101111001101010101001010 10
0101000101010110111011010 11
1111101100101010010101010 10
0101010101010111011000010 10
0000011101010111010100101 01
1101010010011010101010010 10
0101010101010111010010101 00
$g(x) = \{h_1(x), h_2(x), .., h_k(x)\}$, 2^k buckets
Setting k we define $\mathcal{G}$.

Example, Indyk, Motwani 1998

$S =$
010111100110101010100101010
010100010101011011101101011
111110110010101001010101010
010101010101011101100001010
000001110101011101010010101
110101001001101010101001010
010101010101011101001010100
$g(x) = \{h_1(x), h_2(x), .., h_k(x)\}$, 2^k buckets
Setting k we define $\mathcal{G}$.

- If x is a r-near. $Pr[g(x) = g(p)] \geq P_1^k$.
- Using L hash functions from $\mathcal{G}$, at least one collision with Prb. $\geq 1 - (1 - P_1^k)^L$.

r-NN with $\mathcal{H}$

Theorem (Indyk, Motwani, Gionis '99)

Set $L = n^\rho$:

r-NN with $\mathcal{H}$

Theorem (Indyk, Motwani, Gionis '99)

Set $L = n^\rho$:

1. *Hash all points with L functions from $\mathcal{G}$.*

r-NN with $\mathcal{H}$

Theorem (Indyk, Motwani, Gionis '99)

Set $L = n^\rho$:

1. *Hash all points with L functions from $\mathcal{G}$.*
2. *Query: Find buckets from $g_1(q), .., g_L(q)$ in $O(n^\rho)$.*

r-NN with $\mathcal{H}$

Theorem (Indyk, Motwani, Gionis '99)

Set $L = n^{\rho}$:

1. *Hash all points with L functions from $\mathcal{G}$.*
2. *Query: Find buckets from $g_1(q), .., g_L(q)$ in $O(n^{\rho})$.*
3. *Look at up to $3L$ points from those buckets in $O(dn^{\rho})$.*

Given $\mathcal{H}$ this solves $(c, r) - NN$ in $O\left(dn^{\rho}\right)$ time with error $\lambda < 1$.

Developments

Smaller $\rho \Rightarrow$ larger $P1/P2$ gap.

Reference	ρ	Comment
Linear search	1	
Indyk & Motwani STOC 1998	$\frac{1}{c}$	Worst case upper bound
O'Donnell, Wu & Zhou ITCS 2010	$\frac{1}{c}$	Worst case, lower bound

Developments

Smaller $\rho \Rightarrow$ larger $P1/P2$ gap.

Reference	ρ	Comment
Linear search	1	
Indyk & Motwani STOC 1998	$\frac{1}{c}$	Worst case upper bound
O'Donnell, Wu & Zhou ITCS 2010	$\frac{1}{c}$	Worst case, lower bound
Dubiner Trans. Inf. Theory 2010	$\frac{1}{2c-1}$	Assuming random data, upper bound

Developments

Smaller $\rho \Rightarrow$ larger $P1/P2$ gap.

Reference	ρ	**Comment**
Linear search	1	
Indyk & Motwani STOC 1998	$\frac{1}{c}$	Worst case upper bound
O'Donnell, Wu & Zhou ITCS 2010	$\frac{1}{c}$	Worst case, lower bound
Dubiner Trans. Inf. Theory 2010	$\frac{1}{2c-1}$	Assuming random data, upper bound
Andoni & Razenshteyn STOC 2015	$\frac{1}{2c-1}$	Data Dependent, Upper and lower bound

Developments

Smaller $\rho \Rightarrow$ larger $P1/P2$ gap.

Reference	ρ	Comment
Linear search	1	
Indyk & Motwani STOC 1998	$\frac{1}{c}$	Worst case upper bound
O'Donnell, Wu & Zhou ITCS 2010	$\frac{1}{c}$	Worst case, lower bound
Dubiner Trans. Inf. Theory 2010	$\frac{1}{2c-1}$	Assuming random data, upper bound
Andoni & Razenshteyn STOC 2015	$\frac{1}{2c-1}$	Data Dependent, Upper and lower bound
Kapralov PODS 2015	$\frac{4}{c+1}$	Linear space upper bound

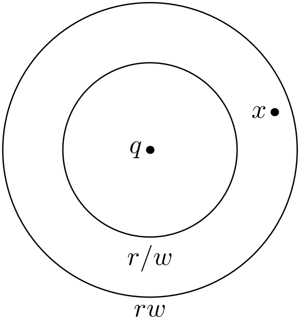
Part 2

- Annulus Query
- Furthest Neighbor

Annulus query

Definition

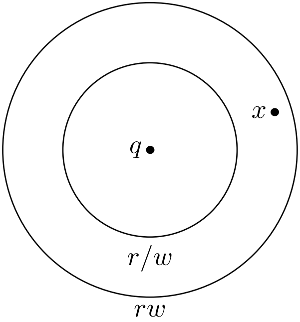
Given $S \subseteq \mathbb{R}^d$ a query point q and parameters $r, w > 1$ return x such that $\frac{r}{w} \leq ||x - q||_2 \leq wr$.



Annulus query

Definition

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



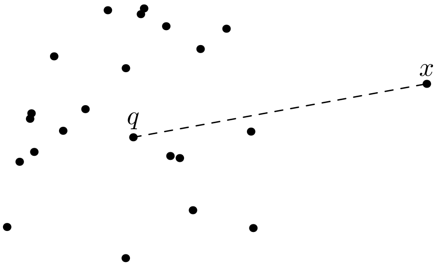
Applications in recommender systems.

The Furthest Neighbor Problem

Joint work with Rasmus Pagh, Matthew Skala and Francesco Silvestri, SISAP '15.

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]

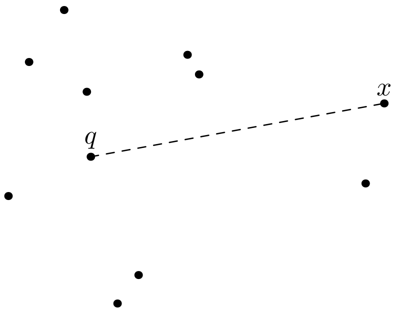
Sublinear Furthest Neighbor

- 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

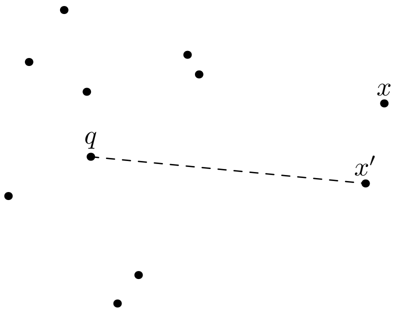
Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$, let x be the furthest neighbor. Return x' such than $||x' - q|| \geq \frac{||x-q||}{c}$. We call this c-FN.



Approximate Furthest Neighbor

Definition

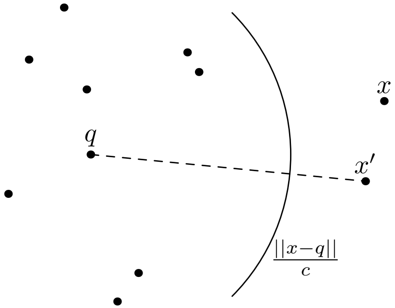
Let $S \subseteq \mathbb{R}^d$. Given some $q \in \mathbb{R}^d$, let x be the furthest neighbor. Return x' such than $||x' - q|| \geq \frac{||x-q||}{c}$. We call this c-FN.



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
1.	Bespamyatnikh '96	$c > 1$	$O((1 + \frac{1}{c-1})^{d-1})$

1. Split tree

Related work

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

	Paper	c	Query Time
1.	Bespamyatnikh '96	$c > 1$	$O((1 + \frac{1}{c-1})^{d-1})$
2.	Goel et al. '01	$c > \sqrt{2}$	$O(d^2)$

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

1. Split tree
2. Minimum enclosing ball
3. Random Projections, binary search

Related work

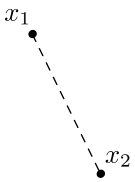
Though not nearly as popular as Nearest Neighbor there is notable prior work on Furthest neighbor.

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

1. Split tree
2. Minimum enclosing ball
3. Random Projections, binary search
4. Random Projections, single query

Random Projections

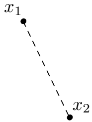
Fix two points in $\mathbb{R}^d$.



Random Projections

Lemma (Distance preservation)

$$a_i \cdot (x_1 - x_2) \sim \mathcal{N}(0,1)||x_1 - x_2||_2 \tag{3}$$

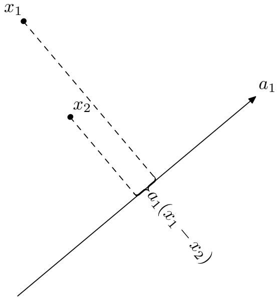


$$a_i = \{g_1, g_2, .., g_d\} \text{ where } g_j \sim \mathcal{N}(0,1)$$

Random Projections

Lemma (Distance preservation)

$$a_i \cdot (x_1 - x_2) \sim \mathcal{N}(0,1)||x_1 - x_2||_2 \tag{3}$$

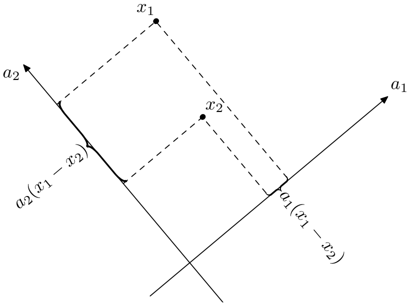


$$a_i = \{g_1, g_2, .., g_d\} \text{ where } g_j \sim \mathcal{N}(0,1)$$

Random Projections

Lemma (Distance preservation)

$$a_i \cdot (x_1 - x_2) \sim \mathcal{N}(0,1)||x_1 - x_2||_2 \quad (3)$$



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

Preprocessing

We would like to examine the points with the largest $|a_i \cdot (x - q)|$.

Preprocessing

We would like to examine the points with the largest $a_i \cdot (x - q)$.

- $a_i \cdot (x - q)$ not know till query time

Preprocessing

We would like to examine the points with the largest $a_i \cdot (x - q)$.

- $a_i \cdot (x - q)$ not know till query time
- $a_i \cdot x \geq a_i \cdot y \Rightarrow a_i(x - q) \geq a_i(y - q)$

Preprocessing

We would like to examine the points with the largest $a_i \cdot (x - q)$.

- $a_i \cdot (x - q)$ not know till query time
- $a_i \cdot x \geq a_i \cdot y \Rightarrow a_i(x - q) \geq a_i(y - q)$
- $\forall x \in S$ we can preprocess $a_i \cdot x$

Preprocessing

We would like to examine the points with the largest $a_i \cdot (x - q)$.

- $a_i \cdot (x - q)$ not know till query time
- $a_i \cdot x \geq a_i \cdot y \Rightarrow a_i(x - q) \geq a_i(y - q)$
- $\forall x \in S$ we can preprocess $a_i \cdot x$

Algorithm:

1. $\forall a_i$: store all x in order of $a_i \cdot x$

Preprocessing

We would like to examine the points with the largest $a_i \cdot (x - q)$.

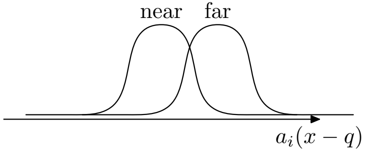
- $a_i \cdot (x - q)$ not know till query time
- $a_i \cdot x \geq a_i \cdot y \Rightarrow a_i(x - q) \geq a_i(y - q)$
- $\forall x \in S$ we can preprocess $a_i \cdot x$

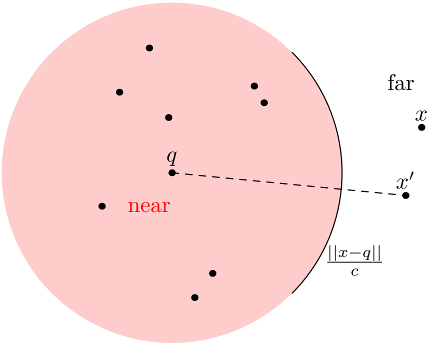
Algorithm:

1. $\forall a_i$: store all x in order of $a_i \cdot x$
2. Query: Examine all points with $a_i \cdot (x - q)$ larger than some threshold.

Finding the threshold

$X = a_i \cdot (x - q)$ is a normal distributed variable:



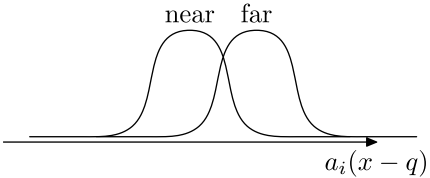


Bounding the Gaussian

Lemma (See Lemma 7.4 in Karger, Motwani, and Sudan '98)

For every $t > 0$, if $X \sim N(0,1)$ then

$$\frac{1}{\sqrt{2\pi}} \cdot \left(\frac{1}{t} - \frac{1}{t^3}\right) \cdot e^{-t^2/2} \le \Pr[X \ge t] \le \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{t} \cdot e^{-t^2/2}$$

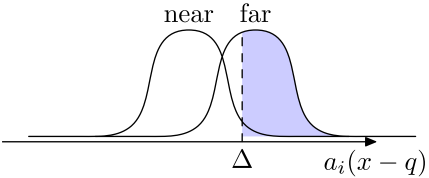


Finding the threshold

Lemma (See Lemma 7.4 in Karger, Motwani, and Sudan '98)

For every $t > 0$, if $X \sim N(0,1)$ then

$$\frac{1}{\sqrt{2\pi}} \cdot \left(\frac{1}{t} - \frac{1}{t^3}\right) \cdot e^{-t^2/2} \leq \Pr[X \geq t] \leq \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{t} \cdot e^{-t^2/2}$$



We want $O(n^{1/c^2})$
Set t so $\Pr[X > t] \geq (1 - o(1))\frac{1}{n^{1/c^2}}$.

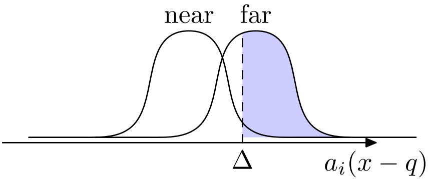
Crossing the threshold

Lemma (Threshold projection)

$\Delta = rt/c$

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

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



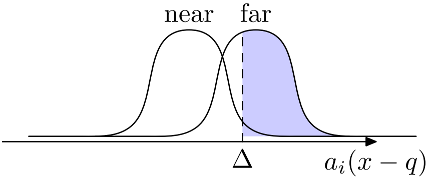
Crossing the threshold

Lemma (Threshold projection)

$\Delta = rt/c$

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

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



What is r?

Two ways to fail

1. No "far" points have any $a_i \cdot (x - q) \geq \Delta$. We will cross the threshold without seeing a valid candidate.

Two ways to fail

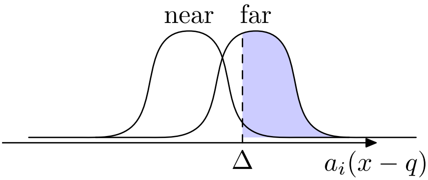
1. No "far" points have any $a_i \cdot (x - q) \geq \Delta$. We will cross the threshold without seeing a valid candidate.
2. Too many "close" points have $a \cdot (x - q) \geq \Delta$. We will not cross the threshold

Bounding the error

Corollary (Failure probability)

Using ℓ random projections.

$$\Pr[\textit{Missing the far point}] \leq (1 - 1/n^{1/c^2})^{\ell}$$

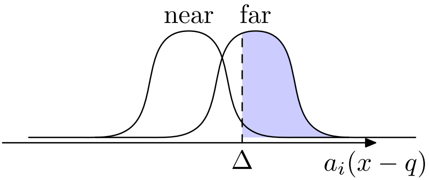


Bounding the error

Corollary (Failure probability)

Using $\ell = 2n^{1/c^2}$ *random projections.*

$$\Pr[\textit{Missing the far point}] \leq (1 - 1/n^{1/c^2})^{\ell} \leq 1/e^2$$



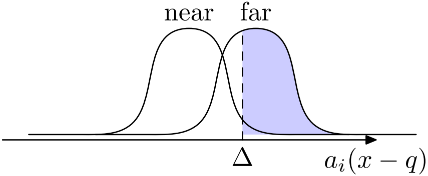
Bounding the error

Corollary (Failure probability)

Using $\ell = 2n^{1/c^2}$ *random projections.*

$$\Pr[\textit{Missing the far point}] \leq (1 - 1/n^{1/c^2})^\ell \leq 1/e^2$$

$$\Pr[\textit{Too many close points}] \leq Pr[\ell \log^{c^2/2-1/3} n > m]$$



Bounding the error

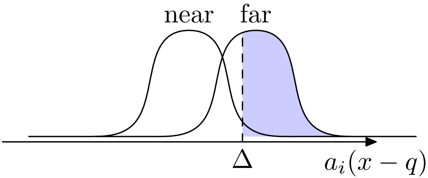
Corollary (Failure probability)

Using $\ell = 2n^{1/c^2}$ *random projections.*

$$\Pr[\textit{Missing the far point}] \leq (1 - 1/n^{1/c^2})^\ell \leq 1/e^2$$

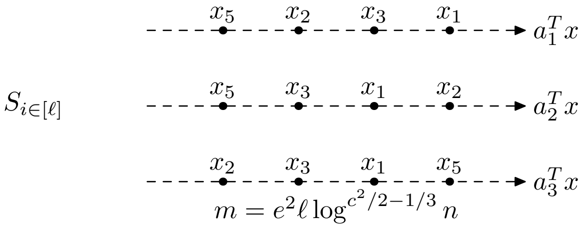
$$\Pr[\textit{Too many close points}] \leq Pr[\ell \log^{c^2/2-1/3} n > m] \leq 1/e^2$$

For $m = e^2 \ell \log^{c^2/2-1/3} n$.
The combined failure probability is $\leq \frac{2}{e^2}$.



Data structure

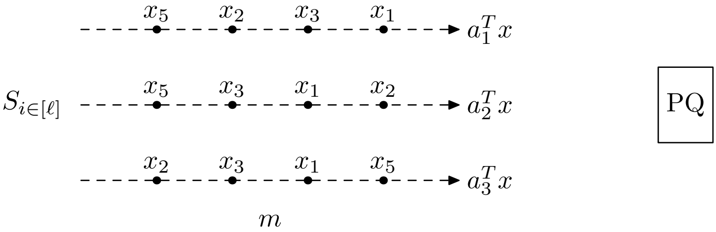
Points are stored in their projection order. We use $\ell = 2n^{1/c^2}$ projections and store the top m points in each.



$O(\ell m d) \approx dn^{2/c^2}$ space.(With some large constants).

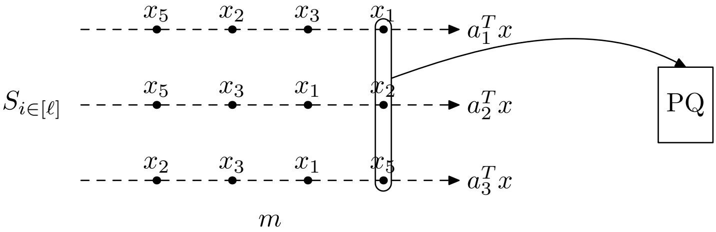
Query procedure

- Create an empty priority queue PQ.



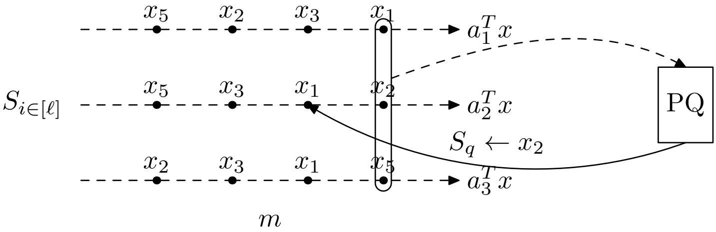
Query procedure

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



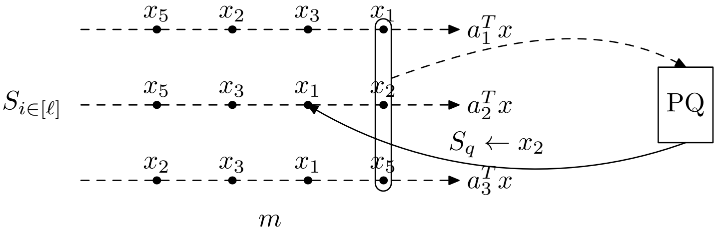
Query procedure

- 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.
- Take its neighbor on the projection it came from, add it to PQ.



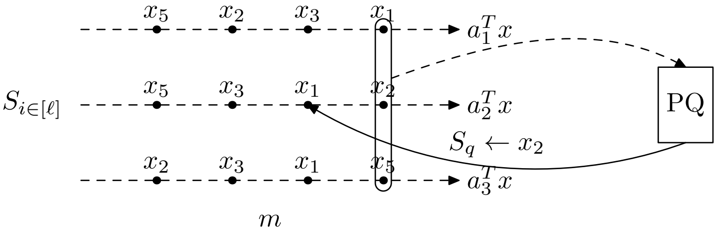
Query procedure

- Create an empty priority queue PQ.
- Add the $\ell = 2n^{1/c^2}$ points .
- Points are added with priority $a_i \cdot x - a_i \cdot q$.
- Take out the top priority element and examine its distance to q.
- Take its neighbor on the projection it came from, add it to PQ.
- We will look at at $m = e^2 \ell \log^{c^2/2-1/3} n$ 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.
- Take its neighbor on the projection it came from, add it to PQ.
- We will look at at $m = e^2 \ell \log^{c^2/2-1/3} n$ points.
- Time $O(\ell + m(d + \log \ell))$ vs. $O(\ell md)$.



Approximate Furthest Neighbor

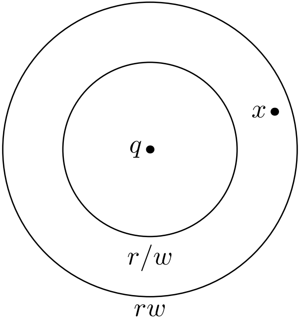
Theorem (Approximate Furthest Neighbor)

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

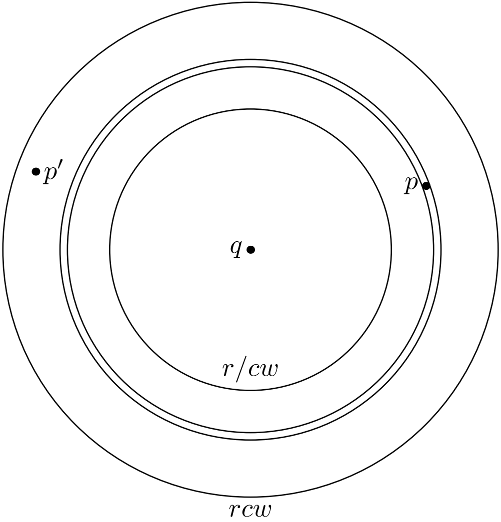
- *Queries take $\widetilde{\mathcal{O}}(n^{1/c^2} d)$ time.*
- *The data structure uses $\widetilde{\mathcal{O}}(n^{2/c^2} d)$ space.*

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

Annulus query



Approximate Annulus query



Annulus Query Datastructure

Natural combination of random projections and LSH
$\forall x \in S$

- $\forall i \in [\ell]$ calculate $p_i(x)x = a_i \cdot x$

Annulus Query Datastructure

Natural combination of random projections and LSH

$\forall x \in S$

- $\forall i \in [\ell]$ calculate $p_i(x)x = a_i \cdot x$
- $\forall l \in [L]$ calculate $g_l(x)$

Annulus Query Datastructure

Natural combination of random projections and LSH
$\forall x \in S$

- $\forall i \in [\ell]$ calculate $p_i(x)x = a_i \cdot x$
- $\forall l \in [L]$ calculate $g_l(x)$
- Store non-empty hash-buckets in order of p_i with PQ query structure.

Annulus Query Datastructure

Natural combination of random projections and LSH

$\forall x \in S$

- $\forall i \in [\ell]$ calculate $p_i(x)x = a_i \cdot x$
- $\forall l \in [L]$ calculate $g_l(x)$
- Store non-empty hash-buckets in order of p_i with PQ query structure.
- Query by PQ design. Early termination now possible

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.*

Failure probability $\lambda < 1$.

Open problems

- Expand the random projection technique to general metric space?

Open problems

- Expand the random projection technique to general metric space?
- Use furthest neighbor to improve LSH output sensitivity?

Open problems

- Expand the random projection technique to general metric space?
- Use furthest neighbor to improve LSH output sensitivity?
- Direct Annulus Query hash functions?

Thank you!

Thank you!

Questions?