Nearest Neighbor Search in high-dimensional spaces

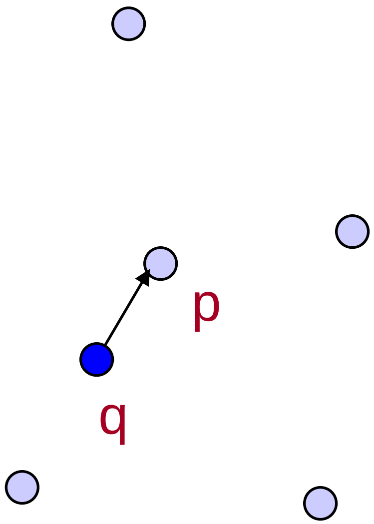
Alexandr Andoni
(Princeton/CCI $\rightarrow$ MSR SVC)

Barriers II

August 30, 2010

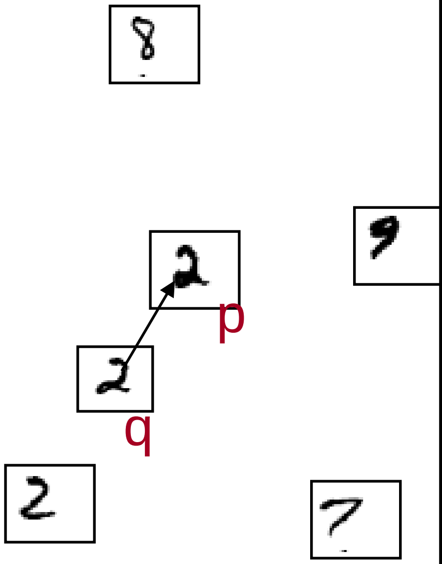
Nearest Neighbor Search (NNS)

- Preprocess: a set D of points in R^d
- Query: given a new point q, report a point $p \in D$ with the smallest distance to q



Motivation

- Generic setup:
  - Points model *objects (e.g. images)*
  - Distance models *(dis)similarity measure*
- Application areas:
  - machine learning, data mining, speech recognition, image/video/music clustering, bioinformatics, etc…
- Distance can be:
  - Euclidean, Hamming, ℓ_∞, edit distance, Ulam, Earth-mover distance, etc…
- Primitive for other problems:
  - find the closest pair in a set D, MST, clustering…

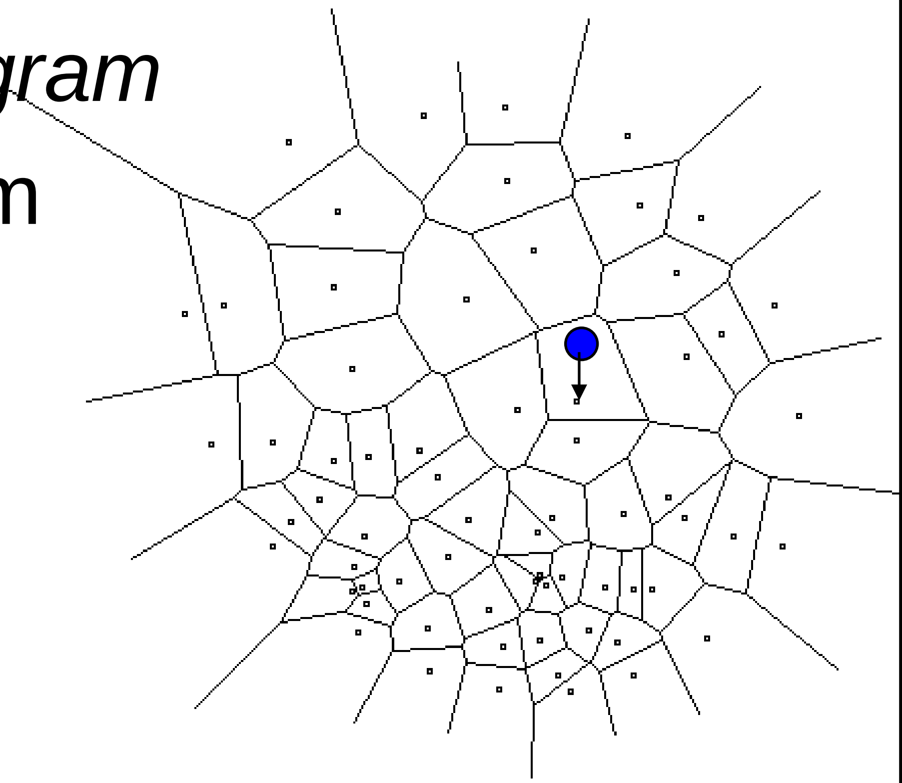


Plan for today

1. NNS for basic distances
2. NNS for advanced distances: embeddings
3. NNS via product spaces

2D case

- Compute *Voronoi diagram*
- Given query q, perform *point location*
- Performance:
  - Space: O(n)
  - Query time: O(log n)



High-dimensional case

- All exact algorithms degrade rapidly with the dimension d

Algorithm	Query time	Space
Full indexing	$O(d*\log n)$	$n^{O(d)}$ (Voronoi diagram size)
No indexing – linear scan	$O(dn)$	$O(dn)$

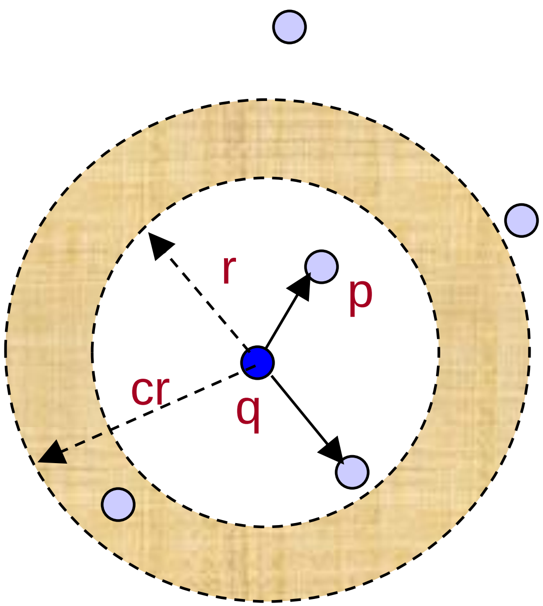
- When d is high, state-of-the-art is unsatisfactory:
  - Even in practice, query time tends to be linear in n

Approximate NNS

c-approximate

- r-near neighbor: given a new point q, report a point p∈D s.t. $||p-q|| \le \boxed{cr}$

  as long as there exists a point at distance ≤r



Approximation Algorithms for NNS

- A vast literature:
  - With exp(d) space or Ω(n) time:

    [Arya-Mount-et al], [Kleinberg'97], [Har-Peled'02],…
  - With poly(n) space and o(n) time:

    [Kushilevitz-Ostrovsky-Rabani'98], [Indyk-Motwani'98], [Indyk'98, '01], [Gionis-Indyk-Motwani'99], [Charikar'02], [Datar-Immorlica-Indyk-Mirrokni'04], [Chakrabarti-Regev'04], [Panigrahy'06], [Ailon-Chazelle'06], [A-Indyk'06]…

The landscape: algorithms

Space: poly(n).
Query: logarithmic

Space	Time	Comment	Reference
$n^{4/\varepsilon^2}+nd$	$O(d*\log n)$	$c=1+\varepsilon$	[KOR'98, IM'98]

Space: small poly (close to linear).
Query: poly (sublinear).

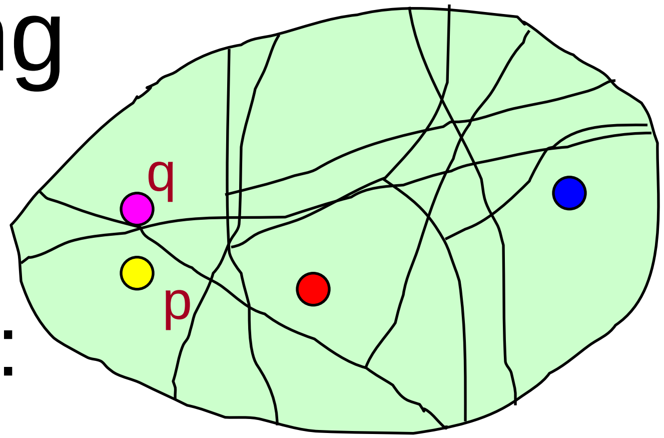
$n^{1+\rho}+nd$	dn^{ρ}	$\rho\approx 1/c$	[IM'98, Cha'02, DIIM'04]
		$\rho=1/c^2+o(1)$	[AI'06]

Space: near-linear.
Query: poly (sublinear).

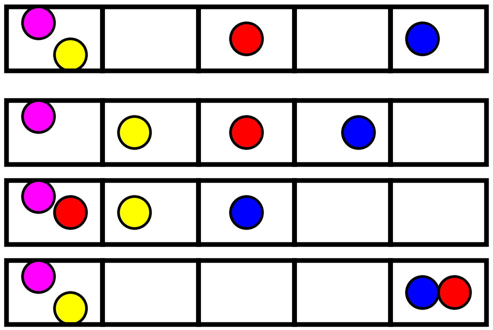
$nd*\log n$	dn^{ρ}	$\rho=2.09/c$	[Ind'01, Pan'06]
		$\rho=O(1/c^2)$	[AI'06]

Locality-Sensitive Hashing

[Indyk-Motwani'98]

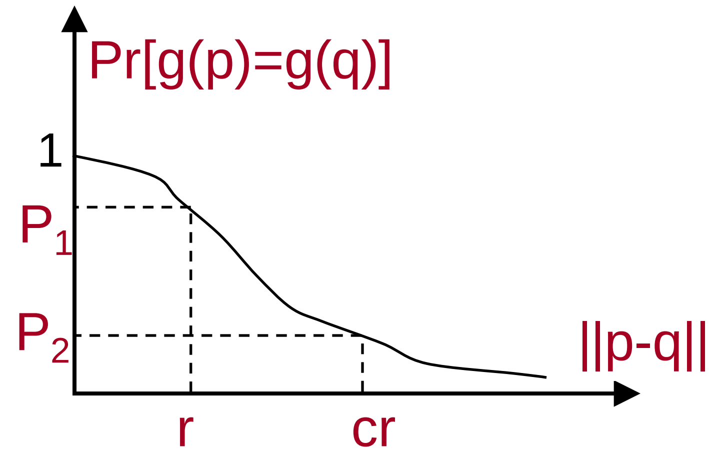


- Random hash function g: $\mathbb{R}^d \to \mathbb{Z}$ s.t. for any points p,q:
  - If $\|p-q\| \leq r$, then $\Pr[g(p)=g(q)]$ is ~~“high”~~ “not-so-small”
  - If $\|p-q\| > cr$, then $\Pr[g(p)=g(q)]$ is “small”



- Use several hash tables: n^ρ, where ρ s.t.

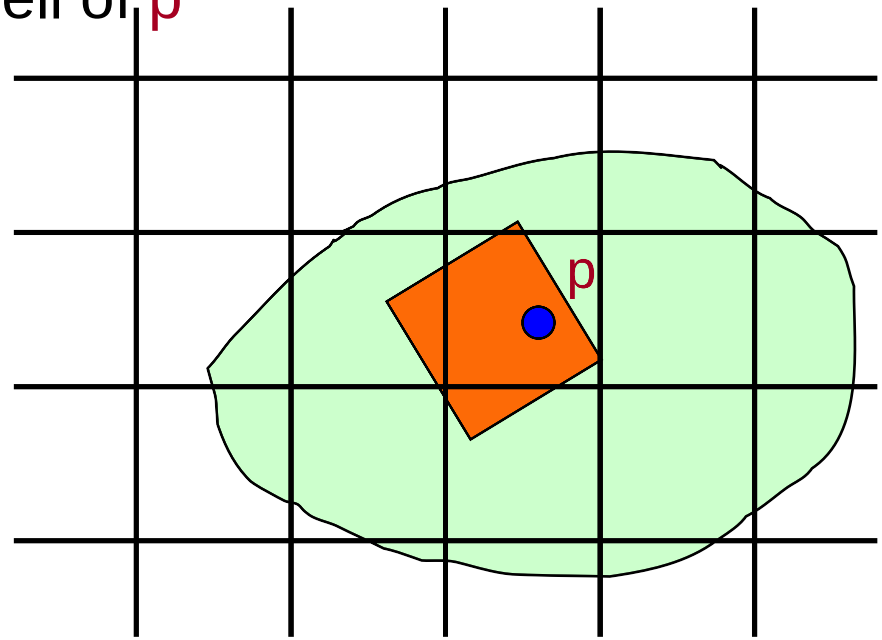
$$\rho = \frac{\log 1/P_1}{\log 1/P_2}$$



Example of hash functions: grids

[Datar-Immorlica-Indyk-Mirrokni'04]

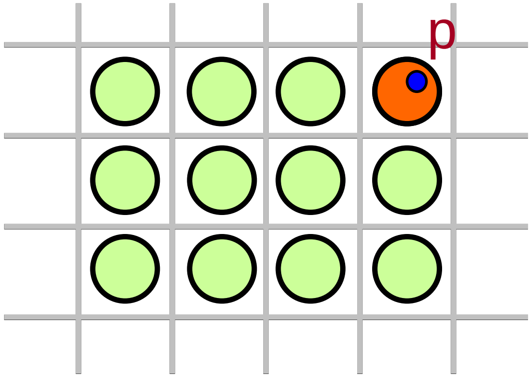
- Pick a regular grid:
  - Shift and rotate randomly
- Hash function:
  - $g(p)$ = index of the cell of p
- Gives $\rho \approx 1/c$



State-of-the-art LSH

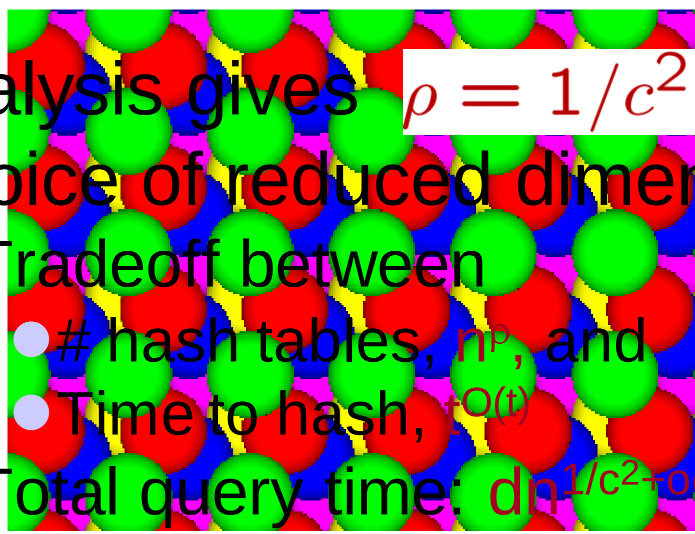
[A-Indyk'06]

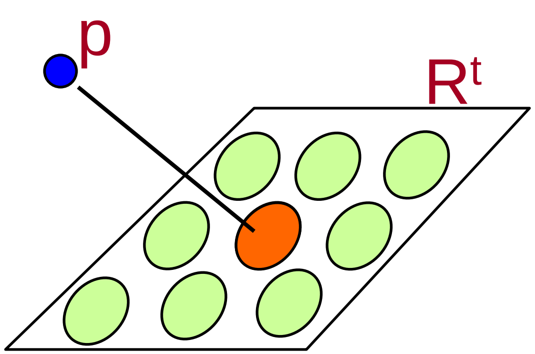
- Regular grid → grid of balls
  - p can hit empty space, so take more such grids until p is in a ball
- Need (too) many grids of balls
  - Start by reducing dimension to t



- Analysis gives $\rho = 1/c^2 + o_t(1)$
- Choice of reduced dimension t?
  - Tradeoff between
    - # hash tables, n^ρ, and
    - Time to hash, $t^{O(t)}$
  - Total query time: $dn^{1/c^2+o(1)}$

2D





Proof idea

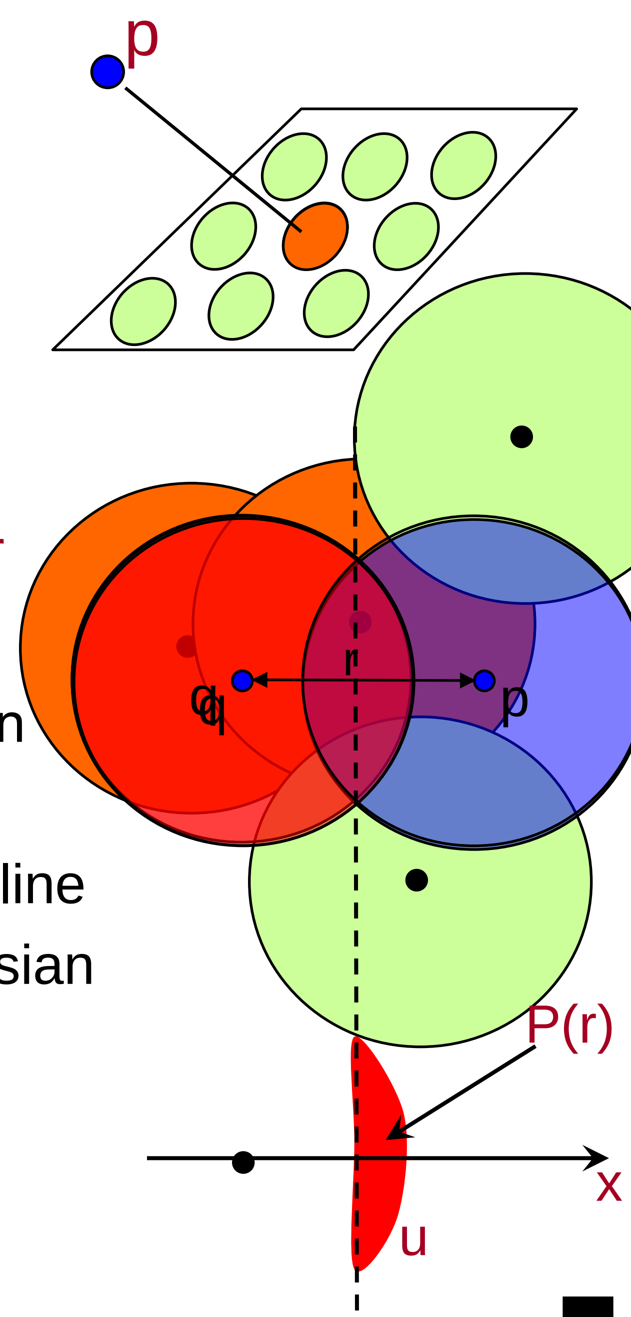
- Claim: $\rho \approx 1/c^2$, where

$$\rho = \frac{\log 1/P(r)}{\log 1/P(cr)}$$

  - P(r)=probability of collision when ||p-q||=r
- Intuitive proof:
  - Let's ignore effects of reducing dimension
  - P(r) = intersection / union
  - P(r)≈random point u beyond the dashed line
  - The x-coordinate of u has a nearly Gaussian distribution

    $\rightarrow$ P(r) ≈ exp(-A·r^2)

$$\rho = \frac{A{\cdot}r^2}{A{\cdot}(cr)^2} = \frac{1}{c^2}$$



∎

The landscape: lower bounds

Space: poly(n).
Query: logarithmic

Space	Time	Comment	Reference
n^{4/ε^2}+nd	O(d*log n)	c=1+ε	[KOR'98, IM'98]
$n^{o(1/\varepsilon^2)}$	ω(1) memory lookups		[AIP'06]

Space: small poly
(close to linear).
Query: poly
(sublinear).

Space	Time	Comment	Reference
$n^{1+\rho}$+nd	dn^{ρ}	$\rho \approx 1/c$	[IM'98, Cha'02, DIIM'04]
		$\rho = 1/c^2 + o(1)$	[AI'06]
		$\rho \geq 1/c^2$	[MNP'06, OWZ'10]
$n^{1+o(1/c^2)}$	ω(1) memory lookups		[PTW'08, PTW'10]

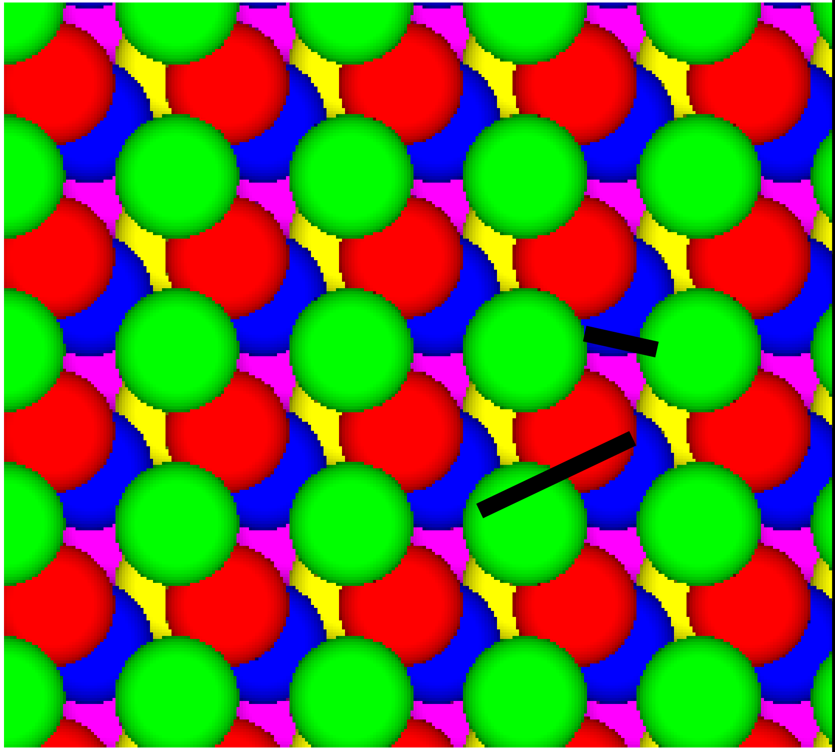
Space: near-linear.
Query: poly
(sublinear).

Space	Time	Comment	Reference
nd*logn	dn^{ρ}	$\rho = 2.09/c$	[Ind'01, Pan'06]
		$\rho = O(1/c^2)$	[AI'06]

Challenge 1:

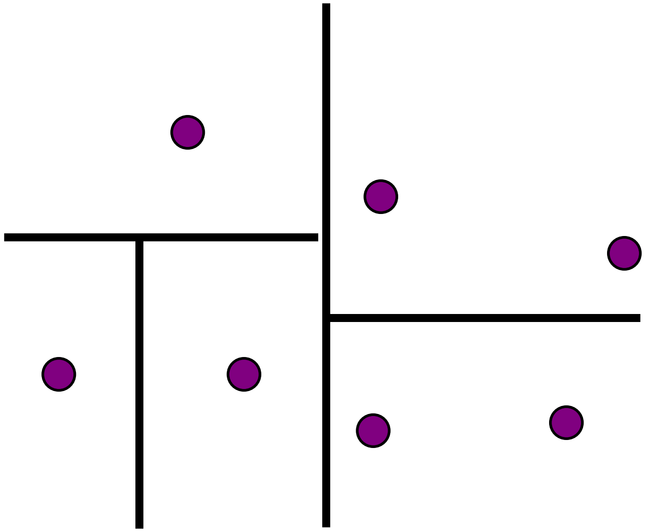
- Design space partitioning of $\mathbb{R}^t$ that is
  - efficient: point location in poly(t) time
  - qualitative: regions are “sphere-like”

$$[\text{Prob. needle of length } 1 \text{ is cut}]^{c^2}$$
$$\geq$$
$$[\text{Prob needle of length } c \text{ is cut}]$$



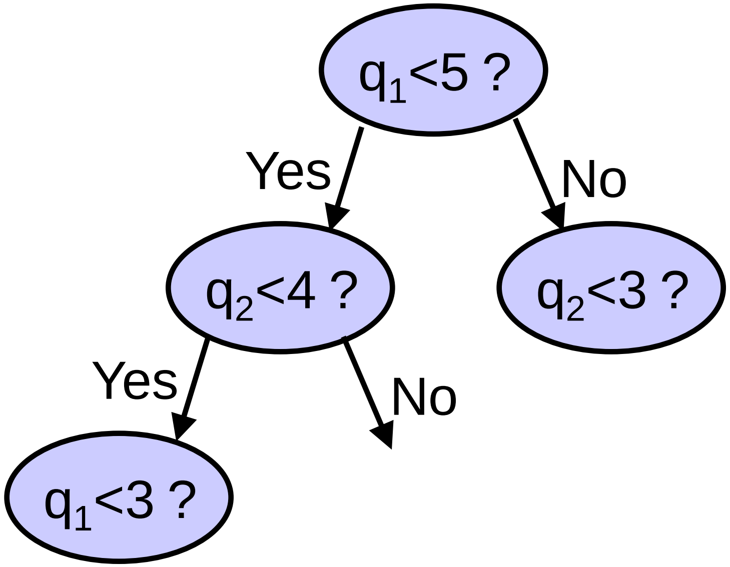
NNS for ℓ_∞ distance

[Indyk'98]



- Thm: for $\rho>0$, NNS for ℓ_∞^d with
  - $O(d * \log n)$ query time
  - $n^{1+\rho}$ space
  - $O(\lg_{1+\rho} \lg d)$ approximation
- The approach:
  - A deterministic decision tree
    - Similar to kd-trees
  - Each node of DT is "$q_i < t$"
  - One difference: algorithms goes down the tree *once*

    (while tracking the list of possible neighbors)
- [ACP'08]: optimal for decision trees!



Plan for today

1. NNS for basic distances
2. NNS for advanced distances: embeddings
3. NNS via product spaces

What do we have?

- Classical ℓ_p distances:
  - Hamming, Euclidean, ℓ_∞

How about other distances, like edit distance?

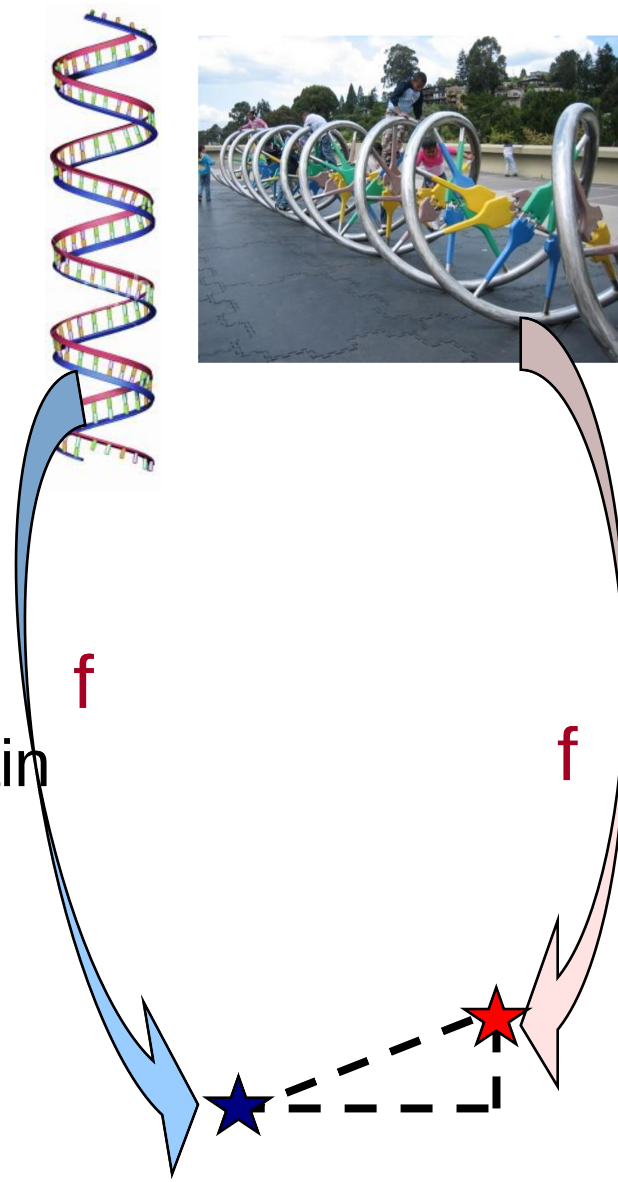
ed(x,y) = number of substitutions/insertions/ deletions to transform string x into y

Hammi…				…]
				…08'10]
Euclidean (ℓ_2)			$\rho \approx 1/c^2$	[AI'06]
			$\rho \geq 1/c^2$	[MNP06,OZW10,PTW'08'10]
ℓ_∞	$n^{1+\rho}$	$O(d \log n)$	$c \approx \log_\rho \log d$	[I'98]
	optimal for decision trees			[ACP'08]

NNS via Embeddings

- An *embedding* of M into a host metric (H,d_H) is a map $f : M \to H$
  - has distortion $A \geq 1$ if $\forall\ x,y \in \mathbf{M}$

    $$d_M(x,y) \leq d_H(f(x),f(y)) \leq A^* d_M(x,y)$$
- Why?
  - If H is Euclidean space, then obtain NNS for the original space M !
- Popular host: $H = \ell_1$



Embeddings of various metrics

- Embeddings into ℓ_1

Metric	Upper bound
Edit distance over $\{0,1\}^d$	$2^{\tilde{O}(\sqrt{\log d})}$ [OR05]
Ulam (edit permutatio	
Block edit d	
Earth-move (s-sized sets in 2D plane)	[Cha02, IT03]
Earth-mover distance (s-sized sets in $\{0,1\}^d$)	O(log s*log d) [AIK08]

Challenge 2:

Improve the distortion of embedding edit distance into ℓ_1

OK, but where's the barrier?

A barrier: ℓ_1 non-embeddability

- Embeddings into ℓ_1

Metric	Upper bound	Lower bound
Edit distance over $\{0,1\}^d$	$2^{\tilde{O}(\sqrt{\log d})}$ [OR05]	$\Omega(\log d)$ [KN05,KR06]
Ulam (edit distance between permutations)	$O(\log d)$ [CK06]	$\tilde{\Omega}(\log d)$ [AK07]
Block edit distance	$\tilde{O}(\log d)$ [MS00, CM07]	4/3 [Cor03]
Earth-mover distance (s-sized sets in 2D plane)	$O(\log s)$ [Cha02, IT03]	$\Omega(\log^{1/2} s)$ [NS07]
Earth-mover distance (s-sized sets in $\{0,1\}^d$)	$O(\log s^* \log d)$ [AIK08]	$\Omega(\log s)$ [KN05]

Other good host spaces?

- What is "good":
  - is algorithmically tractable
  - is rich (can embed into it)

$(\ell_2)^2$, etc	ℓ_∞
✓	✗
✗	✓

$(\ell_2)^2$=real space with distance: $\|x-y\|_2^2$

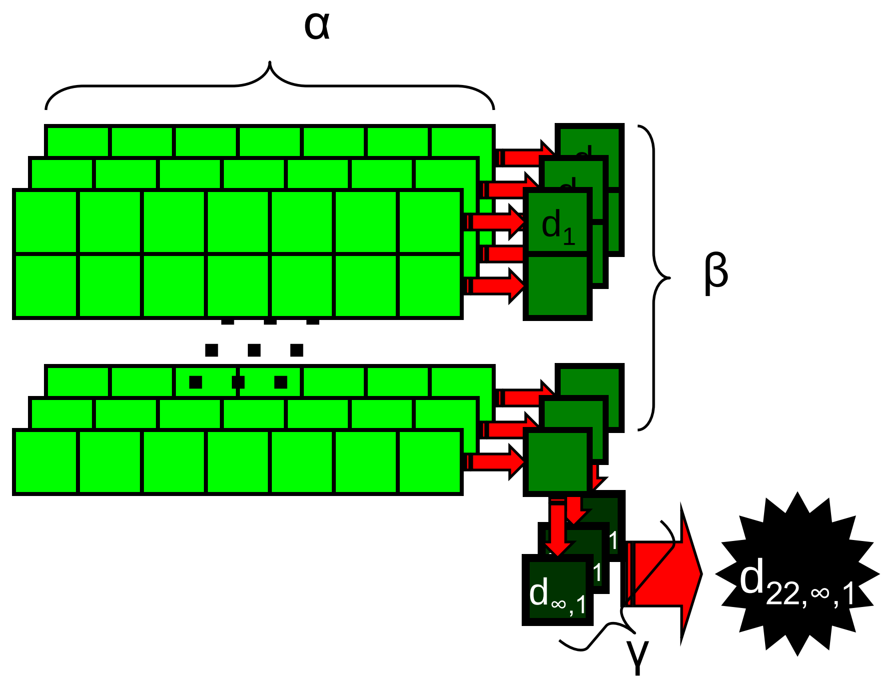
Metric	Lower bound into $\cancel{\ell_1}$	$(\ell_2)^2$, host with v. good LSH (sketching l.b. via communication complexity)
Edit distance over $\{0,1\}^d$	$\tilde{\Omega}(\log d)$ [KN05,KR06]	[AK'07]
Ulam (edit distance between orderings)	$\tilde{\Omega}(\log d)$ [AK07]	[AK'07]
Earth-mover distance (s-sized sets in $\{0,1\}^d$)	$\Omega(\log s)$ [KN05]	[AIK'08]

Plan for today

1. NNS for basic distances
2. NNS for advanced distances: embeddings
3. NNS via product spaces

Meet a new host

Iterated product space, $\mathcal{P}_{22,\infty,1}$=



$$\bigoplus^{\gamma}_{(\ell_2)^2} \bigoplus^{\beta}_{\ell_\infty} \ell_1^\alpha$$

$x = (x_1, \ldots x_a)$ 2 R $^\alpha$
$d_1(x, y) = \sum_{i=1}^{\alpha}$ jx_i ¡ y_ij

$x = (x_1, \ldots x_\beta)$ 2 ℓ_1^α £ ℓ_1^α £ $\ldots \ell_1^\alpha$
$d_{\infty,1}(x, y) = \max_{i=1}^{\beta} d_1(x_i, y_i)$

$x = (x_1, \ldots x_\gamma)$ 2 $\bigoplus^{\beta}_{\ell_\infty} \ell_1^\alpha$ £ $\bigoplus^{\beta}_{\ell_\infty} \ell_1^\alpha$ £ $\ldots \bigoplus^{\beta}_{\ell_\infty} \ell_1^\alpha$
$d_{22,\infty,1}(x, y) = \sum_{i=1}^{\gamma} (d_{\infty,1}(x_i, y_i))^2$

Why $\bigoplus^{\gamma}_{(\ell_2)^2} \bigoplus^{\beta}_{\ell_\infty} \ell_1^{\alpha}$?

[A-Indyk-Krauthgamer'09, Indyk'02]

edit distance between permutations

- Because we can…
- Embedding: …embed Ulam into $\mathcal{P}_{22,\infty,1}$ with constant distortion
    - (small dimensions)
- NNS: Any t-iterated product space has NNS on n points with
    - $(\lg \lg n)^{O(t)}$ approximation
    - near-linear space and sublinear time
- Corollary: NNS for Ulam with $O(\lg \lg n)^2$ approximation
    - cf. each ℓ_p part has logarithmic lower bound!

Embedding into $\bigoplus_{(\ell_2)^2}^{\gamma} \bigoplus_{\ell_\infty}^{\beta} \ell_1^{\alpha}$

- Theorem: Can embed Ulam metric into $\mathcal{P}_{22,\infty,1}$ with constant distortion
  - Dimensions: $\alpha=\beta=\gamma=d$
- Proof intuition
  - Characterize Ulam distance “nicely”:
    - “Ulam distance between x and y equals the number of characters that satisfy a simple property”
  - “Geometrize” this characterization

Ulam: a characterization

- Lemma: Ulam(x,y) approximately equals the number characters a satisfying:
  - there exists K≥1 (prefix-length) s.t.
  - the set of K characters preceding a in x differs much from

    the set of K characters preceding a in y

E.g., a=5; K=4

$$x = \overbrace{1234}^{X[5;4]}56789$$

$$y = 1234\underbrace{6789}_{Y[5;4]}5$$

$$|X[a;K] \Delta Y[a;K]| > K$$

$$\Updownarrow$$

$$\left\| \mathbf{1}_{X[a;K]} - \mathbf{1}_{Y[a;K]} \right\|_1 > K$$

E.g. $\mathbf{1}_{X[5;4]} = (1,1,1,1,0,0,0,0,0)$

Ulam: the embedding

$\overbrace{1234}^{X[5;4]}56789$

$1234\underbrace{6789}_{Y[5;4]}5$

- "Geometrizing" characterization:

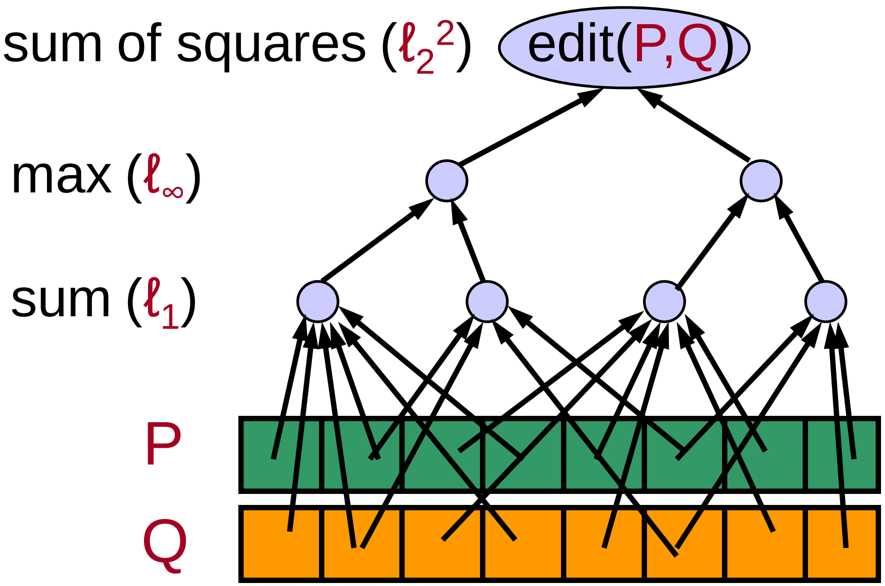
$$Ulam(x,y) \approx \sum_{a=1}^{d} \left(\max_{K=1\ldots d} \frac{\|\mathbf{1}_{X[a;K]} - \mathbf{1}_{Y[a;K]}\|_1}{2K} \right)^2$$

- Gives an embedding

$$f(x) = \left(\left(\tfrac{1}{2K} \mathbf{1}_{X[a;K]} \right)_{K=1\ldots d} \right)_{a=1\ldots d} \in \bigoplus_{(\ell_2)^2}^{d} \bigoplus_{\ell_\infty}^{d} \ell_1^d$$

A view on product metrics: $\bigoplus_{(\ell_2)^2}^{\gamma} \bigoplus_{\ell_\infty}^{\beta} \ell_1^{\alpha}$

- Give more *computational* view of embeddings
- Ulam characterization is related to work in the context of property testing & streaming
  [EKKRV98, ACCL04, GJKK07, GG07, EJ08]



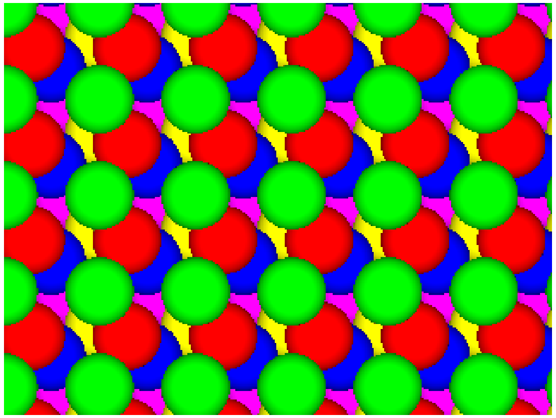
Challenges 3,…

- Embedding into product spaces?
  - Of edit distance, EMD…
- NNS for any norm (Banach space) ?
  - Would help for EMD (a norm)
  - A first target: Schatten norms (e.g., trace of a matrix)
- Other uses of embeddings into product spaces?
  - Related work: *sketching* of product spaces [JW'09, AIK'08, AKO]

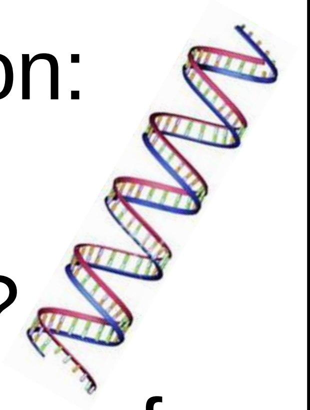
Some aspects I didn’t mention yet

- NNS for spaces with low intrinsic dimension:
  - [Clarkson’99], [Karger-Ruhl’02], [Hildrum-Kubiatowicz-Ma-Rao’04], [Krauthgamer-Lee’04,’05], [Indyk-Naor’07],…
- Cell-probe lower bounds for deterministic and/or exact NNS:
  - [Borodin-Ostrovsky-Rabani’99], [Barkol-Rabani’00], [Jayram-Khot-Kumar-Rabani’03], [Liu’04], [Chakrabarti-Chazelle-Gum-Lvov’04], [Pătraşcu-Thorup’06],…
- NNS for average case:
  - [Alt-Heinrich-Litan’01], [Dubiner’08],…
- Other problems via reductions from NNS:
  - [Eppstein’92], [Indyk’00],…
- Many others !

Summary of challenges



- 1. Design qualitative, efficient space partitioning
- 2. Embeddings with improved distortion: edit into ℓ_1



- 3. NNS for any norm: e.g. trace norm?
- 4. Embedding into product spaces: say, of EMD

$$\bigoplus^{\gamma}_{(\ell_2)^2} \bigoplus^{\beta}_{\ell_\infty} \ell_1^{\alpha}$$