Beyond
Locality Sensitive Hashing

Alex Andoni (MSR SVC)

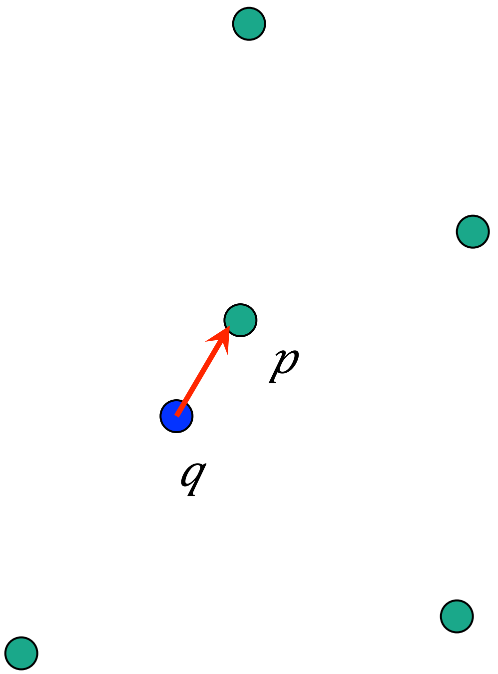
Piotr Indyk (MIT)

Huy L. Nguyen (Princeton)

Ilya Razenshteyn (MIT)

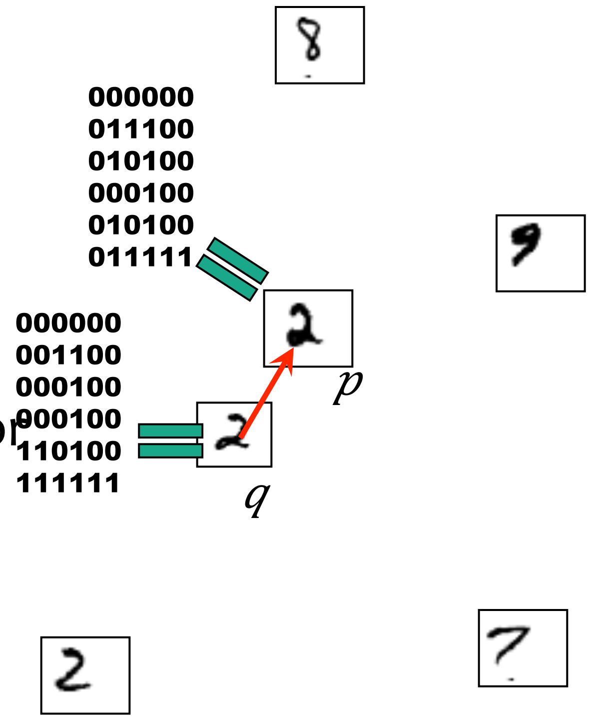
Nearest Neighbor Search (NNS)

- Preprocess: a set D of points
- Query: given a query 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: k-NN rule
  - speech/image/video/music recognition, vector quantization, bioinformatics, etc…
- Distance can be:
  - Hamming, Euclidean, …
- Primitive for other problems:
  - find the similar pairs in a set D, clustering…



Curse of dimensionality

- All exact algorithms degrade rapidly with the dimension d

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

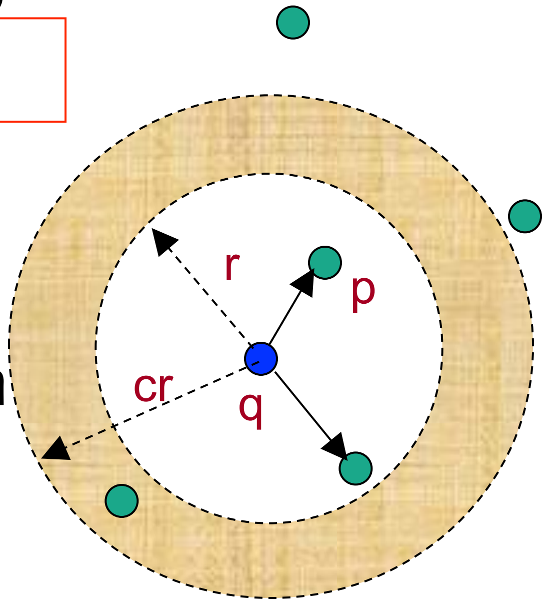
Approximate NNS

c-approximate

- r-near neighbor: given a new point q, report a point $p \in D$ s.t. $\|p-q\| \leq r$ cr

  if there exists a point at distance $\leq r$

- Randomized: a point p returned with 90% probability



Approximation Algorithms

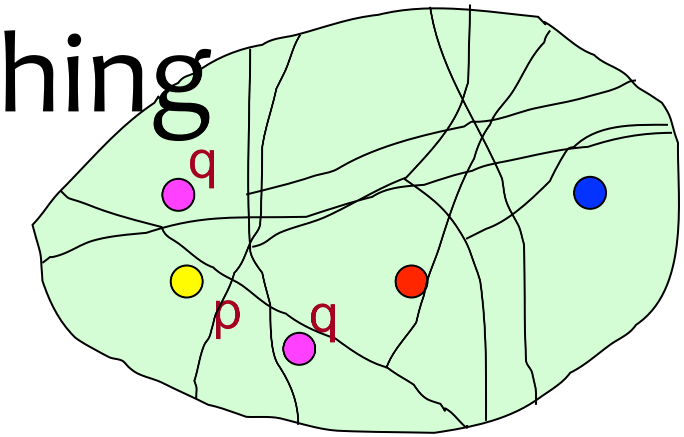
- A vast literature:
  - milder dependence on dimension

    [Arya-Mount'93], [Clarkson'94], [Arya-Mount-Netanyahu-Silverman-We'98], [Kleinberg'97], [Har-Peled'02],…
  - little to no dependence on dimension

    [Indyk-Motwani'98], [Kushilevitz-Ostrovsky-Rabani'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],…

Locality-Sensitive Hashing

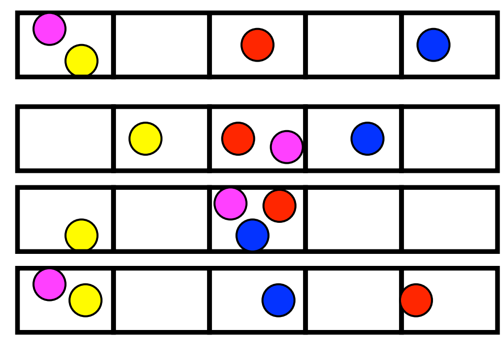
[Indyk-Motwani'98]



- Random hash function g on R^d s.t. for any points p,q:
  - *Close* when $\|p-q\| \leq r$

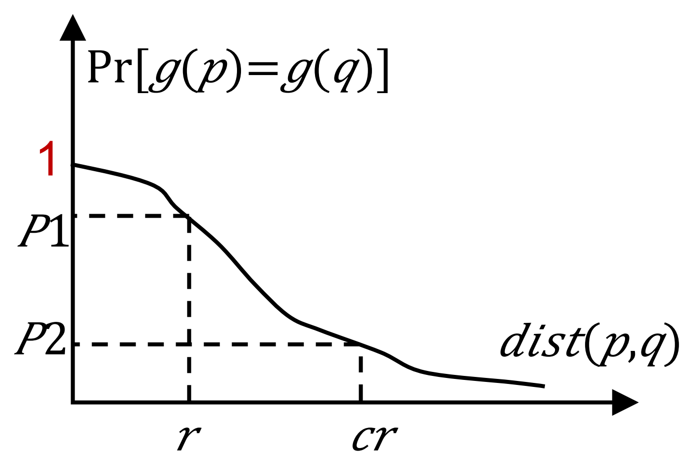
    $P1=$ $\Pr[g(p)=g(q)]$ is “l “not-so-small”
  - *Far* when $\|p-q\| > cr$

    $P2=$ $\Pr[g(p)=g(q)]$ is “small”



- Use several hash tables : n^ρ, where

$$P_1 = P_2^\rho$$



Locality sensitive hash functions

[Indyk-Motwani'98]

- Hash function g is actually a concatenation of "primitive" functions:
  - $g(p) = \langle h_1(p), h_2(p), \ldots, h_k(p) \rangle$
- LSH in Hamming space $\{0,1\}^d$
  - $h(p) = p_j$, i.e., choose j^{th} bit for a random j
  - $\Pr[h(p)=h(q)] = 1 - Ham(p,q)/d$
  - $P_1 = 1 - r/d \approx e^{-r/d}$
  - $P_2 = 1 - cr/d \approx e^{-cr/d}$
  - $\rho = \log 1/P_1 / \log 1/P_2 = r/d / cr/d = 1/c$

Algorithms and Lower Bounds

	Space	Time	Comment	Reference
ℓ_1	$n^{1+\rho}$	n^{ρ}	$\rho=1/c$	[IM'98]
			$\rho\geq 0.5/c$	[MNP'06]
			$\rho\geq 1/c$	[OWZ'11]
	$n^{1+1/c/t}$	$\Omega(t)$ memory lookups		[PTW'08, PTW'10]
ℓ_2	$n^{1+\rho}$	n^{ρ}	$\rho=1/c$	[IM'98]
			$\rho\approx 1/c^2$	[DIIM'04, AI'06]
			$\rho\geq 0.5/c^2$	[MNP'06]
			$\rho\geq 1/c^2$	[OWZ'11]
	$n^{1+1/c^2/t}$	$\Omega(t)$ memory lookups		[PTW'08, PTW'10]

LSH is tight…

leave the rest to cell-probe lower bounds?

Main Result

- NNS in Hamming space (ℓ_1) with $n^{\rho} \cdot d$ query time, $n^{\rho} + nd$ space and preprocessing for
  - $\rho = 7/8/c + O(1/c^{3/2}) + o(1)$
- Improves upon [IM'98]
- NNS in Euclidean space (ℓ_2) with:
  - $\rho = 7/8/c^{2} + O(1/c^{3}) + o(1)$
- Improves upon [AI'06]

A look at LSH lower bounds

- LSH lower bounds in Hamming space
  - Fourier analytic [O'Donnell-Wu-Zhou'11]
- ~~[Motwani-Naor-Panigrahy'06]~~
  - H distribution over hash functions $h:\{0,1\}^d \to U$
  - P_2 = Pr of collision of random ~~p~~,q $q=p+N_\epsilon$
  - P_1 = Pr of collision of random p and ~~$q=p+N_1/c$~~ $q=p+N_\epsilon/c$
  - Get ~~$\rho \geq 0.5/c$~~ $\rho \geq 1/c$

Why not NNS lower bound?

- Suppose we try to generalize [OWZ'11] to NNS
  - Pick random q
  - All the "false near neighbors" are $p = q + N_\epsilon$
  - The dataset is in a small ball of radius $\epsilon d/2$
  - Easy to see at preprocessing: actual near neighbor close to the center of the minimum enclosing ball
- Try $\rho \geq 1/c$ in the distance regime $1/2$ vs $1/2c$?
  - No: $\rho = \ln(1 - 1/2c)/\ln(1/2) \approx 0.72/c$
- Closest pair for *random* data: $n^{1 + 1/(2c-1)}$ [D'10]
  - Improved to $n^{1.79} \cdot (c-1)^{-O(1)}$ [V'12]

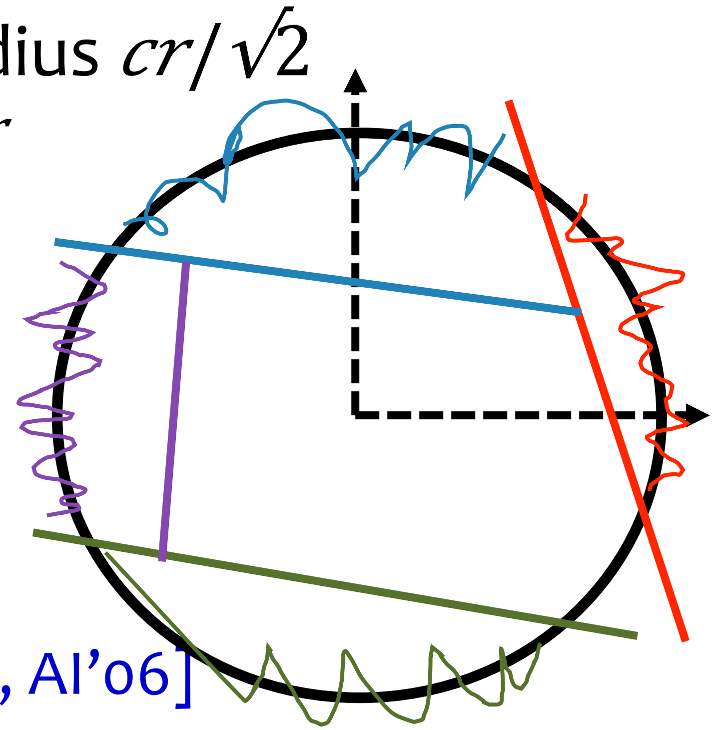
Our algorithm: intuition

- Data dependent LSH:
  - Space partitioning depends on the given dataset!

- Two components:
  - “Nice” geometric configuration with $\rho < 1/c^2$
  - Reduction from general to this “nice” geometric configuration

Configuration: Spherical LSH

- All points are on a sphere of radius $cr/\sqrt{2}$
  - Random points are at distance cr



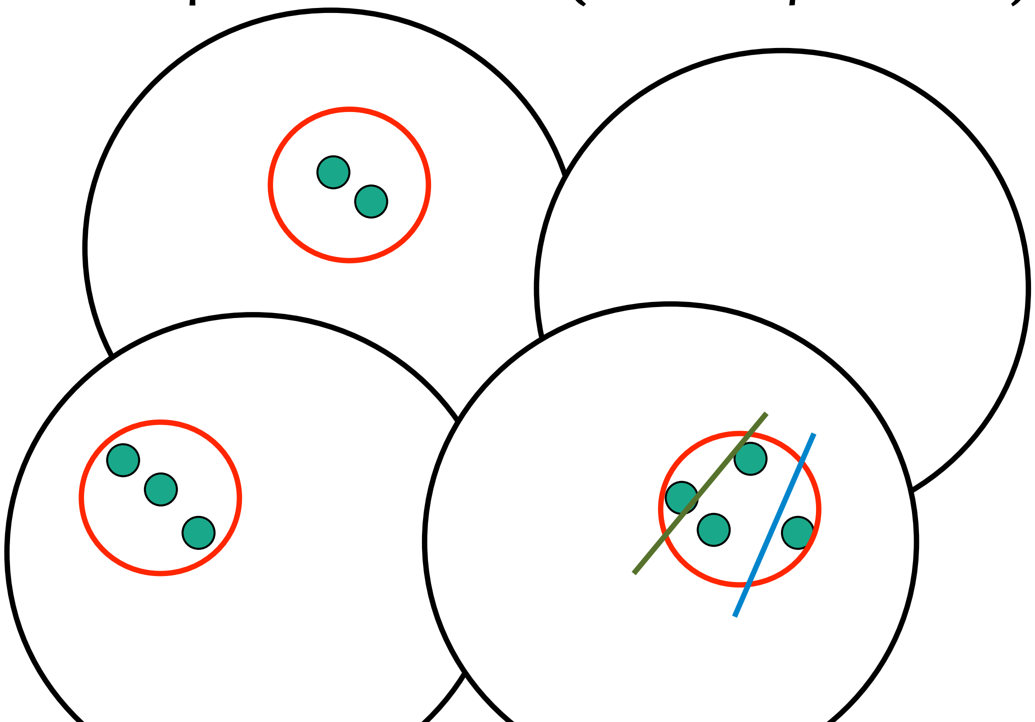
- Lemma 1: $\rho \approx 0.5/c^2$
- "Proof":
  - Obtained via "cap carving"
  - Similar to "ball carving" [KMS'98, AI'06]
- Lemma 1': $\rho \approx (1-1/4\eta^2)1/c^2$ for radius = ηcr

Reduction: into spherical LSH

- Idea: apply a few rounds of “regular” LSH
  - Ball carving [AI’06]
- Intuitively:
  - far points unlikely to collide
  - partitions the data into buckets of small diameter $\approx O(cr)$
  - find the minimum enclosing ball
  - finally apply spherical LSH on this ball!

Two-level algorithm

- n^{ρ} hash tables, each with:
  - hash function $g=(h_1, h_2, \ldots h_l, s_1, \ldots s_m)$
  - h_i's are "ball carving LSH" (data *independent*)
  - s_j's are "spherical LSH" (data *dependent*)



Details

- Analysis:
  - Final ρ is an "average" of ρ from levels 1 and 2
  - Level 1: make pairs at distance τc unlikely to collide
  - Level 2: find minimum enclosing ball of radius $\tau c/\sqrt{2}$
    - use Jung theorem: diameter τc implies MEB radius $\tau c/\sqrt{2}$
- Algorithm inside each bucket (from level 1)
  - Drop all pairs that are further than τc
  - Find approximate MEB
  - Apply spherical LSH on each (approximate) shell of the MEB

Finale

- NNS with n^{ρ} query time:
  - where $\rho \approx 7/8/c^2$ for ℓ_2
  - where $\rho \approx 7/8/c$ for ℓ_1
- Below the lower bounds for LSH/space partitions!
- Idea: *data dependent* space partitions
- Better upper bound?
  - Multi-level improves a bit, but not too much
  - $\rho = 0.5/c^2$ for ℓ_2 ?
- Or data dependent lower bounds?