Data-dependent Hashing for Nearest Neighbor Search

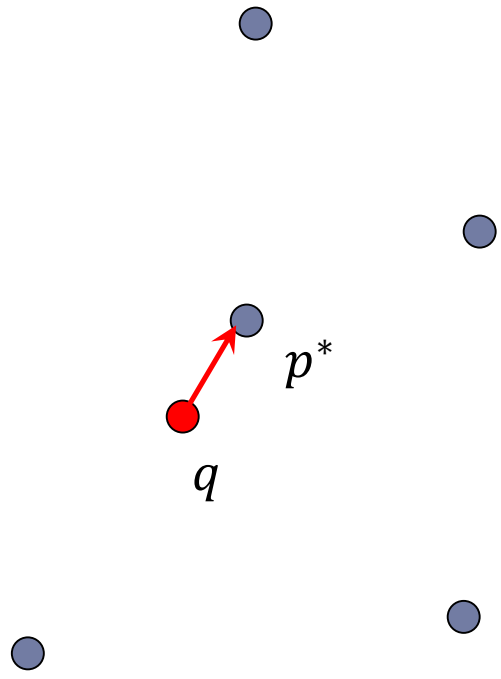
Alex Andoni

(Columbia University)

Based on joint work with: Piotr Indyk, Huy Nguyen, Ilya Razenshteyn

Nearest Neighbor Search (NNS)

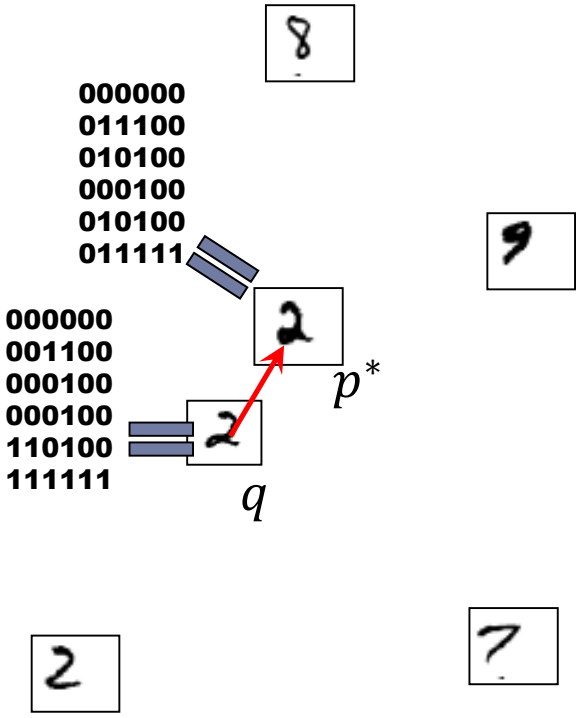
- Preprocess: a set P of points
- Query: given a query point q, report a point $p^* \in P$ with the smallest distance to q



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…
- Distances:
  - Hamming, Euclidean,

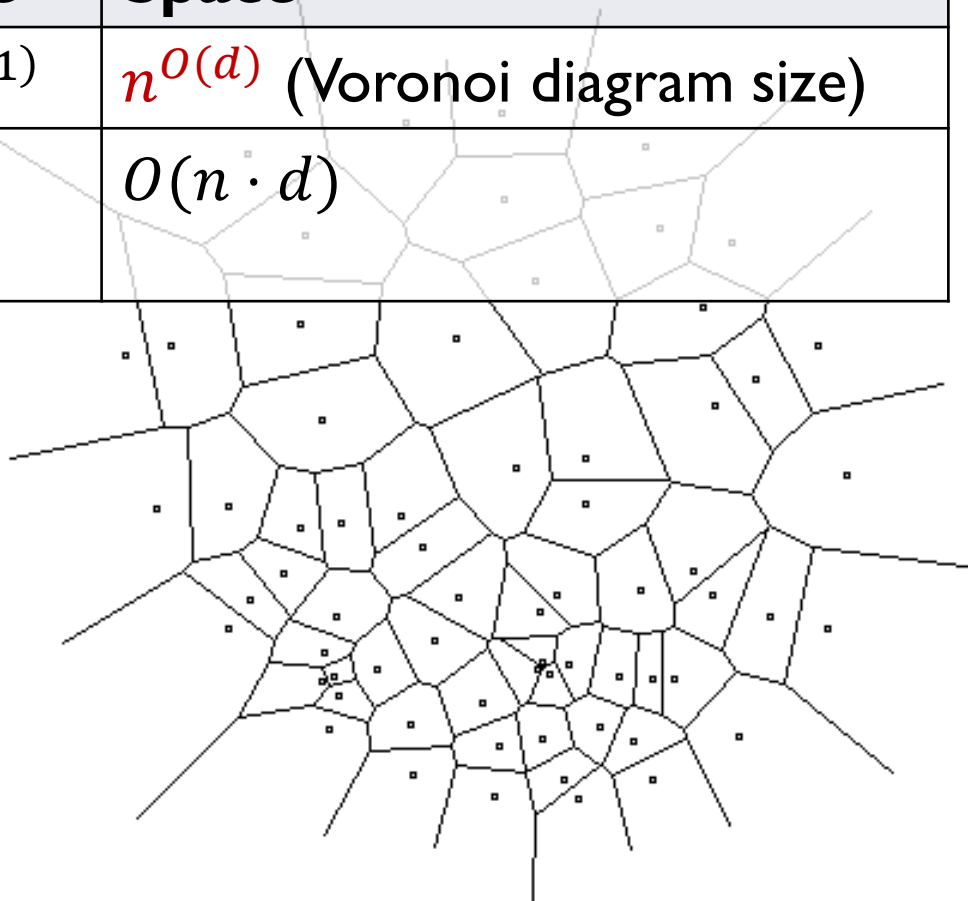
    edit distance, earthmover distance, etc…
- Core primitive: closest pair, clustering, etc…



Curse of Dimensionality

- All exact algorithms degrade rapidly with the dimension d

Algorithm	**Query time**	**Space**
Full indexing	$(d \cdot \log n)^{O(1)}$	$n^{O(d)}$ (Voronoi diagram size)
No indexing – linear scan	$O(n \cdot d)$	$O(n \cdot d)$

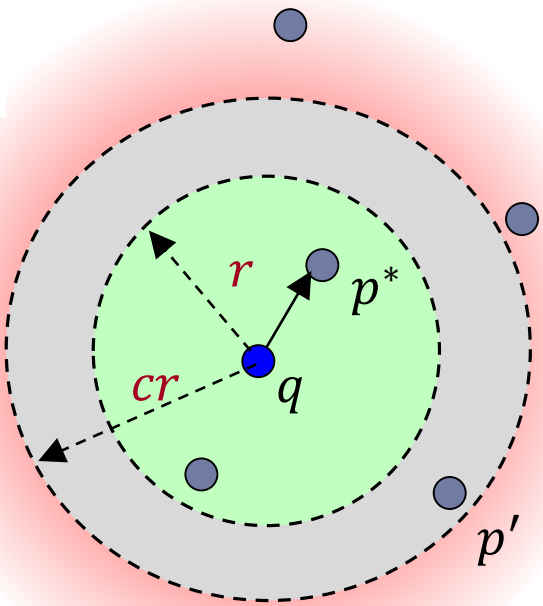


Approximate NNS

c-approximate

- r-near neighbor: given a query point q, report a point $p' \in P$ s.t. $\|p' - q\| \leq cr$
  - as long as there is some point within distance r

- Practice: use for exact NNS
  - *Filtering*: gives a set of candidates (hopefully small)



NNS algorithms

Exponential dependence on dimension

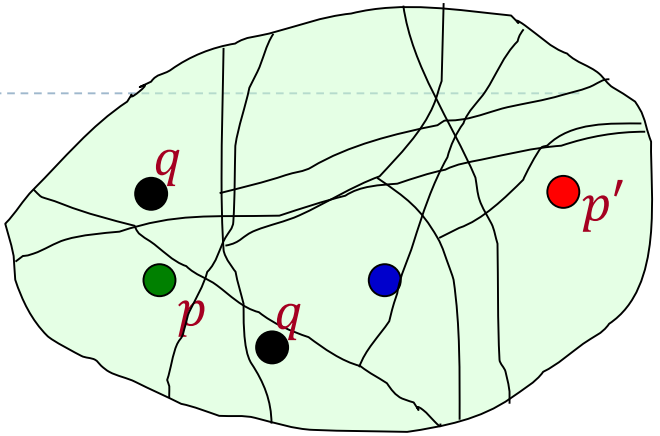
- [Arya-Mount'93], [Clarkson'94], [Arya-Mount-Netanyahu-Silverman-We'98], [Kleinberg'97], [Har-Peled'02],[Arya-Fonseca-Mount'11],…

Linear/poly dependence on dimension

- [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], [A.-Indyk-Nguyen-Razenshteyn'14], [A.-Razenshteyn'15], [Pagh'16],[Laarhoven'16],…

Locality-Sensitive Hashing

[Indyk-Motwani'98]



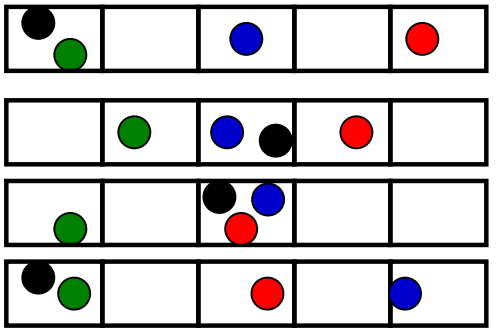
Random hash function h on R^d satisfying:

- for *close pair*: when $\|q - p\| \leq r$

$P_1 =$ $\Pr[h(q) = h(p)]$ is "not-so-small"

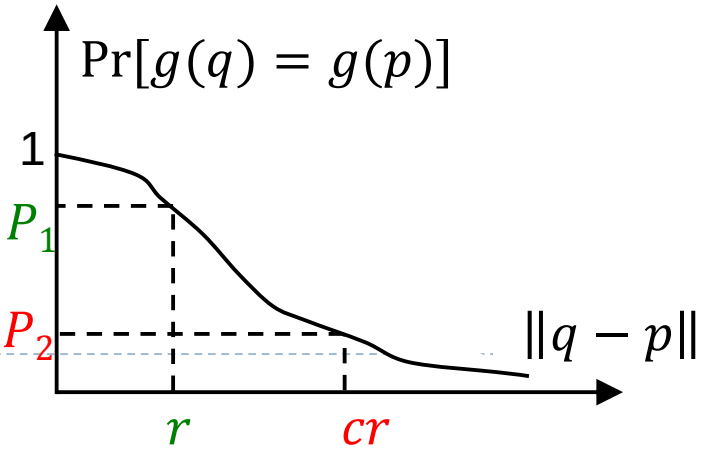
- for *far pair*: when $\|q - p'\| > cr$

$P_2 =$ $\Pr[h(q) = h(p')]$ is "small"



Use several hash tables:

n^{ρ}, where $\rho = \dfrac{\log 1/P_1}{\log 1/P_2}$



LSH Algorithms

	Space	Time	Exponent	$c = 2$	Reference
Hamming space	$n^{1+\rho}$	n^{ρ}	$\rho = 1/c$	$\rho = 1/2$	[IM'98]
			$\rho \geq 1/c$		[MNP'06, OWZ'11]
Euclidean space	$n^{1+\rho}$	n^{ρ}	$\rho = 1/c$	$\rho = 1/2$	[IM'98, DIIM'04]
			$\rho \approx 1/c^2$	$\rho = 1/4$	[AI'06]
			$\rho \geq 1/c^2$		[MNP'06, OWZ'11]

LSH is tight… what's next?

Lower bounds (cell probe)
[A.-Indyk-Patrascu'06, Panigrahy-Talwar-Wieder'08,'10, Kapralov-Panigrahy'12]

Space-time trade-offs
[Panigrahy'06, A.-Indyk'06]

Datasets with additional structure
[Clarkson'99, Karger-Ruhl'02, Krauthgamer-Lee'04, Beygelzimer-Kakade-Langford'06, Indyk-Naor'07, Dasgupta-Sinha'13, Abdullah-A.-Krauthgamer-Kannan'14,…]

But are we really done with basic NNS algorithms?

Beyond Locality Sensitive Hashing

Hamming space

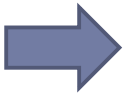
Space	Time	Exponent	$c = 2$	Reference
$n^{1+\rho}$	n^{ρ}	$\rho = 1/c$	$\rho = 1/2$	[IM'98]
		$\rho \geq 1/c$		[MNP'06, OWZ'11]
$n^{1+\rho}$	n^{ρ}	complicated	$\rho = 1/2 - \epsilon$	[AINR'14]
		$\rho \approx \frac{1}{2c-1}$	$\rho = 1/3$	[AR'15]

LSH

Euclidean space

$n^{1+\rho}$	n^{ρ}	$\rho \approx 1/c^2$	$\rho = 1/4$	[AI'06]
		$\rho \geq 1/c^2$		[MNP'06, OWZ'11]
$n^{1+\rho}$	n^{ρ}	complicated	$\rho = 1/4 - \epsilon$	[AINR'14]
		$\rho \approx \frac{1}{2c^2-1}$	$\rho = 1/7$	[AR'15]

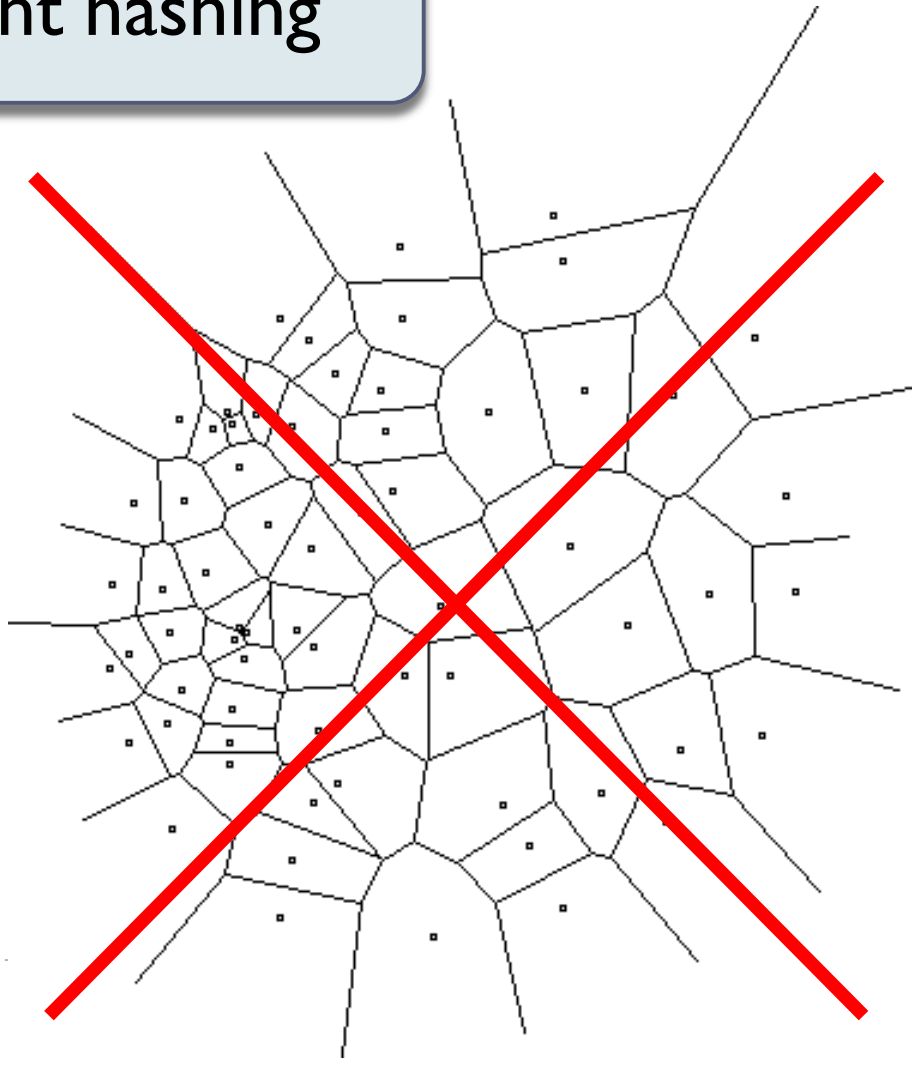
LSH



New approach?

Data-dependent hashing

- A random hash function, chosen after seeing the given dataset
- Efficiently computable

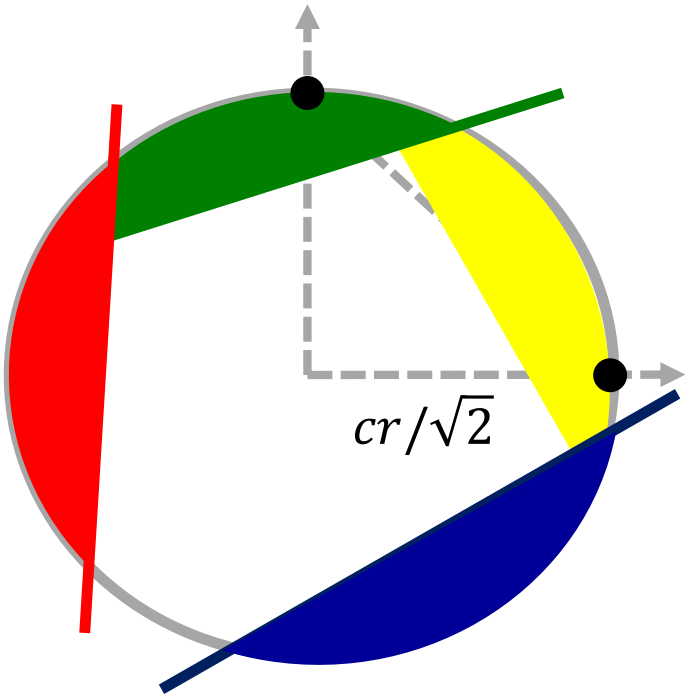


Construction of hash function

- Two components:
  - Nice geometric structure ⟸ has better LSH
  - Reduction to such structure ⟸ data-dependent
- Like a (weak) "regularity lemma" for a set of points

Nice geometric structure: average-case

- Think: random dataset on a sphere
  - vectors perpendicular to each other
  - s.t. random points at distance $\approx cr$
- Lemma: $\rho = \frac{1}{2c^2-1}$
  - via Cap Carving

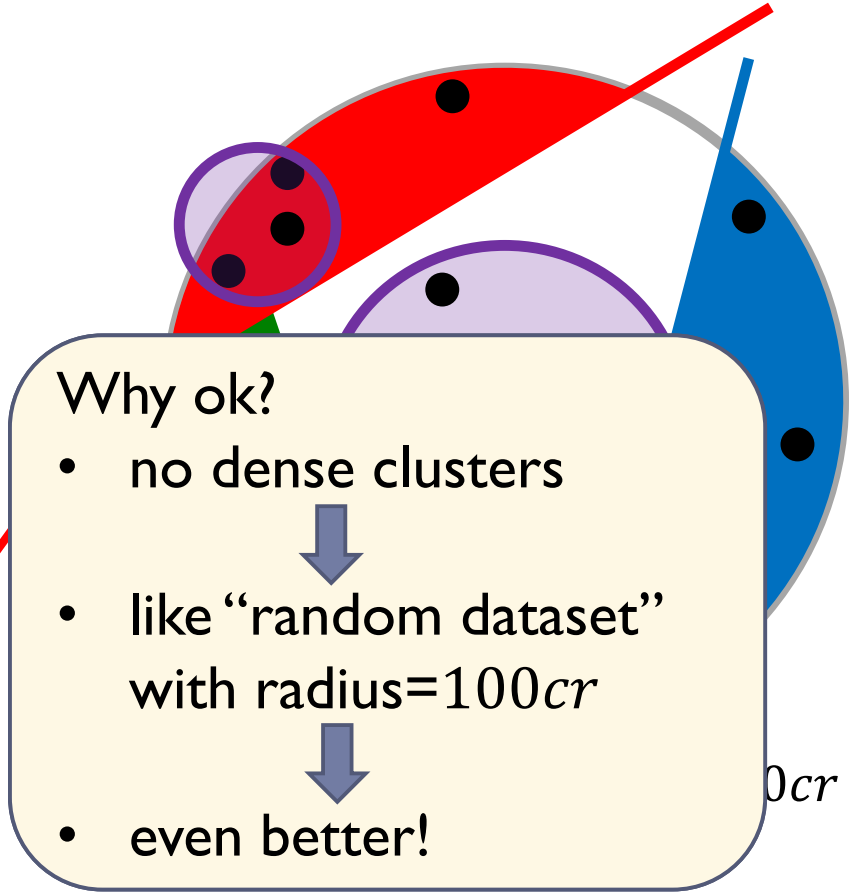


Reduction to nice structure

- Idea:

  iteratively decrease the radius of minimum enclosing ball

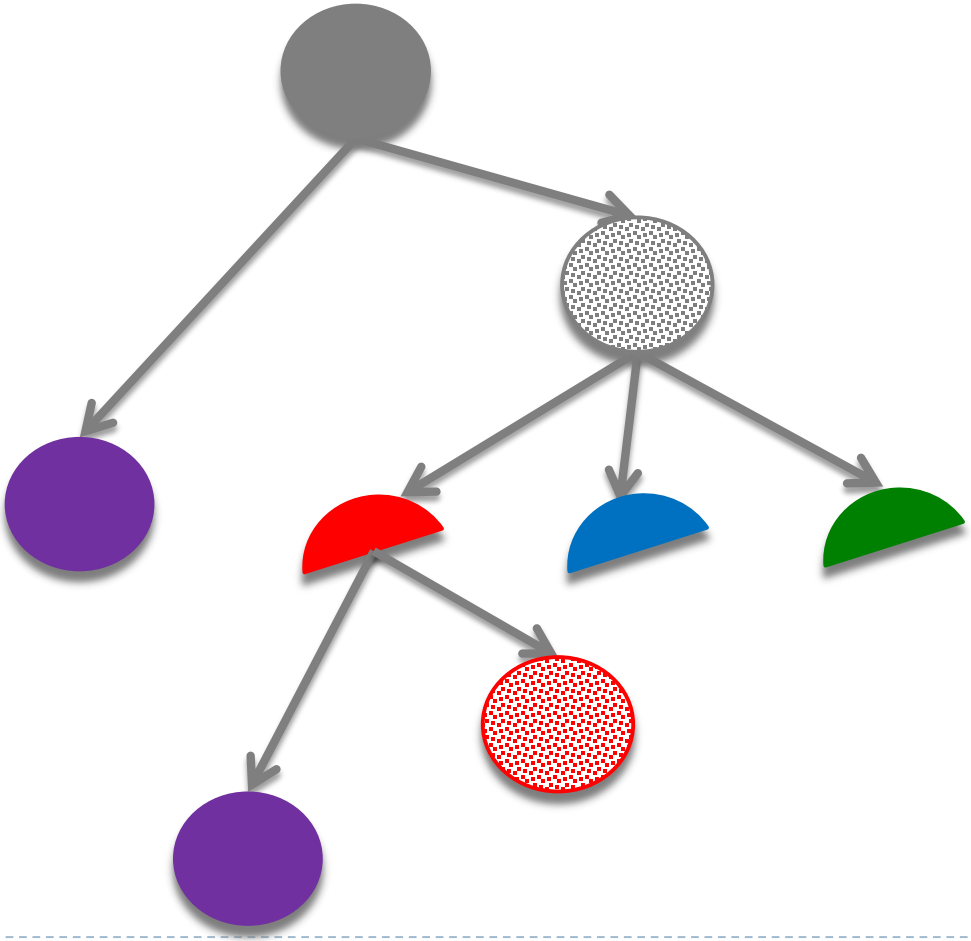
- Algorithm:
  - find dense clusters
    - with smaller radius
    - large fraction of points
  - recurse on dense clusters
  - apply cap carving on the rest
    - recurse on each "cap"
    - eg, dense clusters might reappear

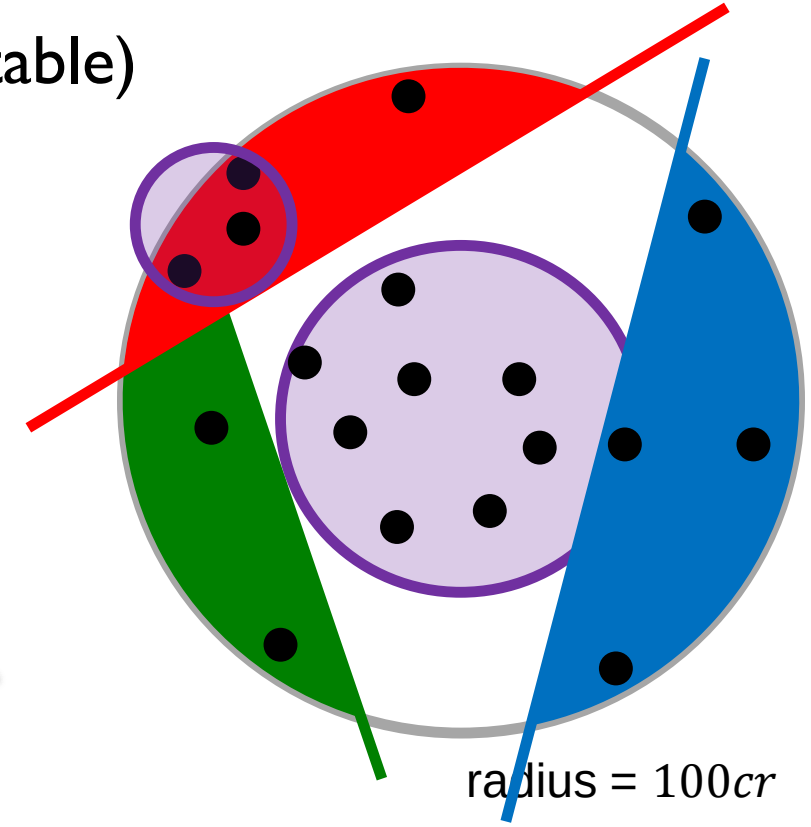


*picture not to scale & dimension

Hash function

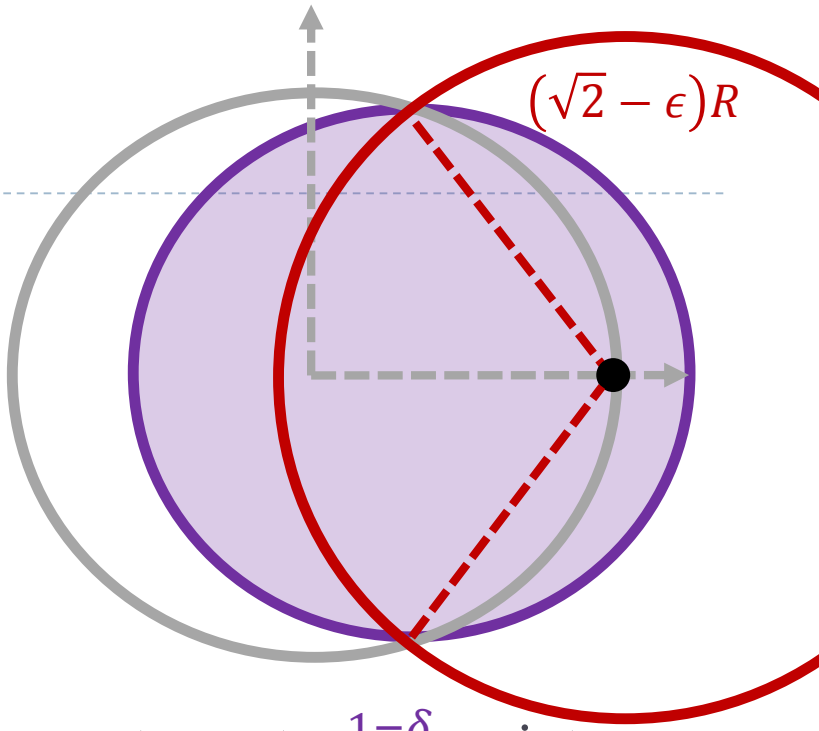
- Described by a tree (like a hash table)





*picture not to scale&dimension

Dense clusters



- Current dataset: radius R
- A dense cluster:
  - Contains $n^{1-\delta}$ points
  - Smaller radius: $(1-\Omega(\epsilon^2))R$
- After we remove all clusters:
  - For any point on the surface, there are at most $n^{1-\delta}$ points within distance $(\sqrt{2}-\epsilon)R$
  - The other points are essentially orthogonal !
- When applying Cap Carving with parameters (P_1, P_2):
  - Empirical number of far pts colliding with query: $nP_2 + n^{1-\delta}$
  - As long as $nP_2 \gg n^{1-\delta}$, the "impurity" doesn't matter!

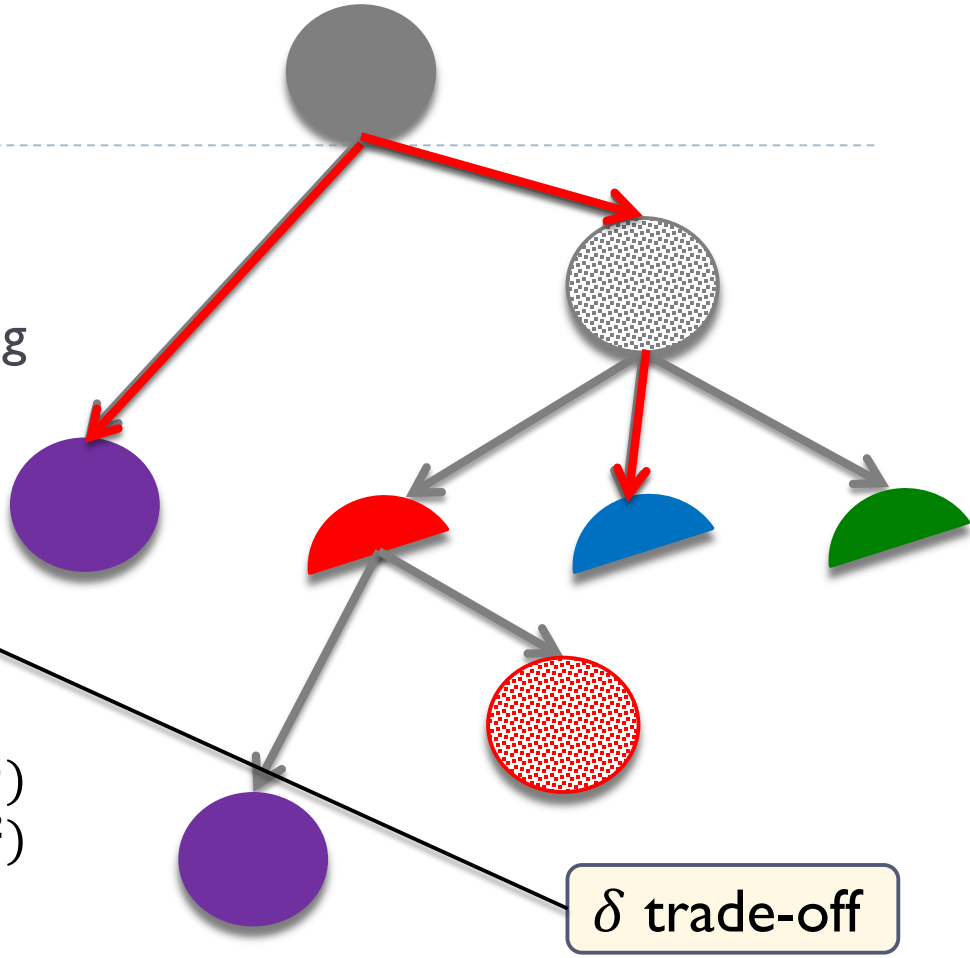
ϵ trade-off

δ trade-off

?

Tree recap

- During query:
  - Recurse in all clusters
  - Just in one bucket in CapCarving
- Will look in >1 leaf!
- How much branching?
  - Claim: at most $\left(n^{\delta}+1\right)^{O(1/\epsilon^2)}$
  - Each time we branch
    - at most n^{δ} clusters $(+1)$
    - a cluster reduces radius by $\Omega(\epsilon^2)$
    - cluster-depth at most $100/\Omega(\epsilon^2)$
- Progress in 2 ways:
  - Clusters reduce radius
  - CapCarving nodes reduce the # of far points (empirical P_2)
- A tree succeeds with probability $\geq n^{-\frac{1}{2c^2-1}-o(1)}$



Beyond "Beyond LSH"

- Practice: often optimize partition to *your* dataset
  - PCA-tree, spectral hashing, etc [S91, McN01, VKD09, WTF08,…]
  - no guarantees (performance or correctness)
- Theory: assume special structure in the dataset
  - low intrinsic dimension [KR'02, KL'04, BKL'06, IN'07, DS'13,…]
  - structure + noise [Abdullah-A.-Krauthgamer-Kannan'14]

Data-dependent hashing helps even when *no a priori structure* !

Data-dependent hashing wrap-up

- Dynamicity?
  - Dynamization techniques [Overmars-van Leeuwen'81]
- Better bounds?
  - For dimension $d = O(\log n)$, can get better ρ! [Laarhoven'16]
  - For $d > \log^{1+\delta} n$: our ρ is optimal even for data-dependent hashing! [A-Razenshteyn'??]:
    - in the right formalization (to rule out Voronoi diagram):
    - description complexity of the hash function is $n^{1-\Omega(1)}$
- Practical variant [A-Indyk-Laarhoven-Razenshteyn-Schmidt'15]
- NNS for ℓ_∞
  - [Indyk'98] gets approximation $O(\log \log d)$ (poly space, sublinear qt)
    - Cf., ℓ_∞ has no non-trivial sketch!
    - Some matching lower bounds in the relevant model [ACP'08, KP'12]
  - Can be thought of as data-dependent hashing
- NNS for any norm (eg, matrix norms, EMD) ?