Nearest Neighbor Search (3)

Alex Andoni
(Columbia University)

MADALGO Summer School on Streaming Algorithms 2015

Time-Space Trade-offs (Euclidean)

space	query time	Space	Time	Comment	Reference
low	high	$\approx n$	n^{σ}	$\sigma = 2.09/c$	[Ind'01, Pan'06]
				$\sigma = O(1/c^2)$	[AI'06]
medium	medium	$n^{1+\rho}$	n^{ρ}	$\rho = 1/c$	[IM'98, DIIM'04]
				$\rho = 1/c^2$	[AI'06]
				$\rho \geq 1/c^2$	[MNP'06, OWZ'11]
		$n^{1+o(1/c^2)}$	ω(1) memory lookups		[PTW'08, PTW'10]
high	low	n^{4/ϵ^2}	$\cancel{O(d \log n)}$ 1 mem lookup	$c = 1 + \epsilon$	[KOR'98, IM'98, Pan'06]
		$n^{o(1/\epsilon^2)}$	ω(1) memory lookups		[AIP'06]

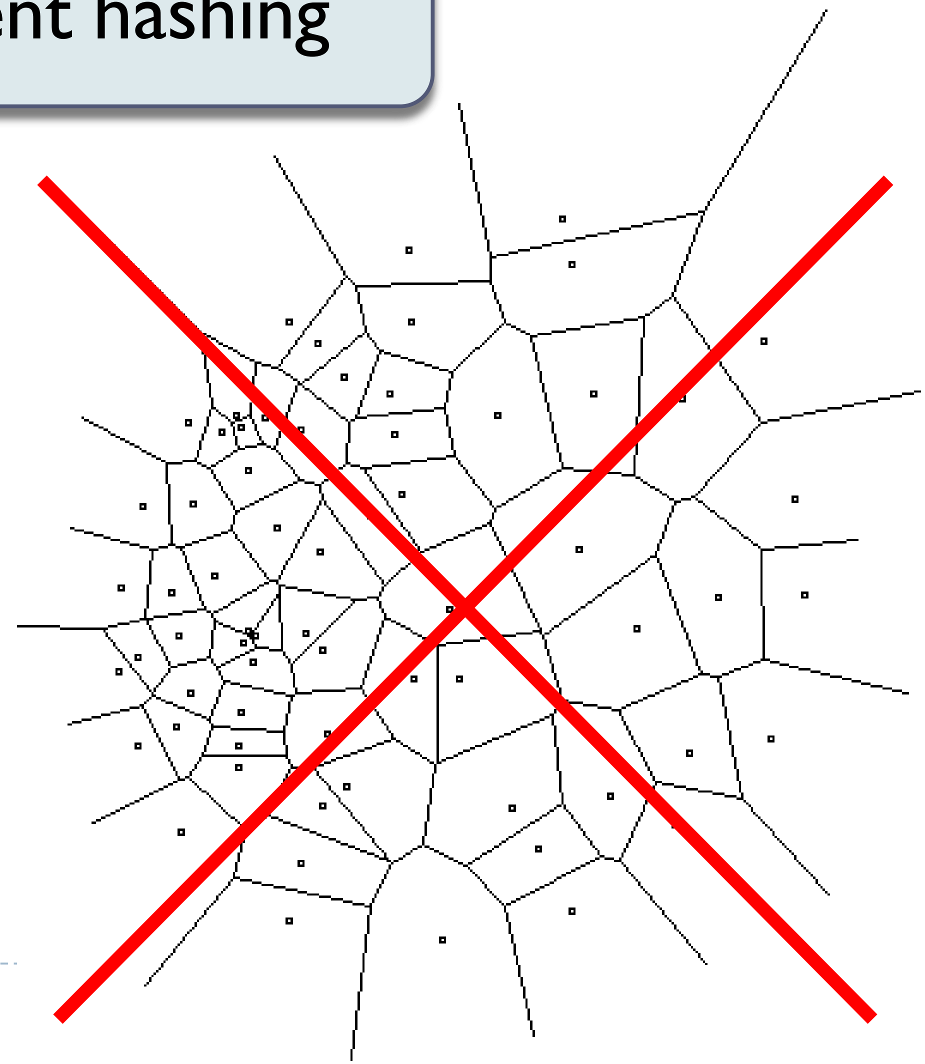
Beyond Locality Sensitive Hashing

	Space	Time	Exponent	$c = 2$	Reference	
Hamming space	$n^{1+\rho}$	n^{ρ}	$\rho = 1/c$	$\rho = \mathbf{1/2}$	[IM'98]	LSH
			$\rho \geq 1/c$		[MNP'06, OWZ'11]	LSH
	$n^{1+\rho}$	n^{ρ}	$\rho \approx \frac{1}{2c-1}$	$\rho = \mathbf{1/3}$	[AINR'14, AR'15]	
Euclidean space	$n^{1+\rho}$	n^{ρ}	$\rho \approx 1/c^2$	$\rho = \mathbf{1/4}$	[AI'06]	LSH
			$\rho \geq 1/c^2$		[MNP'06, OWZ'11]	LSH
	$n^{1+\rho}$	n^{ρ}	$\rho \approx \frac{1}{2c^2-1}$	$\rho = \mathbf{1/7}$	[AINR'14,AR'15]	

New approach?

Data-dependent hashing

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



A look at LSH lower bounds

- LSH lower bounds in Hamming space
  - Fourier analytic
- ~~[Motwani-Naor-Panigrahy'06]~~ [O'Donnell-Wu-Zhou'11]
  - H distribution over hash functions $h: \{0,1\}^d \to U$
  - Far pair: p, q random, distance $= \cancel{d/2}$ ϵd
  - Close pair: p, q random at distance = $\cancel{\frac{d/2}{c}}$ $\frac{\epsilon d}{c}$
  - Get $\cancel{\rho \geq 0.5/c}$
    $\rho \geq 1/c$

Why not NNS lower bound?

- Suppose we try to apply [OWZ'11] to NNS
  - Pick random q
  - All the "far point" are concentrated 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

Construction of hash function

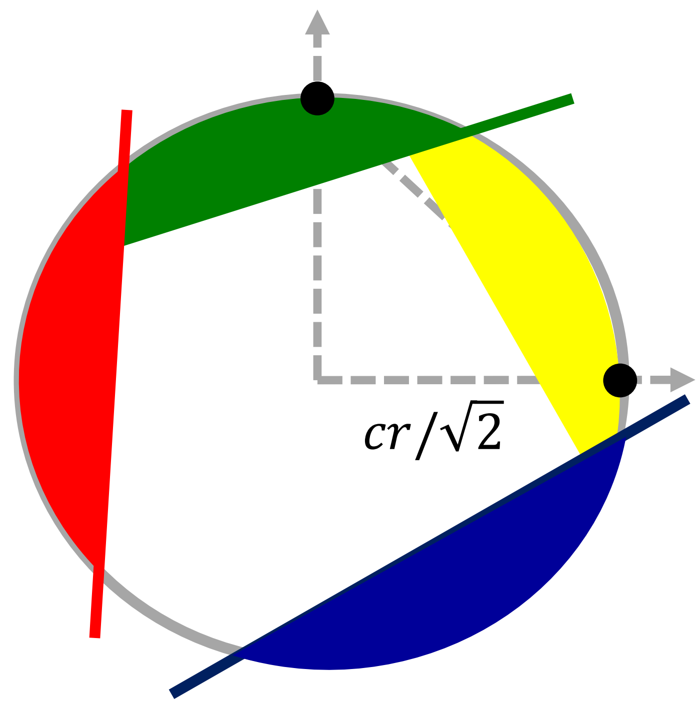
[A.-Indyk-Nguyen-Razenshteyn'14, A.-Razenshteyn'15]

- Two components:
  - Nice geometric structure ⟸ has better LSH
  - Reduction to such structure ⟸ data-dependent
- Like a "Regularity Lemma" for a set of points

Nice geometric structure

- Like a random dataset on a sphere
  - s.t. random points at distance $\approx cr$

- Lemma: $\rho = \frac{1}{2c^2-1}$
  - via Cap Carving
  - curvature

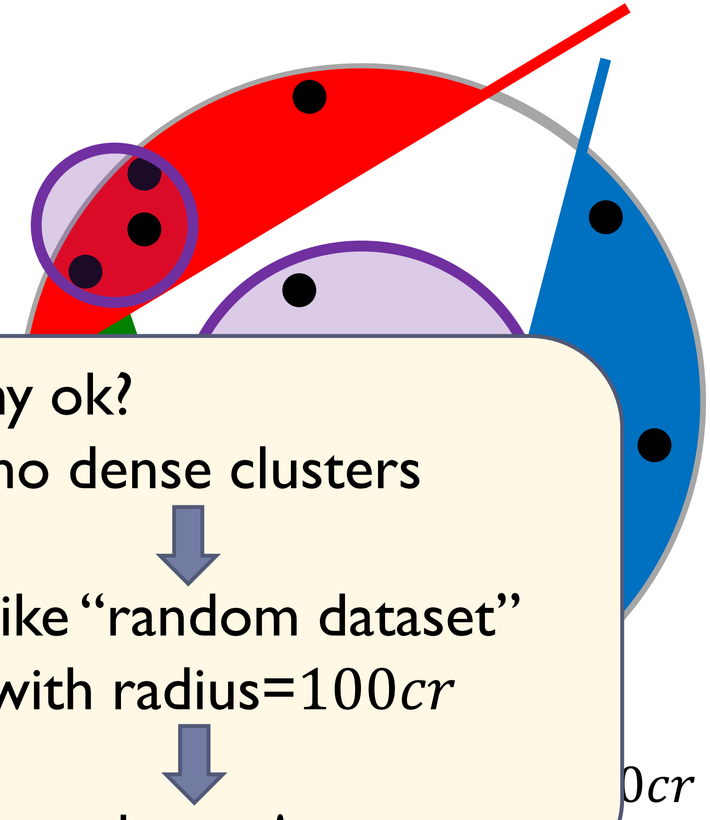


Reduction to nice structure

- 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



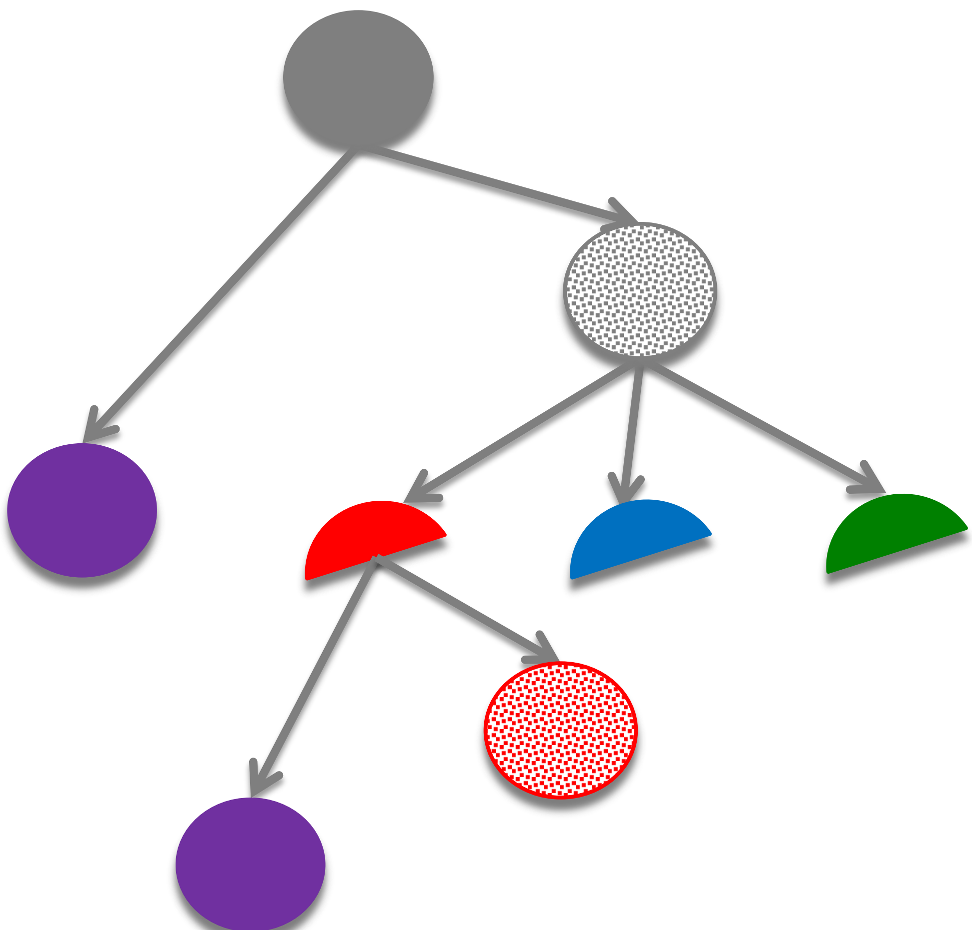
Why ok?
- no dense clusters
- like "random dataset" with radius=$100cr$
- even better!

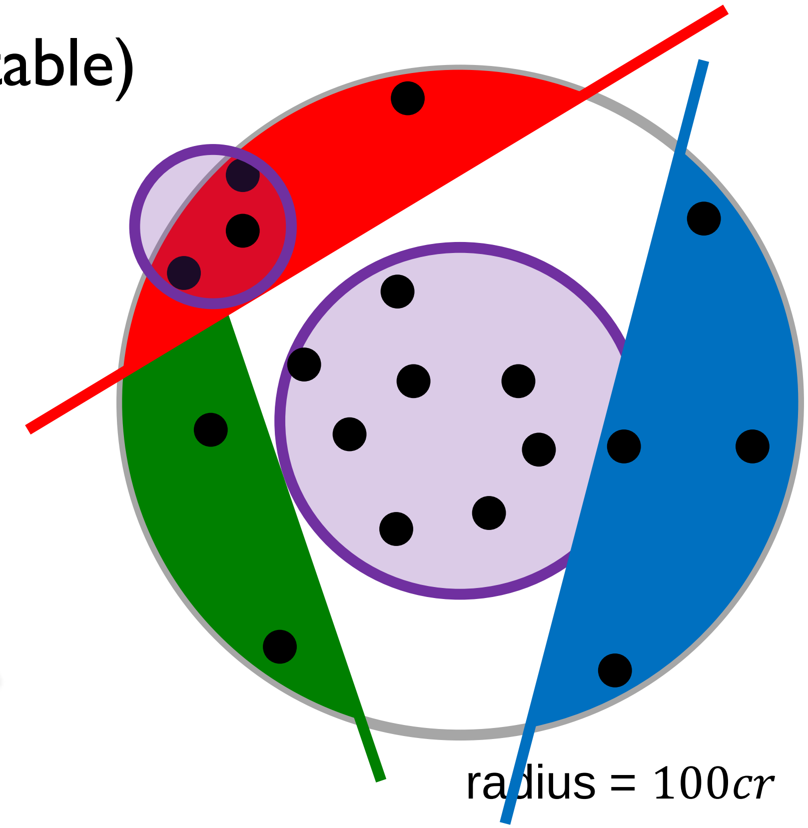
radius = $99cr$

*picture not to scale & dimension

Hash function

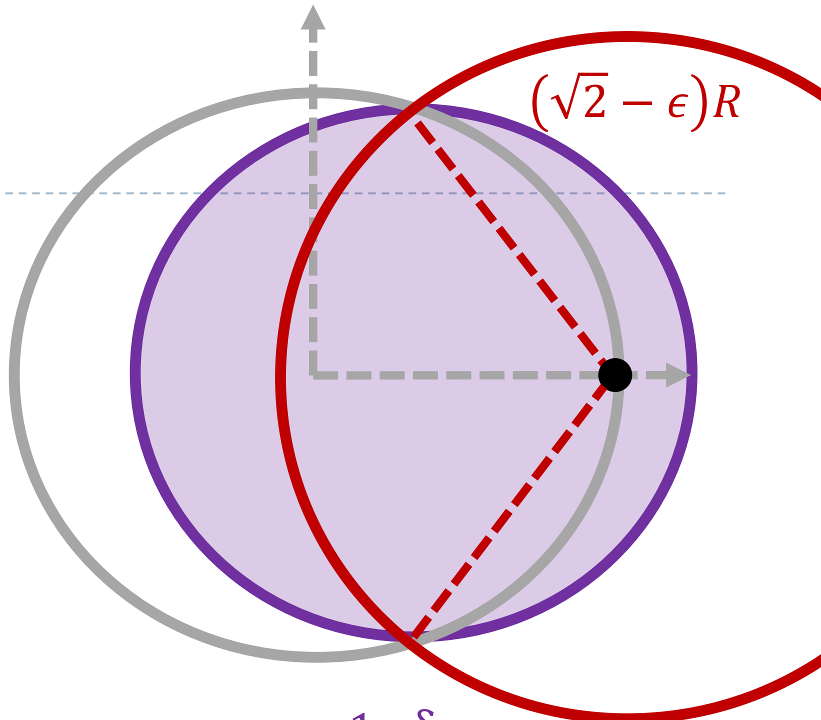
- Described by a tree (like a hash table)





*picture not to scale&dimension

Dense clusters



- Current dataset: radius R
- A dense cluster:
  - Contains $n^{1-\delta}$ points
  - Smaller radius: $\left(1-\Omega(\epsilon^2)\right)R$
- After we remove all clusters:
  - For any point on the surface, there are at most $n^{1-\delta}$ points within distance $\left(\sqrt{2}-\epsilon\right)R$
  - 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!

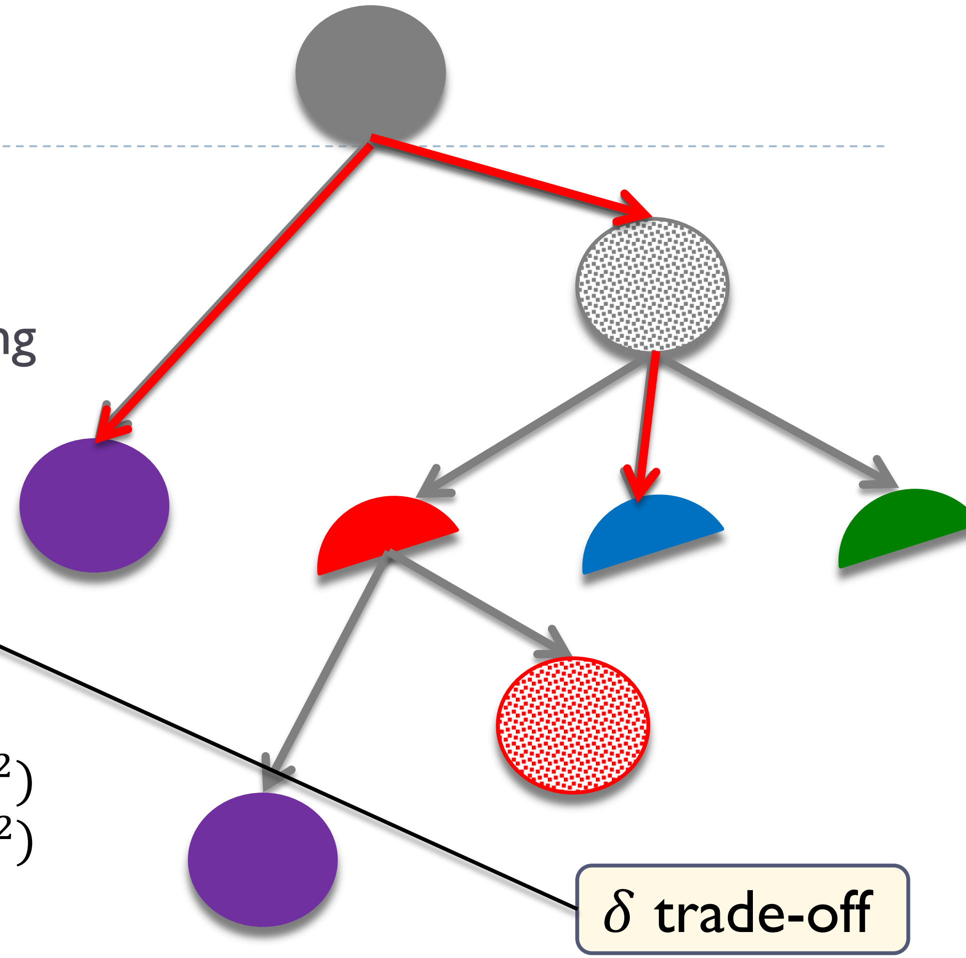
ϵ trade-off

δ trade-off

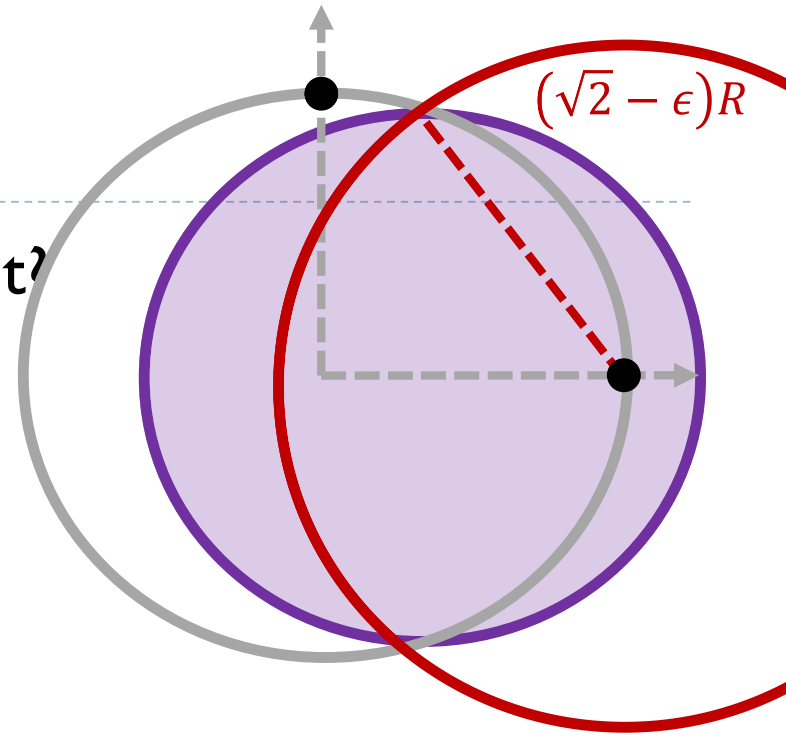
?

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 $\geq n^{-\frac{1}{2c^2-1}-o(1)}$



Fast preprocessing



- How to find the dense clusters fast?
- Step 1: reduce to $O(n^2)$ time.
  - Enough to consider centers that are data points
- Step 2: reduce to near-linear time.
  - Down-sample!
  - Ok because we want clusters of size $n^{1-\delta}$
  - After downsampling by a factor of $\sqrt{n}$, a cluster is still somewhat heavy.

Other details

- In the analysis,
  - Instead of working with "probability of collision with far point" P_2, work with "empirical estimate" (the actual number)
  - A little delicate: interplay with "probability of collision with close point", P_1
    - The empirical P_2 important only for the bucket where the query falls into
    - Need to condition on collision with close point in the above empirical estimate
  - In dense clusters, points may appear *inside* the balls
    - whereas CapCarving works for points on the sphere
    - need to partition balls into thin shells (introduces more branching)

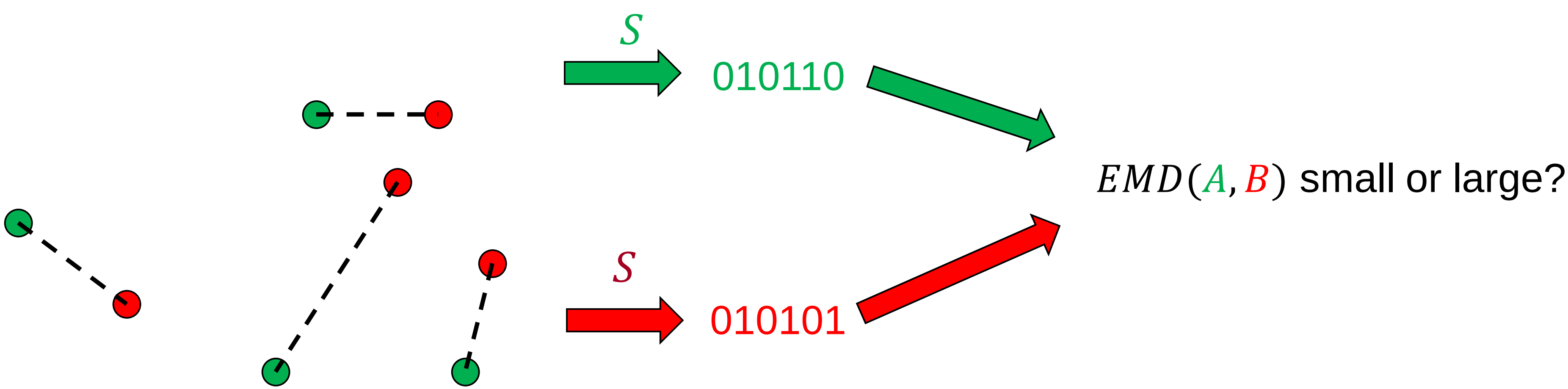
Data-dependent hashing wrap-up

- Dynamicity?
  - Dynamization techniques [Overmars-van Leeuwen'81]
- Better bounds?
  - [AR]: optimal even for data-dependent hashing!
  - In the right formalization (to rule out Voronoi diagram)
    - Description complexity of the hash function is $n^{1-\Omega(1)}$
    - High dimension
- 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?

Beyond ℓ_p's ?

Sketching/NNS for other distances?

- Earth Mover Distance:
  - Given two sets A, B of points in a metric space
  - EMD(A,B) = min **cost** bipartite matching between A and B
- Applications in image vision



Algorithms via embedding

- Embedding:
  - Map sets into vectors in ℓ_1 preserving the distance (approximately)
  - Then use algorithms for ℓ_1 !
- Say, EMD over $[s]^2$
- Theorem [Cha02, IT03]: Exists a map f mapping all $A \subset [s]^2$ into ℓ_1 with distortion $O(\log s)$:
  - i.e., for any $A, B \subset [s]^2$ we have:
  $EMD(A,B) \le ||f(A) - f(B)||_1 \le O(\log s) \cdot EMD(A,B)$

- Sketch: $O(\log s)$-approximation in $\tilde{O}(1)$ space

Embeddability into ℓ_1

Metric	Upper bound
Earth-mover distance (s-sized sets in 2D plane)	$O(\log s)$ [Cha02, IT03]
Earth-mover distance (s-sized sets in $\{0,1\}^d$)	$O(\log s \cdot \log d)$ [AIK08]
Edit distance over $\{0,1\}^d$ (#indels to transform x->y)	$2^{\tilde{O}(\sqrt{\log d})}$ [OR05]
Ulam (edit distance between permutations)	$O(\log d)$ [CK06]
Block edit distance	$\tilde{O}(\log d)$ [MS00, CM07]

edit(banana ,

ananas) = 2

edit(1234567,

7123456) = 2

Non-embeddability into ℓ_1

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

- Other hosts possible, with worse sketching complexity
- EMD: α-approximation in $O\left(s^{1/\alpha}\right)$ space [ADIW'09]
  - embed into a more complex space, and use Precision Sampling

Theory of sketching

- When is sketching possible?

- [BO'10]: characterize when can sketch for "generalized frequency moments":
  - $\sum_i F(x_i)$ for increasing functions F
- [LNW'14]: in general streams (insertions and deletions), for estimating any $f(x)$, might as well have f which is *linear*
- [AKR'15]: in the case of a norm X
  - X has very efficient sketch: $O(1)$ size and approximation (like for ℓ_p for $p \leq 2$)

    if and only if X embeds into some ℓ_p for $p \leq 2$!
  - Eg, using [NS07], EMD does not admit very efficient sketches
- Characterization in other cases?