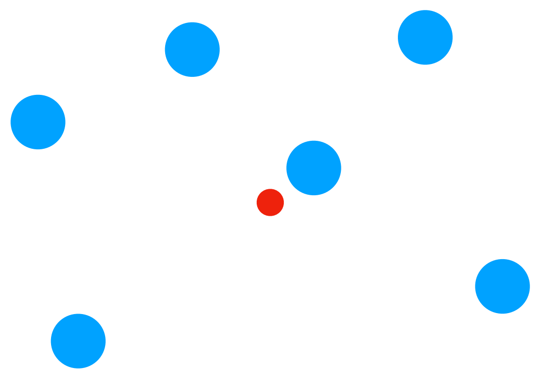
Efficient Centroid-Linkage Clustering

MohammadHossein Bateni, Laxman Dhulipala, Willem Fletcher,
Kishen N Gowda, D Ellis Hershkowitz, Rajesh Jayaram, Jakub Łącki

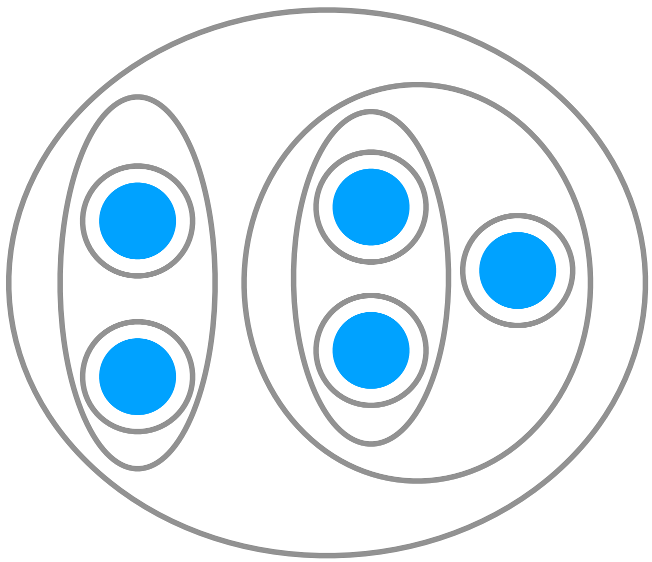
Centroid

Given data $C \subset \mathbb{R}^d$ in we define the centroid of C as $\text{cent}(C) = \dfrac{\sum_{p \in C} p}{|C|}$



Hierarchical Agglomerative Clustering (HAC)

- Commonly used form of hierarchical clustering
- Use cases include computational biology and computer vision

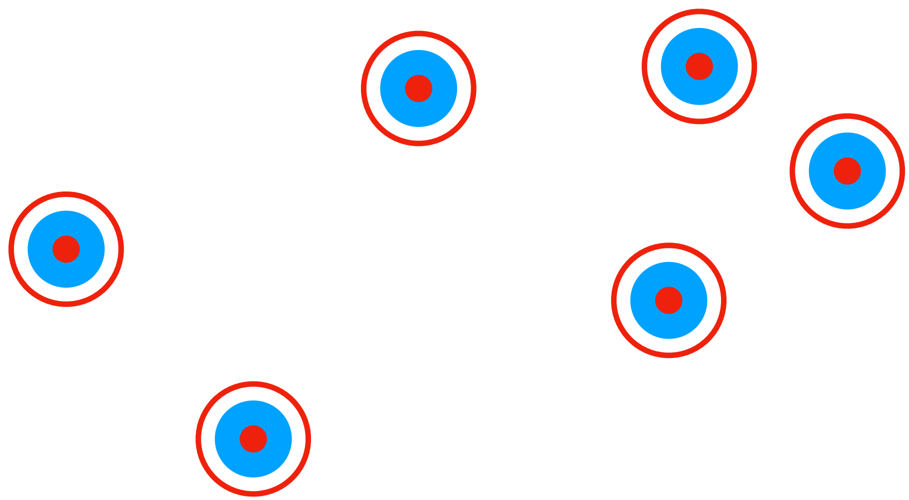


Centroid HAC

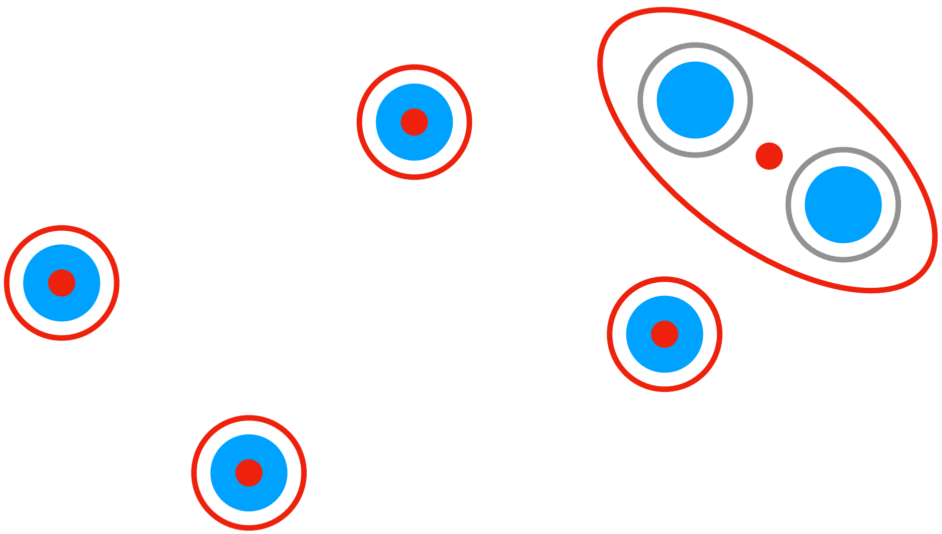
- Initialize $\mathscr{C} = \{\{p\} \mid p \in P\}$
- While $|\mathscr{C}| > 1$:
  - Merge some $\hat{C}_1$ and $\hat{C}_2$ where

$$D\left(\text{cent}(\hat{C}_1), \text{cent}(\hat{C}_2)\right) = \min_{C_1, C_2 \in \mathscr{C}} D\left(\text{cent}(C_1), \text{cent}(C_2)\right)$$

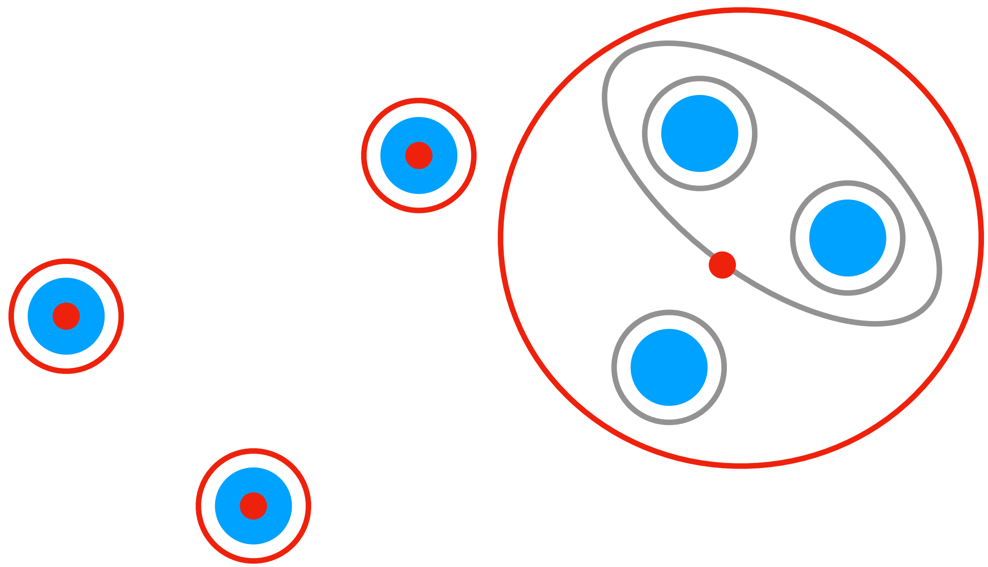
Centroid Hierarchical Agglomerative Clustering



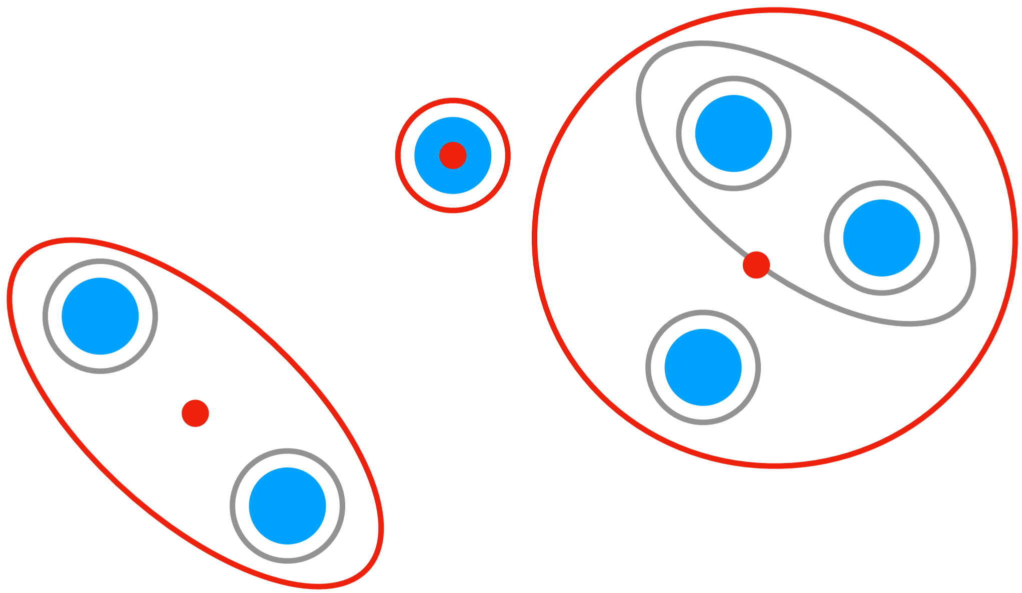
Centroid Hierarchical Agglomerative Clustering



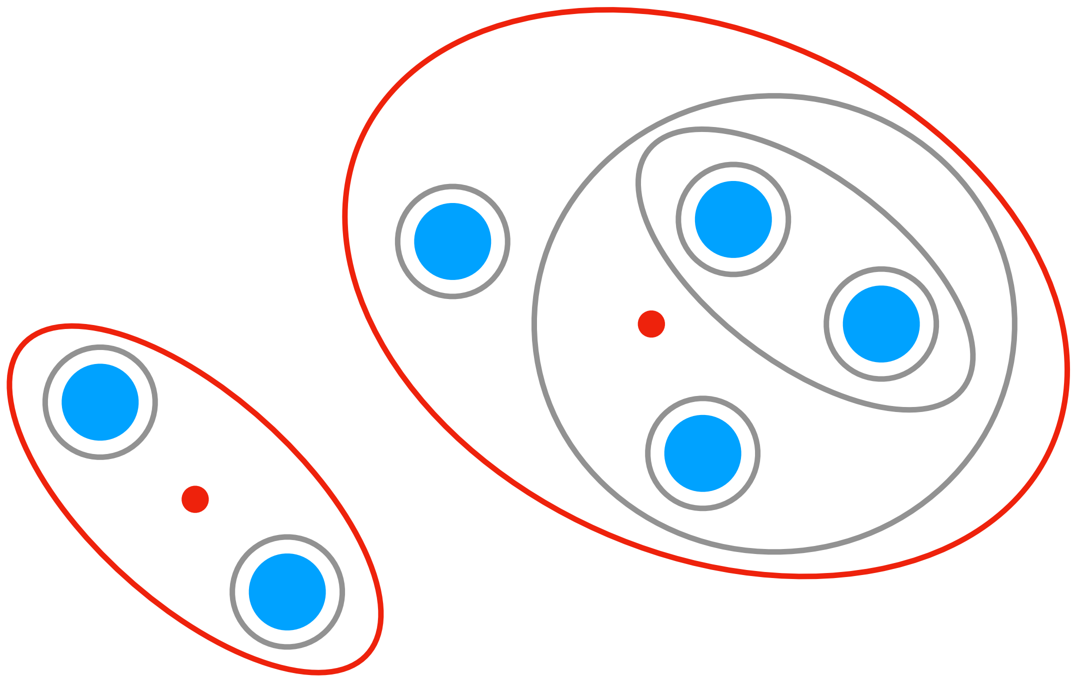
Centroid Hierarchical Agglomerative Clustering



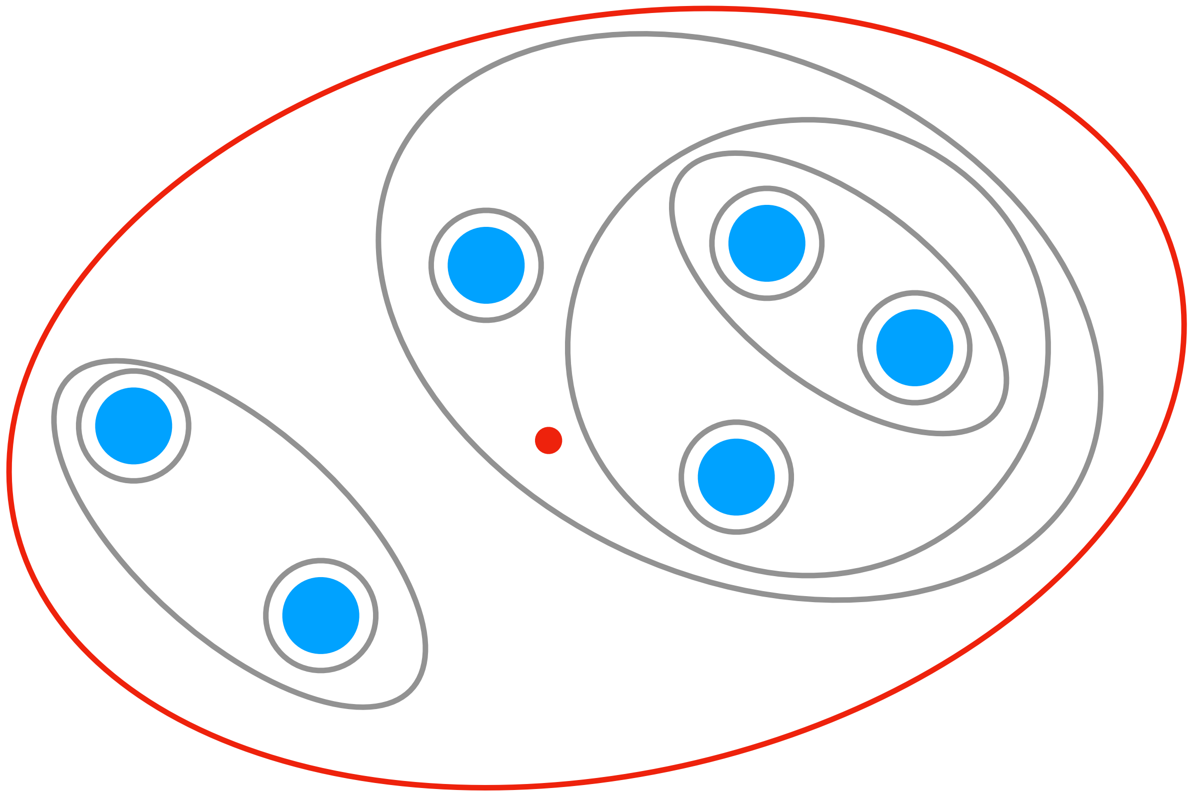
Centroid Hierarchical Agglomerative Clustering



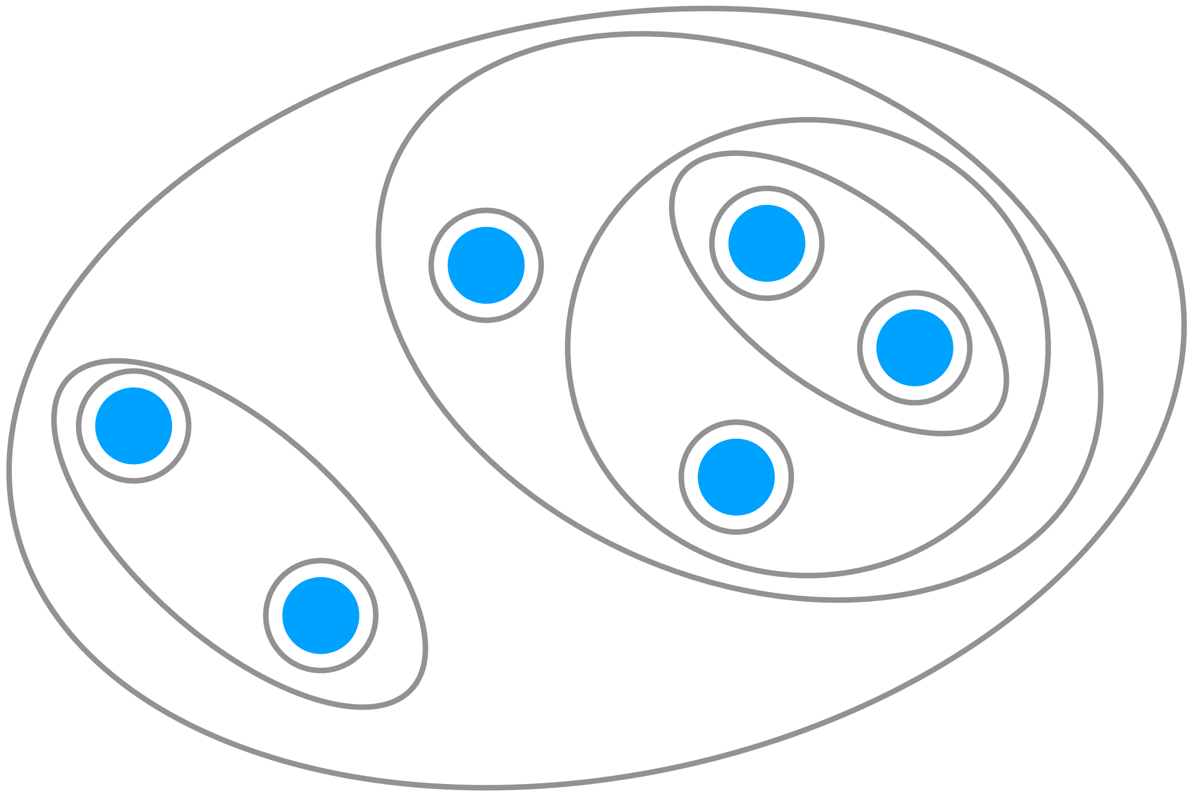
Centroid Hierarchical Agglomerative Clustering



Centroid Hierarchical Agglomerative Clustering



Centroid Hierarchical Agglomerative Clustering



c-Approximate HAC

- Generally impossible to beat quadratic time with exact HAC
- Approximate HAC lets us find sub-quadratic algorithms

c-Approximate HAC

Approximate HAC:

- Initialize $\mathscr{C} = \{\{p\} \mid p \in P\}$
- While $|\mathscr{C}| > 1$:
  - Merge some $\hat{C}_1$ and $\hat{C}_2$ where

$$D\left(\text{cent}(\hat{C}_1), \text{cent}(\hat{C}_2)\right) = c \cdot \min_{C_1, C_2 \in \mathscr{C}} D\left(\text{cent}(C_1), \text{cent}(C_2)\right)$$

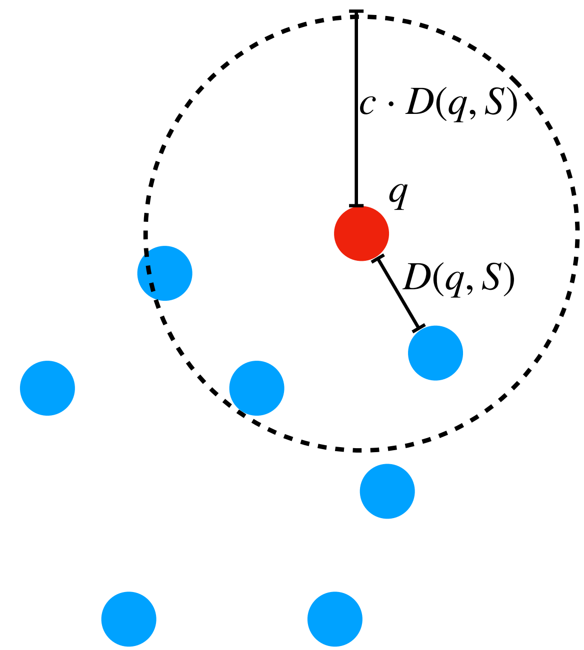
Fast Centroid HAC

Theorem: For n points in $\mathbb{R}^d$ there exists an algorithm for centroid HAC that runs in time $\tilde{O}(n^{1+O(1/c^2)})$.

Dynamic Approximate Nearest Neighbor Search (ANNS)

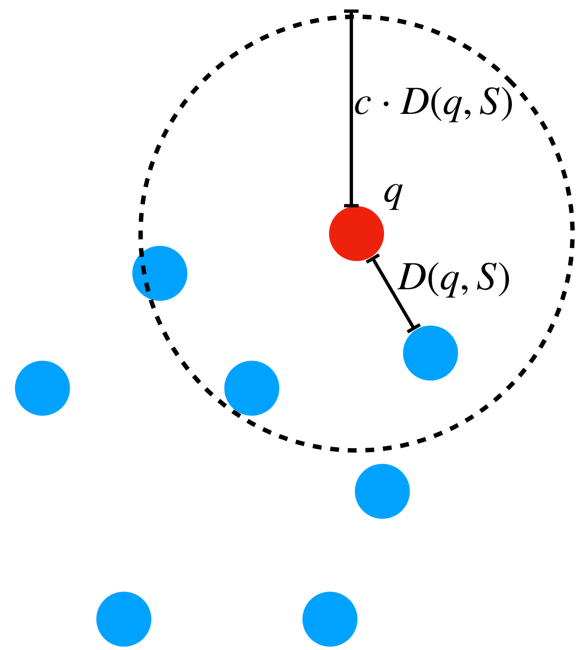
A Dynamic c-approximate NNS data structure $\mathcal{N}$ maintains a dynamically updated set $S \in \mathbb{R}^d$ and given a query point q returns a point $p \in S$ such that:

$$D(q, p) \leq c \cdot D(q, S)$$



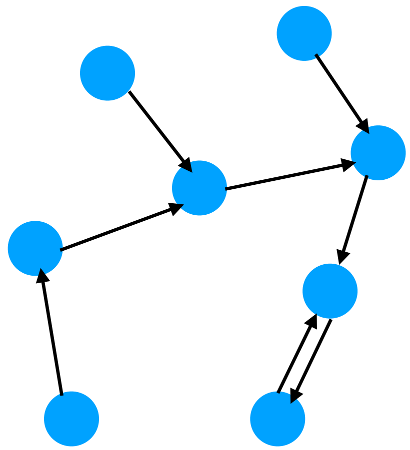
Dynamic Approximate Nearest Neighbor Search (ANNS)

Theorem: There exists a dynamic c-approximate ANNS data structure that supports insertions, deletion, and queries in time $\tilde{O}(n^{1/c^2+o(1)})$.

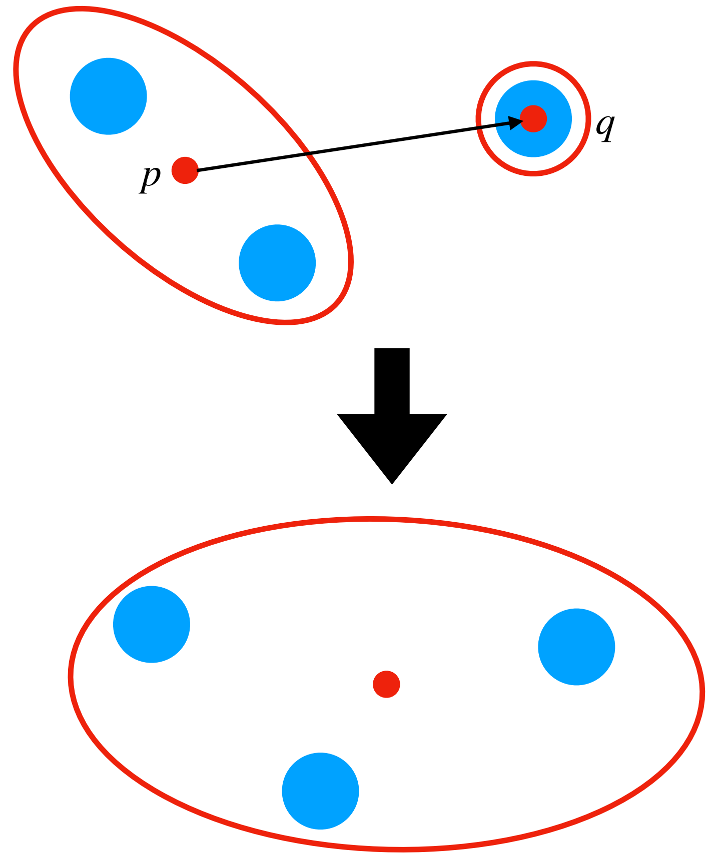


Fast Approximate Centroid HAC

- Let $\mathcal{N}$ be a dynamic ANNS data structure
- For every point $p \in P$, find a near neighbor q and insert this pair into a priority queue Q based on $D(p, q)$
- While Q is not empty:
  - Dequeue the shortest distance $(p, q, D(p, q))$ from Q
  - …



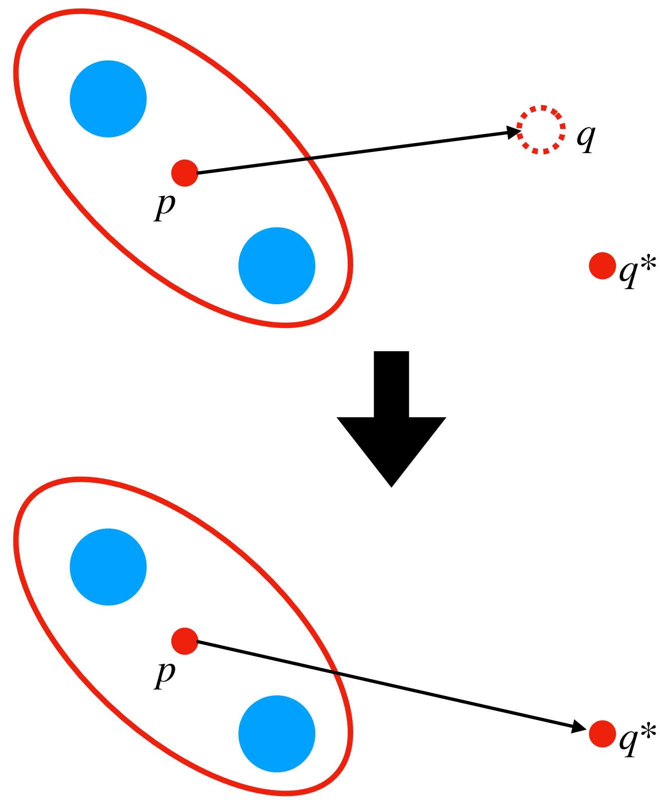
Fast Approximate Centroid HAC



- If p and q are both active centroids:
  - Merge p and q

Fast Approximate Centroid HAC

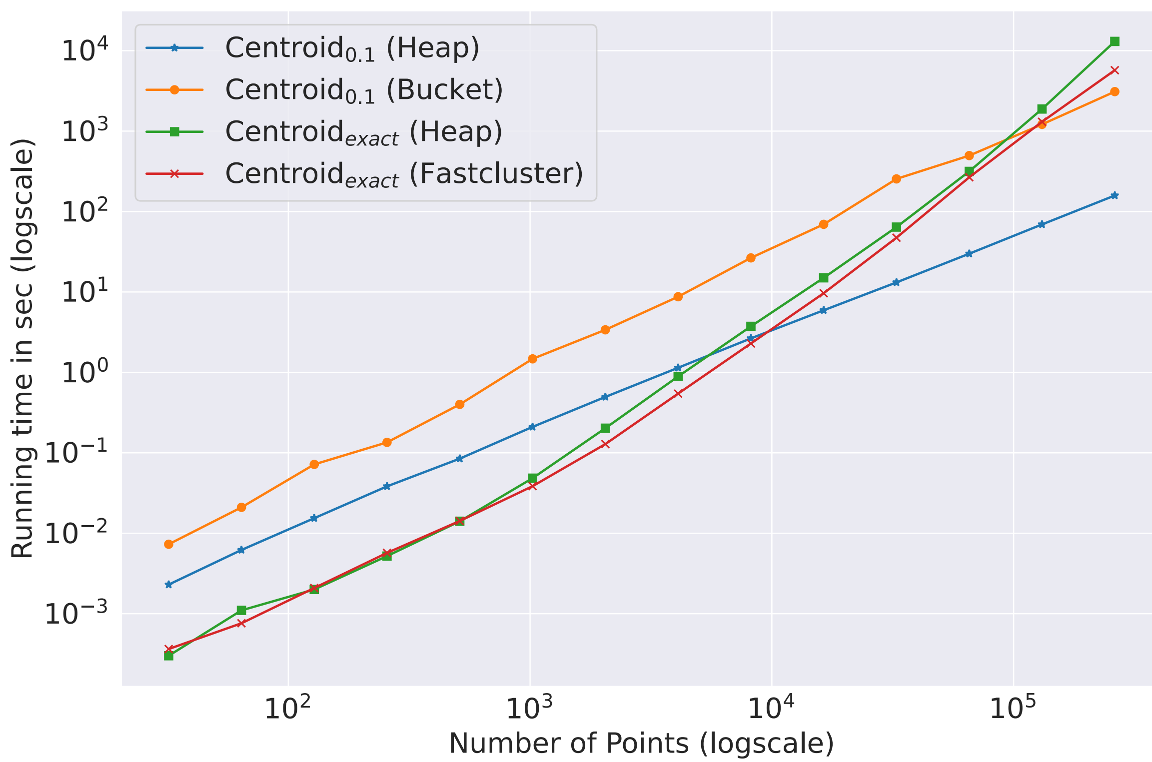
- If p is an active centroid:
  - Find a new near neighbor $q*$
  - If $D(p, q*) \leq (1 + \epsilon) \cdot D(p, q)$:
    - Merge p and $q*$
    - Else: Add $(p, q*, D(p, q*))$ to Q



Experiments: Methods

- We use the same meta-algorithm as for our theoretical results
- Replace the ANNS data structure with one the works well in practice
- Practical ANNS based on DiskANN

Experiments: Runtime



Experiments: Quality

	Dataset	$\text{Centroid}_{0.1}$	$\text{Centroid}_{0.2}$	$\text{Centroid}_{0.4}$	$\text{Centroid}_{0.8}$	Exact Centroid
ARI	iris	**0.759**	0.746	0.638	0.594	**0.759**
	wine	0.352	0.352	**0.402**	0.366	0.352
	cancer	0.509	0.526	0.490	**0.641**	0.509
	digits	0.589	0.571	0.576	**0.627**	0.559
	faces	0.370	0.388	**0.395**	0.392	0.359
	mnist	**0.270**	0.222	0.218	0.191	0.192
	birds	0.449	0.449	0.442	**0.456**	0.441
	Avg	**0.471**	0.465	0.452	0.467	0.453
NMI	iris	**0.803**	0.795	0.732	0.732	**0.803**
	wine	0.424	0.424	0.413	0.389	0.424
	cancer	0.425	0.471	0.459	**0.528**	0.425
	digits	0.718	0.726	0.707	**0.754**	0.727
	faces	0.539	0.534	0.549	0.549	**0.556**
	mnist	0.291	0.282	0.306	**0.307**	0.250
	birds	0.748	0.747	0.756	**0.764**	0.743
	Avg	0.564	0.569	0.560	**0.575**	0.561