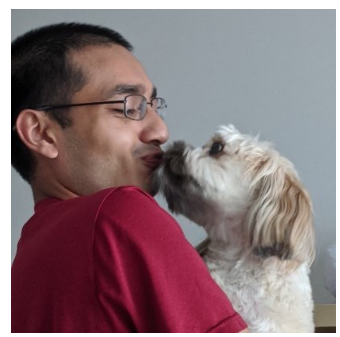
A Bi-metric Framework for Fast Similarity Search

Piotr Indyk (MIT)



Haike Xu
MIT

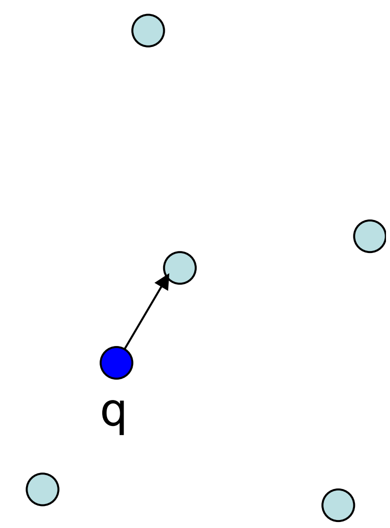
Sandeep Silwal
MIT→U Wisconsin

Nearest Neighbor Search

- **Given:** a set P of n points in some space X under some metric d
- **Goal:** data structure which, given any query q returns $p' \in P$, where

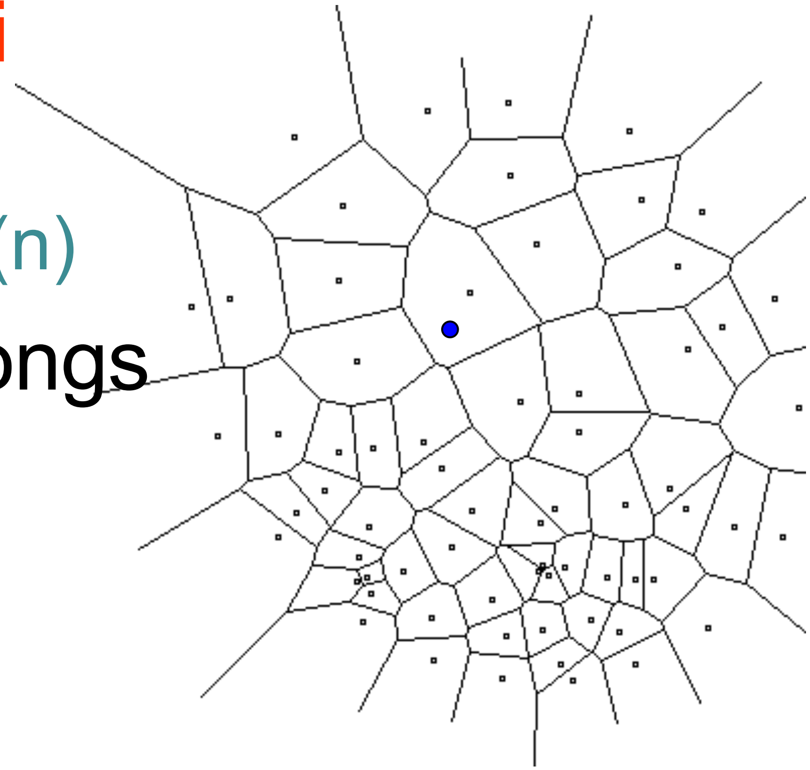
$$d(p',q) \leq \min_{p \in P} d(p,q)$$

- Many applications
- Text retrieval:
  - P={doc1, doc2, doc3, ….}
  - q = query



Example: d=2

- Space partitioning: Voronoi diagram
  - Combinatorial complexity O(n)
- Given q, find the cell q belongs to (point location)
- Performance:
  - Query time: O(log n)
  - Space: O(n)



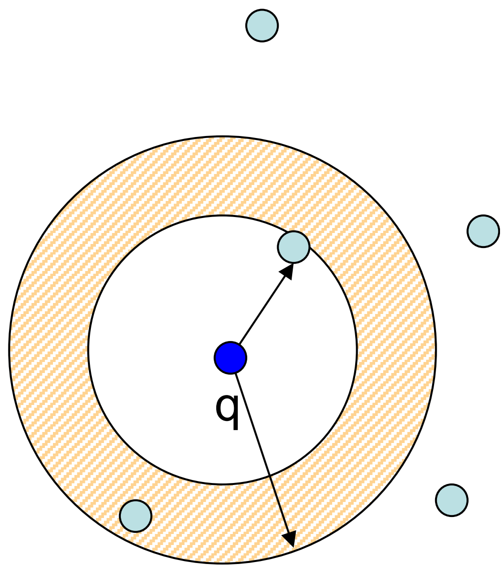
The case of d>2

- Voronoi diagram has size $n^{[d/2]}$
  - NNS data structure with $n^{O(d)}$ space, $(d+\log n)^{O(1)}$ time [Dobkin-Lipton'78,Meiser'93,Clarkson'88]
- We can also perform a linear scan: $O(dn)$ space, $O(dn)$ time
  - Can speedup the scan time by roughly $O(n^{1/d})$
- These are pretty much the only known **general** solutions !
- In fact, exact algorithm with $n^{1-\beta}$ query time for some $\beta>0$ and poly(n) preprocessing would violate certain complexity-theoretic conjecture (Strong Exponential Time Hypothesis)

Relaxation: Theory

- **Given:** a set P of n points in some space X under some metric d, parameter $\varepsilon>0$
- **Goal:** data structure which, given any query q returns $p' \in P$, where

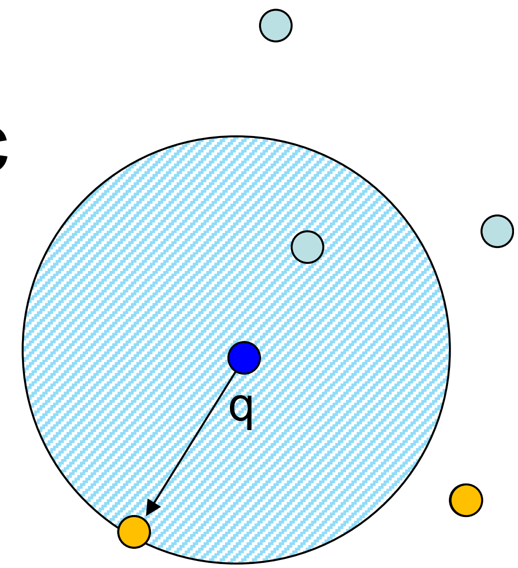
$$d(p',q) \leq (1+\varepsilon) \min_{p \in P} d(p,q)$$



"$(1+\varepsilon)$-approximate nearest neighbor"

Relaxation: Practice

- **Given:** a set P of n points in some space X under some metric d, parameter k
- **Goal:** data structure which returns as many top k nearest neighbors as possible
  - Recall@k: the fraction of top k nearest neighbors returns

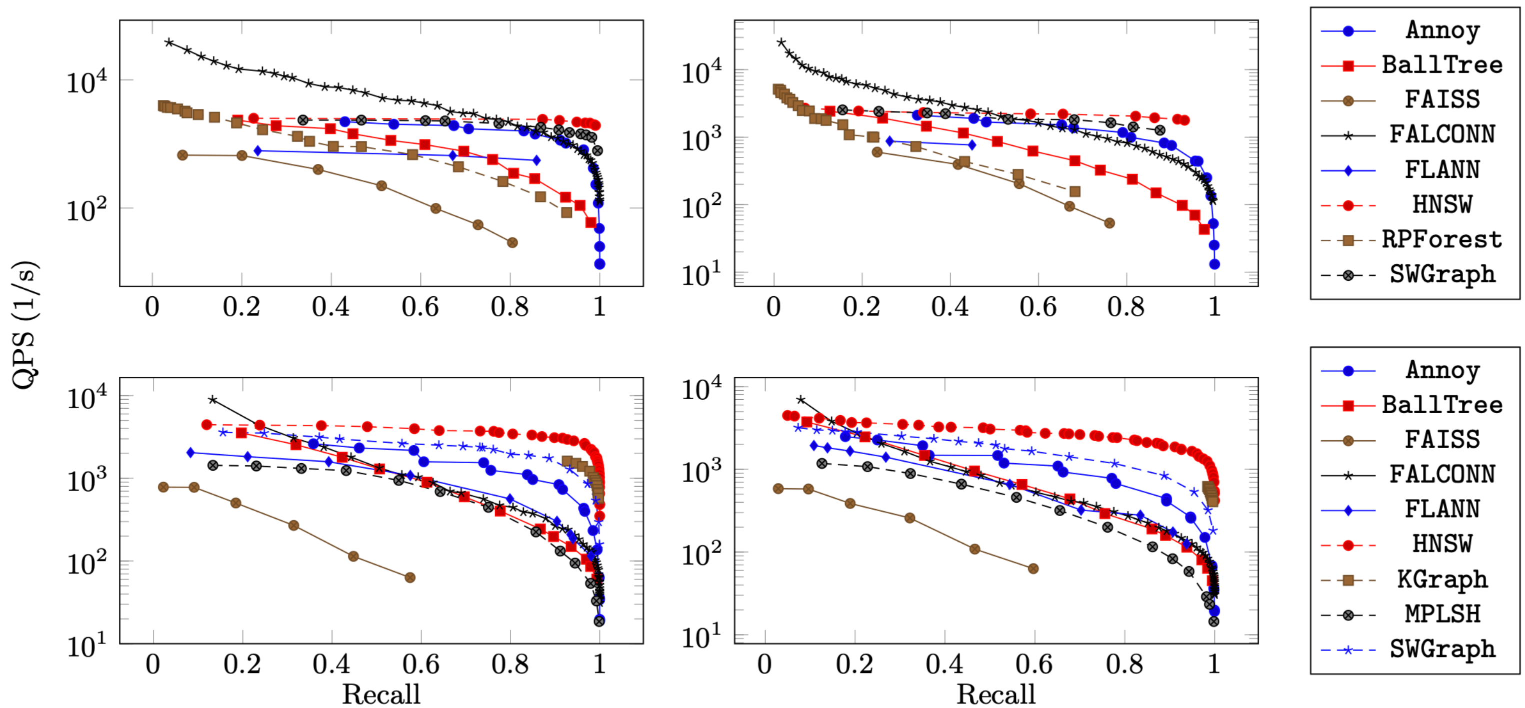


k=2

Recall@k=0.50

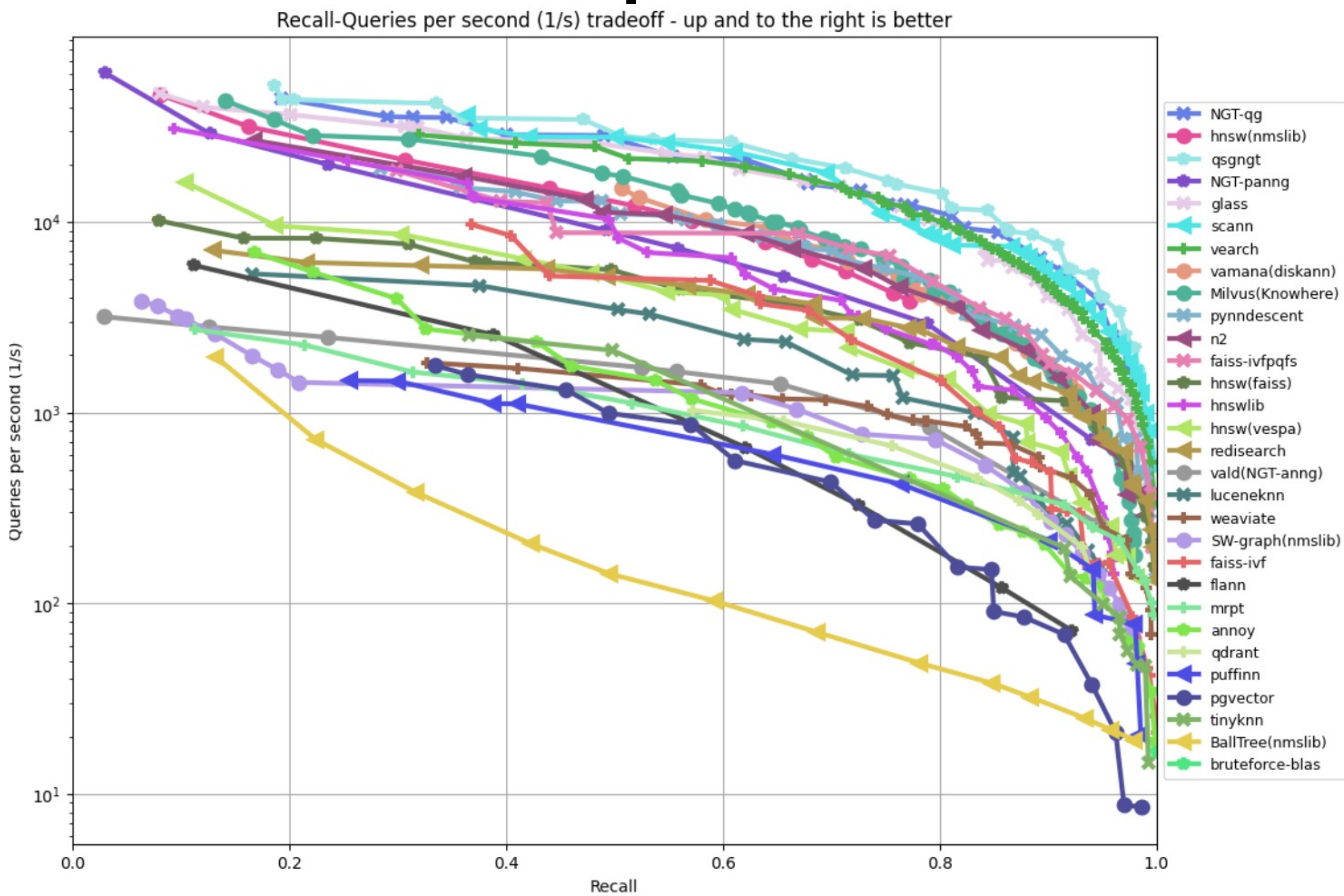
Nearest neighbor algorithms

Landscape in 2017



From: M Aumüller, E Bernhardsson, A Faithfull, ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms, SISAP 2017

Landscape in 2024



From: https://ann-benchmarks.com

Graph-based algorithms

- Recent “empirical wave”: NGT , HNSW, NSG, DiskANN, SSG, Kgraph, DPG, NSW, SPTAG-KDT, EFANNA ….
- Main ideas:
  - **Preprocessing:** create a graph $G=(P,E)$ over points P
  - **Query time:** greedy search, i.e., in each step, move from p to $\text{argmin}_{u \in N(p)} d(q,p)$

Analysis:
Algorithms for doubling metrics

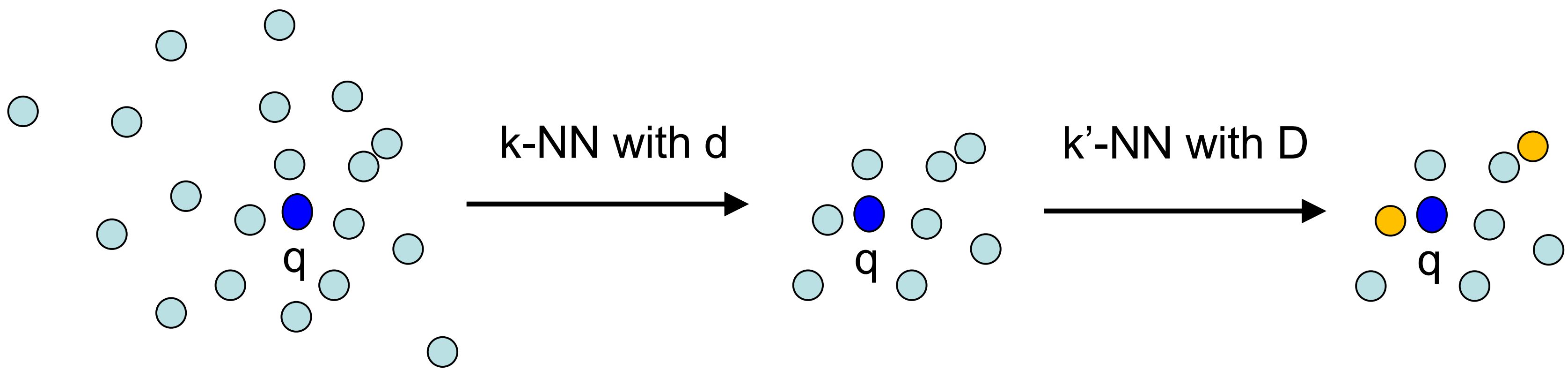
Authors	Space	Query Time
Krauthgamer, Lee'04	$2^{O(dim)}$ n log Δ	$2^{O(dim)}$ log Δ
Krauthgamer, Lee'04	n^2	$2^{O(dim)}$ $\log^2$ n
Har-Peled, Mendel'05	$2^{O(dim)}$ n log n	$2^{O(dim)}$ log n
Beygelzimer, Kakade, Langford'06	n	$2^{O(dim)}$ log Δ
Cole, Gottlieb;06	n	$2^{O(dim)}$ log n
Indyk, Xu'23: DiskANN (slow preprocess)	$2^{O(dim)}$ n log Δ	$2^{O(dim)}$ $\log^2$ Δ
Indyk, Xu'23: all other algorithms		Linear in n (empirically)

Constant approximation factor; bounds up to O(.). Notation:

- dim = logarithm of the number of r-balls needed to cover any 2r-ball (doubling dimension)
- Δ = ratio of max distance to min distance

Nearest neighbor search - usage

Using nearest neighbor search: filtering/reranking



- d – proxy metric (less accurate, cheap to compute)
- D – ground-truth metric (more accurate, expensive)
  - E.g., for our text retrieval experiments, 2-3 orders of magnitude difference between computation times
- Benefits:
  - k' ≪ k, so query time must faster than if k'-NN was done directly on D
  - Preprocessing on d, not D. So cheap to construct; also no need to update when D changes
- Drawbacks:
  - Theory: Suppose that $d \leq D \leq c\, d$. Then cannot guarantee $<c$-approximation
  - Practice: need to do a linear scan over the output of the first stage
- Can we keep benefits and ameliorate drawbacks ?

Results

Improving over reranking: theory

Informal Theorem: Suppose we have a "graph-based" $(1+\varepsilon)$-approximate algorithm, with space $S(n,\varepsilon)$ and query time $Q(n,\varepsilon)$.

Then we get $(1+\varepsilon)$-approximation using space $S(n,\varepsilon/C)$ and query time $Q(n,\varepsilon/C)$.

Authors	Space	Query Time
Beygelzimer, Kakade, Langford	n	$(2/\varepsilon)^{O(dim)} \log \Delta$
DiskANN (slow preprocessing)	$(2/\varepsilon)^{O(dim)}\, n \log \Delta$	$(2/\varepsilon)^{O(dim)} \log^2 \Delta$

↓

n	$(C/\varepsilon)^{O(dim)} \log \Delta$
$(C/\varepsilon)^{O(dim)}\, n \log \Delta$	$(C/\varepsilon)^{O(dim)} \log^2 \Delta$

Improving over reranking: practice

- Text retrieval application
- Experimented with DiskANN algorithm on MTEB benchmark data sets
- For several data sets, state-of-the-art retrieval accuracy using up 4x fewer evaluations of expensive D compared to reranking

Theory - intuition

Informal Theorem expanded: for graph-based $(1+\varepsilon)$-approximate algorithms with space $S(n,\varepsilon)$ and query time $Q(n,\varepsilon)$, if we perform **preprocessing using d** and answer **queries using D** (and modify the algorithms slightly) then we get $(1+\varepsilon)$-approximation using space $S(n,\varepsilon/C)$ and query time $Q(n,\varepsilon/C)$.

Main proof idea:

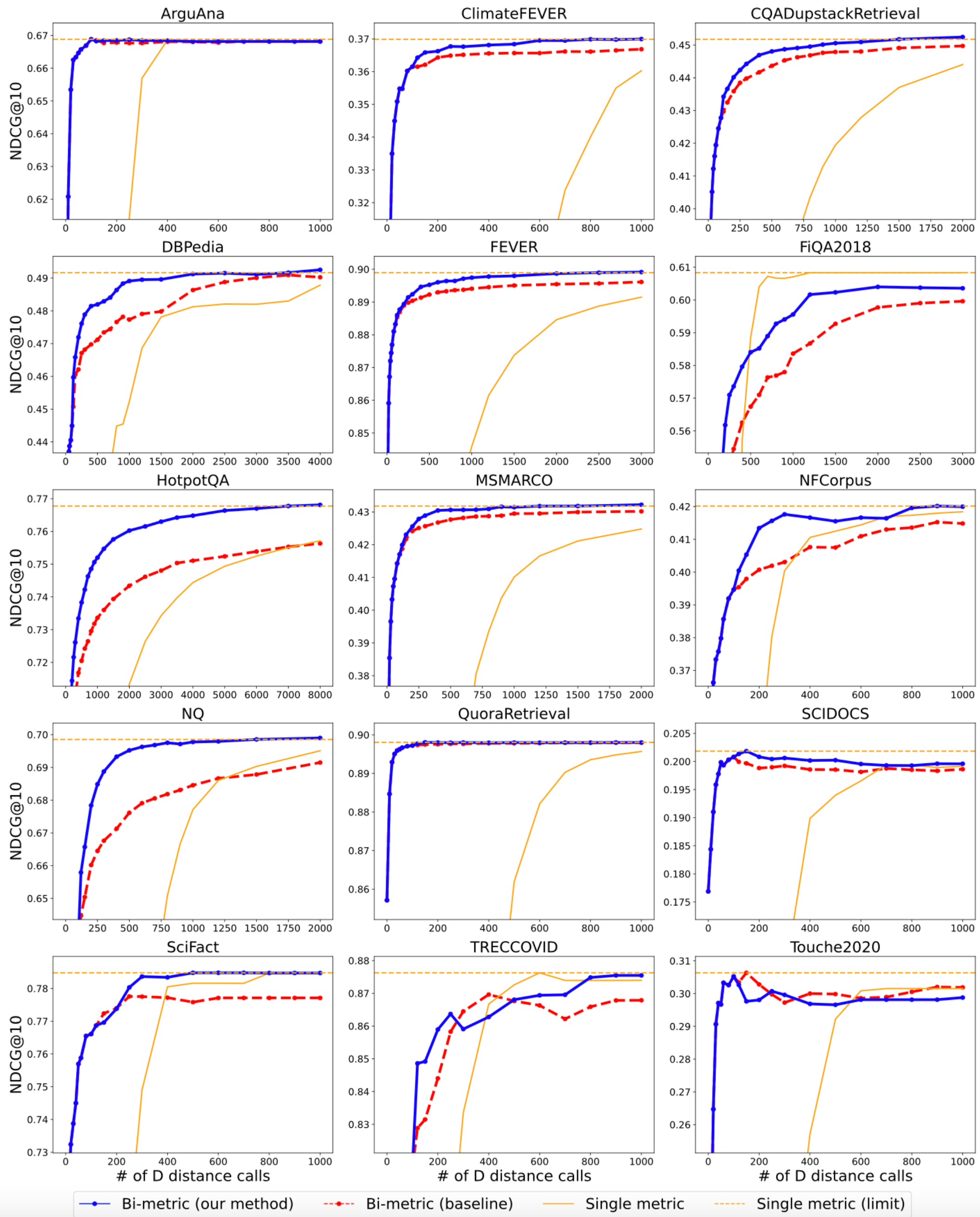
- Graph-based algorithms rely (explicitly or implicitly) on r-nets, i.e., coverings using r-balls
- r-net constructed for d is a Cr-net for D

Text retrieval - setup

- Mostly focused on DiskANN algorithm
- Modify the algo so that it also uses d answering queries
- MTEB benchmark data sets, models from Hugging Face leaderboard (below)

	Model	Model Size (Million Parameters)	Memory Usage (GB, fp32)	Average	ArguAna	ClimateFEVER	CQADupstackRetrieval	DBPedia	FEVER	FiQA2018	HotpotQA
	Linq-Embed-Mistral	7111	26.49	60.19	69.65	39.11	47.27	51.32	92.42	61.2	76.24
	NV-Embed-v1	7851	29.25	59.36	68.2	34.72	50.51	48.29	87.77	63.1	79.92
D	SFR-Embedding-Mistral	7111	26.49	59	67.17	36.41	46.49	49.06	89.35	60.4	77.02
	voyage-large-2-instruct			58.28	64.06	32.65	46.6	46.03	91.47	59.76	70.86
	gte-large-en-v1.5	434	1.62	57.91	72.11	48.36	42.16	46.3	93.81	63.23	68.18
	GritLM-7B	7242	26.98	57.41	63.24	30.91	49.42	46.6	82.74	59.95	79.4
	e5-mistral-7b-instruct	7111	26.49	56.89	61.88	38.35	42.97	48.89	87.84	56.59	75.72
	LLM2Vec-Meta-Llama-3-supervis	7505	27.96	56.63	62.78	34.27	48.25	48.34	90.2	55.33	71.76
	voyage-lite-02-instruct	1220	4.54	56.6	70.28	31.95	46.2	39.79	91.35	52.51	75.51
	gte-Qwen1.5-7B-instruct	7099	26.45	56.24	62.65	44	40.64	48.04	93.35	55.31	72.25
	LLM2Vec-Mistral-supervised	7111	26.49	55.99	57.48	35.19	48.84	49.58	89.4	53.11	74.07
						...					
d	bge-micro-v2	17	0.06	42.56	55.31	25.35	35.07	32.25	74.99	25.59	53.91

Text retrieval - results

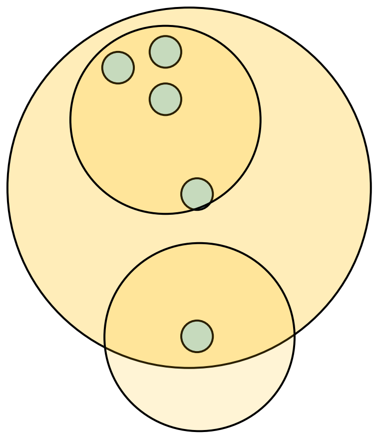


Conclusions

- Bi-metric framework for nearest neighbor search
- Questions:
  - Theory
  - Applications
  - Connections, e.g., to learning-augmented algorithms

Doubling constant

- Consider a metric $M=(X,D)$
- A **doubling constant** of M is the smallest value C such that any ball $B(p,2r)$ can be covered using at most C balls $B(p_1,r)\ldots B(p_C,r)$
  - $d=\log C$ is called **doubling dimension**



- We will also use Δ to denote the ratio of diameter to closest pair distance