Graph-based algorithms for similarity search: challenges and opportunities

Piotr Indyk (MIT)

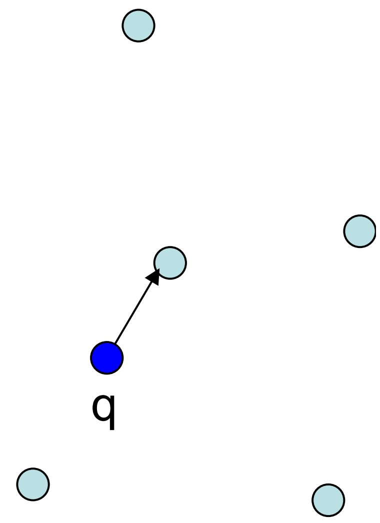
Plan

- Nearest neighbor search: definitions
- Graph-based algorithms: definitions, examples
- Two research vignettes
- Conclusions and open problems

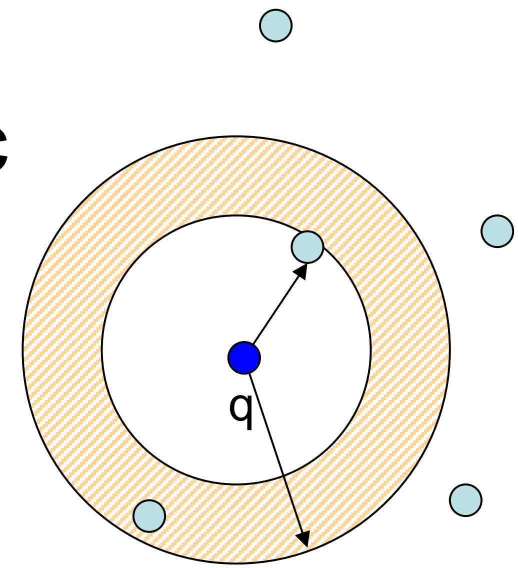
Nearest Neighbor Search

The nearest neighbour search problem arises in numerous fields of application, including:

- Pattern recognition – in particular for optical character recognition
- Statistical classification – see k-nearest neighbor algorithm
- Computer vision
- Computational geometry – see Closest pair of points problem
- Databases – e.g. content-based image retrieval
- Coding theory – see maximum likelihood decoding
- Data compression – see MPEG-2 standard
- Robotic sensing[2]
- Recommendation systems, e.g. see Collaborative filtering
- Internet marketing – see contextual advertising and behavioral targeting
- DNA sequencing
- Spell checking – suggesting correct spelling
- Plagiarism detection
- Similarity scores for predicting career paths of professional athletes.
- Cluster analysis – assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense, usually based on Euclidean distance



Relaxation: Theory



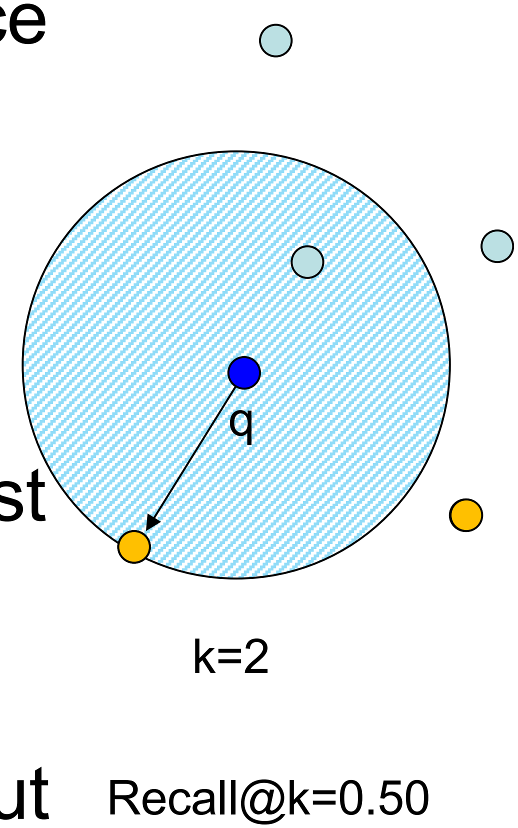
- **Given:** a set P of n points in some space X under some metric d, parameter $\varepsilon>0$
- **Goal:** data structure which, given any query q returns $p' \in P$, where

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

"$(1+\varepsilon)$-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

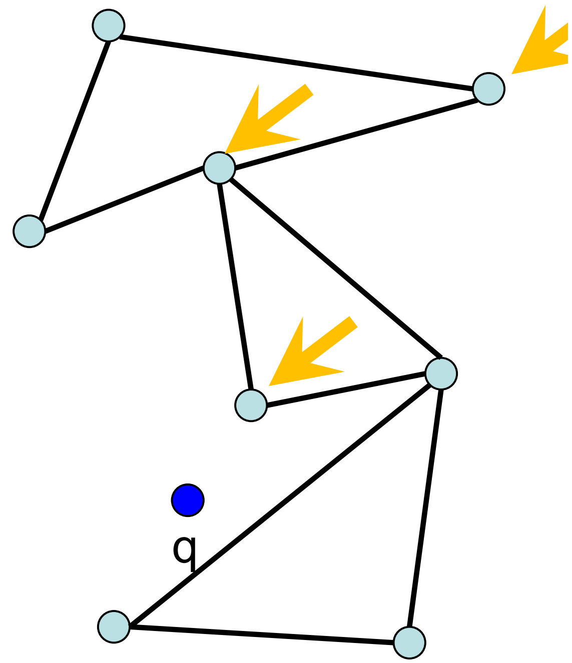


- These two relaxations are correlated, but distinct
- We will use either, depending on the context

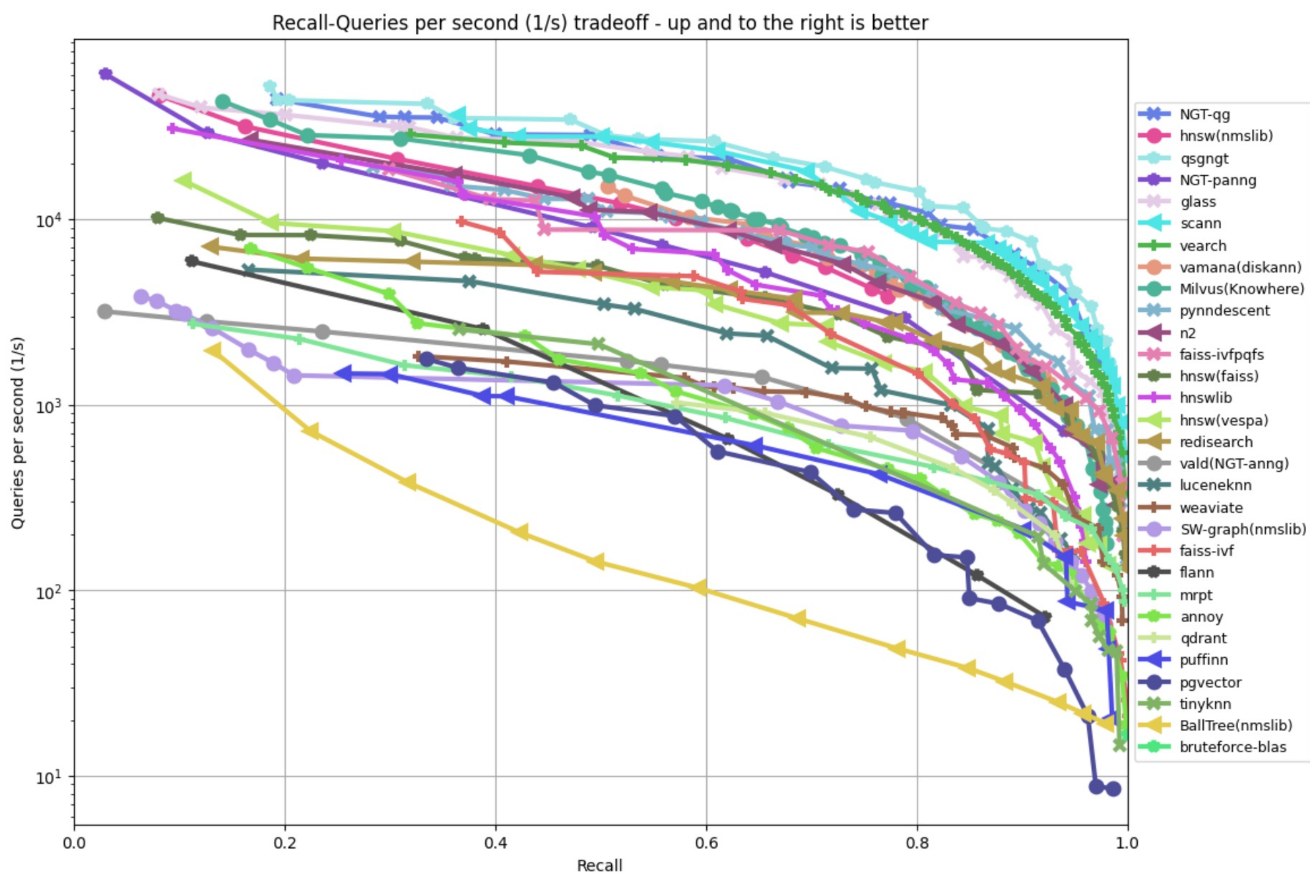
Graph-based algorithms

Graph-based algorithms

- Main ideas:
  - Create a graph G=(P,E) over points P
  - **Greedy search:** in each step, move from p to $\text{argmin}_{(p,u)\in E}\, d(q,p)$
    - Generalized version uses bounded priority queue of size L
- Ideas go back to:
  - Orchard'91 (complete graph)
  - Arya-Mount'93 (for the Euclidean space)
  - Navarro'02 (for exact search)
  - Krauthgammer-Lee'04 (not quite greedy search)
  - See Clarkson'06 for a survey
- Recent wave: HNSW, NSG, DiskANN, NGT, SSG, Kgraph, DPG, NSW, SPTAG-KDT, EFANNA ….



Landscape in 2025



From: https://ann-benchmarks.com

Vignettes/challenges

1. Correctness and/or performance guarantees (Indyk-Xu, NeurIPS'23)
2. Diversity-aware search (Anand, Indyk, Krishnaswamy, Mahabadi, Raykar, Shiragur, Xu, ICML'25)
3. Exploit the power of searching in arbitrary metrics to improve re-ranking (Xu-Silwal-Indyk'24)

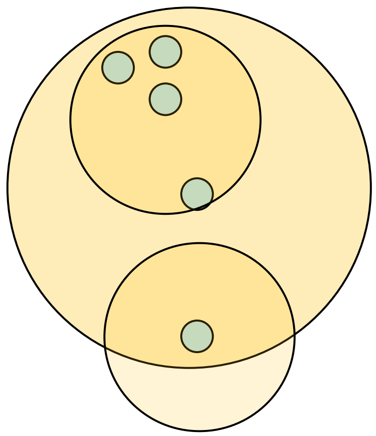
Correctness/performance guarantees

with Haike Xu

MIT

Quantifying intrinsic dimension: doubling constant

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



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

Past results

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

Constant approximation factor; bounds up to O(.)

Can we obtain similar approximation/performance guarantees for popular graph-based algorithms ?

Indyk, Xu'23: DiskANN (slow preprocessing)	$2^{O(dim)}$ n log Δ	$2^{O(dim)} \log^2 \Delta$

Can we obtain guarantees for popular graph-based algorithms ?

- Two answers:
  - – Yes: for DiskANN with “slow” preprocessing
  - – No (empirically): for everything else

DiskANN

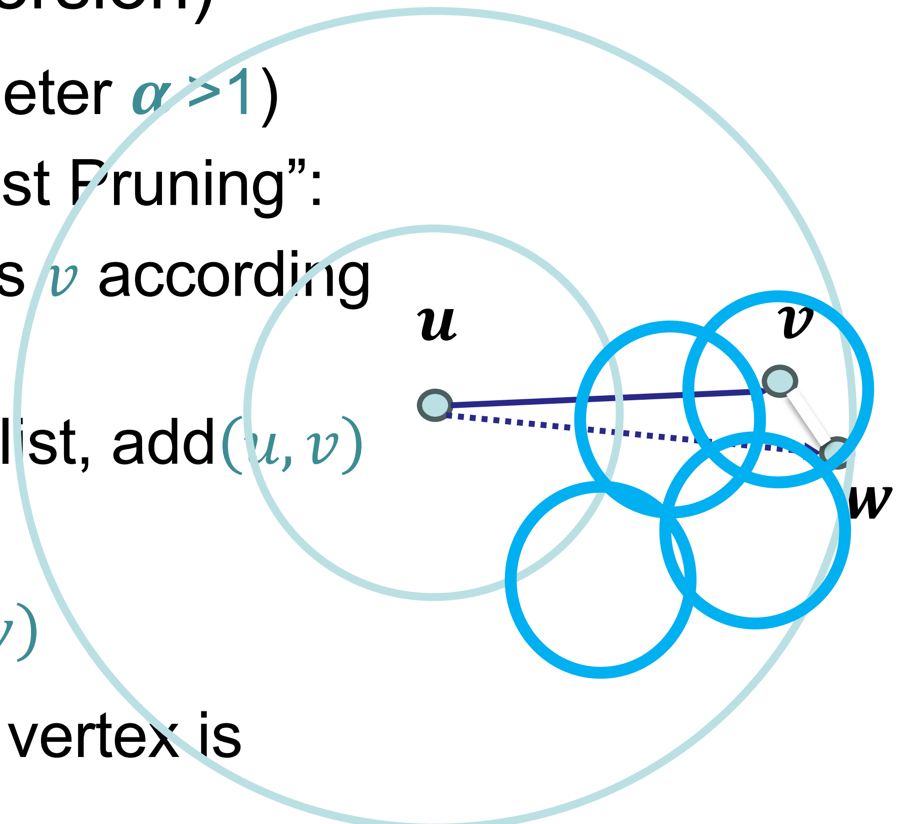
(slow version)

Building the Graph (with parameter $\alpha > 1$)

For each point u, perform "Robust Pruning":

- Create a sorted list of all points v according to $d(u,v)$
- For each point v in the sorted list, add (u,v) and prune all w such that

$$d(v,w) \leq \frac{1}{\alpha} \cdot d(u,w)$$



Space: The out degree of each vertex is

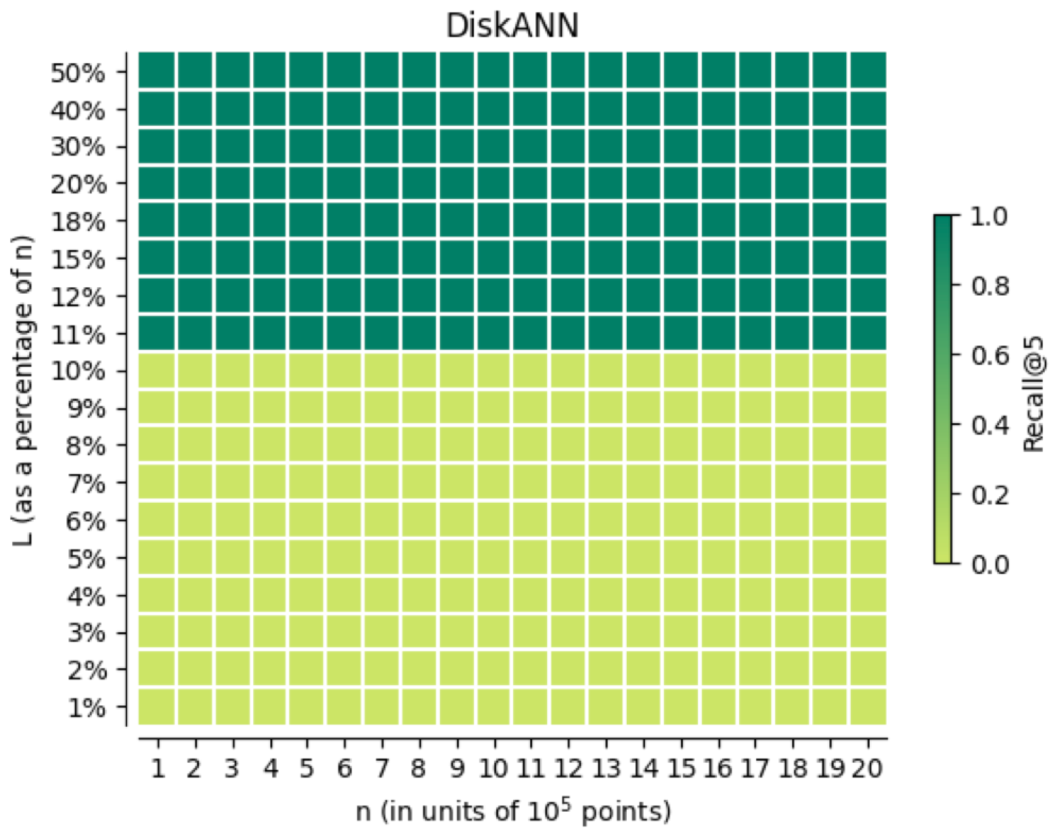
$$\leq (4\alpha)^{dim} \cdot \log \Delta$$

Greedy search: yields a $\left(\frac{\alpha+1}{\alpha-1} + \epsilon\right)$ – approx. solution in $\log_\alpha \frac{\Delta}{\alpha(1-\epsilon)}$ iterations

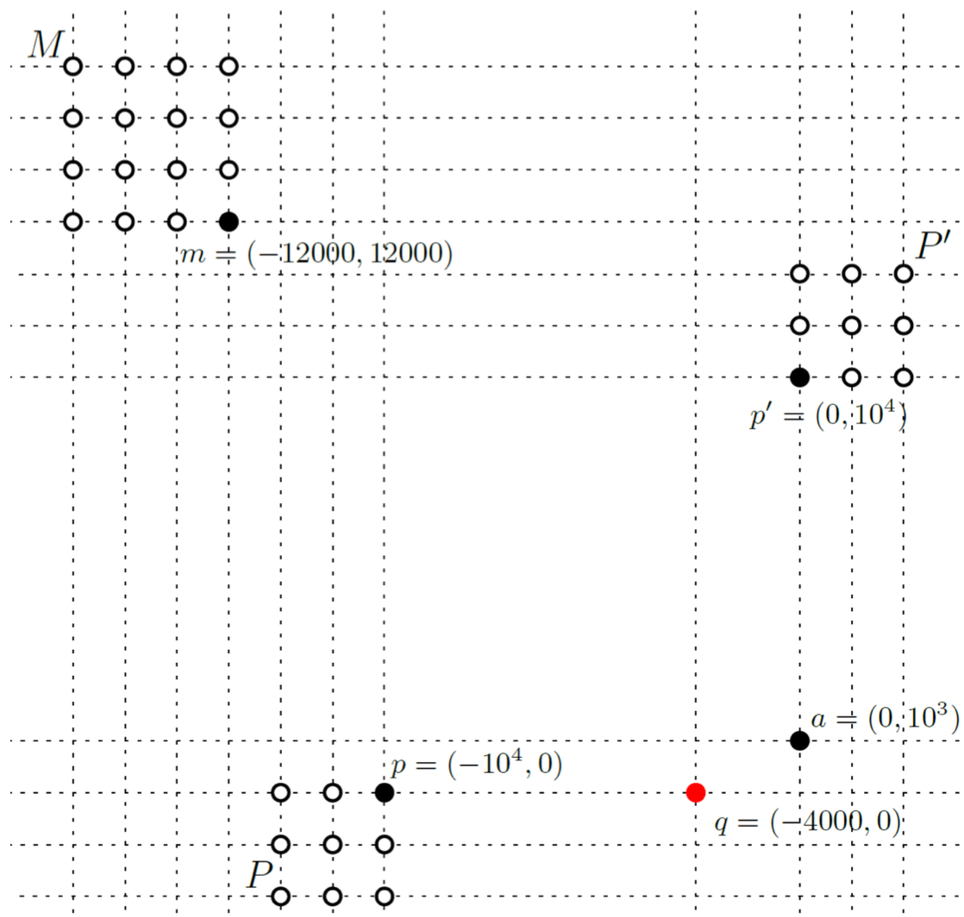
“Fast” DiskANN: Preprocessing

- Initialize the graph to a random R-out graph
- Repeat twice
  - For each $p \in P$
    - Perform greedy search for the nearest neighbor of p
    - Connect p to/from the vertices scanned by greedy search
    - Perform **Robust Pruning** on any node that has >R neighbors, keeping at most R of them
- Does this offer provable guarantees?

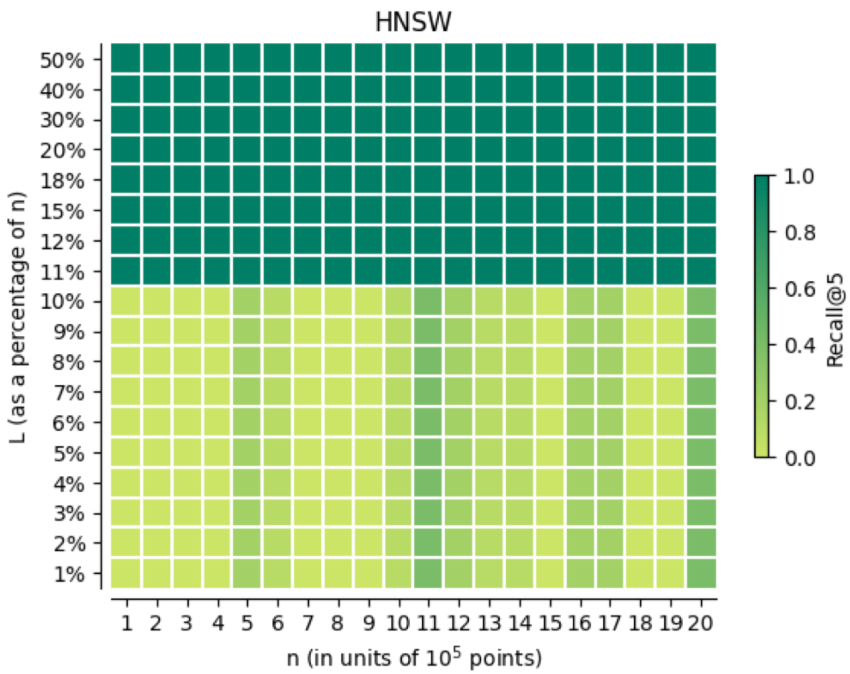
DiskANN with fast preprocessing

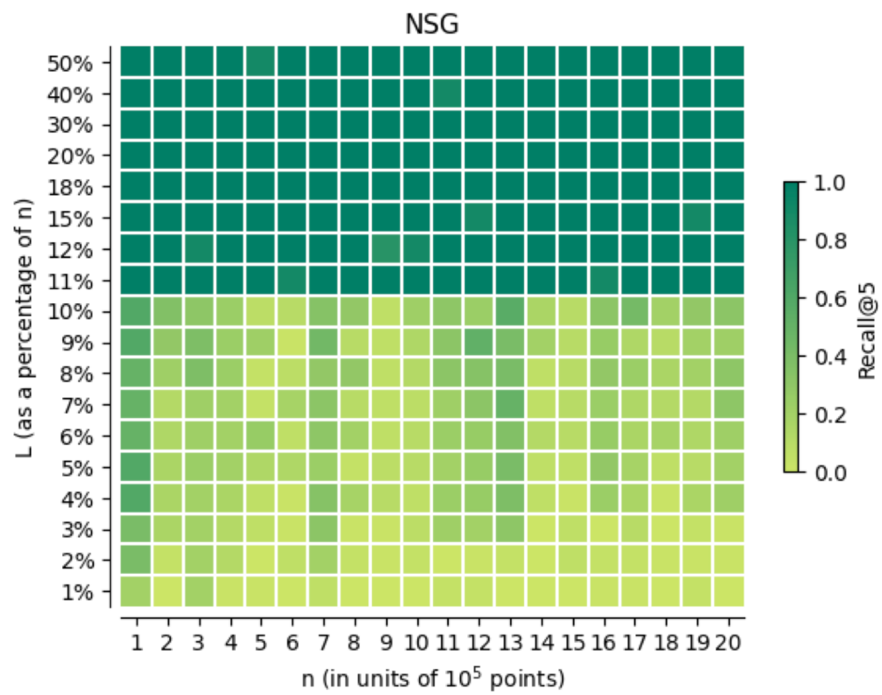


Hard data set for DiskANN



HNSW, NSG



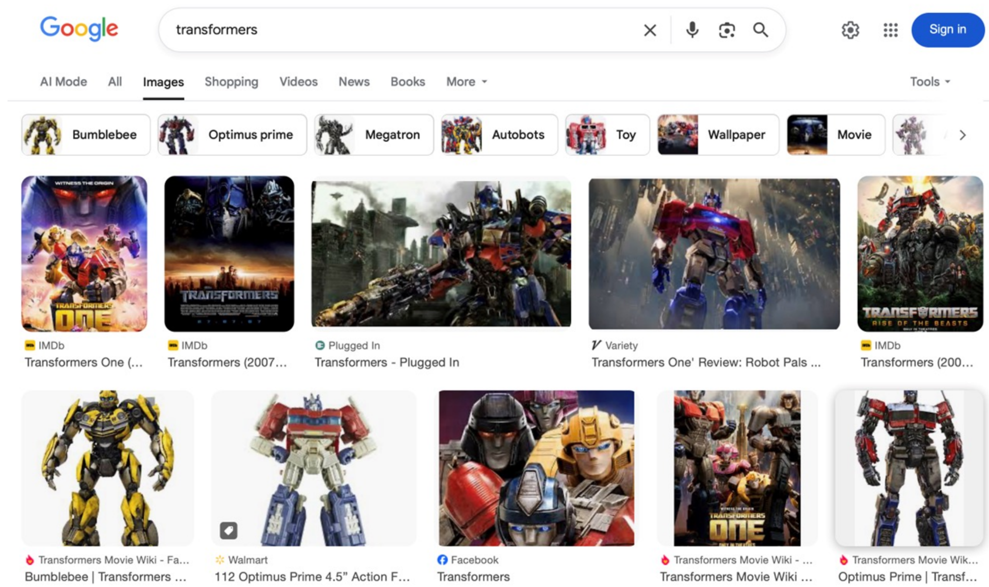


- Similar results hold for NGT, SSG, Kgraph, DPG, NSW, SPTAG-KDT, EFANNA

Wrap up correctness

- Correctness/runtime guarantees exist for **one** variant of **one** graph-based algorithm
- …but not (empirically) for many others
- Questions:
  - **Empirically fast** algorithm with **provable** guarantees ?
  - Other notions of dimensionality (e.g., LID ?)
  - **Generic** counterexamples with **proofs** ?

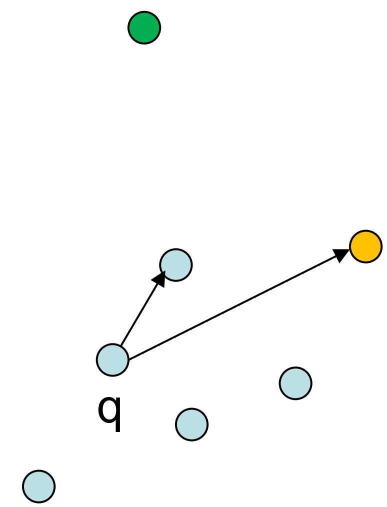
Diversity-aware search

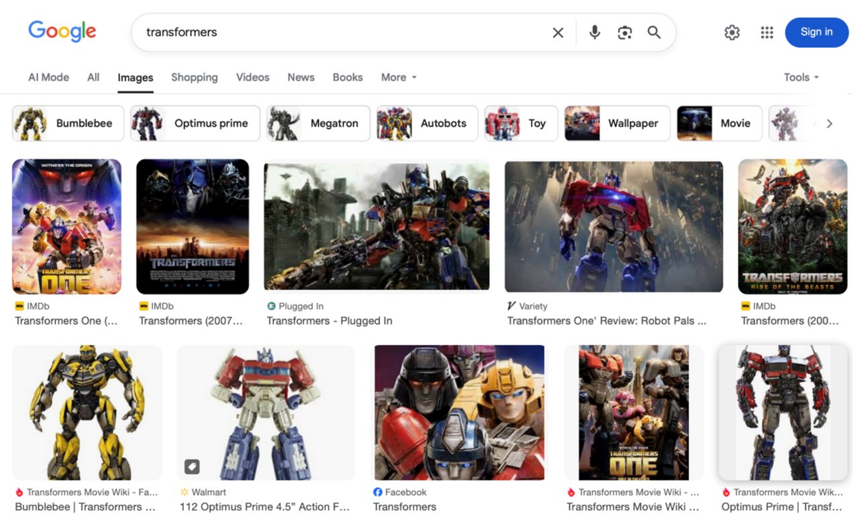


with P. Anand, R. Krishnaswamy,
S. Mahabadi, Raykar, K. Shiragur, H. Xu

Diverse Nearest Neighbor Search

- **Given:** a set P of n points in metric space , with colors and a parameter k
- **Goal:** build a data structure which, given any query q returns k points of different color minimizing the distance to q
- Motivation?





Prior work and our results

Authors	Comment	Space	Query Time
Abbar, Amer-Yahia, Indyk, Mahabadi, Varadarajan'13	d=Hamming approx=O(c)	$\log k \; n^{1+1/c}$	$k^2 \log k \; n^{1/c}$

We were able to modify existing non-diverse DiskANN graph-based algorithm to give the first graph-based algorithms for the diverse problem.

- Space: multiplied by k
- Time: multiplied by k^2 (or k for the colorful version)

Diverse DiskANN

Building the Graph:

- Pruning
  - If $d(v,w) \leq \frac{1}{\alpha} \cdot d(u,w)$
    - Prune the edge (u,w) only if either
      1. $col[v] = col[w]$
      2. Or we have connected u to at least k different colors in the ball of radius $\frac{1}{\alpha} \cdot d(u,w)$ around w

Diverse DiskANN

- **Query answering algorithm:**
  - Start from k points that all have different colors $a_1, \ldots, a_k$
  - In each iteration, **swap one point** a_i with a neighbor point a' that
    - is closer to the query
    - has a different color from the rest of the points

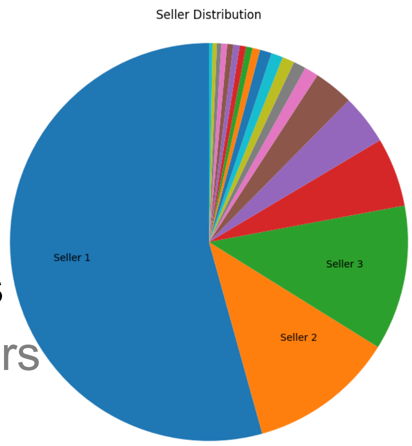
Experiments

Algorithms:

Baseline: Standard DiskANN + Postprocessing to ensure diversity

Our algo 1: Standard DiskANN Build + Diverse DiskANN Search

Our algo 2: Diverse DiskANN Build + Diverse DiskANN Search



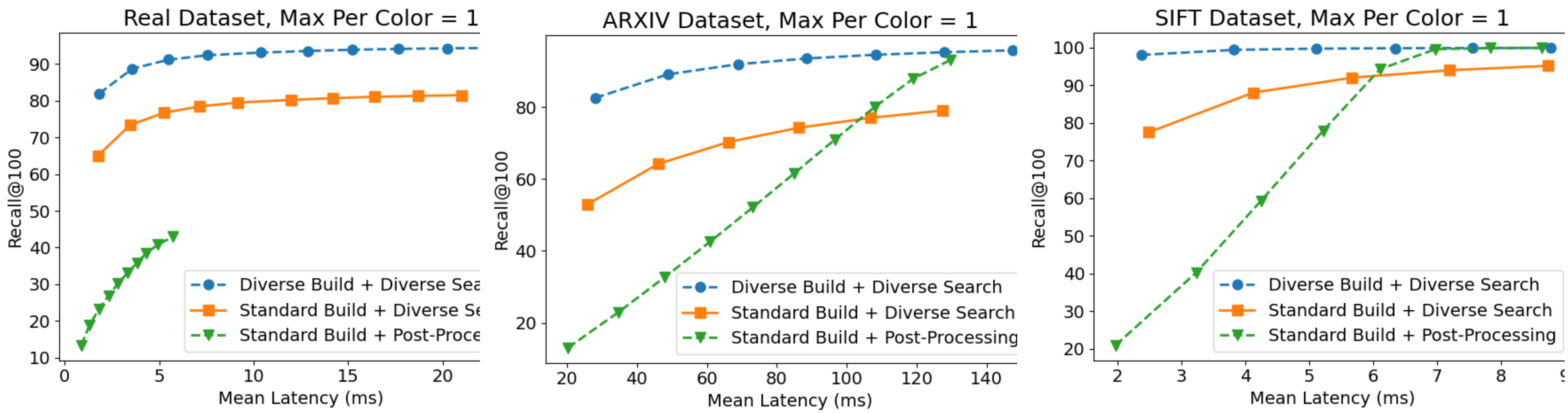
Datasets:

- **Ads dataset**: 20 Million vectors, 5000 queries, 64 dimensions
- **Semi-Synthetic Arxiv**: 2 Million, 1536 dimensions, 1000 colors (uniform on {1,2,3} w.p. 0.9 and uniform on the rest w.p. 0.1)
- **Semi-Synthetic SIFT**: 1 Million, 128 dimensions, 1000 colors (one color w.p. 0.8 and uniform on the rest w.p. 0.2)

Parameters:

- $k = 100$

Experiments: Recall vs Latency



Baseline: Standard DiskANN + Postprocessing to ensure diversity

Our algo 1: Standard DiskANN Build + Diverse DiskANN Search

Our algo 2: Diverse DiskANN Build + Diverse DiskANN Search

Wrap up diversity

- Extended (slow) DiskANN to diverse nearest neighbor
  - Space: multiplied by k. Time: multiplied by k^2 (or k for colorful variant).
- Questions:
  – Better space/time bounds? Recent result by Samson Zhou's group for the colorful version:
    - Space: multiplied by $\log k$. Time: multiplied by k. (but randomized)
    - Constant space overhead?

FOCS 2025 workshop

(with Raj Jayaram, Ravi Krishnaswamy)

Sunday, December 14, 2025 • FOCS, Sydney

Approximate Nearest Neighbor Search

Bridging Theoretical Foundations and Industrial Frontiers. A workshop bringing together researchers and practitioners to discuss the latest advancements in ANNS.

View Schedule

Presenters

THANKS!