Approximate Nearest Neighbors in Limited Space

Piotr Indyk
MIT

Tal Wagner
MIT

Introduction

What is the space complexity of the (Euclidean) Approximate Nearest Neighbor problem?

Problem: Compress a dataset $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ into a small size data structure (sketch) that can answer $(1+\epsilon)$-approximate nearest neighbor queries:

$$\textit{Given } y \in \mathbb{R}^d, \ \textit{return } i^* \in \{1,\dots,n\} \ \textbf{s.t. } \|y - x_{i^*}\| \le (1+\epsilon) \cdot \min_{i \in \{1,\dots,n\}} \|y - x_i\|.$$

Benefits of compression:
- **Time:** Speed-up linear scan of data.
- **Space:** Fit on memory-limited devices like GPUs (*Johnson, Douze, Jégou (2017)*).
- **Communication:** Facilitate distributed architectures.

Context:
- Nearest neighbor classifiers are popular in Machine Learning (eg. *Efros (2017)*).
- Large body of **empirical** work on the above problem (see survey at *Wang et al. (2016)*).
- Yet, no better **theoretical** bounds than the dimension reduction theorem due to *Johnson & Lindenstrauss (1984)* were previously known.

Our Results

Problem 1 – Approximate Nearest Neighbor:
Answer query with success probability $1 - 1/n^{O(1)}$.

Method	Size in bits per point*	What can it approximate?
No compression	$O(d \log n)$	Distances between any y and all $x \in X$
Johnson & Lindenstrauss (1984)	$O\left(\frac{\log^2 n}{\epsilon^2}\right)$	Distances between any y and all $x \in X$
Kushilevitz, Ostrovski, Rabani (2000)	$O\left(\frac{\log n}{\epsilon^2} \cdot \log R\right)$	Distances between any y and all $x \in X$, **assuming** $\|x - y\| \in [r, Rr]$
Indyk & Wagner (2017; 2018)	$O\left(\frac{\log n}{\epsilon^2}\right)$	Distances between all $x, y \in X$, **no out-of-sample query support**
This work	$O\left(\frac{\log n}{\epsilon^2} \cdot \log(1/\epsilon)\right)$	Nearest neighbor of any y in X

Problem 2 – Approximate Distance Queries:
Compress X such that for any query set $Y \subset \mathbb{R}^d$ with q query points, the sketch can estimate all distances $\|x - y\|$ for $x \in X$ and $y \in Y$, up to distortion $(1 \pm \epsilon)$.

Reference	# queries	Size in bits per point*
Molinaro, Woodruff, Yaroslavtsev (2013)	$q \ge n$	$\Omega\left(\frac{\log^2 n}{\epsilon^2}\right)$ matches the *Johnson-Lindnestrauss (1984)* upper bound for $q = n^{O(1)}$.
This work	$1 \le q \le n$	$O\left(\frac{\log n}{\epsilon^2}(\log q + \log(1/\epsilon))\right)$
		$\Omega\left(\frac{\log n}{\epsilon^2} \cdot \log q\right)$

* *For simplicity, the bounds stated in this poster assume that all points coordinates in X are represented by $O(\log n)$ bits. See the paper for the full dependence on all parameters.*

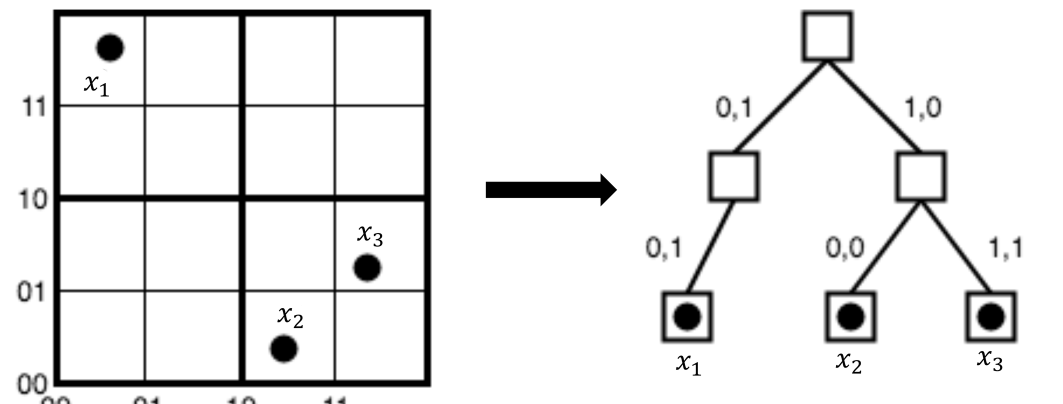
Overview of Techniques

For this poster, we use a simplified sketch due to *Indyk, Razenshteyn, Wagner (2017)*.

➢ Lossier than *Indyk & Wagner (2017)* by $O(\log \log n)$, but simpler and captures main ideas.

The dataset X is represented by a hierarchical clustering tree.

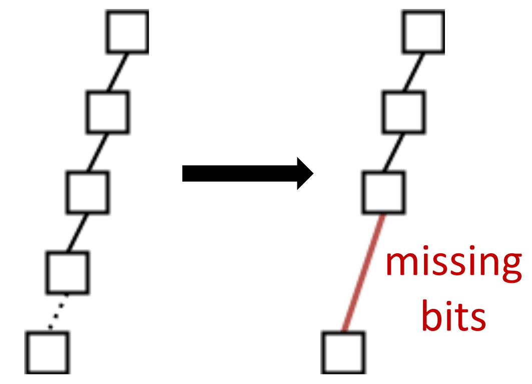
Tree edges are annotated with binary precision bits of point coordinates in X.



How to compress the tree?

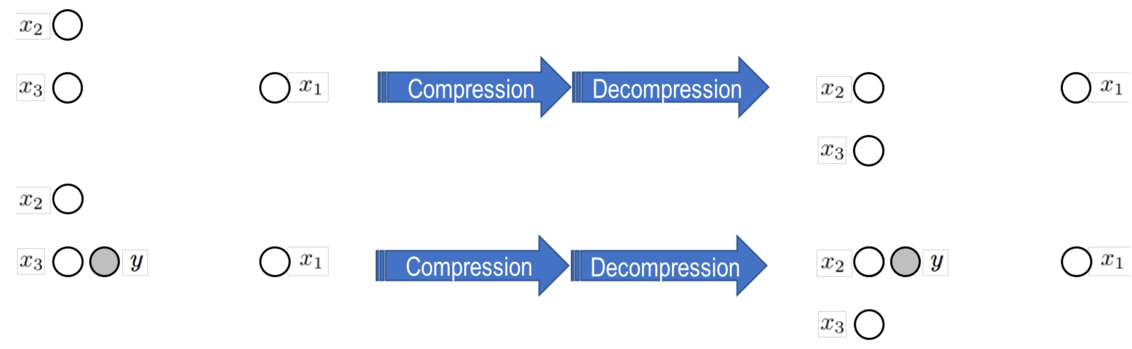
Prior work: "Bottom-out Compression"
Remove every non-branching path from the tree, except its **top** edges.

➢ Stores most significant bits of each cluster.
➢ Preserves global cluster structure.



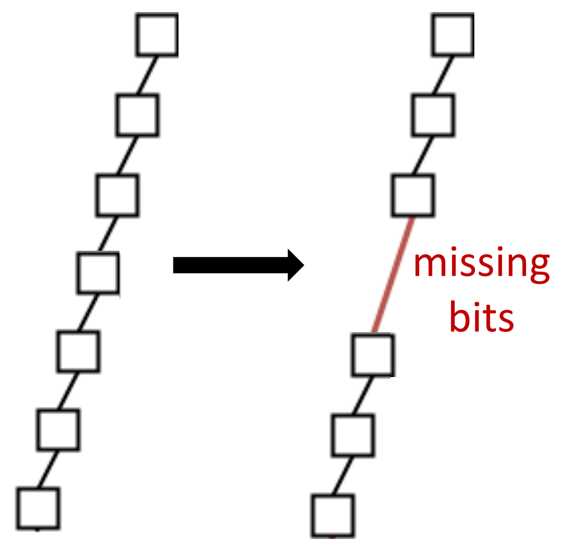
This preserves distances within X:

but not the nearest neighbor of a new query point y:



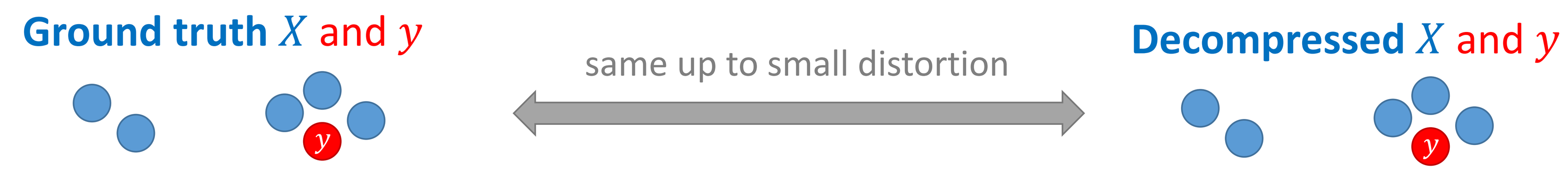
This work: "Middle-Out Compression"
Remove every non-branching path from the tree, except its **top and bottom** edges.

➢ Also stores least significant bits of each cluster.
➢ Also preserves local cluster structure.

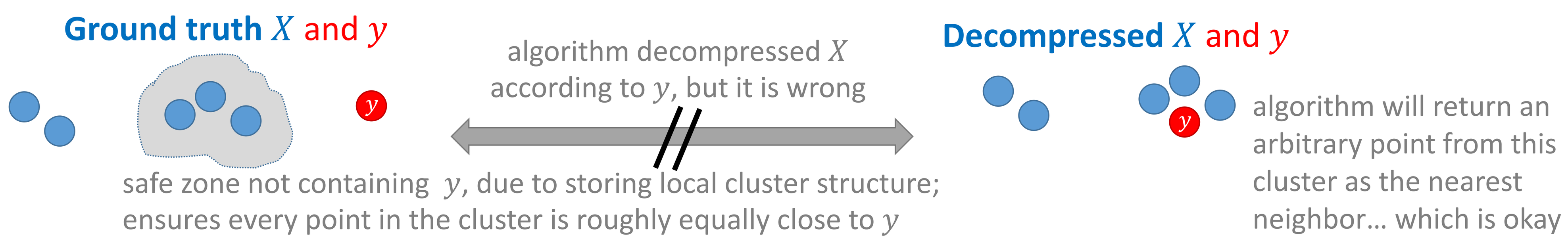


Overview of Analysis

Approximate nearest neighbor algorithm for a query point $y \in \mathbb{R}^d$:
- Search for y down the tree, by the bits on the tree edges, until reaching a leaf.
- Return the point in X represented by that leaf.
- How to handle missing bits in the tree? **Guess they are the same as** y.
- **Guessed right? Yay!** The algorithm learned the right absolute location of X from y.



- **Guessed wrong? It's okay.** The algorithm doesn't know it learned X wrong, but any point from now on is a good approximate nearest neighbor.



References

A Efros, **How to stop worrying and learn to love Nearest Neighbors**. Talk at NIPS workshop, 2017.
P Indyk, I Razenshteyn, T Wagner, **Practical data-dependent metric compression with provable guarantees**. NIPS 2017.
P Indyk, T Wagner, **Near-optimal (Euclidean) metric compression**. SODA 2017.
P Indyk, T Wagner, **Optimal (Euclidean) metric compression**. Unpublished preprint, 2018.
J Johnson, M Douze, H Jégou, **Billion-scale similarity search with GPUs**. ArXiv preprint, 2017.
W Johnson, J Lindenstrauss, **Extensions of Lipschitz mappings into a Hilbert space**. Contemporary Mathematics, 1984.
E Kushilevitz, R Ostrovsky, Y Rabani, **Efficient search for approximate nearest neighbor in high dimensional spaces**. SIAM Journal on Computing, 2000.
M Molinaro, DP Woodruff, G Yaroslavtsev. **Beating the direct sum theorem in communication complexity with implications for sketching**. SODA 2013.
J Wang, W Liu, S Kumar, SF Chang, **Learning to hash for indexing big data—a survey**. Proceedings of the IEEE, 2016.