Approximate Nearest Neighbors in Limited Space

Piotr Indyk
MIT

Tal Wagner
MIT

Introduction

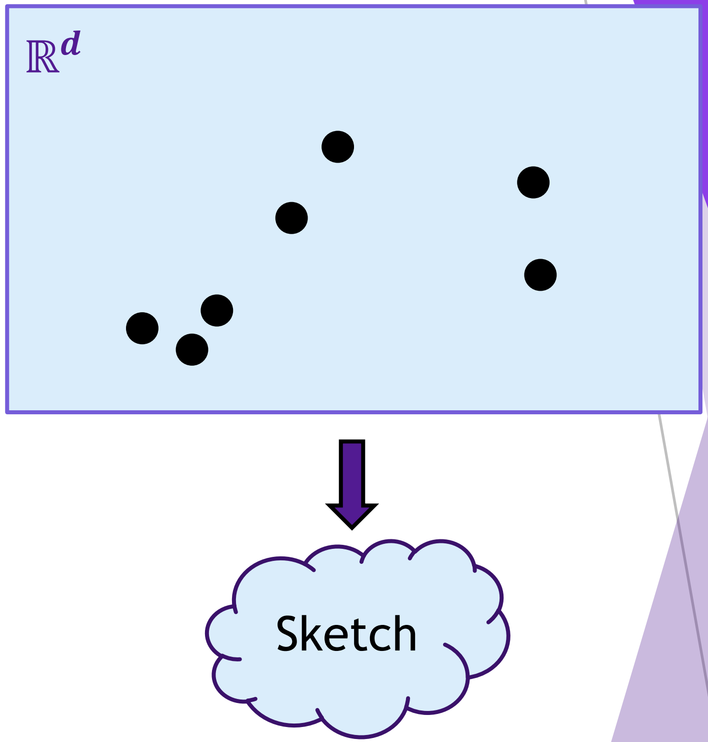
What is the space complexity of the (Euclidean) Approximate Nearest Neighbor Problem?

Introduction

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

$(1+\epsilon)$-Approximate Nearest Neighbor problem:

- Preprocess:
  - Input: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$
  - Output: small-size sketch



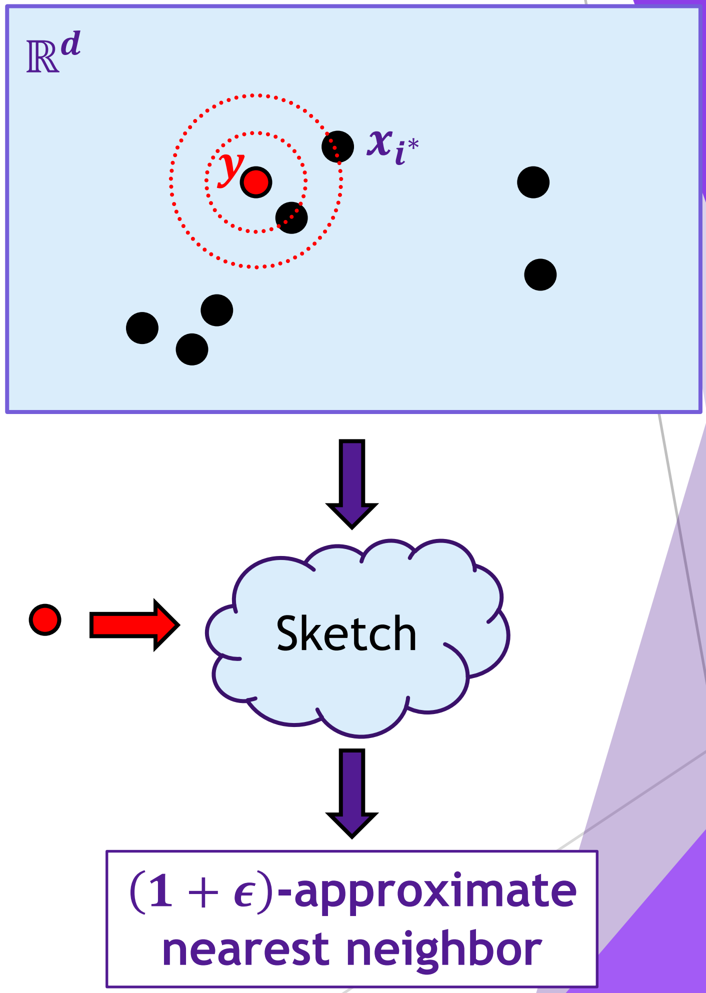
Introduction

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

$(1+\epsilon)$-Approximate Nearest Neighbor problem:

- **Preprocess:**
  - Input: $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$
  - Output: small-size sketch
- **Query:**
  - Input: $\boldsymbol{y} \in \mathbb{R}^d$
  - Output: $\boldsymbol{i}^* \in \{\mathbf{1}, \dots, \boldsymbol{n}\}$

  s.t. $\|\boldsymbol{y} - \boldsymbol{x}_{i^*}\| \le (\mathbf{1}+\epsilon) \min_{j \in \{1,\dots,n\}} \|\boldsymbol{y} - \boldsymbol{x}_j\|$



Introduction

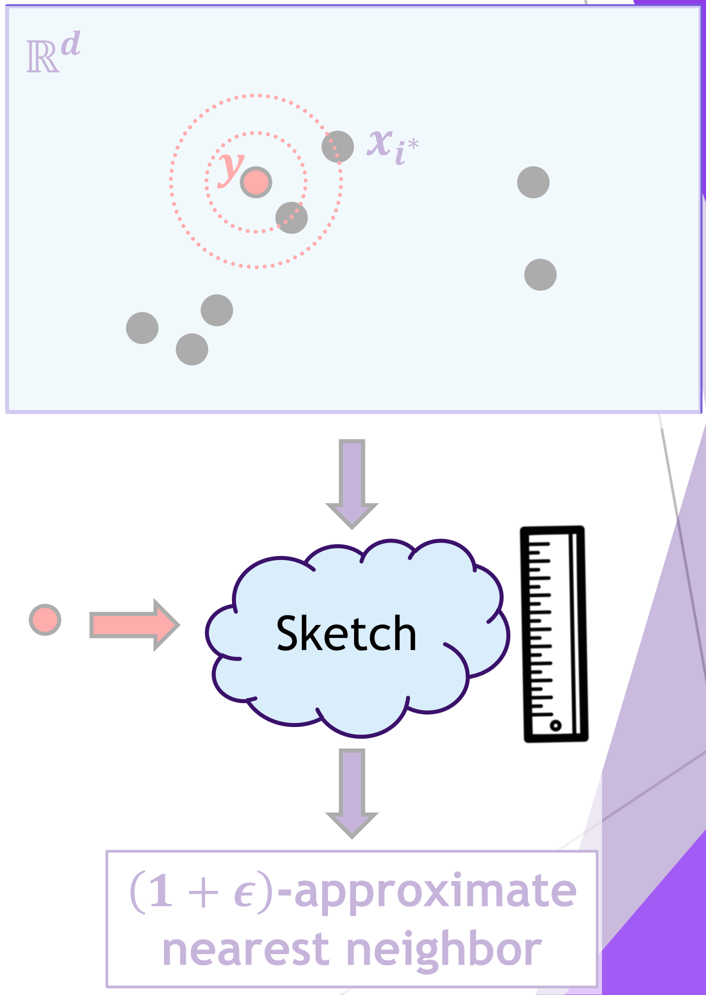
What is the space complexity of the (Euclidean) Approximate Nearest Neighbor Problem?

$(1+\epsilon)$-Approximate Nearest Neighbor problem:

- **Preprocess:**
  - Input: $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$
  - Output: small-size sketch
- **Query:**
  - Input: $\boldsymbol{y} \in \mathbb{R}^d$
  - Output: $\boldsymbol{i}^* \in \{\mathbf{1}, \dots, \boldsymbol{n}\}$

  s.t. $\|\boldsymbol{y} - \boldsymbol{x}_{i^*}\| \le (\mathbf{1}+\boldsymbol{\epsilon}) \min_{j \in \{1,\dots,n\}} \|\boldsymbol{y} - \boldsymbol{x}_j\|$

This talk: Minimize sketch size



Context

- Nearest neighbor classifiers are fundamental in machine learning
  - Eg. [Efros'17, NIPS 2017 workshop]
- Compression is useful:
  - Fast linear scan
  - Fit on GPU [Johnson-Douze-Jegou'17]
- Huge amount of empirical literature: **Quantization, Learning-to-Hash**
  - Surveys and tutorials: [Li'15, Moran'16, Wang-Zhang-Son-Sebe-Shen'16]
  - Partial sample of references: **NIPS**: [Weiss-Torralba-Fergus'08, Raginsky-Lazebnik'09, Kulis-Darrel'09, Kong-Li'12, …] **ICML**: [Norouzi-Blei'11, Norouzi-Fleet'11, Liu-Wang-Kumar-Chang'11, Gong-Kuma-Verma-Lazebnik'12, Li-Lin-Shen-Hengel-Dick'13, Zhang-Du-Wang'14, …] **CVPR**: [Grauman-Darrel'07, Gong-Lazebnik'11, Heo-Li-He-Chang-Yoon'12, Norouzi-Fleet'13, Gong-Kumar-Rowley-Lazebnik'13, He-Wen-Sun'13, Kalantidis-Avrithis'14, …] **TPAMI**: [Jegou-Douze-Schmid'11, Ge-He-Ke-Sun'14, …] **AAAI**: [Kong-Li'12, Wang-Duan-Huang-Gao'16, …] **KDD**: [He-Liu-Chang'10, …] **IJCAI**: [Xu-Bu-Lin-Chen-He-Cai'13, Wang-Duan-Lin-Wang-Huang-Gao'15, …] **SIGIR**: [Moran-Lavrenko-Osborne'13, Moran'16, …]
- ...and in theory?

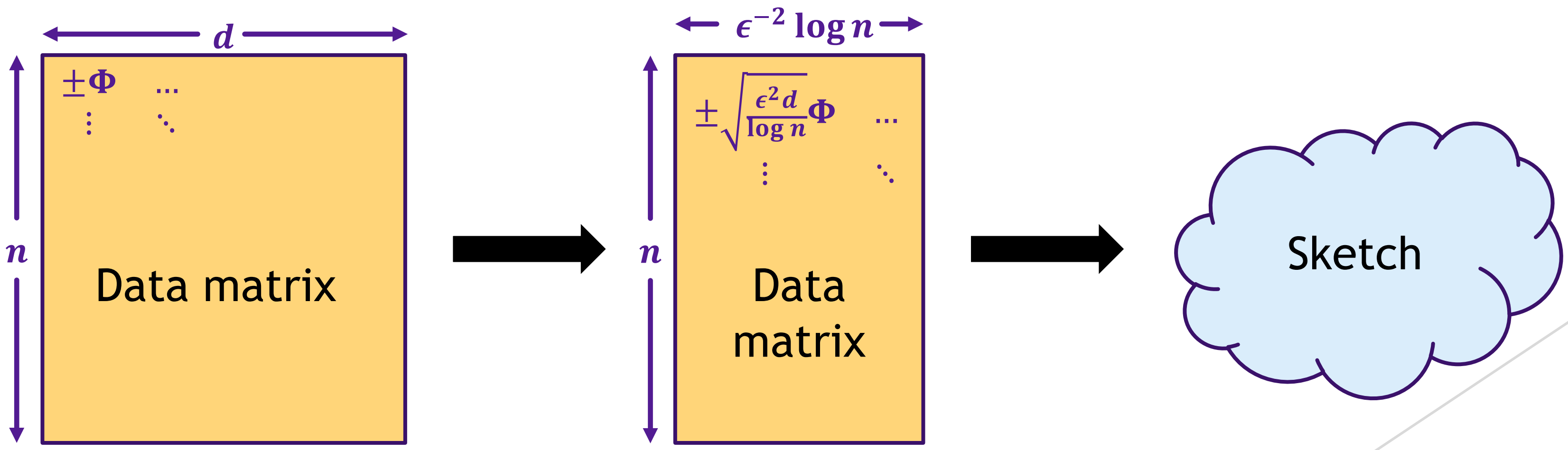
Euclidean Metric Compression

Goal: Compress a dataset $x_1, \dots, x_n \in \mathbb{R}^d$ with coordinates in $\{-\Phi, \dots, \Phi\}$

- **Dimension reduction:** $d \dashrightarrow O(\epsilon^{-2} \log n)$ [Johnson-Lindenstrauss'84]
- $\Rightarrow$ **Space:** $d \log \Phi \dashrightarrow O(\epsilon^{-2} \log n \cdot \log(d\Phi))$ bits per point [Achlioptas'03]

Can we do better?

- The [Johnson-Lindenstrauss'84] bound is tight [Larsen-Nelson'17, Alon'03]
  … for dimension reduction
- **What about space?**



Compression Beyond Dimension Reduction

- Input: $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, coordinates in $\{-\Phi, \dots, \Phi\}$, distortion $(1+\epsilon)$
- For presentation: $\Phi = n^{O(1)}$, $\epsilon = \Omega(1)$

Method	Bits per point	Returns $(1+\epsilon)$-approximate…
No compression	$d \log n$	Distances between X and $y \in \mathbb{R}^d$ (exact)
Dimension reduction	$\log^2 n$	Distances between X and $y \in \mathbb{R}^d$
[Kushilevitz-Ostrovski-Rabani'00]	$\log n \cdot \log R$	Distances between X and $y \in \mathbb{R}^d$ **assuming** $\|x_i - y\| \in [r, Rr]$
[Indyk-W'17,'18]	$\log n$, tight	Distances within X **no out-of-sample queries**
This work	$\log n$	**Nearest neighbor** of $y \in \mathbb{R}^d$ in X

Compression Beyond Dimension Reduction

- **Input:** $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, **coordinates in** $\{-\Phi, \ldots, \Phi\}$, **distortion** $(1+\epsilon)$
- For presentation: $\Phi = n^{O(1)}$, $\epsilon = \Omega(1)$

Method	Bits per point	Returns $(1+\epsilon)$-approximate…
No compression	$d \log n$	Distances between X and $y \in \mathbb{R}^d$ (exact)
Dimension reduction	[illegible]	Distances between X and $y \in \mathbb{R}^d$
[Kushilevitz-Ostrovski-Rabani'00]	[illegible]	Distances between X and $y \in \mathbb{R}^d$ **assuming** $\|x - y\| \in [r, Rr]$
[Indyk-W'17,'18]	$\log n$, tight	Distances within X **no out-of-sample queries**
This work	[illegible]	**Nearest neighbor of** $y \in \mathbb{R}^d$ **in** X

Return all distances

Returns nearest neighbor ID, not distance

Compression Beyond Dimension Reduction

- **Input:** $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$**, coordinates in** $\{-\Phi, \dots, \Phi\}$**, distortion** $(1+\epsilon)$
- For presentation: $\Phi = n^{O(1)}$, $\epsilon = \Omega(1)$

Method	Bits per point	Returns $(1+\epsilon)$-approximate…
No compression	$d \log n$	Distances between X and $y \in \mathbb{R}^d$ (exact)
Dimension reduction	$\log^2 n$	Distances between X and $y \in \mathbb{R}^d$
[Kushilevitz-Ostrovski-Rabani'00]	$\log n \cdot \log R$	Distances between X and $y \in \mathbb{R}^d$ **assuming** $\|x_i - y\| \in [r, Rr]$
[Indyk-W'17,'18]	$\log n$, tight	Distances within X **no out-of-sample queries**
This work	$\log n$	**Nearest neighbor of** $y \in \mathbb{R}^d$ in X
[Molinaro-Woodruff-Yaroslavtzev'13]	$\log^2 n$ lower bound	Distances between X and $y_1, \dots, y_n \in \mathbb{R}^d$

Compression Beyond Dimension Reduction

- **Input:** $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, **coordinates in** $\{-\Phi, \dots, \Phi\}$, **distortion** $(1+\epsilon)$
- For presentation: $\Phi = n^{O(1)}$, $\epsilon = \Omega(1)$

Method	Bits per point	Returns $(1+\epsilon)$-approximate…
No compression		Distances between X and $y \in \mathbb{R}^d$ (exact)
Dimension reduction		Distances between X and $y \in \mathbb{R}^d$
[Kushilevitz-Ostrovsky-Rabani'00]		Distances between X and $y \in \mathbb{R}^d$ **assuming** $\|x_i - y\| \in [r, Rr]$
[Indyk-W'17,'18]		Distances within X **no out-of-sample queries**
This work	$\log n$	**Nearest neighbor of** $y \in \mathbb{R}^d$ **in** X
[Molinaro-Woodruff-Yaroslavtzev'13]	$\log^2 n$ lower bound	Distances between X and $y_1, \dots, y_n \in \mathbb{R}^d$ **n query points**

Support up to $n^{O(1)}$ queries

Tight if |dataset| $\cong$ |query set|

What if |dataset| $\gg$ |query set|?

Compression Beyond Dimension Reduction

- Input: $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, coordinates in $\{-\Phi, \dots, \Phi\}$, distortion $(1+\epsilon)$
- For presentation: $\Phi = n^{O(1)}$, $\epsilon = \Omega(1)$

Method	Bits per point	Returns $(1+\epsilon)$-approximate…
No compression	$d \log n$	Distances between X and $y \in \mathbb{R}^d$ (exact)
Dimension reduction	$\log^2 n$	Distances between X and $y \in \mathbb{R}^d$
[Kushilevitz-Ostrovski-Rabani'00]	$\log n \cdot \log \mathrm{R}$	Distances between X and $y \in \mathbb{R}^d$ assuming $\|x_i - y\| \in [r, Rr]$
[Indyk-W'17,'18]	$\log n$, tight	Distances within X no out-of-sample queries
This work	$\log n$	**Nearest neighbor of $y \in \mathbb{R}^d$ in X**
[Molinaro-Woodruff-Yaroslavtzev'13]	$\log^2 n$ lower bound	Distances between X and $y_1, \dots, y_n \in \mathbb{R}^d$ **n query points**
This work	$\log n \cdot \log q$	Distances between X and $y_1, \dots, y_q \in \mathbb{R}^d$ **$q \le n$ query points**

Practical Variant

[Indyk-W’17]
Size: $\log n$
No query support
Impractical algorithm

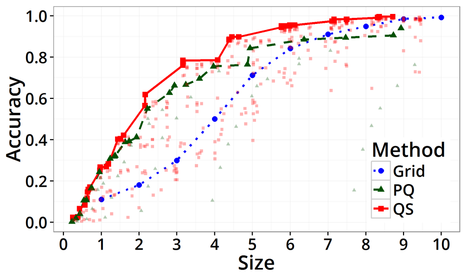
→

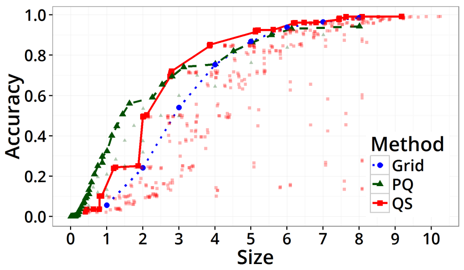
This work
Size: $\log n$
Nearest neighbor query support
Impractical algorithm

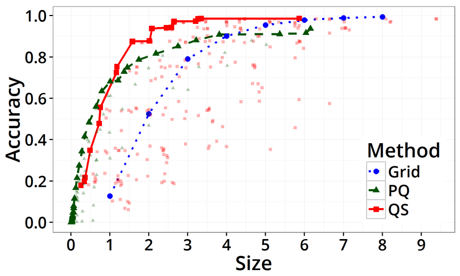
[Indyk-Razenshteyn-W’17]
Size: $\log n \cdot \log\log n$
No query support
Practical algorithm

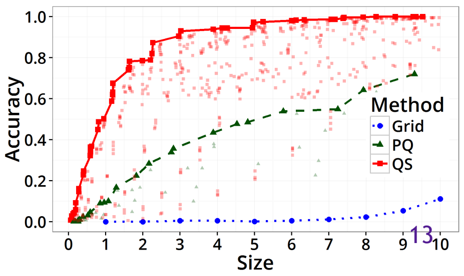
→

This work
Size: $\log n \cdot \log\log n$
Nearest neighbor query support
Practical algorithm







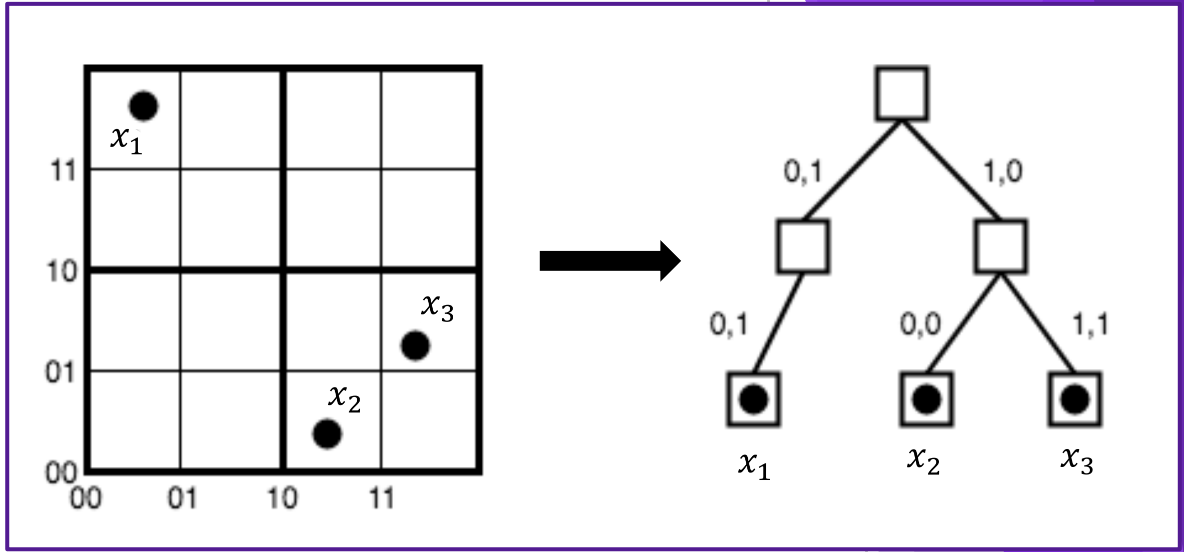


Techniques

- Prior work:

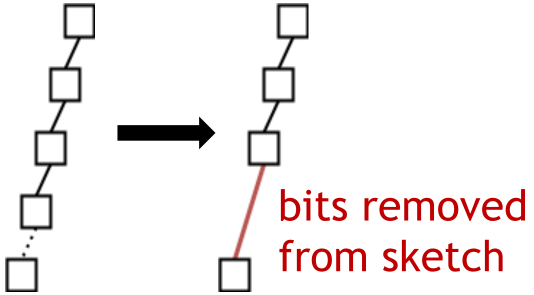
  Step 1: Hierarchical clustering

  - Eg., quadtree
  - Tree edges ↔ precision bits



Step 2: "Bottom-out compression"

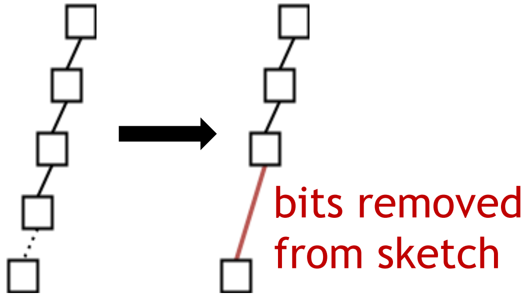
- Stores most significant bits per cluster

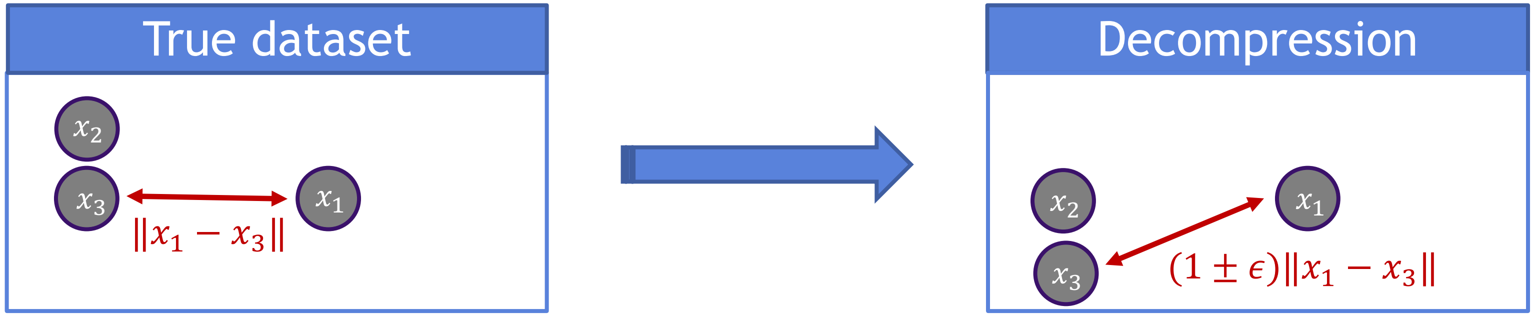


Techniques

"Bottom-out compression":

- Preserves global cluster structure
- Recovers dataset distances

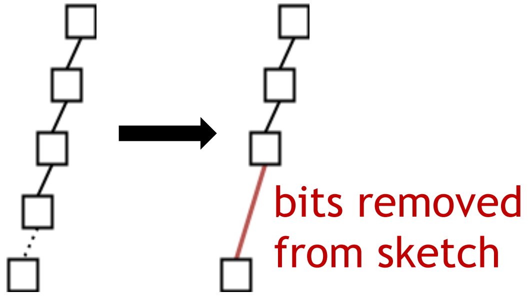


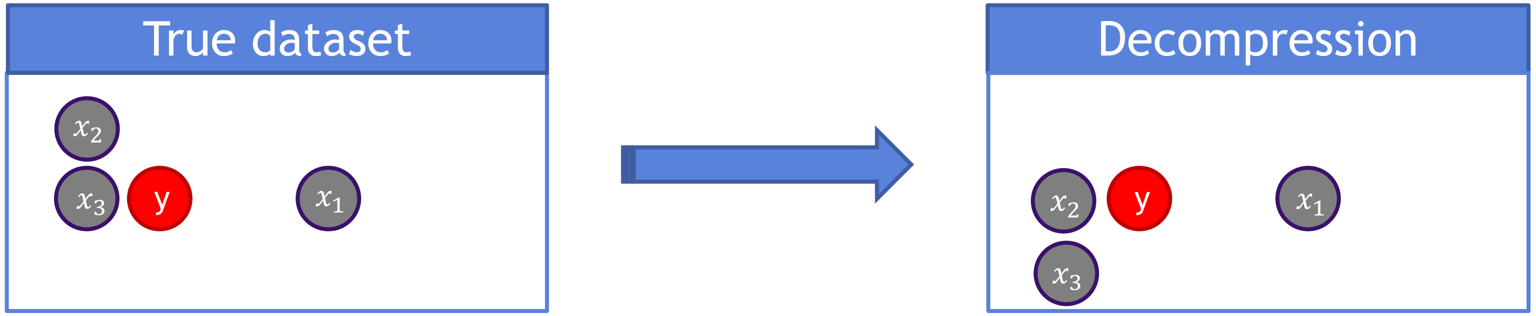


Techniques

“Bottom-out compression”:

- Preserves global cluster structure
- Recovers dataset distances
- **But not nearest neighbor queries**

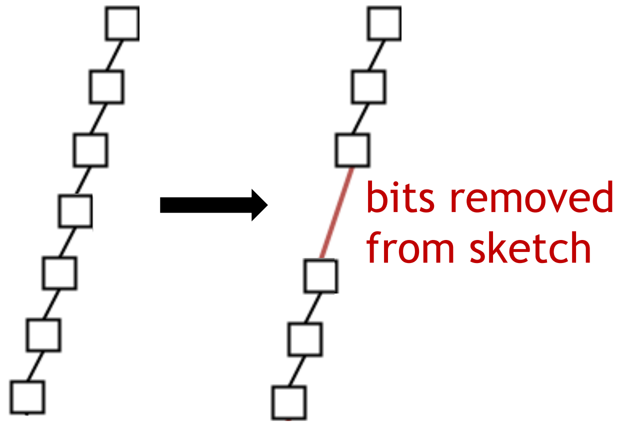




Techniques

This work: "Middle-out compression":

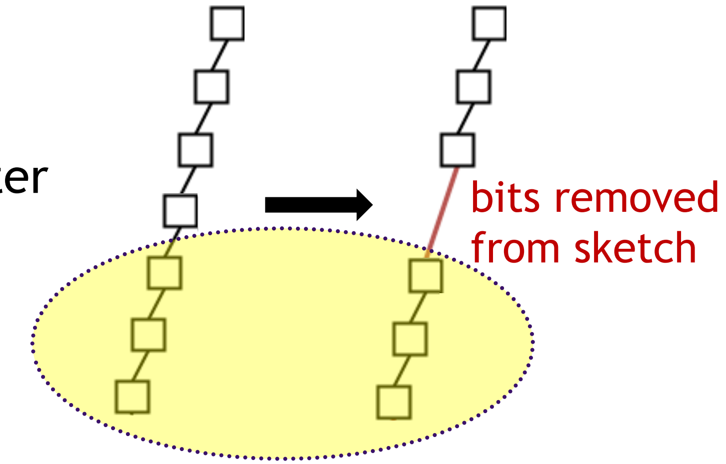
- Stores **most and least** significant bits per cluster

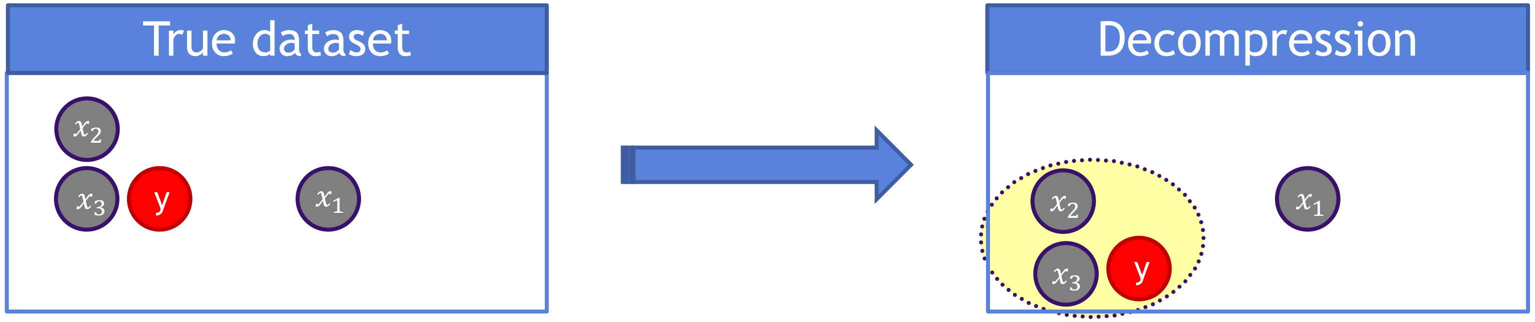


Techniques

This work: “Middle-out compression”:

- Stores **most and least** significant bits per cluster
  - Sketch is only twice as big
- Recovers **global and local** cluster structure





Thank you

Questions?