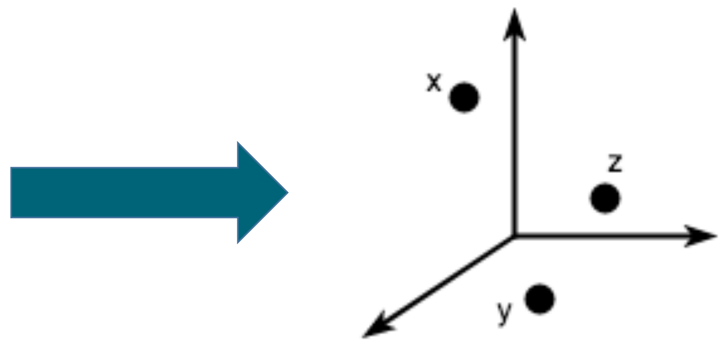
Practical Data-Dependent Metric Compression with Provable Guarantees

Piotr Indyk — MIT
Ilya Razenshteyn — Columbia University
Tal Wagner — MIT

Code available at:
github.com/talwagner/quadsketch

Introduction

Metric embedding:
Starting point of many algorithms



Real-world objects
(images, text, etc.)

High-dimensional feature vectors
(image descriptors, word2vec, etc.)

Goal: Compress vectors while approximately preserving distances.

- Many algorithms for data analysis and machine learning rely on distances
- **E.g.:** Nearest neighbor queries

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

Contribution

Our algorithm:

- **Simple** to describe and implement
- **Provable** pointwise guarantees
- **Matches or outperforms** state-of-the-art in the high-precision regime

Previous work: Either *heuristic* or *impractical*.

Heuristic algorithms:

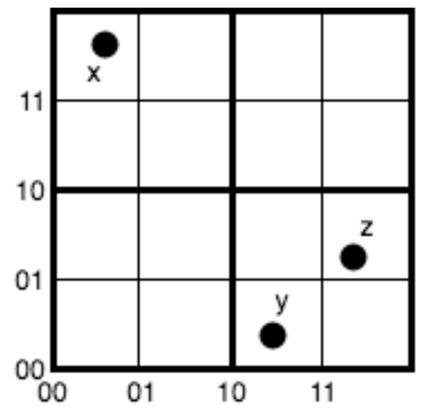
- Lack provable guarantees -
  may be unsuitable for non-standard datasets
- Optimize for average accuracy -
  may perform undesirably on individual queries
- Solve a global optimization problem on the dataset (e.g. k-means) -
  slow or infeasible in high precision regime

Theoretical algorithms:
Unsuitable for implementation despite asymptotic guarantees, due to large hidden constants, underlying combinatorial complexity, etc.

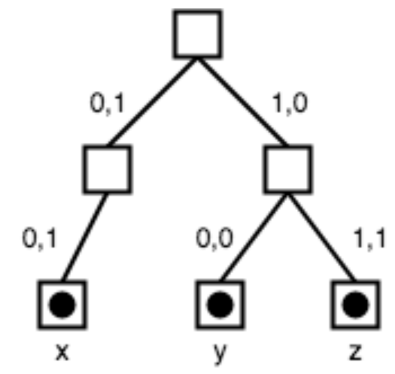
QuadSketch: Algorithm Description

Construction

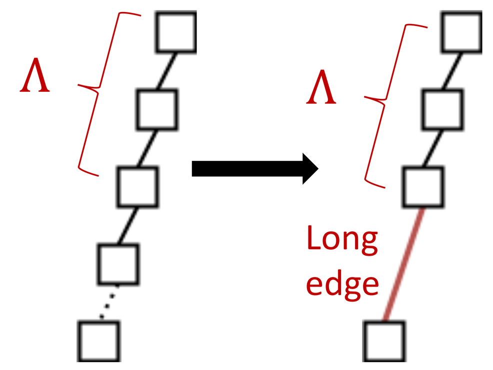
- **Step 1: Randomly shifted grids**
  Enclose points in hypercube.
  Refine into sub-cubes by halving each dimension.
  Repeat refinement for L levels.
  Shift grids by a uniformly random vector.



- **Step 2: Quadtree**
  Construct high-dimensional quadtree from grids:
  - The root is the enclosing hypercube.
  - For every non-empty sub-hypercube, add child node.



- **Step 3: Pruning**
  For every tree path longer than Λ:
  Replace the path after the top Λ nodes with a long edge.



The compressed representation is the pruned quadtree.

Recovery

To recover the approximation $\tilde{\boldsymbol{x}}$ of a point $\boldsymbol{x}$:

- Follow path from root to leaf containing $\boldsymbol{x}$.
- In each dimension, concatenate bits along edges in path.
- If long edge, concatenate zeros instead.

$\tilde{\boldsymbol{x}} = (00, 11)$
$\tilde{\boldsymbol{y}} = (10, 00)$
$\tilde{\boldsymbol{z}} = (11, 01)$

Theoretical Results

Parameters:
n - num. points; d - dimension; Φ - ratio of maximum to minimum distance *(captures numerical range)*

Theorem: Given $\epsilon, \delta > 0$, set

$$\Lambda = \log\left(\frac{16 \cdot d^{1.5} \cdot \log\Phi}{\epsilon \cdot \delta}\right) \quad \text{and} \quad L = \Lambda + \log\Phi .$$

QuadSketch guarantees: For every point $\boldsymbol{x}$,

$$\Pr[\forall \boldsymbol{y}\ \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{y}}\| \in (1 \pm \epsilon)\|\boldsymbol{x} - \boldsymbol{y}\|] > 1 - \delta .$$

- In particular, $(1+\epsilon)$-approximate nearest neighbors are preserved with probability $1 - \delta$.
- Construction time: $\tilde{O}(ndL)$.
- Compressed size: $O(nd\Lambda + n\log n)$ bits.

Comparison with prior work:
For $d = \Theta(\epsilon^{-2}\log n)$ by dimension reduction, and $\Phi = poly(n)$

Reference	Bits per coordinate	Construction time
Vanilla bound	$O(\log n)$	--
(Indyk, Wagner 2017)	$O(\log(1/\epsilon))$	$\tilde{O}(n^{1+\alpha} + \epsilon^{-2}n)$ for $\alpha \in (0,1]$
This work	$O(\log\log n + \log(1/\epsilon))$	$\tilde{O}(\epsilon^{-2}n)$

Experiments

We compare:

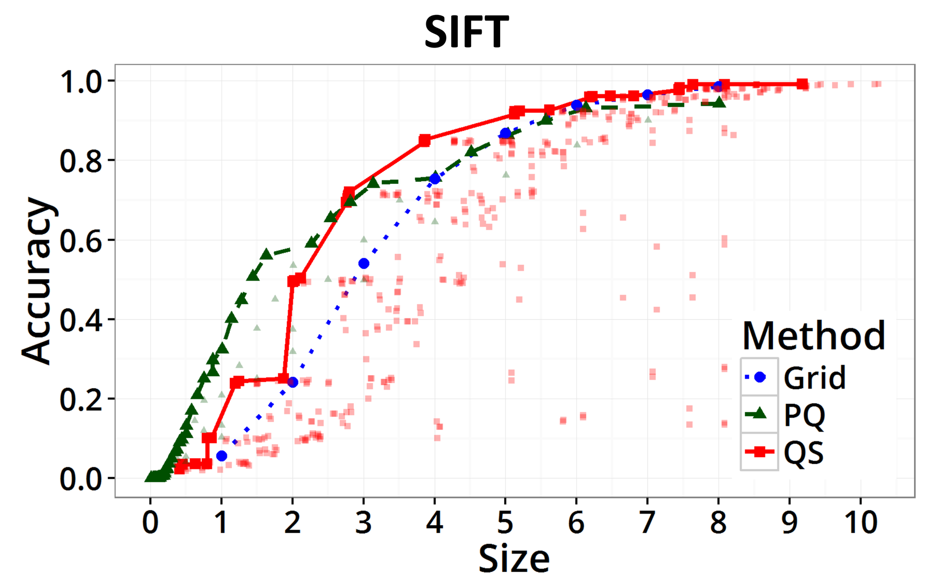
- **QS:** Product QuadSketch
  Partition into blocks, QuadSketch in each
- **PQ:** Product Quantization *(Jégou, Douze, Schmid 2011)*
  Partition into blocks, k-means in each
- **Grid:** Uniform scalar quantization (baseline)

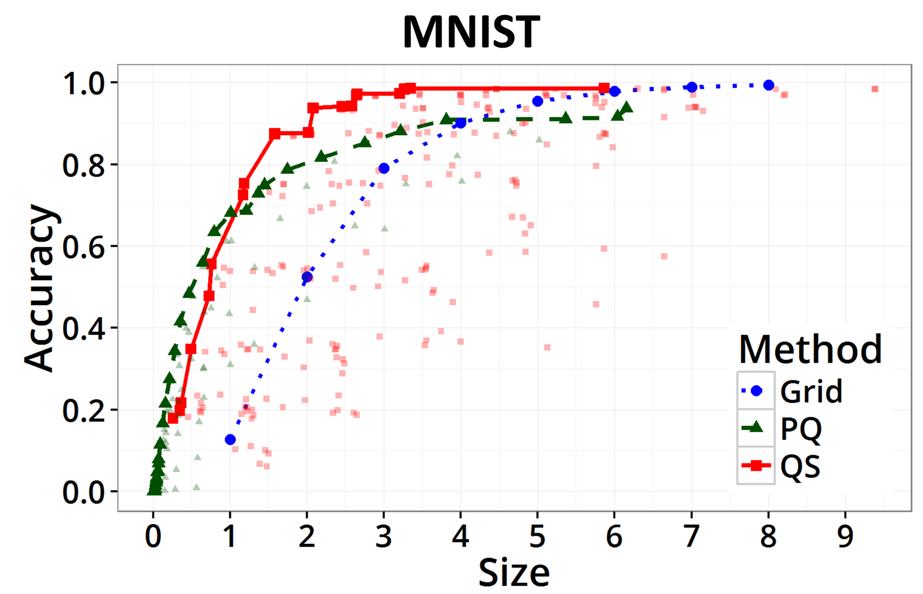
We report:

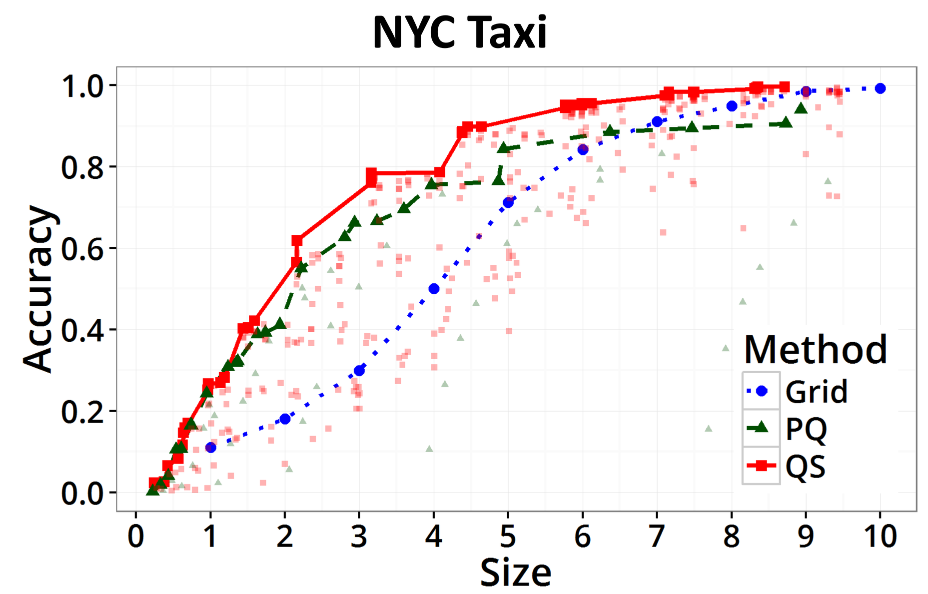
- **Accuracy** - fraction of correct nearest neighbors
- **Size** - bits per coordinate

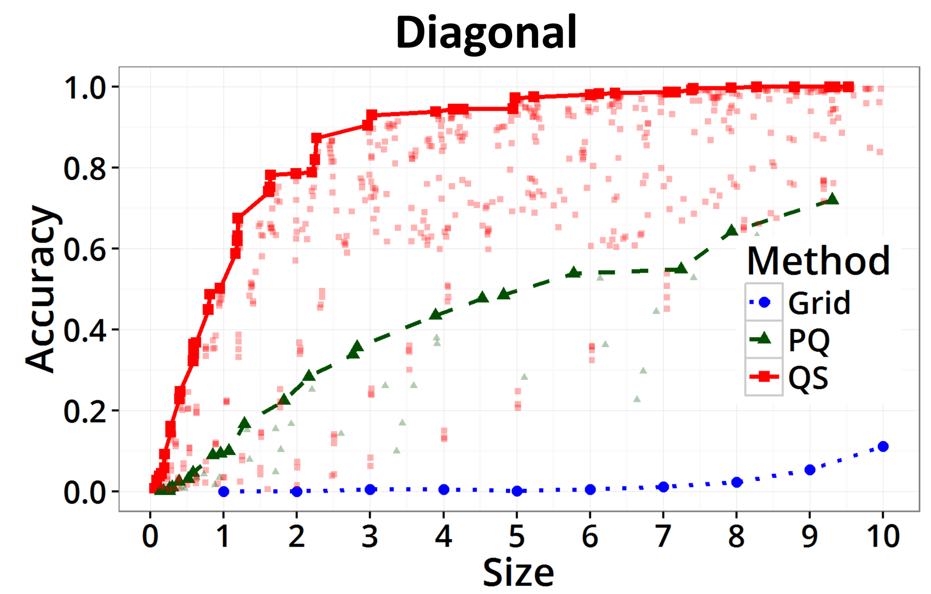
Datasets:	n	d	Φ
SIFT	1M	128	≥ 83.2 *
MNIST	60K	784	≥ 9.2 *
NYC Taxi ridership	8,874	48	49.5
Diagonal (synthetic) **	10K	128	20,478,740.2

* Estimated on a random sample.
** Random points on a line, embedded in a 128-dimensional space.









References:

P. Indyk, T. Wagner, *Near-optimal (Euclidean) metric compression.* ACM-SIAM Symposium on Discrete Algorithms, 2017.

H. Jégou, M. Douze, C. Schmid, *Product quantization for nearest neighbor search.* IEEE transactions on pattern analysis and machine intelligence, 2011.

J. Johnson, M. Douze, H. Jégou, *Billion-scale similarity search with GPUs.* ArXiv preprint, 2017.