Learning Space Partitions for Nearest Neighbor Search

Yihe Dong
MSR

Piotr Indyk
MIT

Ilya Razenshteyn
MSR

Tal Wagner
MIT
Nearest Neighbor Search

• Given:
  • Dataset of points in $\mathbb{R}^d$.

• Query:
  • $q$ in $\mathbb{R}^d$.

• Goal:
  • $k$-nearest neighbors from dataset.
Method: Space Partitions of $\mathbb{R}^d$

Advantages:

- **Sublinear** query time
  - Compute distance from query to a subset of candidate data points
- **Distributed** computation
  - Put each bin on different machine
Space Partition Desiderata

• Want a partition of $\mathbb{R}^d$ that:
  • Returns accurate nearest neighbors
  • Approximately balanced
    • w.r.t. data points
  • Algorithmically simple
Methods for Space Partitions

• Data independent:
  • Classical Locality-sensitive hashing (LSH)

• Data dependent:
  • Data dependent LSH
  • Quantization (k-means)
  • Supervised hyperplane partitions

• Our goal: Use modern supervised learning (like neural networks) to learn better space partitions
Our Contribution

• New method to partition $\mathbb{R}^d$

• Two stage process:
  1. Combinatorial graph partitioning
  2. Supervised learning

• Empirically better than prior methods for nearest neighbor search

We use KaHIP (Sanders and Schultz 2013)

We use small neural networks (“Neural LSH”)
Our Method: Preprocessing

- Create $k$-NN graph of dataset
- Find balanced partition of graph
- **Train learning model** to generalize partition from graph nodes to all of $\mathbb{R}^d$
Our Method: Query

• Run **inference** on query to **classify** into bin, or to get **ranking** of likely bins
• Search for nearest neighbors in highest ranking bins
Select Experimental Results

- Partition into 256 bins

![Graphs showing k-NN accuracy for GloVe and SIFT datasets](image)

- **GloVe** (1.2M points, 100 dimensions)
  - Neural LSH (average)
  - Neural LSH (0.95-quantile)
  - k-means (average)
  - k-means (0.95-quantile)

- **SIFT** (1M points, 128 dimensions)
  - Neural LSH (average)
  - Neural LSH (0.95-quantile)
  - k-means (average)
  - k-means (0.95-quantile)

Number of distance computations ("candidates")

![GitHub button](image)

Thank you