A Theory-Based Evaluation of Nearest Neighbor Models Put into Practice

Hendrik Fichtenberger and Dennis Rohde

**Part of Data Processing Pipeline**

given: point-set in euclidean space
build nearest neighbor model [black box]
output

query: give k nearest points to

What is the quality of the output?

**Query Complexity**

Testing k-nearest Neighborhood

- $O(k^{d-1}/\epsilon)$ queries sufficient in bounded average-degree graphs
- $\Omega(\frac{d^2}{\epsilon^2} + k\psi)$ queries required in general graphs

**Property Testing Algorithm**

- accepts every k-nearest neighborhood graph with high probability
- rejects every graph that is $\epsilon$-far from being a k-nearest neighborhood graph with high probability
- at least an $\epsilon$-fraction of edges are faulty
- can freely answer otherwise

**Our Algorithm**

- sample $O(k^{d-1}/\epsilon)$ vertices uniformly at random
- throw away vertices with high degree
- sample $O(k\psi n)$ vertices uniformly at random
- for every vertex in first sample check if any vertex from second sample lies nearer than any neighbor

**Example Graphs**

- brute force 3-nn graph
- graph from Annoy(Spotify) model

- 60 vertices are incident with faulty edges

**Experiments**

all models were built ten times for each parameterization, then tested once

recall of algorithm by $\epsilon$-distance of model

**Our Code**

The algorithm: github.com/derohde/knn_test
Extension of ann-benchmarks: github.com/hfichtenberger/ann-benchmarks

**Models**

AAALgo: github.com/aaalgo/kgraph
NMSLIB: github.com/nmslib/nmslib
Annoy: github.com/spotify/annoy

**Full Paper**

on arXiv

no need to compute full graph

vertices = points

edges = query results

has query-access to

1. implicit conversion of k-nn model to geometric graph
2. test if given graph is k-nearest neighbor graph
   - directed edges
   - regular (out-degree = k)
   - edges point to k-nearest neighbors of vertex

buckeled distance

datasets: MNIST, Fashion-MNIST, Sift
models: KGraph(Ann Arbor Algorithms), hnsw and SWGraph(NMSLib)

parameters
0.001, 0.05
0.001, 0.5
0.001, 5
0.01, 0.05
0.01, 0.5
0.01, 5
0.1, 0.05
0.1, 0.5
0.1, 5

number of points
d-dimensional kissing number

$\epsilon$-distance of model