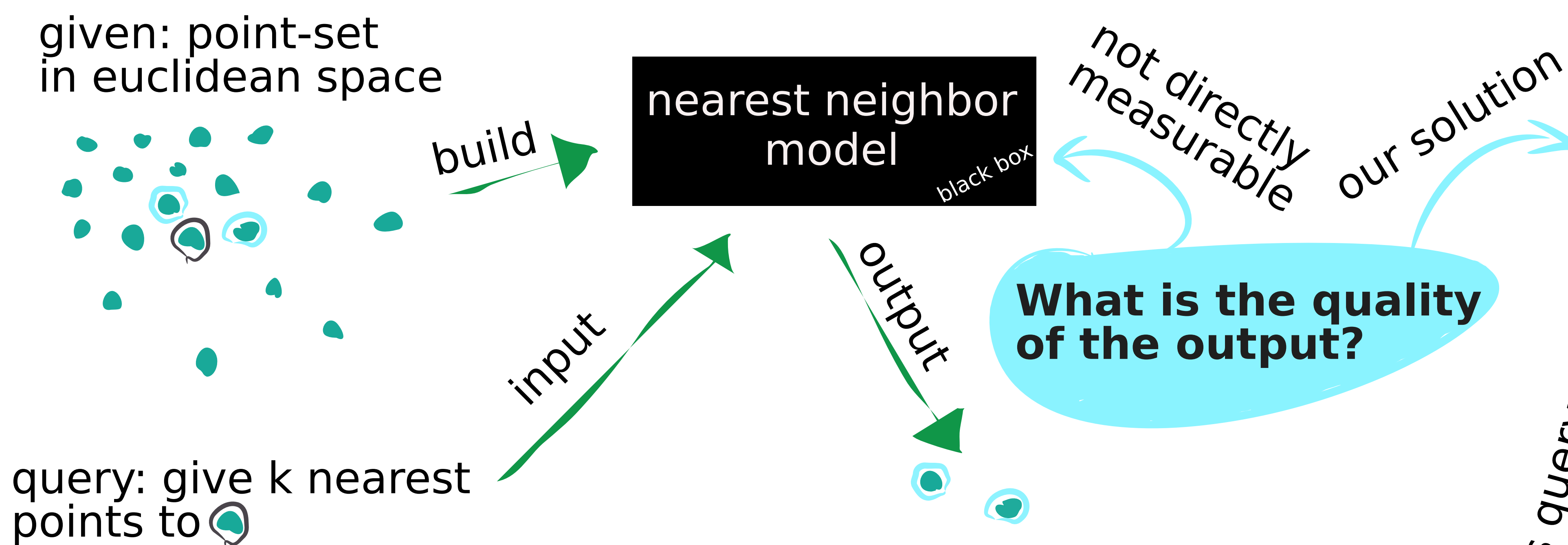


A Theory-Based Evaluation of Nearest Neighbor Models Put Into Practice

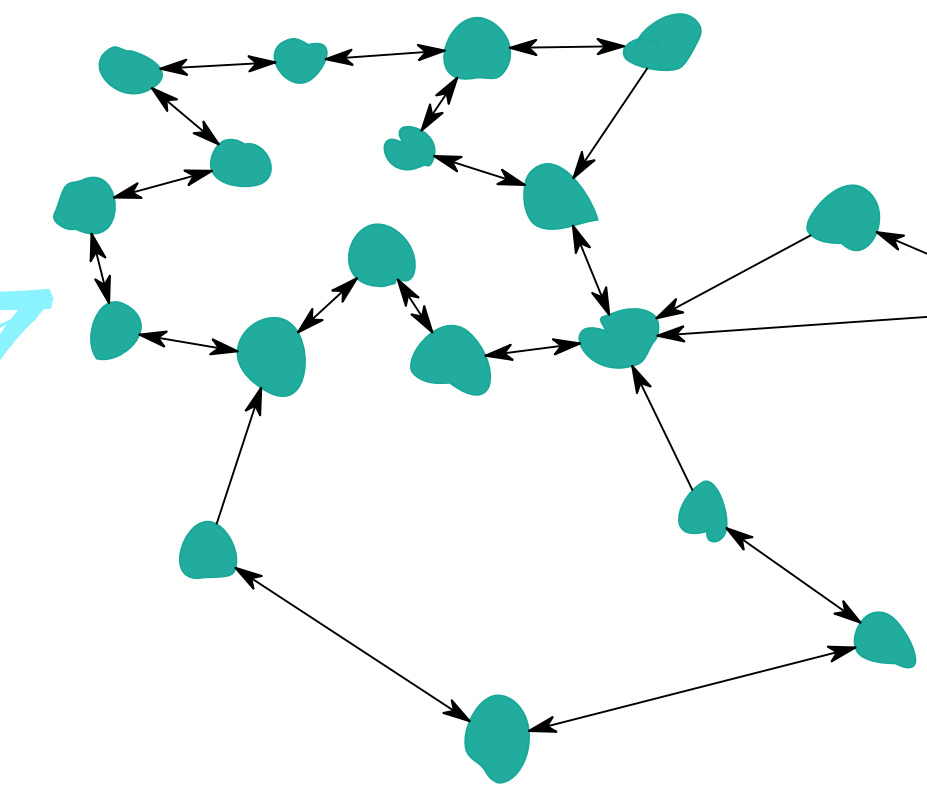
HENDRIK FICHTENBERGER AND DENNIS ROHDE

PART OF DATA PROCESSING PIPELINE



no need to compute full graph

1. implicit conversion of k-nn model to geometric graph



vertices = points
edges = query results

2. test if given graph is k-nearest neighbor graph

- directed edges
- regular (out-degree = k)
- edges point to k-nearest neighbors of vertex

has query-access to

QUERY COMPLEXITY

measure of efficiency

Testing k-nearest Neighborhood

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

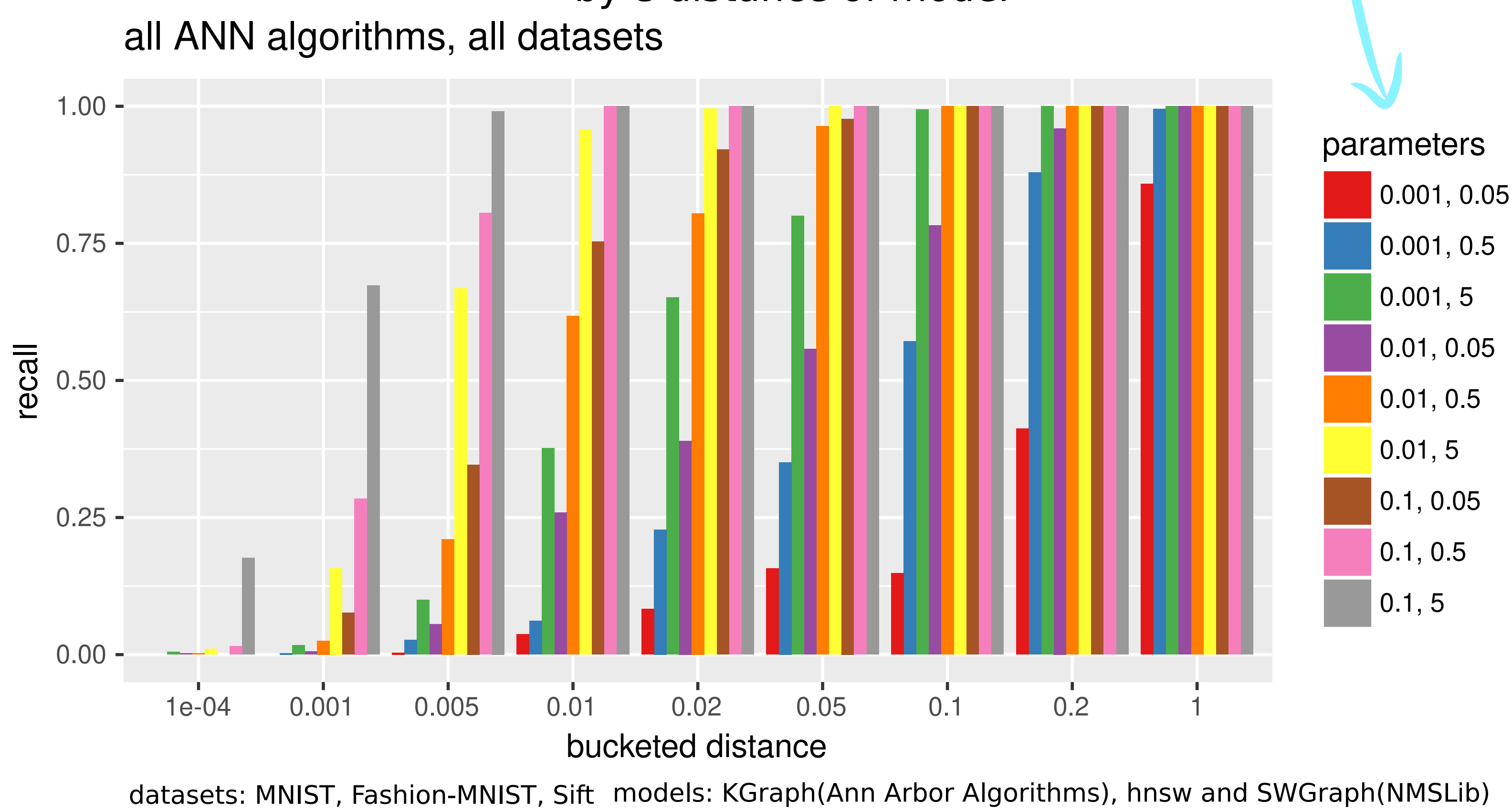
fixed here
d-dimensional kissing number
number of points

EXPERIMENTS

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

recall of algorithm
by ϵ -distance of model

constants of O-notation



PROPERTY TESTING ALGORITHM

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

is a

OUR ALGORITHM

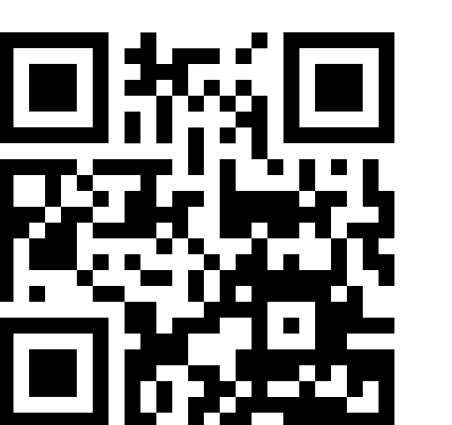
- sample $O(k \frac{1}{\epsilon} \sqrt{n})$ vertices uniformly at random
- throw away vertices with high degree
- sample $O(k \psi_d \sqrt{n})$ vertices uniformly at random
- for every vertex in first sample check if any vertex from second sample lies nearer than any neighbor

Yes
Reject

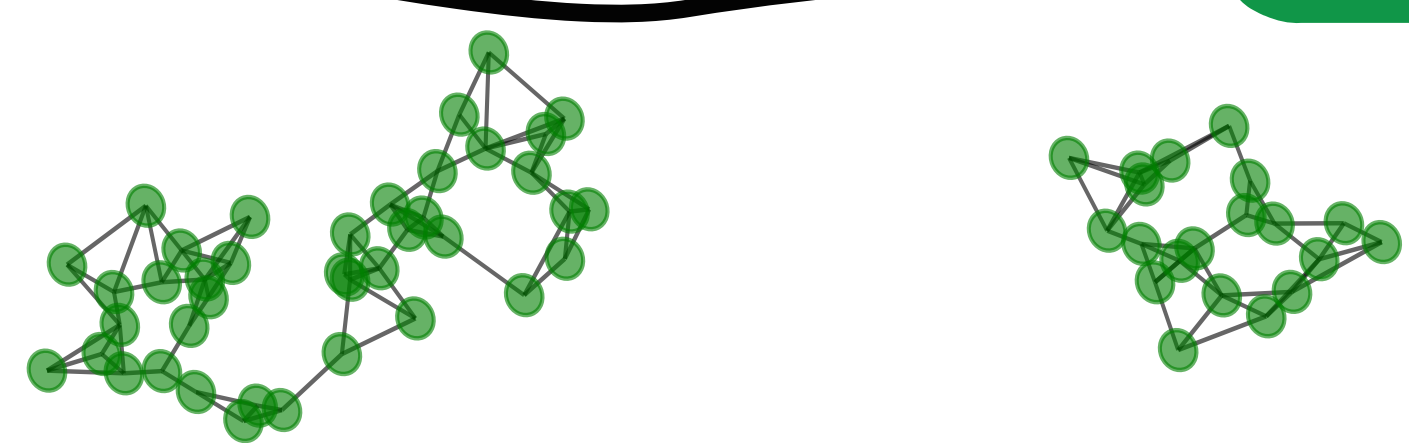
Never
Accept

full paper on arXiv

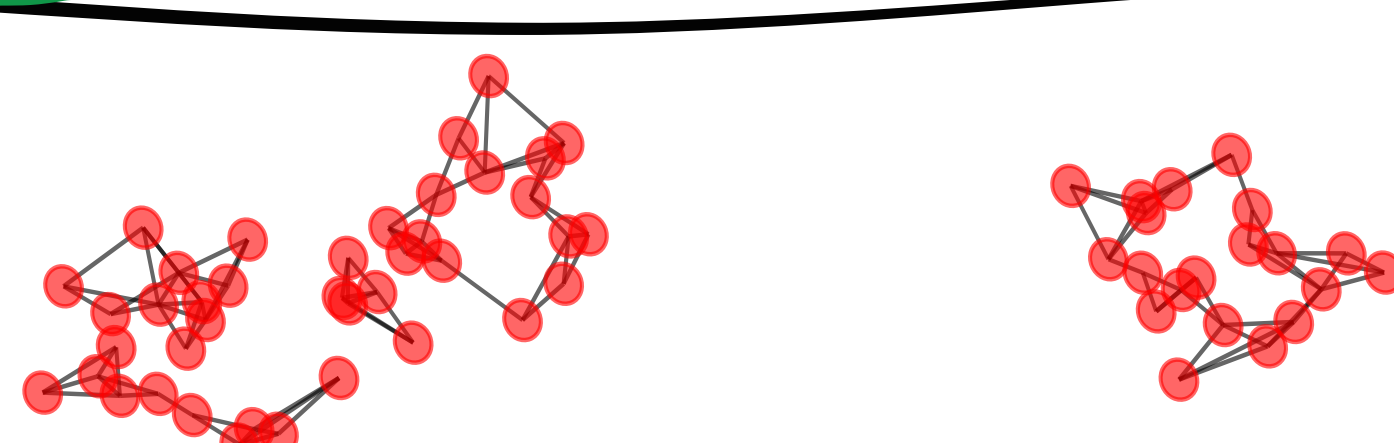
140 random points
420 edges
4 clusters



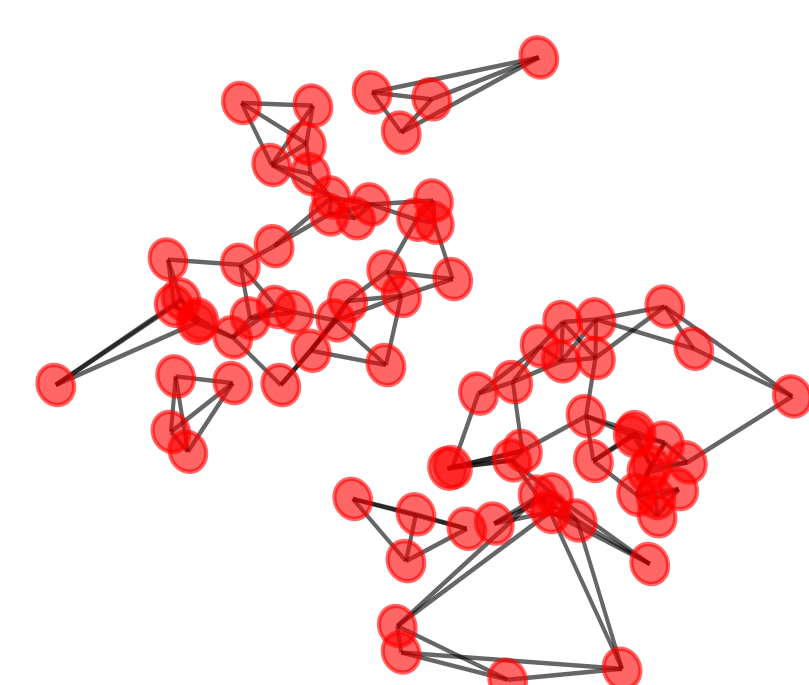
EXAMPLE GRAPHS



brute force 3-nn graph



graph from Annoy(Spotify) model



60 vertices are incident with faulty edges

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