# Approximate nearest neighbor search using the Hierarchical Navigable Small World (HNSW) algorithm

Sebastian Björkqvist

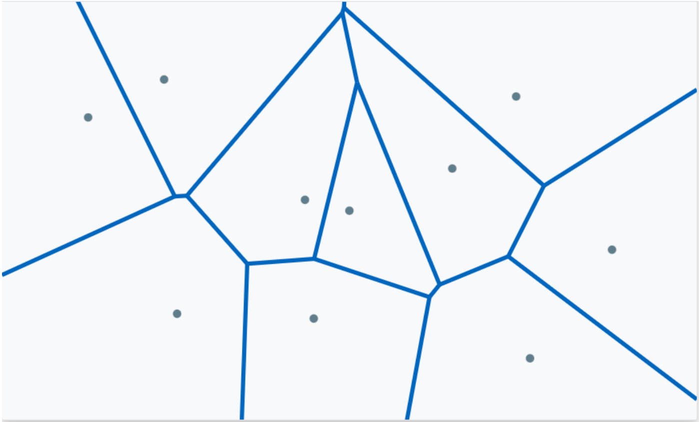
Lead AI Developer, IPRally

May 12, 2023

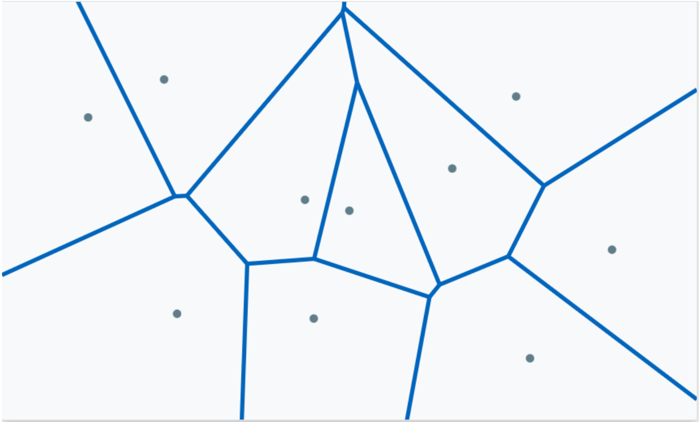
# Outline

# Voronoi diagram for a set of points

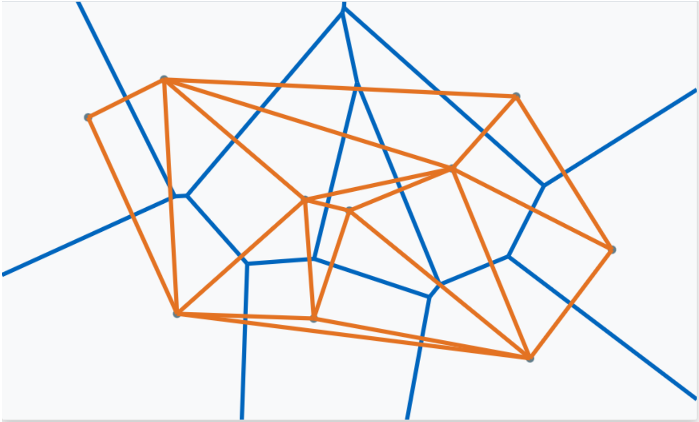
# Voronoi diagram for a set of points



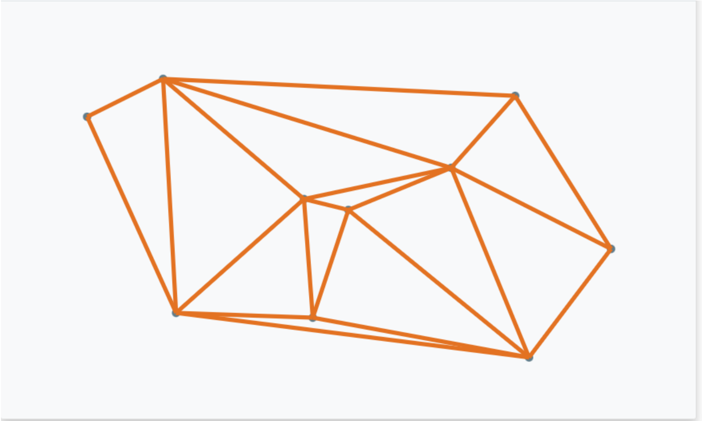
# Voronoi diagram to Delaunay graph



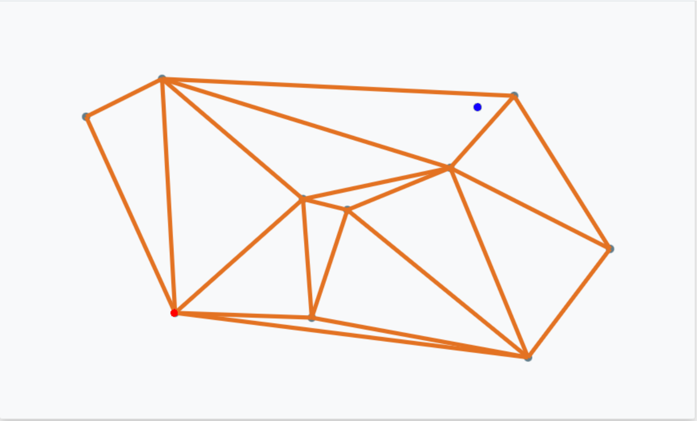
# Voronoi diagram to Delaunay graph



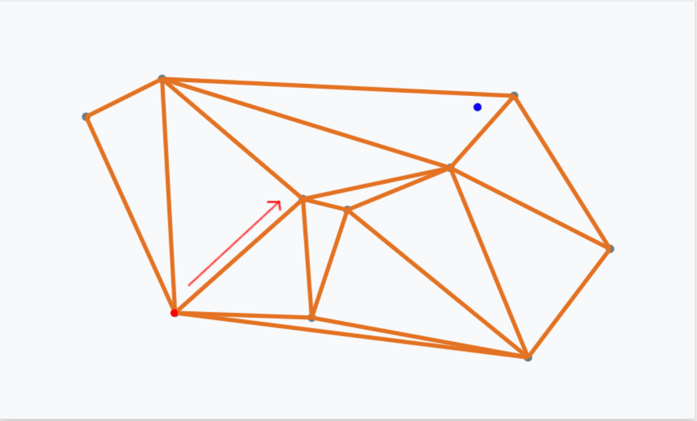
# Delaunay graph



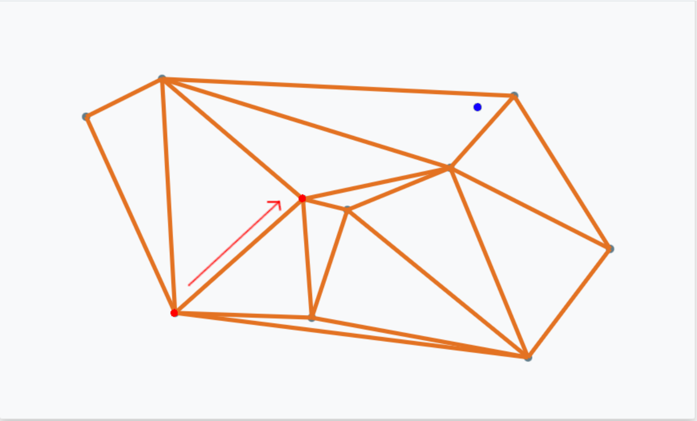
# Greedy NN search start - Query and entry point



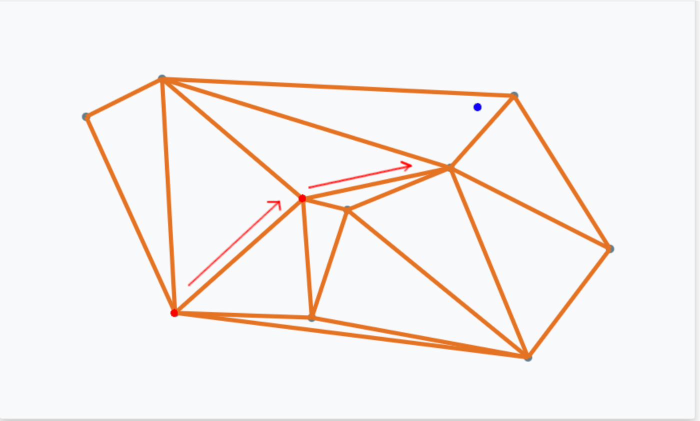
# Greedy NN search - iteration



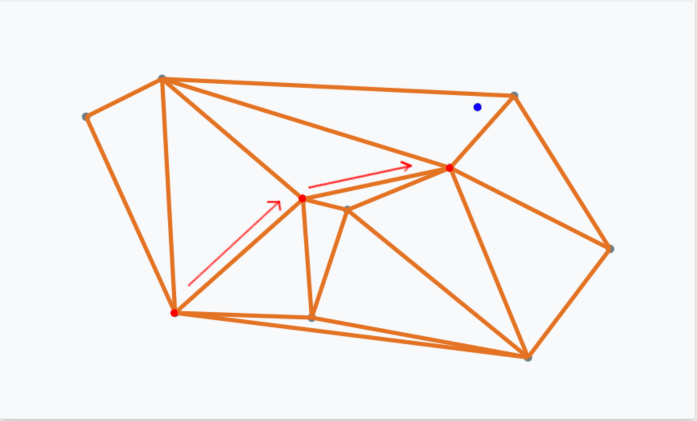
# Greedy NN search - iteration



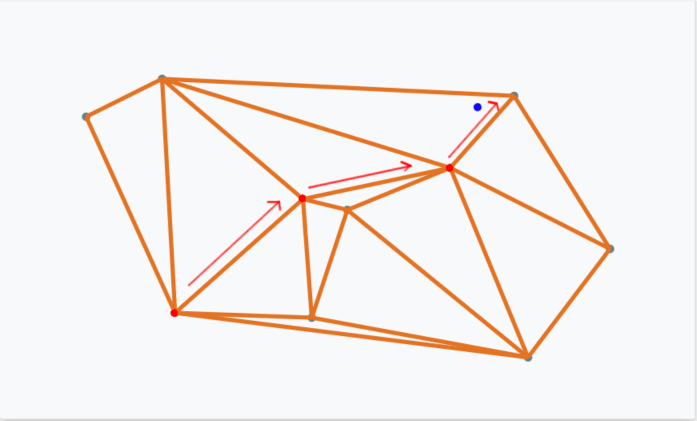
# Greedy NN search - iteration



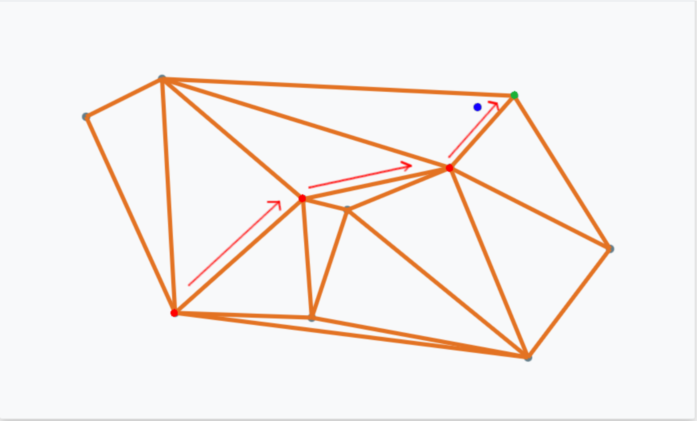
# Greedy NN search - iteration



# Greedy NN search - iteration



# Greedy NN search done!



# Drawbacks

- Delaunay graph intractable to construct for large, high-dimensional data sets
- Greedy search might be slow if graph is large

# Navigable small world (NSW) graph

# Navigable small world (NSW) graph

- Small world graph

# Navigable small world (NSW) graph

- Small world graph
  - Distance of two random nodes is $log\ N$, where $N$ is the number of nodes in graph

# Navigable small world (NSW) graph

- Small world graph
  - Distance of two random nodes is $log\ N$, where $N$ is the number of nodes in graph
  - Neighbors of a given node are likely to be neighbors of another (clustering coefficient is high)

# Navigable small world (NSW) graph

- Small world graph
  - Distance of two random nodes is $log\ N$, where $N$ is the number of nodes in graph
  - Neighbors of a given node are likely to be neighbors of another (clustering coefficient is high)
- Navigability

# Navigable small world (NSW) graph

- Small world graph
  - Distance of two random nodes is $log\ N$, where $N$ is the number of nodes in graph
  - Neighbors of a given node are likely to be neighbors of another (clustering coefficient is high)
- Navigability
  - Greedy search algorithm has logarithmic scalability

# Why is an NSW useful for nearest neighbor search?

# Why is an NSW useful for nearest neighbor search?

- Logarithmic distance allows us to get anywhere in the graph quickly

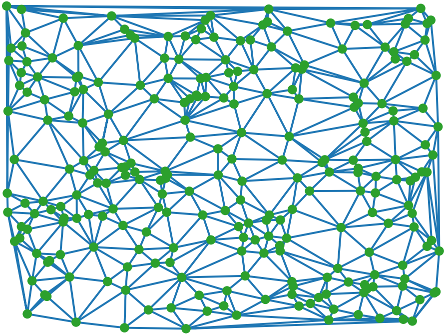
## Why is an NSW useful for nearest neighbor search?

- Logarithmic distance allows us to get anywhere in the graph quickly
- Navigability ensures that the greedy algorithm finds the logaritmic path

# Why is an NSW useful for nearest neighbor search?

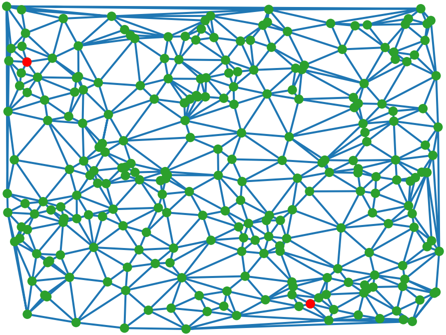
- Logarithmic distance allows us to get anywhere in the graph quickly
- Navigability ensures that the greedy algorithm finds the logaritmic path
- High clustering coefficient lets us zoom in on the actual correct node when we're in the right area

# Making Delaunay graph navigable

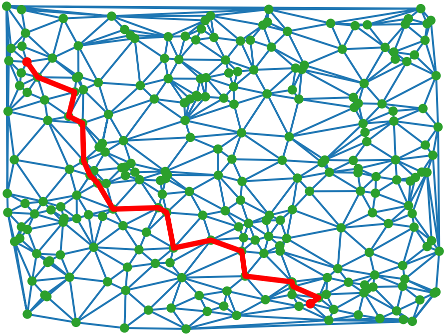


256 nodes

# Making Delaunay graph navigable

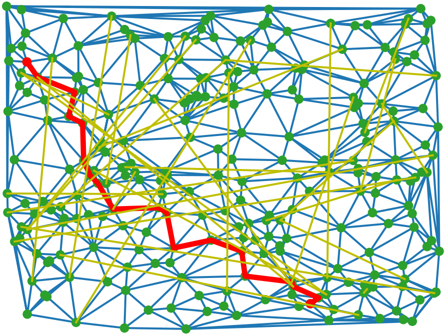


# Making Delaunay graph navigable



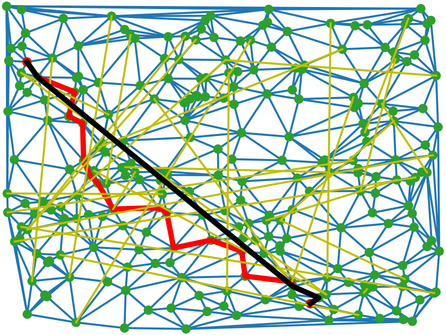
Length of path: 19

# Making Delaunay graph navigable



32 random edges added

# Making Delaunay graph navigable



Length of path: 5

# Properties of NSW graph

# Properties of NSW graph

- An NSW graph is not necessarily a Delaunay graph (or have one as a subgraph)

## Properties of NSW graph

- An NSW graph is not necessarily a Delaunay graph (or have one as a subgraph)
- Thus the greedy algorithm doesn't always return the actual nearest neighbor

# Properties of NSW graph

- An NSW graph is not necessarily a Delaunay graph (or have one as a subgraph)
- Thus the greedy algorithm doesn't always return the actual nearest neighbor
- Ok since we're doing approximate nearest neighbor search!

# Constructing NSW graph

## Constructing NSW graph

- Goal: Construct a graph that has the Delaunay graph as a subgraph, but also has longer connections to make it navigable

# Constructing NSW graph

- Goal: Construct a graph that has the Delaunay graph as a subgraph, but also has longer connections to make it navigable
- Approximation of Delaunay graph is sufficient

# Constructing NSW graph

# Constructing NSW graph

1. Randomize order of data points

# Constructing NSW graph

1. Randomize order of data points
2. Add data point to graph

# Constructing NSW graph

1. Randomize order of data points
2. Add data point to graph
3. Add edges from data point to its $k$ nearest neighbors that are already present in the graph

# Constructing NSW graph

1. Randomize order of data points
2. Add data point to graph
3. Add edges from data point to its $k$ nearest neighbors that are already present in the graph
4. Repeat 2 and 3 until all data points have been added

# Why does NSW graph creation algorithm work?

# Why does NSW graph creation algorithm work?

- Adding enough nearest neighbor edges approximates Delaunay graph

# Why does NSW graph creation algorithm work?

- Adding enough nearest neighbor edges approximates Delaunay graph
- The edges added for the early nodes give long-range connections, enabling navigability

# References


- *Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs (Malkov et al.* `https://arxiv.org/abs/1603.09320`
- *Approximate nearest neighbor algorithm based on navigable small world graphs (Malkov et al* `https://doi.org/10.1016/j.is.2013.10.006`
- *Voronoi diagrams—a survey of a fundamental geometric data structure (Aurenhammer)* `https://dl.acm.org/doi/10.1145/116873.116880`
- *Hierarchical Navigable Small Worlds (HNSW) (Pinecone blog)* `https://www.pinecone.io/learn/hnsw/`