

Graph-based Approximate Nearest Neighbors Search

Silvio Martinico



Overview

- From Similarity Search to Nearest Neighbors Search
- Graph-based ANN Search
- PROs and CONs of Graph-Based ANN Search
- Non-Sequential Nature of Graphs

From Similarity Search to Nearest Neighbors Search

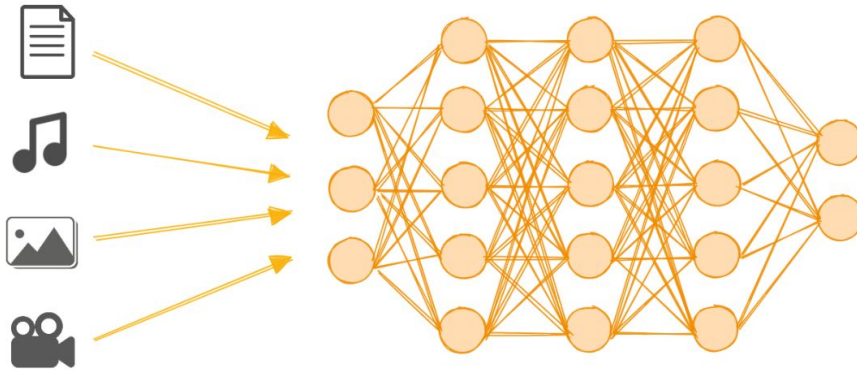


Similarity Search and Neural Networks

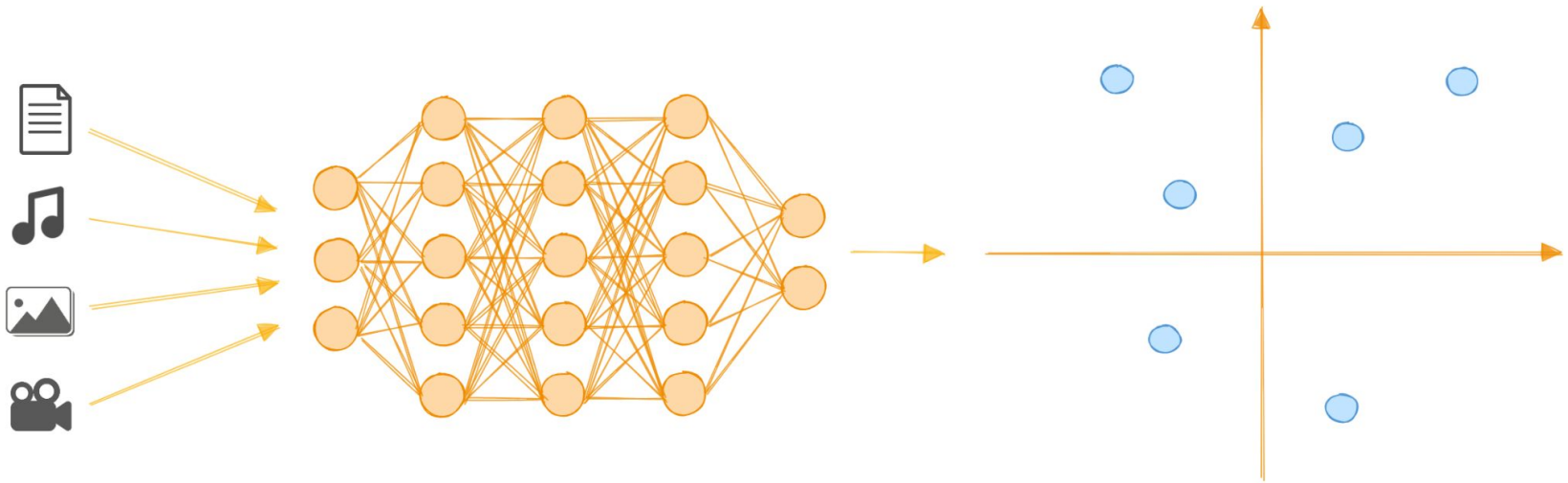
Similarity Search and Neural Networks



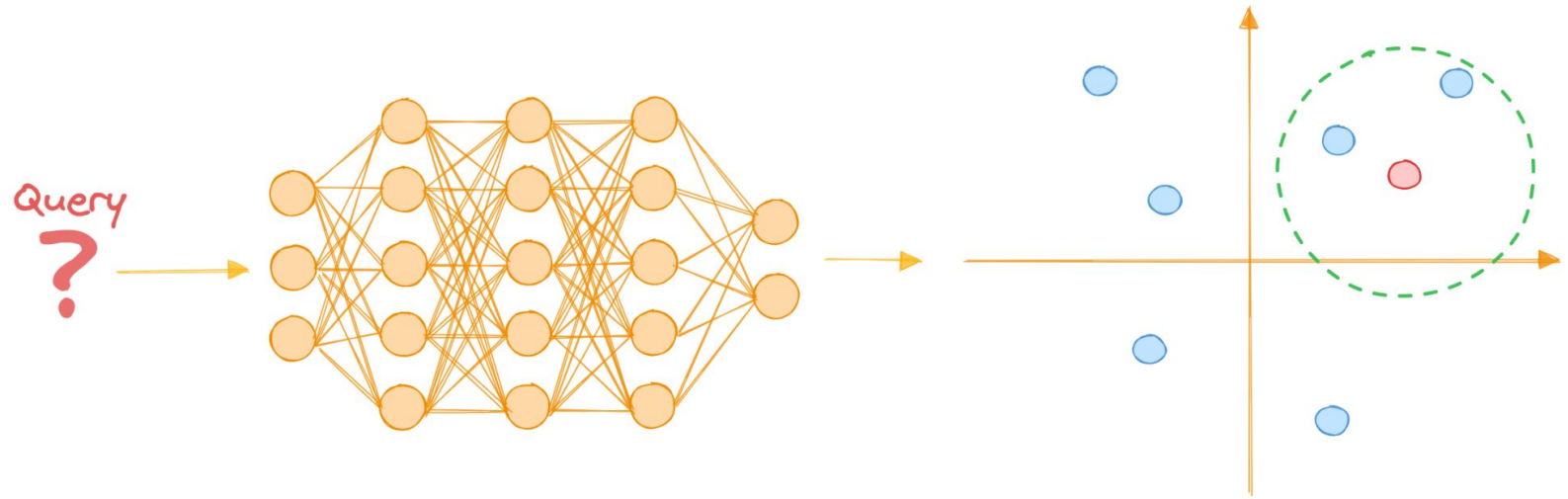
Similarity Search and Neural Networks



Similarity Search and Neural Networks



k-Nearest Neighbors (kNN) Search



Approximate Nearest Neighbors (ANN) Methods

Approximate Nearest Neighbors (ANN) Methods

Three main categories:

Approximate Nearest Neighbors (ANN) Methods

Three main categories:

- Hashing-based methods (LSH^[1])



[1] Indyk, Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. Theory Comput. 1998.

Approximate Nearest Neighbors (ANN) Methods

Three main categories:

- Hashing-based methods (LSH^[1])
- Space partitioning approaches (IVF Index^[2])



[1] Indyk, Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. Theory Comput. 1998.

[2] Jégou, Douze, Schmid. Product quantization for nearest neighbor search. IEEE transactions on pattern analysis and machine intelligence. 2011.

Approximate Nearest Neighbors (ANN) Methods

Three main categories:

- Hashing-based methods (LSH^[1])
- Space partitioning approaches (IVF Index^[2])
- Graph-based methods



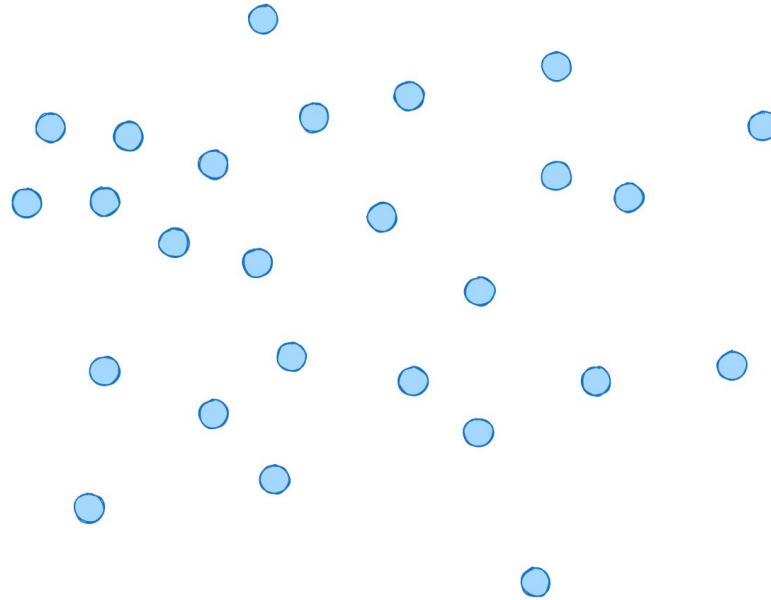
[1] Indyk, Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. Theory Comput. 1998.

[2] Jégou, Douze, Schmid. Product quantization for nearest neighbor search. IEEE transactions on pattern analysis and machine intelligence. 2011.

Graph-Based ANN Search

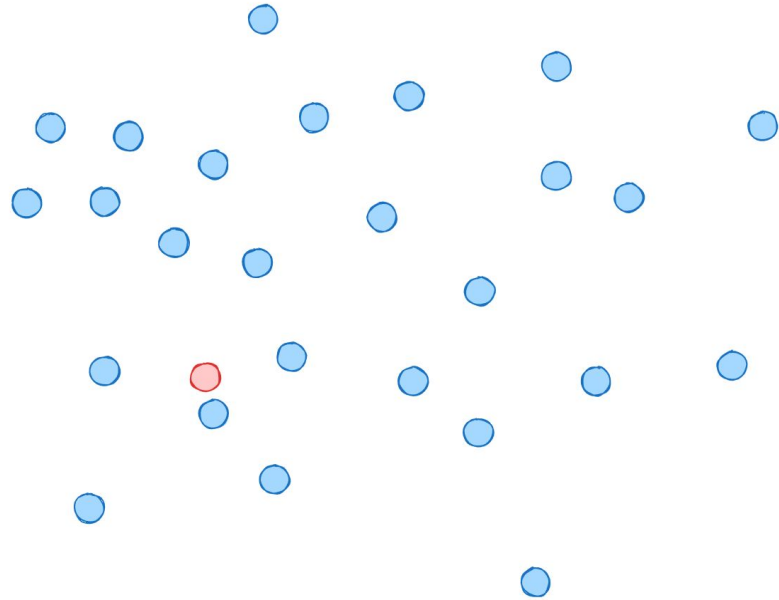


Spatial Approximation

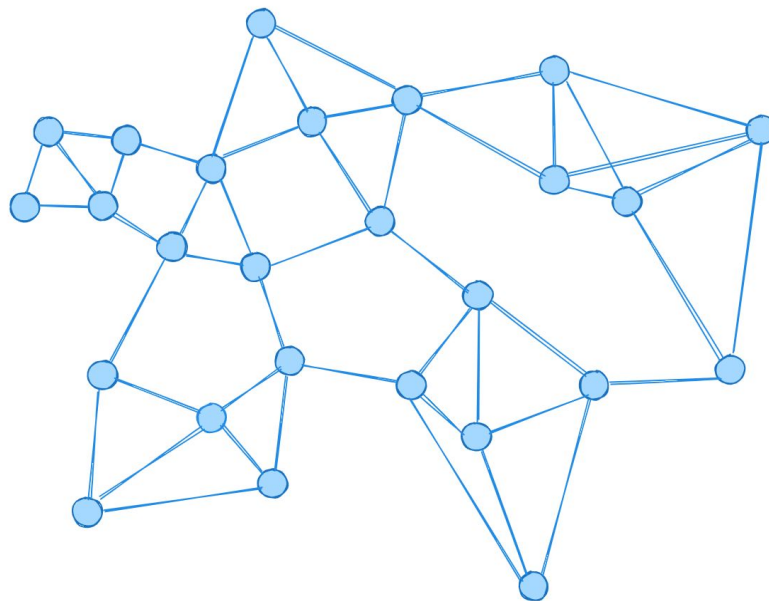


[3] Navarro. Searching in metric spaces by spatial approximation. 6th International Symposium on String Processing and Information Retrieval. 5th International Workshop on Groupware. 1999

Spatial Approximation



KNN-Graph



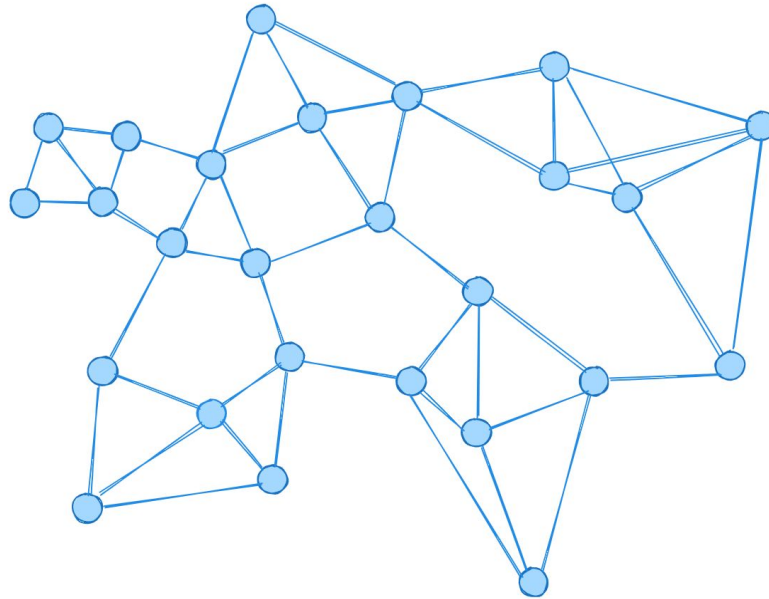
Navigable Small-World Network



[4] Milgram, Travers. An experimental study of the small world problem. *Sociometry* 1969.

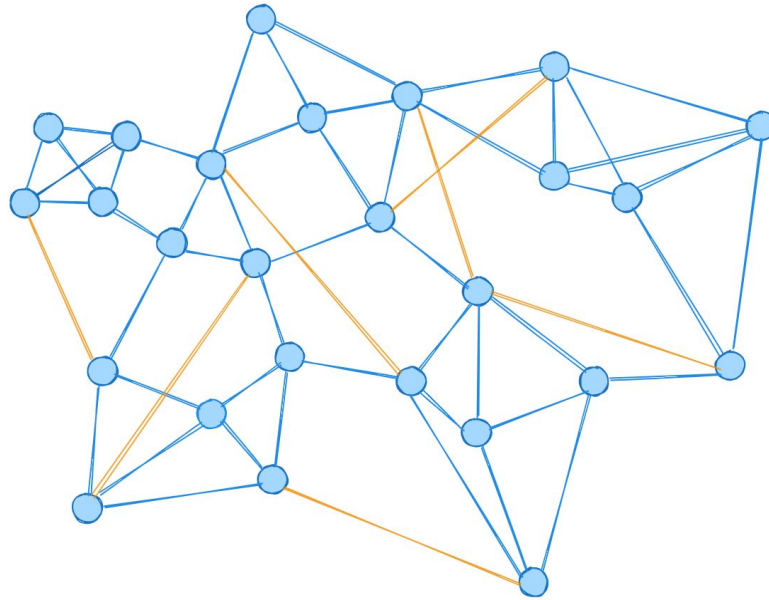
[5] Kleinberg. The small-world phenomenon: An algorithmic perspective. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*. 2001.

Navigable Small-World Graph



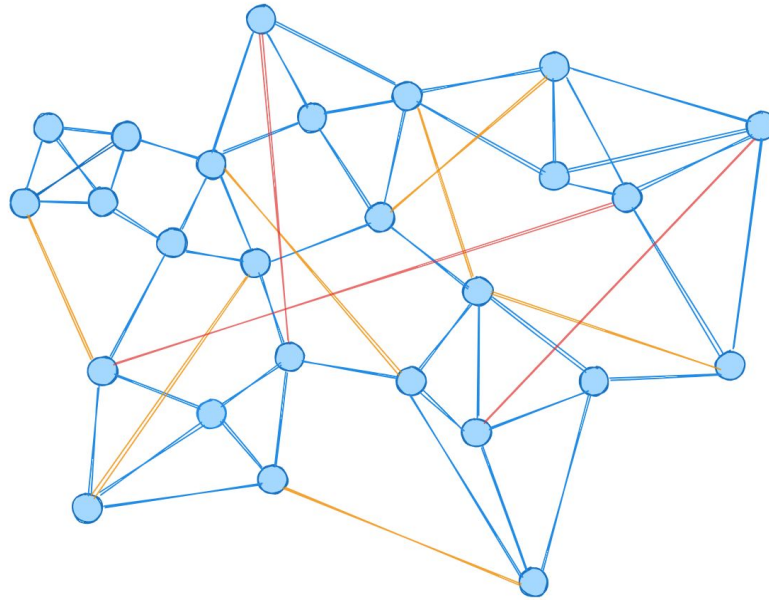
[6] Malkov, Ponomarenko, Logvinov, Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems 45. 2014.

Navigable Small-World Graph



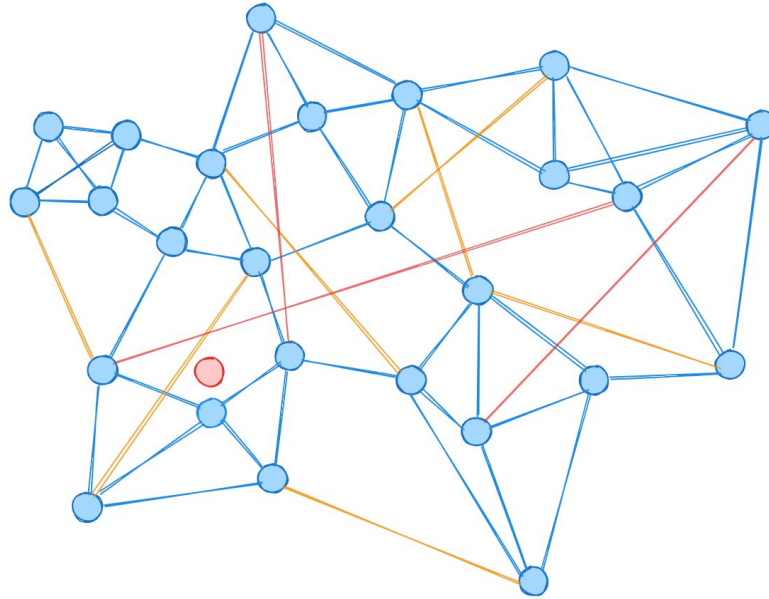
[6] Malkov, Ponomarenko, Logvinov, Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems 45. 2014.

Navigable Small-World Graph

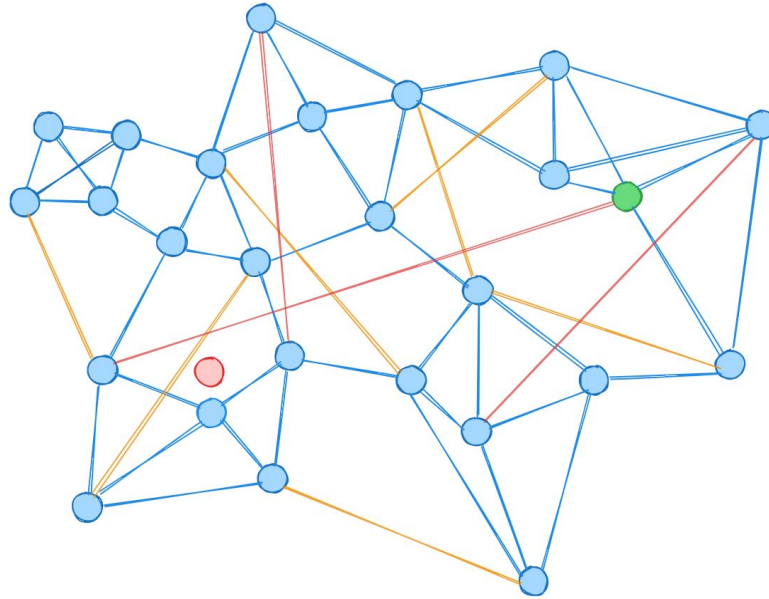


[6] Malkov, Ponomarenko, Logvinov, Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems 45. 2014.

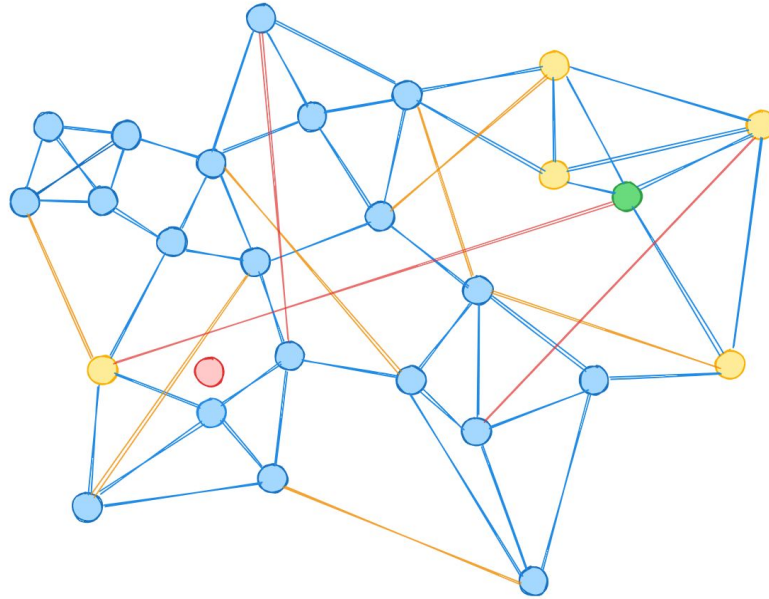
Greedy Search



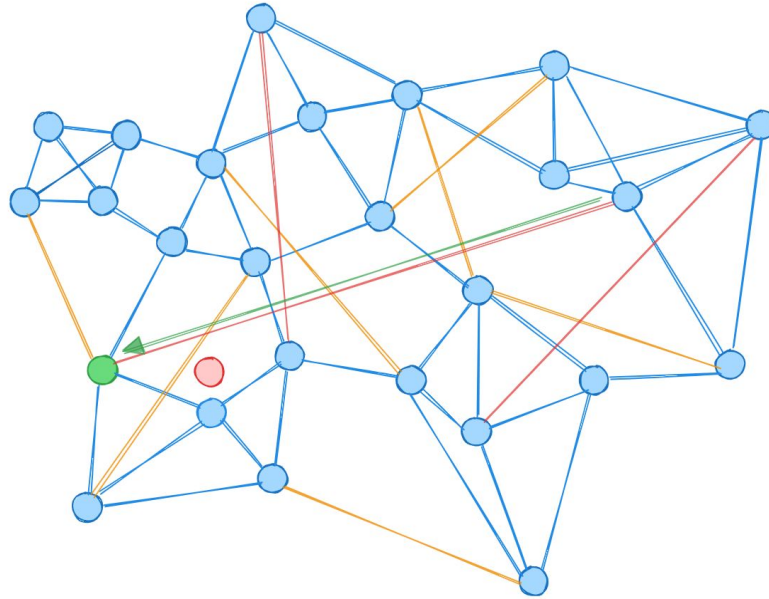
Greedy Search



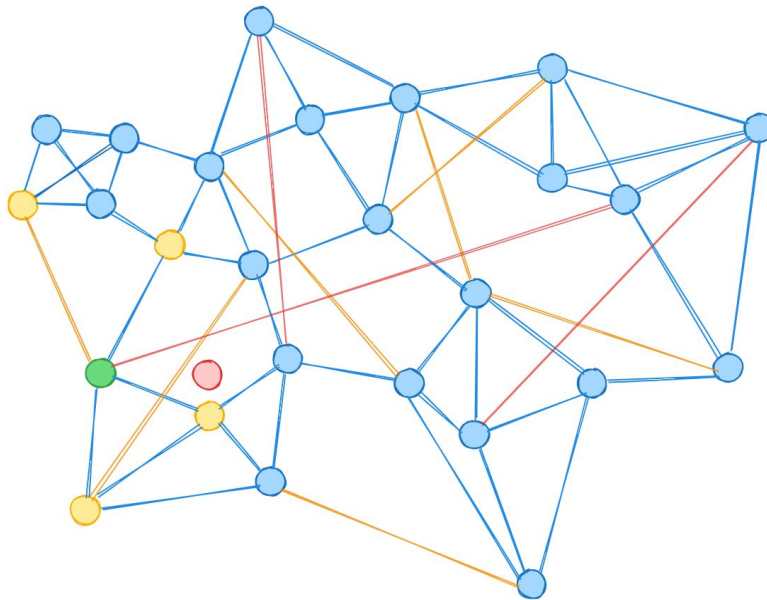
Greedy Search



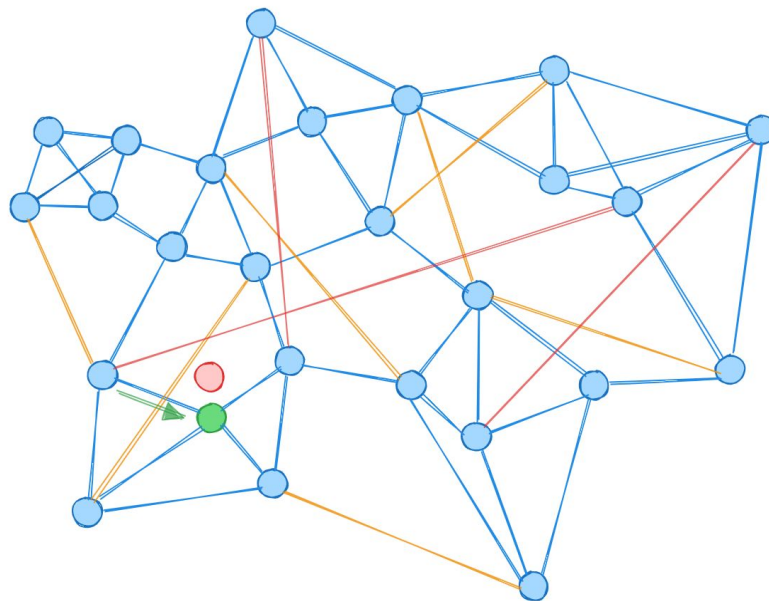
Greedy Search



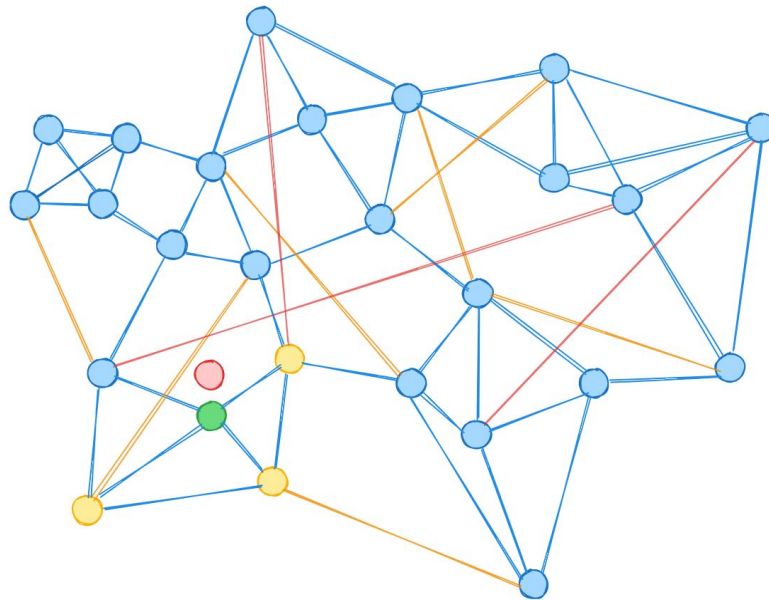
Greedy Search



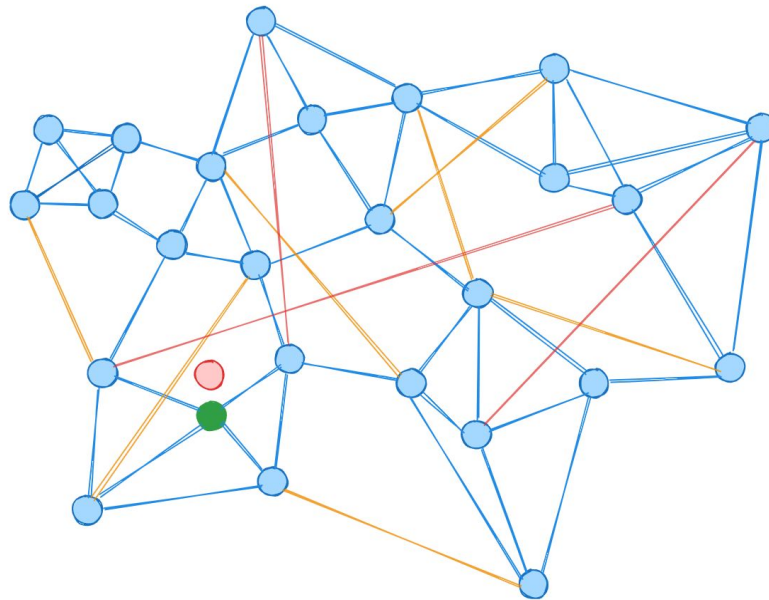
Greedy Search



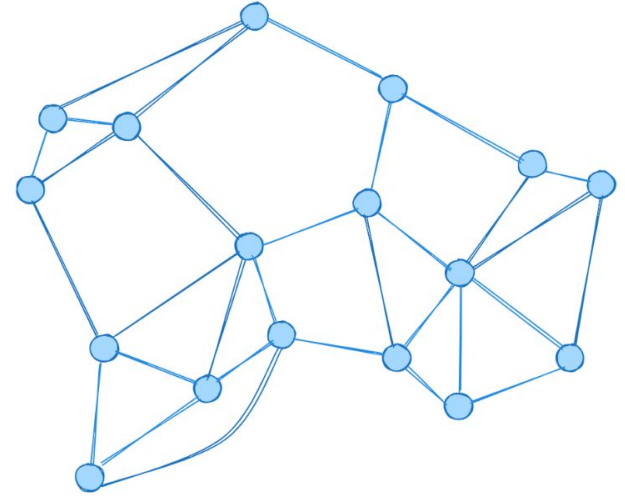
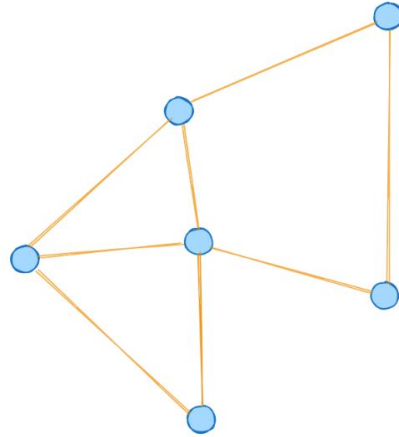
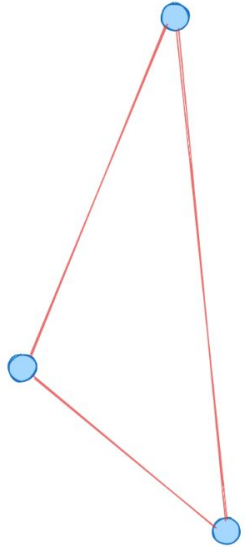
Greedy Search



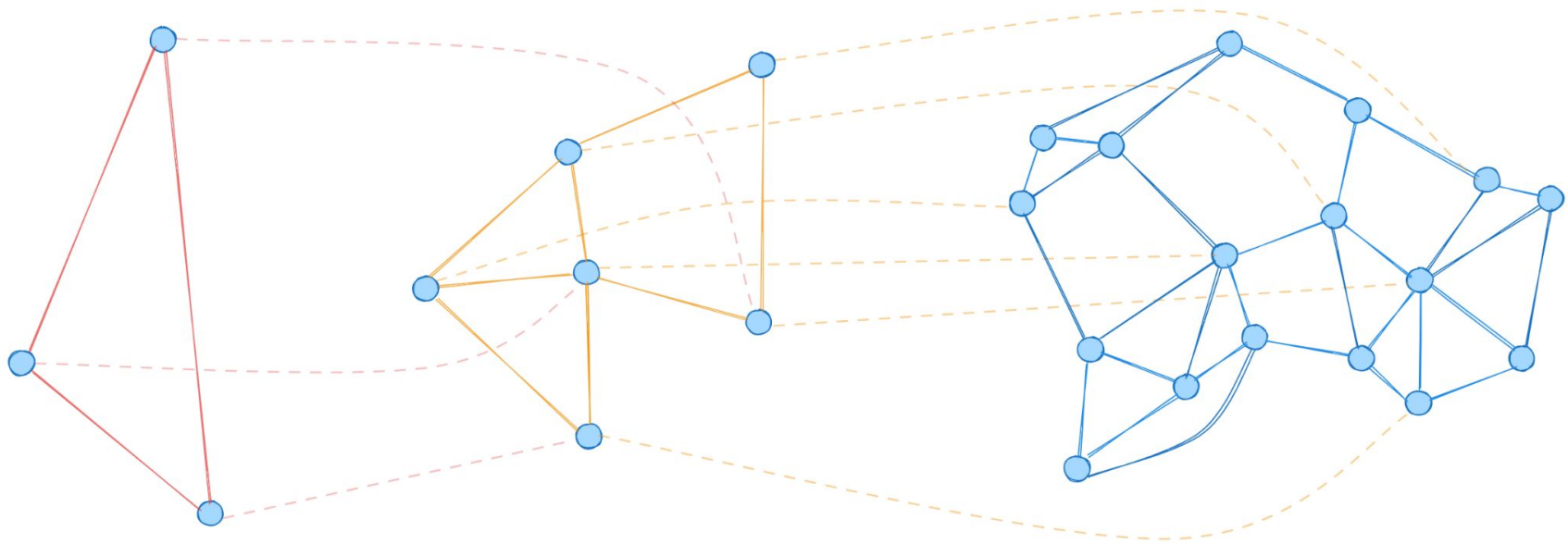
Greedy Search



Introducing Hierarchy

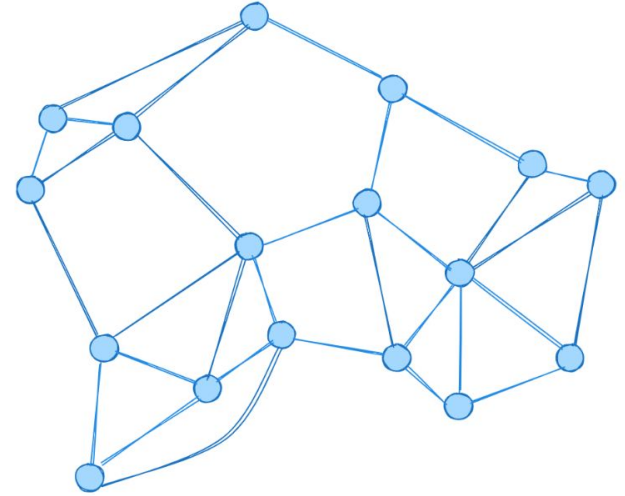
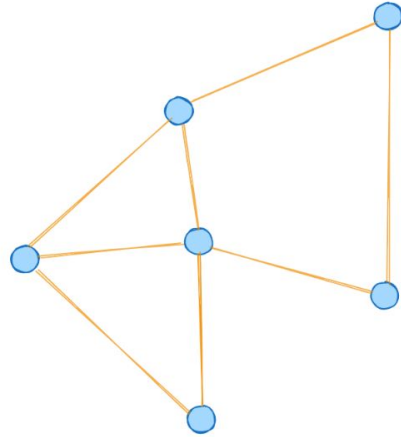
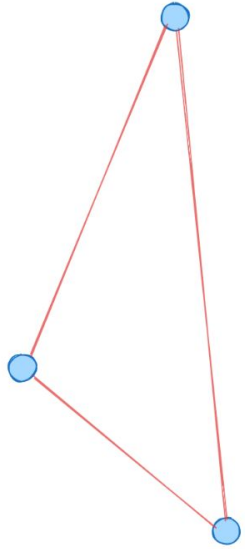


Introducing Hierarchy

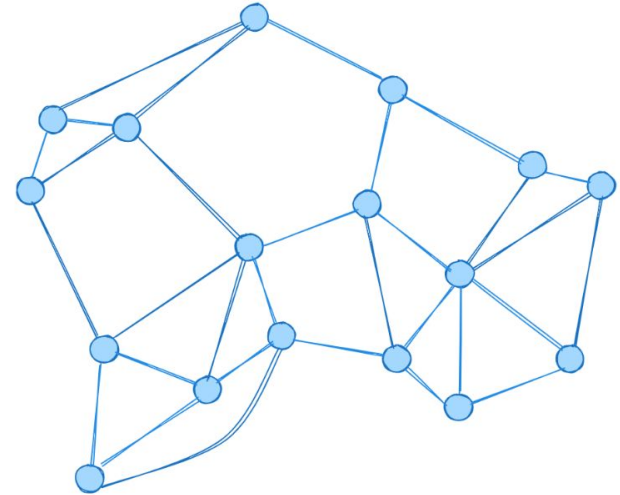
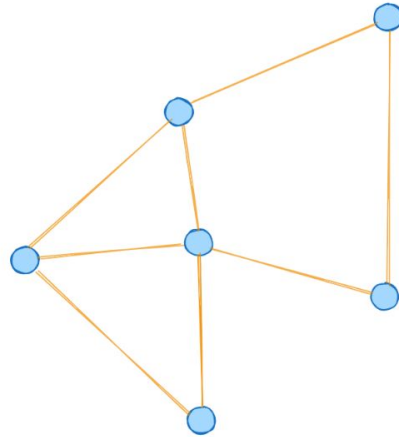
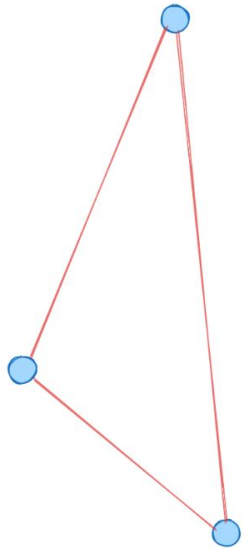


[7] Malkov, Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans. Pattern Anal. Mach. Intell. 2020.

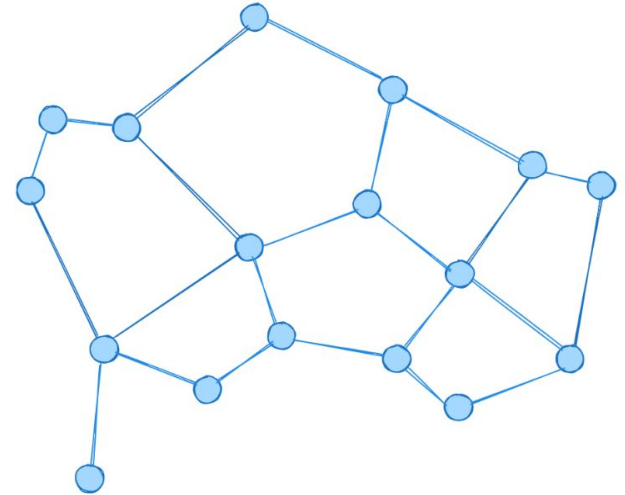
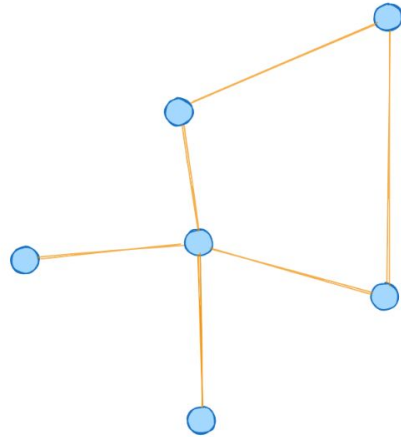
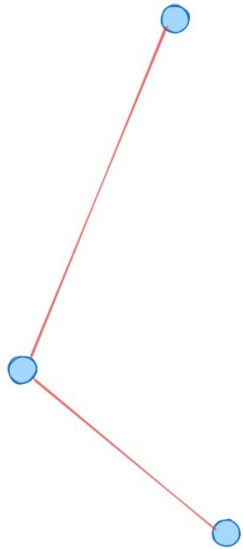
Introducing Hierarchy



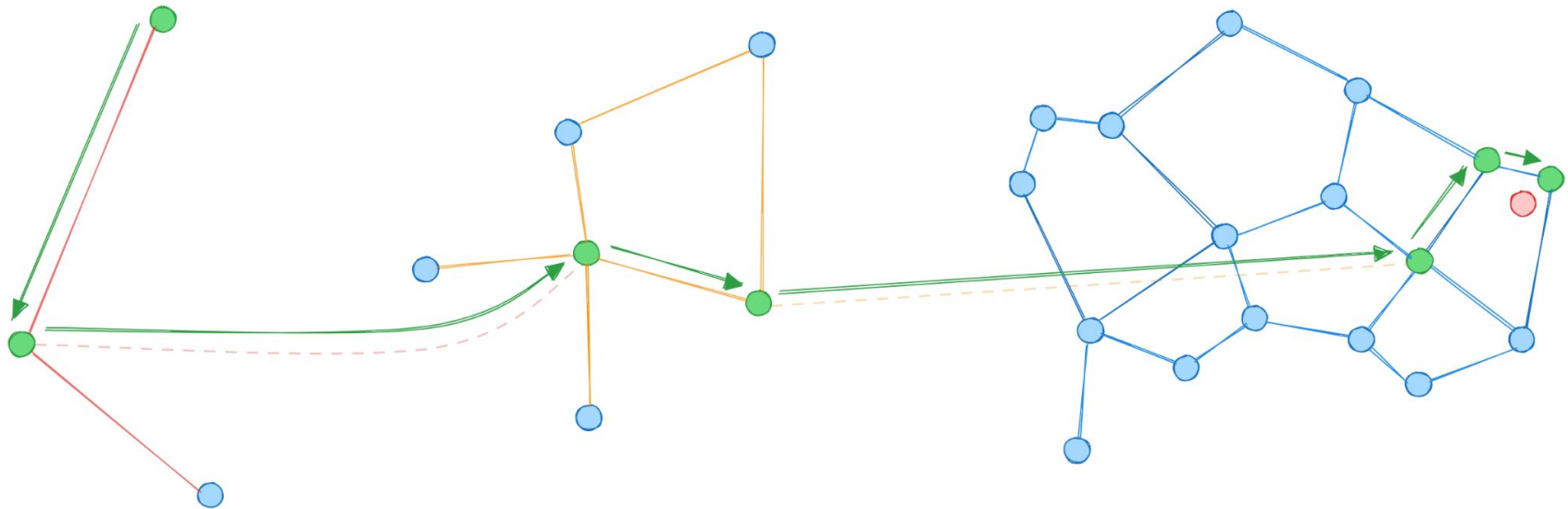
Relative Neighborhood Graph



Relative Neighborhood Graph



Hierarchical Navigable Small-World Graph



[7] Malkov, Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans. Pattern Anal. Mach. Intell. 2020.

PROs and CONs of Graph-Based ANN Search



Graph-based ANN: Advantages

- Fast Search

Graph-based ANN: Advantages

- Fast Search
 - Avoids visiting entire regions of search space

Graph-based ANN: Advantages

- Fast Search
 - Avoids visiting entire regions of search space
 - Drastically reduces the number of visited elements

Graph-based ANN: Advantages

- Fast Search
 - Avoids visiting entire regions of search space
 - Drastically reduces the number of visited elements
- High Efficacy

Graph-based ANN: Advantages

- Fast Search
 - Avoids visiting entire regions of search space
 - Drastically reduces the number of visited elements
- High Efficacy
 - Avoids partitioning the space

Graph-based ANN: Advantages

- Fast Search
 - Avoids visiting entire regions of search space
 - Drastically reduces the number of visited elements
- High Efficacy
 - Avoids partitioning the space
 - Not sensible to small changes

Graph-based ANN: Advantages

- Fast Search
 - Avoids visiting entire regions of search space
 - Drastically reduces the number of visited elements
- High Efficacy
 - Avoids partitioning the space
 - Not sensible to small changes
 - Intrinsic notion of neighborhood

Graph-based ANN: Open Problems

Graph-based ANN: Open Problems

- High Construction Time

Graph-based ANN: Open Problems

- High Construction Time
 - For each node we have to find the right connections

Graph-based ANN: Open Problems

- High Construction Time
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge

Graph-based ANN: Open Problems

- High Construction Time
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge
- High Space Occupancy

Graph-based ANN: Open Problems

- High Construction Time
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge
- High Space Occupancy
 - For each vector we must store a lot of edges

Graph-based ANN: Open Problems

- High Construction Time
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge
- High Space Occupancy
 - For each vector we must store a lot of edges
 - To guarantee high search efficiency, indexes must stay in memory

Graph-based ANN: Open Problems

- High Construction Time
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge
- High Space Occupancy
 - For each vector we must store a lot of edges
 - To guarantee high search efficiency, indexes must stay in memory
- Non-Sequential Nature

Graph-based ANN: Open Problems

- High Construction Time
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge
- High Space Occupancy
 - For each vector we must store a lot of edges
 - To guarantee high search efficiency, indexes must stay in memory
- Non-Sequential Nature
 - Random memory accesses to visit neighbors

Graph-based ANN: Open Problems

- High Construction Time —————> Approximate construction
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge
- High Space Occupancy
 - For each vector we must store a lot of edges
 - To guarantee high search efficiency, indexes must stay in memory
- Non-Sequential Nature
 - Random memory accesses to visit neighbors

Graph-based ANN: Open Problems

- High Construction Time —————> **Approximate construction**
 - For each node we have to find the right connections
 - We must ensure non-redundancy when adding an edge
- High Space Occupancy —————> **Quantization**
 - For each vector we must store a lot of edges
 - To guarantee high search efficiency, indexes must stay in memory
- Non-Sequential Nature
 - Random memory accesses to visit neighbors

Non-Sequential Nature of Graphs



The Problem of Non-Sequentiality



The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞
 - Can be indirectly faced by reducing the number of retrieved nodes 😊

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞
 - Can be indirectly faced by reducing the number of retrieved nodes 😊
 - Early stop strategies

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞
 - Can be indirectly faced by reducing the number of retrieved nodes 😊
 - Early stop strategies
 - Filtering strategies

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞
 - Can be indirectly faced by reducing the number of retrieved nodes 😊
 - Early stop strategies
 - Filtering strategies
 - Reducing number of neighbors

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞
 - Can be indirectly faced by reducing the number of retrieved nodes 😊
 - Early stop strategies } Shorter search path
 - Filtering strategies } Reduced number of distances computed per visited node
 - Reducing number of neighbors }

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞
 - Can be indirectly faced by reducing the number of retrieved nodes 😊
 - Early stop strategies
 - Filtering strategies
 - Reducing number of neighbors
- } Orthogonal strategies

The Problem of Non-Sequentiality

- Non-Sequential Storage of Neighbors
 - Intrinsic in the nature of a Graph 😞
 - Can be indirectly faced by reducing the number of retrieved nodes 😊
 - Early stop strategies
 - Filtering strategies
 - Reducing number of neighbors
- } Improvements to existing graph indexes

Reducing Number of Neighbors

- Optimizing connections (e.g. pruning)

Reducing Number of Neighbors

- Optimizing connections (e.g. pruning)
 - Small improvement margin, state-of-the-art graph structures are already highly optimized

Reducing Number of Neighbors

- Optimizing connections (e.g. pruning)
 - Small improvement margin, state-of-the-art graph structures are already highly optimized
- **Novel graph structure** which is sparser and more navigable by construction

Reducing Number of Neighbors

- Optimizing connections (e.g. pruning)
 - Small improvement margin, state-of-the-art graph structures are already highly optimized
- **Novel graph structure** which is sparser and more navigable by construction

Thanks for your attention!

Q&A Time

