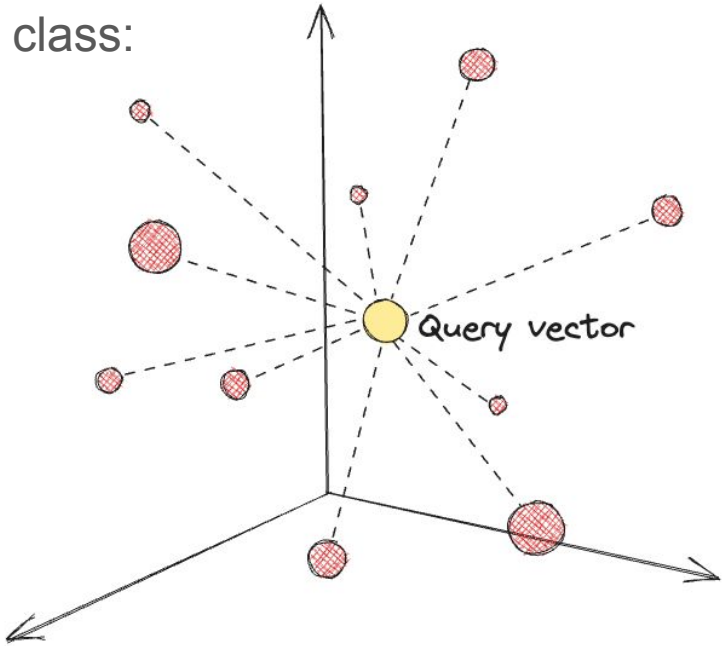
K-NN and A-NN

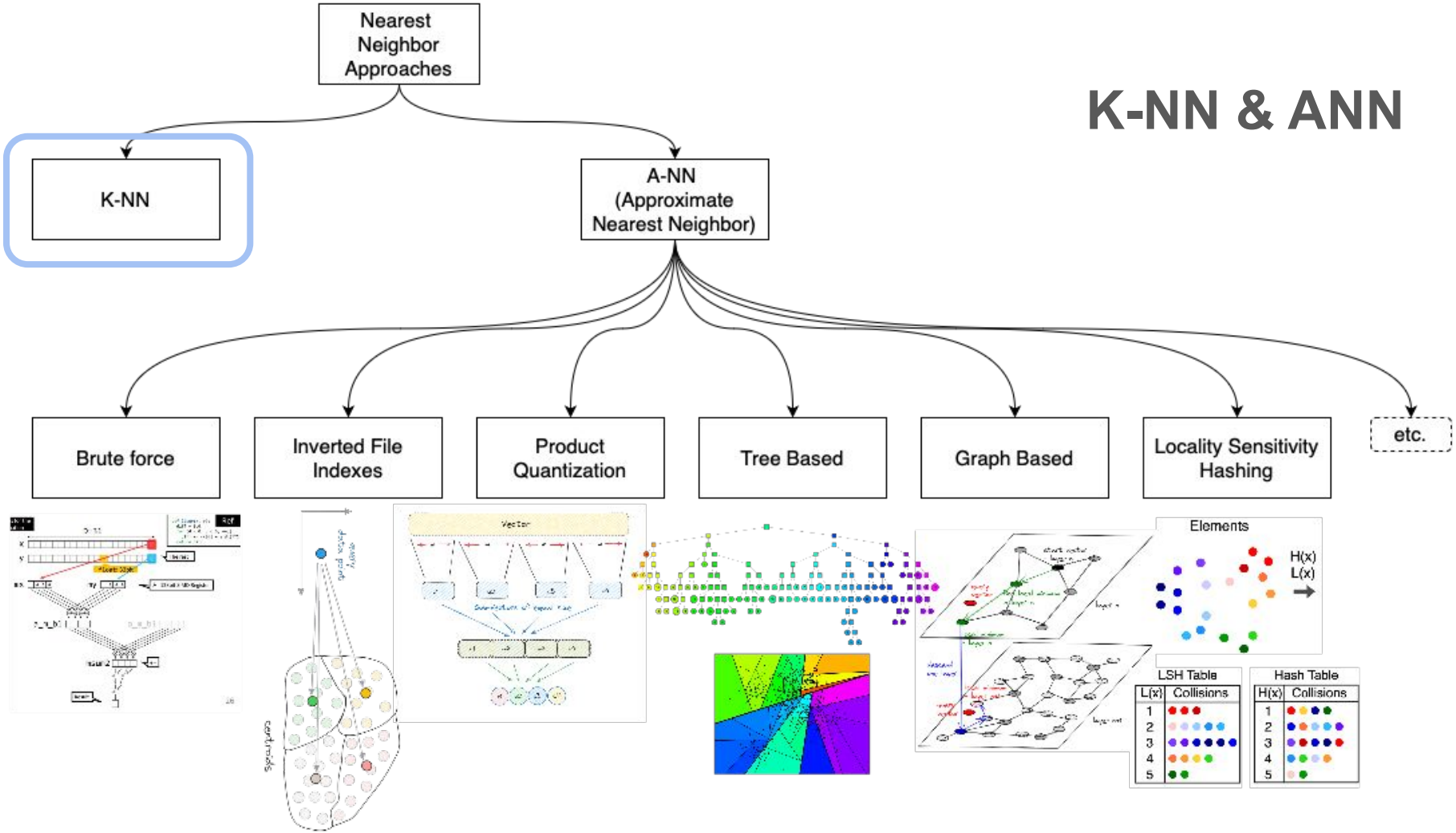
Characteristics, strengths and weaknesses

Vector Space Classification

- Documents as points in a vector-space
- Contiguity hypothesis
- Vector based classification methods from this class:
  - Rocchio
  - SVM
  - **K-NN**
  - **A-NN**

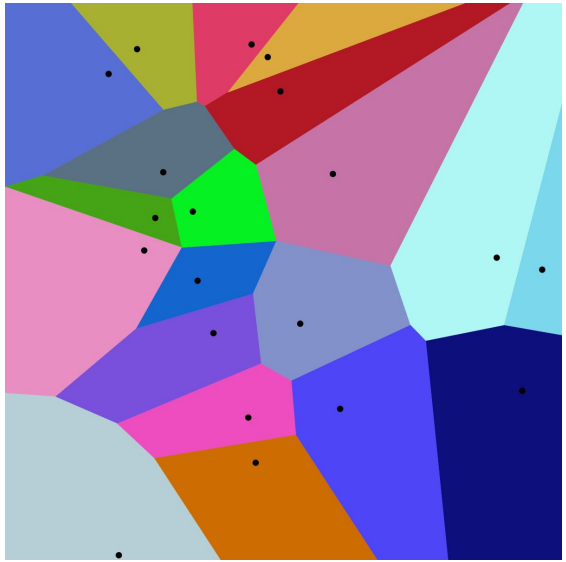


K-NN & ANN


Nearest Neighbor Approaches
K-NN
A-NN (Approximate Nearest Neighbor)
Brute force
Inverted File Indexes
Product Quantization
Tree Based
Graph Based
Locality Sensitivity Hashing
etc.
Vector
Elements
H(x)
L(x)
LSH Table
L(x)
Collisions
Hash Table
H(x)
Collisions

K-NN

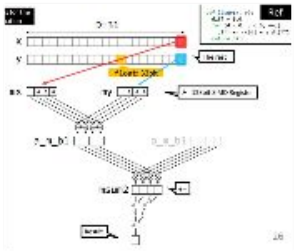
- Documents as points in the vector space
- Measure the distance to each document
- Finding the K nearest
- Assigning a class

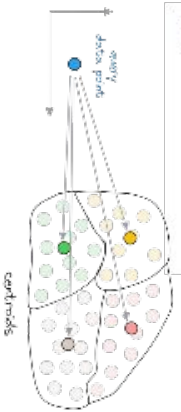


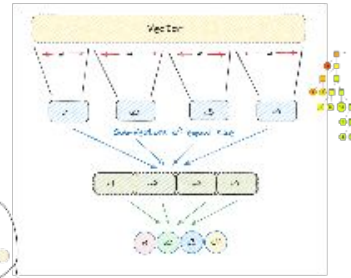
Voronoi

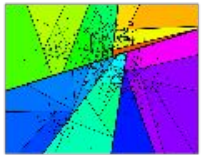
K-NN & ANN

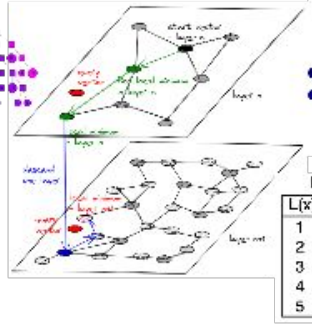
- Nearest Neighbor Approaches
  - K-NN
  - A-NN (Approximate Nearest Neighbor)
    - Brute force
    - Inverted File Indexes
    - Product Quantization
    - Tree Based
    - Graph Based
    - Locality Sensitivity Hashing
    - etc.

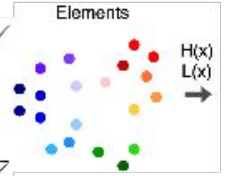


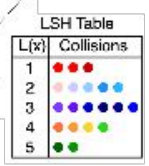


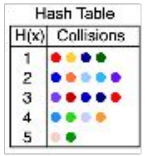












Approximate Nearest Neighbor

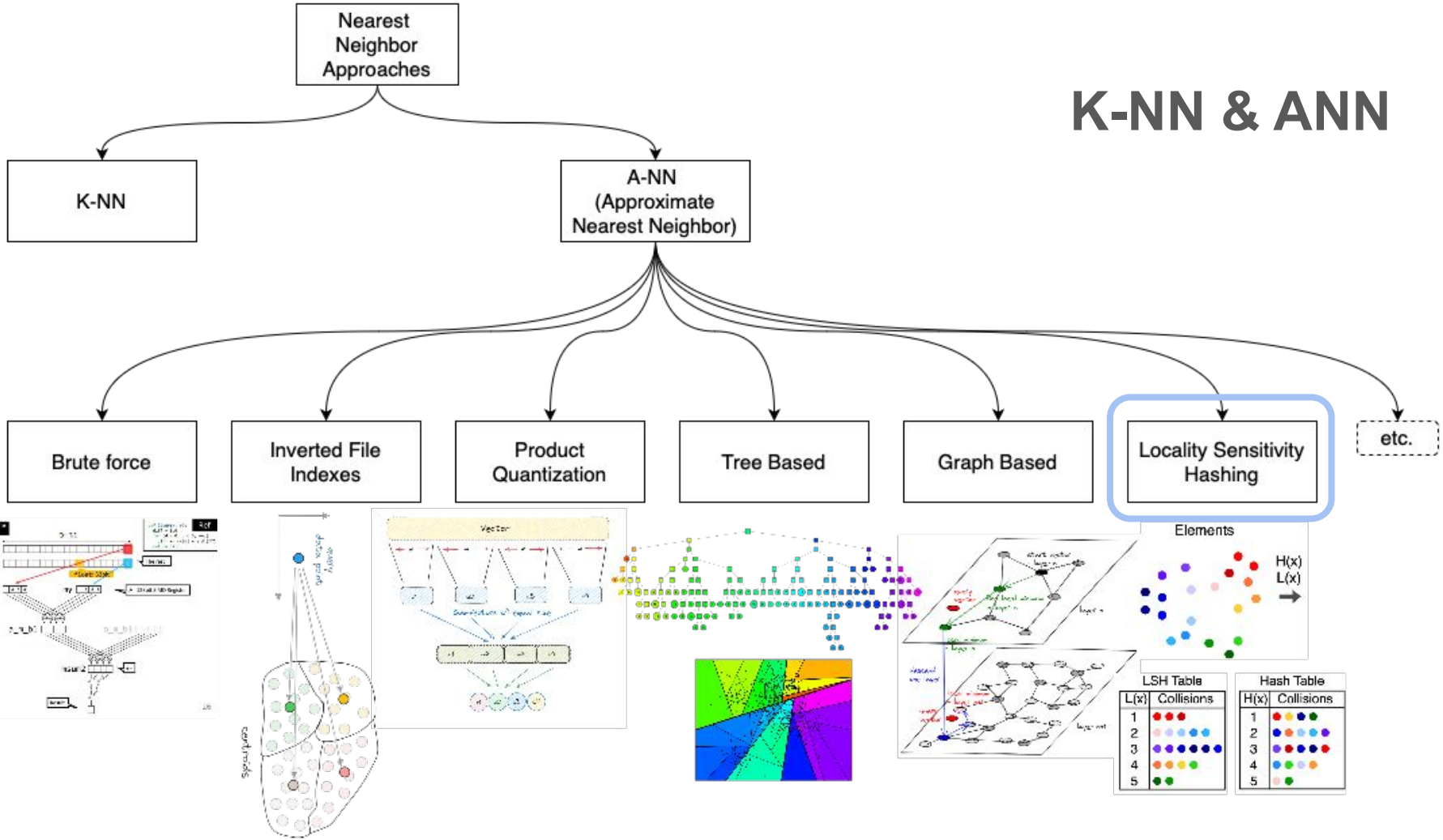
Speeding up KNN with approximation

- Multiple techniques for approximation
- ANN sacrifices accuracy for speed
- Efficient for large-scale similarity search

Idea:

Reduce the number of vectors/documents you have to compare your new document with.

K-NN & ANN


Nearest Neighbor Approaches
K-NN
A-NN (Approximate Nearest Neighbor)
Brute force
Inverted File Indexes
Product Quantization
Tree Based
Graph Based
Locality Sensitivity Hashing
etc.
Vector
Elements
H(x)
L(x)
LSH Table
L(x)
Collisions
Hash Table
H(x)
Collisions

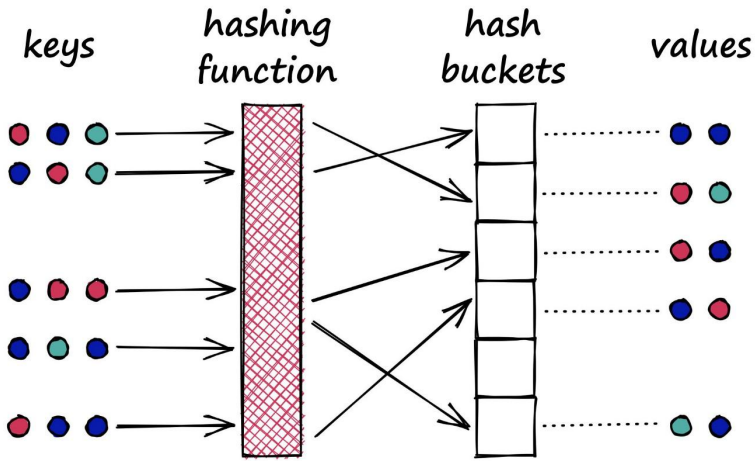
Locality-Sensitive Hashing

Pre-processing:

- All the documents in the collection are hashed into buckets.

Classification:

- Do the same with new document *d*.
- Only calculate the distance to the documents which are in the same buckets as *d* hashed to.



https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing/

Strength and weaknesses

K-NN

- Simple and easy to understand
- Interpretable (based on real observed data points)
- Accurate, but costly
- Do not scale

When precision is important

A-NN

- Often more complex and less interpretable
- Less accurate
- Scalable
- Adjust approximation method to your need
- Pros and cons with each approximation method

When efficiency is important

Nearest Neighbor Approaches

K-NN

A-NN (Approximate Nearest Neighbor)

Brute force

Inverted File Indexes

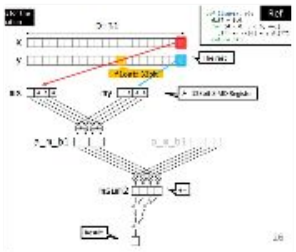
Product Quantization

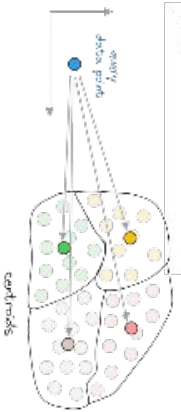
Tree Based

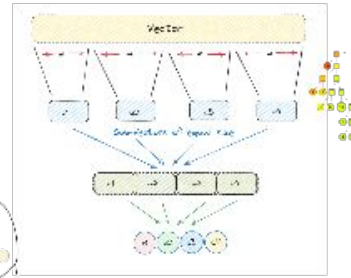
Graph Based

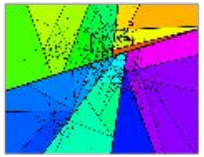
Locality Sensitivity Hashing

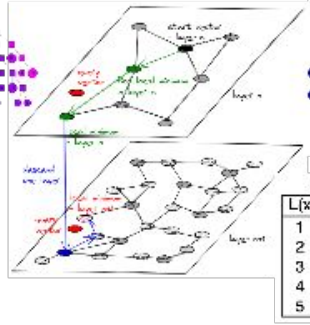
etc.

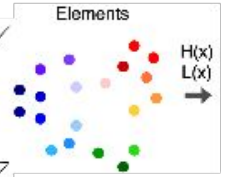


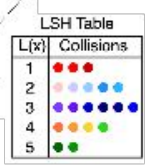


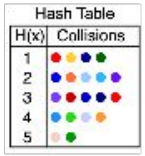












Sources

- https://www.elastic.co/blog/ann-vs-knn
- https://towardsdatascience.com/comprehensive-guide-to-approximate-nearest-neighbors-algorithms-8b94f057d6b6
- https://thedataquarry.com/posts/vector-db-3/
- Lecture slides on ANN and KNN