Approximate k-NN Graph in Chameleon Clustering

M. Sai Akshay Reddy, Rishi Parsai
BTP : End-Semester

Supervisor:
Prof. Kapil Ahuja

Mentor: Priyanshu Singh

Computer Science and Engineering
Indian Institute of Technology Indore

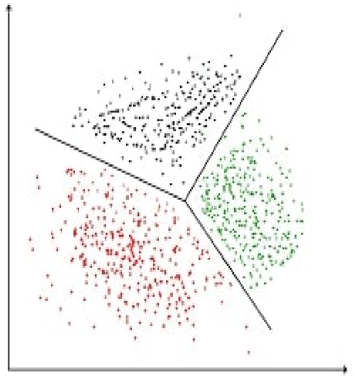
November 24, 2023

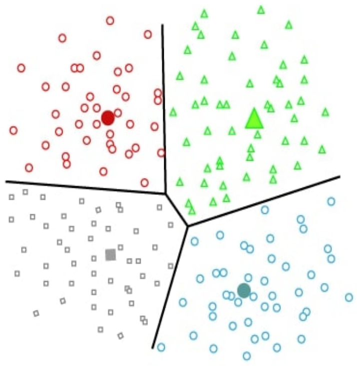
Table of Contents

What is Clustering?

- It is a process that organizes data into clusters, maximizing similarity within clusters and minimizing similarity between them.
- Used in various domains like engineering, sciences, marketing, technology, humanities, medical sciences, etc.

Clustering





Limitations of Traditional Clustering Algorithms?

- Common clustering algorithms like K-means, DBSCAN, CURE, ROCK, etc are tailored for static models.
- These algorithms can encounter issues if the selected parameters fail to capture the true characteristics of the clusters.

Drawback of K-means Algorithm

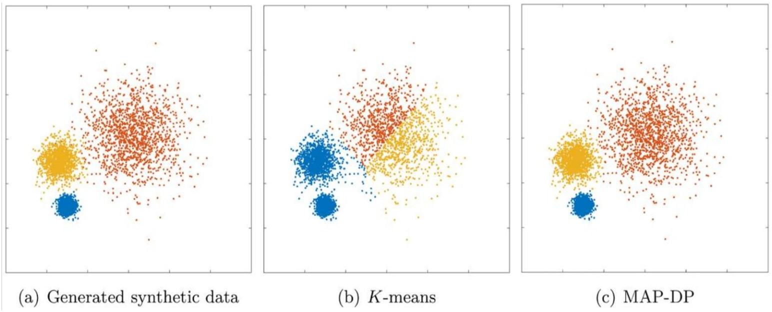


Figure: K-means fails to properly identify the clusters having varying radius and density

Chameleon Clustering

- Ch. Clustering finds cluster similarity using **dynamic model**.
- Clusters are merged only if they satify the merging criteria, computed using RI and RC metrics. [3]

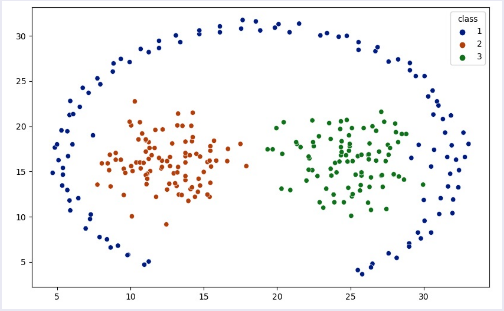


Figure: pathbased dataset, Ch2 : 0.887 , DBSCAN : 0.725

Continued...

- CHAMELEON is a two-phase algorithm .
  1. **K-NN Construction (Pre-Computation)**
  2. **Partitioning Phase (hMETIS Library)**
  3. **Merging Phase**

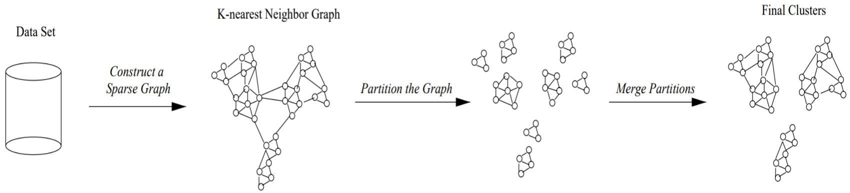
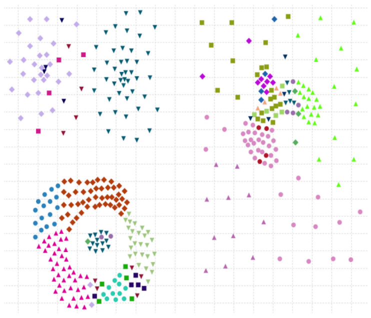
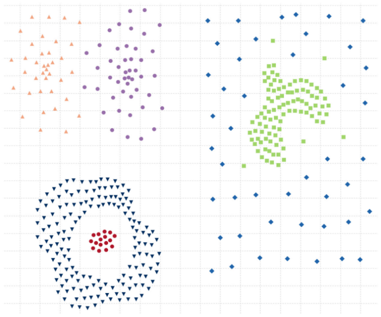


Figure: Two Phase Algorithm

Drawbacks of Chameleon Algorithm

- The main drawback of Chameleon 1 lies in its inability to manage small clusters unlike DBSCAN and CURE, Chameleon doesn't handle noise properly.


(a) Ch1 (NMI = 0.64, 20 clusters)


(d) Ch2 (NMI = 0.96, 6 clusters)

Chameleon 2 Algorithm

- Chameleon 2 (Ch2) is an improved version of Chameleon Algorithm (Ch1)
- Symmetrical K-NN Graph is constructed and this eliminates many between cluster connections and lead to better results.
- Flood Fill Algorithm is applied to refine the partition done recursive bisection or hMETIS.
- Similarity Measures are improved compared to Chameleon 1. [1]

Overview of Chameleon family Approaches

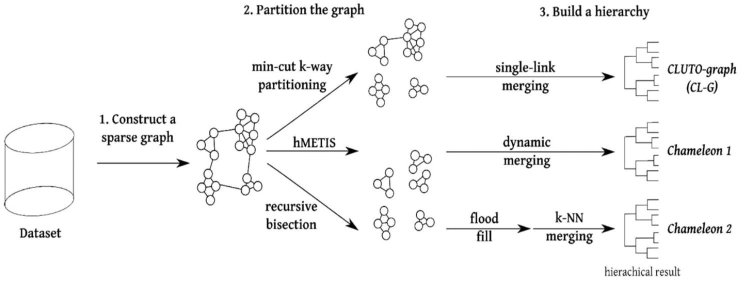


Figure: Overview of Chameleon 2

Insights into ANNoy

ANNoy It divides the data space into regions and organize data points in structures like trees (e.g: KD-Trees, RP-Trees, Ball Trees, etc).

- **Use Case :** It is suitable for medium dimensional data.
- **Pros :** ANNoy provides a good balance between accuracy and speed.
- **Cons :** But it suffers with the curse of dimensionality, experiencing a decrease in effectiveness as the number of dimensions increases.

Approximate Nearest Neigbours Oh Yeah: ANNoy

Searching in ANNoy

- Annoy constructs multiple random projection trees.
- Each tree is created independently through recursive partitioning of data points.
- In each iteration, two centers are generated using a simplified clustering algorithm on a subset of input data samples.
- The two centers establish a partition hyperplane equidistant from them. Data points are then divided into two sub-trees based on this hyperplane.

Insights into FLANN

FLANN is a space partitioning based method and it divides the data space into regions and organize data points in structures like trees.

- **Use Case :** It is suitable for medium dimensional data.
- **Pros :** FLANN can provide a good balance between accuracy and speed.
- **Cons :** But it suffers from curse of dimensionality. As the number of dimension increase, the effectiveness decreases.

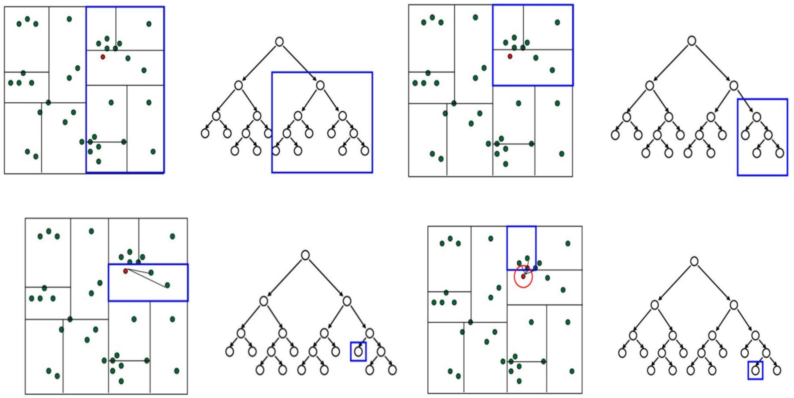
Fast Library for Approximate Nearest Neighbour: FLANN

Introduction

- This Library offers three different algorithms for ANN search, they are: *Randomized k-d trees*, *Hierarchical k-means tree* and *Standard linear scan*.
- We have incorporated *Randomized k-d trees* method because of its *higher accuracy* and *easier fine tuning*.
- For lower dimensions it has been shown that the time complexity of k-d trees method for finding NNs for n points comes out to be $O(n \log n)$. [2]

Finding Nearest Neighbour in k-d tree

- Explore the branch of tree closest to the query point.
- On reaching the leaf node, distance between the points in region and query point is calculated.



Insights into HNSW

HNSW is a method which is based on graph structures and uses the concept of navigable small world and probability skip list

- **Use Case :** It is used when maintaining the quality of one's neighbors is critical.
- **Pros :** It often provides very high-quality neighbors.
- **Cons :** But the graph construction phase can bee time taking.

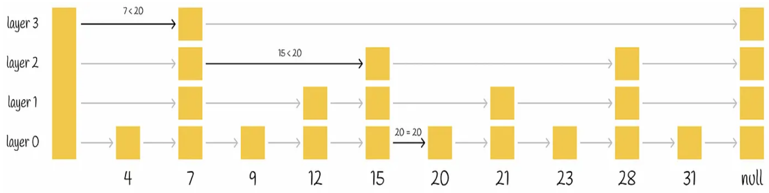
Hierarchical Navigable Small World: HNSW

Introduction

- HNSW is a proximity graph based method.
- HNSW is an efficient in high-dimensional space since its index size is independent of d, but other traditional ANN search methods become inefficient due to *curse of dimensionality*. [4]
- The two fundamental techniques that contributed most heavily to HNSW: are **probability skip list**, and **navigable small world graphs**.

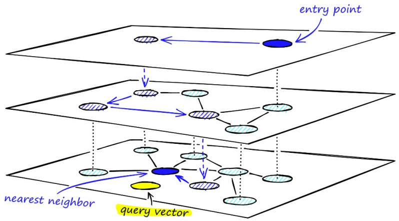
Probability Skip List

- Skip list is a data structure that allows inserting and searching elements within a sorted list for O(logn) on average.
- A skip list is constructed by several layers of linked lists, with lowest layer as the original linked list having all elements.
- When moving to higher levels, the number of skipped elements increases, thus decreasing the number of connections.

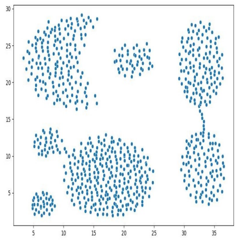


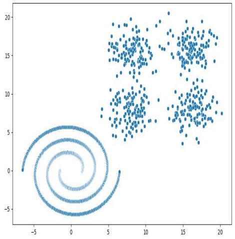
Coming back to HNSW

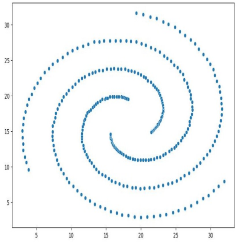
- HNSW adds hierarchy to NSW and produces a graph where links are separated across different layers.
- At the top layer, we have the longest links, and at the bottom layer, we have the shortest.
- Process starts from the top layer and proceeds to one level below every time the local NN is greedily found among the layer nodes. The local NN at bottom layer is the answer.

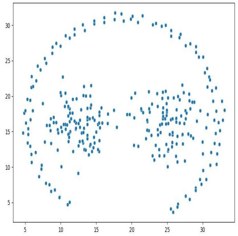


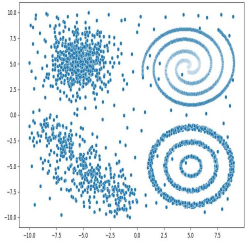
Representation of some Benchmark datasets

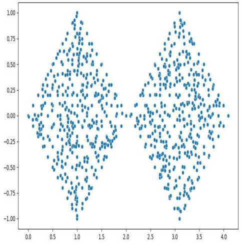












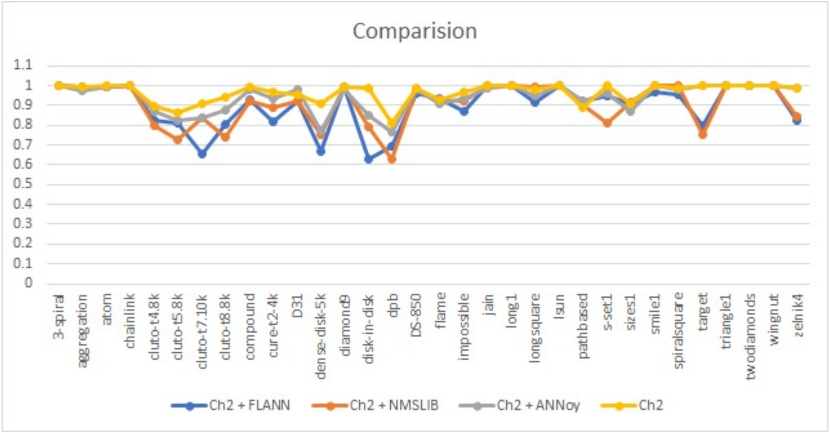
Results: NMI Values on Ch2 Benchmark Datasets

Dataset	Ch2 + FLANN	Ch2 + HNSW	Ch2 + ANNoy	Ch2
3-spiral	1	1	1	1
jain	1	1	0.987	1
long1	1	1	1	1
lsun	1	1	1	1
smile1	0.967	1	0.997	1
target	1	1	1	1
triangle1	1	1	0.997	1
twodiamonds	1	1	1	1
wingnut	1	1	1	1
atom	0.994	0.991	0.993	1
chainlink	1	1	1	1
s-set1	0.95	0.9	0.96	0.997
diamond9	0.996	0.991	0.982	0.993
aggregation	0.996	0.993	0.974	0.992
compound	0.927	0.955	0.978	0.992
disk-in-disk	0.631	0.873	0.85	0.99

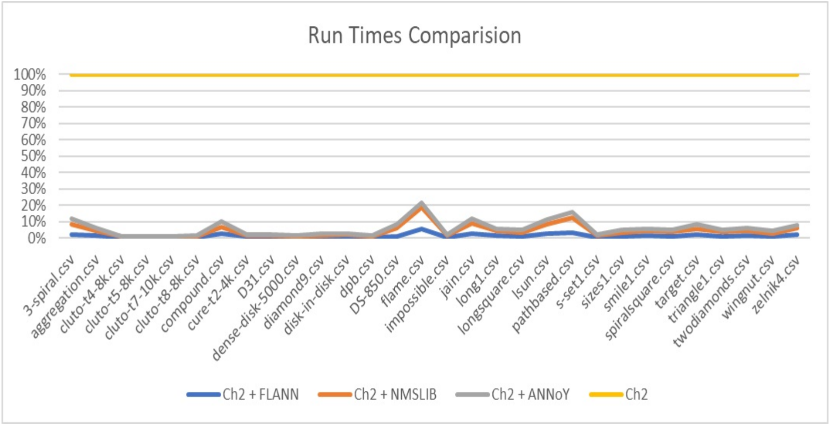
Results: Continued

Dataset	Ch2 + FLANN	Ch2 + HNSW	Ch2 + ANNoy	Ch2
spiralsquare	0.953	0.998	0.983	0.987
zelnik4	0.827	0.845	0.986	0.987
DS-850	0.963	0.979	0.983	0.984
longsquare	0.916	0.993	0.95	0.981
impossible	0.872	0.922	0.938	0.969
cure-t2-4k	0.817	0.912	0.938	0.967
D31	0.921	0.922	0.978	0.957
cluto-t8.8k	0.804	0.812	0.875	0.944
flame	0.935	0.927	0.907	0.927
cluto-t7.10k	0.652	0.838	0.84	0.912
sizes1	0.911	0.915	0.87	0.909
dense-disk-5k	0.67	0.779	0.775	0.908
cluto-t4.8k	0.825	0.821	0.871	0.893
pathbased	0.919	0.905	0.924	0.887
cluto-t5.8k	0.812	0.822	0.824	0.864
dpb	0.692	0.649	0.767	0.81
AVG.	**0.905**	**0.929**	**0.941**	**0.964**
SD.	**0.113**	**0.086**	**0.07**	**0.049**

Graphical Comparison of NMI Values



Graphical Comparison of Runtime



Results: Percentage difference on Ch2 Benchmark Datasets

Dataset	FLANN - Ch2(%)	HNSW - Ch2(%)	ANNoy - Ch2(%)
3-spiral	0	0	0
aggregation	0.37	0.07	-1.8
atom	-0.6	-0.9	-0.7
chainlink	0	0	0
cluto-t4.8k	-7.61	-8.06	-2.2
cluto-t5.8k	-6.02	-4.86	-4
cluto-t7.10k	-28.51	-8.11	-7.2
cluto-t8.8k	-14.83	-13.98	-6.9
compound	-6.59	-3.73	-1.4
cure-t2-4k	-15.51	-5.69	-2.9
D31	-3.8	-3.67	2.1
dense-disk-5k	-26.25	-14.21	-13.3
diamond9	0.26	-0.19	-1.1
disk-in-disk	-36.3	-11.82	-14
dpb	-14.57	-19.88	-4.3
DS-850	-2.13	-0.48	-0.1

Results: Continued

Dataset	FLANN - Ch2(%)	HNSW - Ch2(%)	ANNoy - Ch2(%)
flame	0.91	-0.01	-2
impossible	-9.99	-4.89	-3.1
jain	0	0	-1.3
long1	0	0	0
longsquare	-6.62	1.18	-3.1
lsun	0	0	0
pathbased	3.59	2.04	3.7
s-set1	-4.71	-9.73	-3.7
sizes1	0.26	0.63	-3.9
smile1	-3.28	0	-0.3
spiralsquare	-3.49	1.07	-0.4
target	0	0	0
triangle1	0	0	-0.3
twodiamonds	0	0	0
wingnut	0	0	0
zelnik4	-16.21	-14.39	-0.1
AVG.	**-5.94**	**-4.61**	**-2.26**
SD.	**6.41**	**5.03**	**2.08**

Conclusions

- **FLANN** shows significant performance drops exceeding 5% on datasets like cluto-t7.10k, dense-disk-5k, etc.
- **HNSW** also exhibits notable decreases exceeding 5% on some datasets and has an average decrease of 4.61%.
- **ANNoy**, with generally smaller decreases, experiences notable declines of over 5% on datasets like cluto-t7.10k, dense-disk-5k.

Conclusions: Continued

- In summary, FLANN and HNSW face notable performance drops on specific datasets, with FLANN experiencing more significant declines on average.
- In contrast, ANNoy generally exhibits smaller decreases (2.26%) across the evaluated datasets than FLANN (5.94%) and HNSW (4.61%).

Future Work

- A novel approach can be developed by incorporating techniques from modern-day graph theory.
- There are few optimizations that can be done in the merging phase as well, like tweaking the values of α and β.
- The improved algorithm can be applied to real-world datasets, including astronomical data.

References I

[1] Barton, T., Bruna, T., and Kordik, P. Chameleon 2: an improved graph-based clustering algorithm. *ACM Transactions on Knowledge Discovery from Data (TKDD) 13*, 1 (2019), 1–27.

[2] Friedman, J. H., Bentley, J. L., and Finkel, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS) 3*, 3 (1977), 209–226.

[3] Karypis, G., Han, E.-H., and Kumar, V. Chameleon: Hierarchical clustering using dynamic modeling. *computer 32*, 8 (1999), 68–75.

References II

[4] Li, W., Zhang, Y., Sun, Y., Wang, W., Li, M., Zhang, W., and Lin, X. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering 32*, 8 (2019), 1475–1488.

Thank You