Mutual k-Nearest Neighbor Graph for Data Analysis: Application to Metric Space Clustering

Edgar Chavez[1(✉)], Stephane Marchand-Maillet[2], and Adolfo J. Quiroz[3]

[1] CICESE, Ensenada, Mexico
elchavez@cicese.mx
[2] University of Geneva, Geneva, Switzerland
stephane.marchand-maillet@unige.ch
[3] Universidad de los Andes, Bogotá, Colombia
aj.quiroz1079@uniandes.edu.co

Abstract. In this paper, we delve into the Mutual k-Nearest Neighbor Graph (mkNNG) and its significance in clustering and outlier detection. We present a rigorous mathematical framework elucidating its application and highlight its role in the success of various clustering algorithms. Building on Brito et al.'s findings, which link the connected components of the mkNNG to clusters under specific density bounds, we explore its relevance in the context of a wide range of density functions.

Keywords: Mutual k-nearest neighbor · Neighborhood graph · Cluster analysis · Connectivity

1 Introduction

Clustering demands a nuanced understanding of data and a varied toolset tailored to each unique problem. Successfully clustering hinges on formalizing intuitions into clear theorems. We aim to develop a robust mathematical framework for mkNNG clustering use and explain the success of algorithms that use this principle.

Our literature review underscores the recurring emphasis on the mkNN concept, and our study provides insights into its application in clustering.

This paper bridges intuition and mathematics, advancing clustering techniques. We offer a robust exploration of the Mutual k-Nearest Neighbor Graph, laying groundwork for improved clustering and outlier detection. The formalization is covered in Sect. 3.

2 The mkNN Graph in Clustering

Gowda and Krishna were, to our knowledge, the first to propose using the mkNN relationship for agglomerative clustering in [8]. Starting with $k = 1$, fine-grained

O. Pedreira and V. Estivill-Castro (Eds.): SISAP 2023, LNCS 14289, pp. 33–40, 2023.
https://doi.org/10.1007/978-3-031-46994-7_3

clusters are created and merged with increasing k values until only one cluster remains. This was later refined with probabilistic arguments in [5] under a density hypothesis; a detailed discussion on this is deferred to Sect. 3.

The approach in [11] offers a unique mkNN relationship, where each point connects to exactly k-neighbors, proceeding from closest to farther pairs. Once a point connects to its allotted neighbors, it's excluded from further links. Unlike other mkNN methods, distant points could form a mutual kNN relationship. Due to the necessity of computing, storing, and sorting distance pairs, this method is super-quadratic in memory and storage. The clustering approach identifies dense clusters first, claiming to detect clusters of varying densities.

In [2], the authors introduce CMUNE, emphasizing linkage based on mkNN. Dense areas are identified by measuring common points in their k-nearest neighbor neighborhoods. Seed points from these dense regions then connect unclustered points, excluding those in sparse areas. Further refinements are discussed in [1].

In [9], a regressor based on the mkNN relationship is introduced. For a given query x and value of K, it outputs the expected value of values linked to the Mutual K-nearest neighbors of x. If the mkNN result for x is empty, indicating an outlier, the regressor doesn't produce a value.

[17] details the construction of an mkNN graph, drawing parallels with earlier works [5,8]. The aim is cluster detection via the graph's connected components. Spurious connections, possibly arising from noise or outliers, are pruned by weighting the graph's edges based on an affinity measure. Edges below a set affinity threshold are discarded, and clusters are recognized from the graph's residual components.

In [14], a survey on using mkNN for cluster detection is presented. The authors' proposed algorithm hinges on two concepts: extending the mkNN relation to point groups and incorporating density considerations.

[1] unveils DenMune, a clustering algorithm leveraging mkNN to discern centrality and density, as required in the Density Peak-like algorithm [13]. Points are categorized into strong, weak, and noise points based on their reverse kNN count ($\mathcal{R}^{\mathsf{k}}_{\mathcal{X}}(\mathbf{x})$). The DenMune algorithm operates through a voting framework where points garner votes via their membership in the kNN of other points. High-vote recipients are tagged as dense or seed points, while noise points are excluded. Post noise-point removal, the points segregate into dense (seeds) and low-density (non-seeds). Clusters primarily form around seed points, with the low-density points accommodated subsequently. Remaining weak points are then aligned to the most suitable cluster. Remarkably, the authors use mkNN consistently for both density estimation and membership assessment.

A salient feature of the literature review is the dual application of the mkNN test: determining cluster membership and performing three-tier density estimation (dense, weakly-dense, sparse) as illustrated in [1]. While some algorithms appear intricate, they yield compelling experimental results. Our goal is to present a theoretically sound understanding of these heuristics, introducing a streamlined tool for clustering tasks.

3 Data Model

We consider a N-sized D-dimensional dataset $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_\mathsf{N}\}$ from domain $\Omega \subseteq \mathbb{R}^\mathsf{D}$. From the perspective of the continuous domain Ω, a statistical generative model defines the dataset $\mathcal{X}$ as a N-sized i.i.d sample of a probability density function (pdf) $\mathsf{f}_\mathcal{X} : \Omega \to \mathbb{R}^+$ with respect to the Lebesgue measure on Ω. $\{\mathbf{x}_\mathsf{i}\}_{\mathsf{i}=1}^\mathsf{N}$ is then one realization of a set of N independent random variables $\{\mathsf{X}_\mathsf{i}\}_{\mathsf{i}=1}^\mathsf{N}$ identically distributed according to this pdf ($\mathsf{X}_\mathsf{i} \sim \mathsf{f}_\mathcal{X}$, $\mathsf{i} \in [\![\mathsf{N}]\!]$)

Let $\Xi \subseteq \Omega$ be the support of $\mathsf{f}_\mathcal{X}$ defined as the closure of the set where $\mathsf{f}_\mathcal{X}$ is non-zero ($\Xi = \mathsf{supp}(\mathsf{f}_\mathcal{X}) = \overline{\{\mathbf{x} \in \Omega \,\|\, \mathsf{f}_\mathcal{X}(\mathbf{x}) \neq 0\}}$). The fact that Ξ is not connected indicates the presence of localized structures in the domain Ω such as (continuous) *clusters*. (Continuous) clustering is therefore defined as the labeling of connected components of the set Ξ. By (discrete) extension, given $\mathcal{X}$ as a sample of Ξ, clustering is the decision for every pair $(\mathbf{x}_\mathsf{i}, \mathbf{x}_\mathsf{j}) \in \mathcal{X} \times \mathcal{X}$ whether both data belong to the same connected component of Ξ or not (see Definition 3).

Assuming now that (Ω, d) is a metric space equipped with distance function $\mathsf{d} : \Omega \times \Omega \to \mathbb{R}$, then neighborhood systems may be defined over $\mathcal{X}$.

Definition 1 (Mutual *k-nearest neighbor relationship).* *If* $\mathbf{x}_\mathsf{j} \in \mathcal{V}_\mathcal{X}^\mathsf{k}(\mathbf{x}_\mathsf{i})$ *and* $\mathbf{x}_\mathsf{i} \in \mathcal{V}_\mathcal{X}^\mathsf{k}(\mathbf{x}_\mathsf{j})$ *then* $\mathbf{x}_\mathsf{i}$ *and* $\mathbf{x}_\mathsf{j}$ *are said to be* mutual k-nearest neighbors.

Based on the mutual k-nearest neighbor relationship, one can then build the non-directed *mutual* k*-nearest neighbor* (mkNN) graph $\mathsf{G}_\mathsf{k} = (\mathcal{X}, \mathcal{E}_\mathsf{k})$ where data $\mathcal{X}$ serves as nodes and an edge $(\mathbf{x}_\mathsf{i}, \mathbf{x}_\mathsf{j}) \in \mathcal{E}_\mathsf{k}$ exists if $\mathbf{x}_\mathsf{i}$ and $\mathbf{x}_\mathsf{j}$ are mutual k-nearest neighbors. In this paper, we wish to argue for the interest in exploiting the mkNN graph and use data clustering as a natural application domain where using the mkNN is beneficial.

Here, we are particularly interested in the connectivity of G_k given dataset $\mathcal{X}$.

Definition 2 ($k_{D,N}$). *Given a dataset* $\mathcal{X} \subset \Omega$, $\mathsf{k}_{\mathsf{D},\mathsf{N}}$ *is the smallest integer* k *such* G_k *is connected.*

Authors in [5] provide us with Theorem 1 bridging the continuous and discrete domains.

Theorem 1 (rephrased from [5], Thm 2.1).
Let the i.i.d sample $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_\mathsf{N}\} \subset \Omega \subseteq \mathbb{R}^\mathsf{D}$ *come from a distribution* P *with support* $\Xi \subseteq \Omega$ *and density* $\mathsf{f}_\mathcal{X}$. *Assume that* Ξ *is connected and grid compatible, and that for constants* $\mathfrak{a}_1$ *and* $\mathfrak{a}_2$, *we have, on* Ξ, $0 < \mathfrak{a}_1 \leqslant \mathsf{f}_\mathcal{X}(\mathbf{x}) \leqslant \mathfrak{a}_2$. *Then, there exists a constant* c *such that, almost surely,* $\mathsf{k}_{\mathsf{D},\mathsf{N}} \leqslant \mathsf{c} \log \mathsf{N}$, *for large enough* N.

The proof of the theorem is detailed in [5]. Constants $0 < \mathfrak{a}_1 \leqslant \mathfrak{a}_2$ make sure that values of $\mathsf{f}_\mathcal{X}$ have finite non-zero extremes. Similarly, the *grid compatibility* criterion makes sure that Ξ contains a (dominated) connected cover of cubes of finite side length, each with a sufficient intersection with Ξ. Theorem 1 relates

the connectivity of the mutual kNN graph (via $k_{D,N}$) and the continuity (via grid compatibility) of the local density (via constants a_1 and a_2 and "large enough N"). According to this theorem, the mutual kNN graph is connected in parts where the density of the dataset allows to estimate a support that is grid compatible and where this density is bounded away from 0 and Infinity.

That is, the size $k \geqslant k_{D,N}$ to consider for G_k to be connected (and therefore this discrete structure to represent faithfully the connected continuous domain) is bounded by a sub-linear function of N. As the local density at $\mathbf{x}_i$ may be estimated by $\hat{f}_{\mathcal{X}}(\mathbf{x}_i) = \frac{k}{\mathsf{vol}(\mathcal{V}^k_{\mathcal{X}})}$, Theorem 1 states formally that as long as this local estimated density is large enough, G_k remains connected. We will exploit that interpretation to use the mkNN graph to perform data clustering. Since we rely on the formal model given by Theorem 1, we give data clustering a formal and simple definition.

Definition 3 (Data cluster). *Given dataset* $\mathcal{X}$*, data clusters are defined as* connected components *of the mutual nearest neighbor graph* G_k *built on* $\mathcal{S}$*, as the result of the data filtering operation*

Theorem 1 provides guarantees that data clusters defined as above asymptotically represent the connected subsets of the support of the continuous density $f_{\mathcal{X}}$. However, since no parametric model is used for $f_{\mathcal{X}}$, this definition accommodates arbitrary distributions, i.e arbitrary clusters shapes and arrangements.

4 Data Analysis with the Mutual k-Nearest Neighbor Graph

Theorem 1 formally relates the local estimated density of $\mathcal{X}$ to the mkNN graph connectivity. This creates a natural link to clustering operations. Reversing the argument, we state that parts of low estimated density represent potential *fords* thru which spurious (weak) connections in the mkNN graph may appear. Hence we propose to remove these spurious low-density parts matching the "bounded away from 0 and Infinity" continuous counterpart argument. By construction, the resulting data subset guarantees a minimal local density, which by Theorem 1 guarantees mkNN graph connectivity. Starting from dataset $\mathcal{X}$, we propose a generic clustering algorithm below.

Algorithm 1. Data clustering based on mkNN graph

1: **procedure** DataAnalysis($\mathcal{X}$, k)
2: Remove low density parts of the dataset $\mathcal{X} \rightarrow$ subset $\mathcal{S}$ ▷ Data filtering
3: Build G_k as mkNN graph over $\mathcal{S}$ ▷ Structure analysis
4: Connected components of G_k represent the pieces of connected support for $f_{\mathcal{X}}$

Data Filtering. Filtering low-density regions is a staple in the *denoizing* domain. In many contexts, sparse regions within datasets are interpreted as low-likelihood samples, which are typically smoothed over by MSE-based regression models. However, in this study, our primary concern revolves around the density $f_{\mathcal{X}}$, especially under the parameters delineated in Theorem 1 and subsequent discussions.

From this perspective, the objective becomes highlighting areas where the estimated density not only reaches a minimal threshold (a_1) but also stands as statistically reliable, backed by adequate local samples. Notably, due to the intrinsic property $\int_{\Omega} f_{\mathcal{X}} = 1$, identifying "low-density" zones becomes a comparative exercise, relative to the entire dataset's distribution. Such an approach closely aligns with the objective of outlier detection; an outlier's definition invariably hinges on its relation to the surrounding data [4,10].

Given this, our preliminary strategy entails employing outlier detection for data filtration, sidestepping traditional methods like KDE [15] which tends to falter in sparsely populated datasets. It's worth noting that contemporary density estimation techniques, such as generative normalizing flows [12], might find applicability in this scenario.

From the input dataset $\mathcal{X}$, the filtration process yields $\mathcal{S}$. This step intriguingly presents a dual relationship with its successor. Even though filtering frequently mandates a predefined k value, techniques spanning outlier detection to density estimation can potentially help pinpoint an optimal k.

Complexity: The construction of a compressed cover tree to solve the kNN exhibits a time complexity of $O(c_m(\mathcal{X})^8 c((\mathcal{X})^2 N \log(N))$, where c_m and c represent expansion constants pertinent to the dataset. When tasked with deriving the k nearest neighbor table for a novel point, q, the time complexity is set at $O(c(\mathcal{X} \cup q)^2 \log(k)[c_m(\mathcal{X})^{10} \log N + c(\mathcal{X} \cup q)k])$. This infers that formulating the k nearest neighbor table remains a sub-quadratic challenge. The detailed analysis can be found in the recent PhD thesis [6]. With the $N \times k$ k-nearest neighbor table established, the determination of whether the k-nearest neighbor relationship is mutual or otherwise is achievable through a single table traversal, bearing a complexity of $O(Nk^2)$. Hence, the derivation of the mutual k-nearest neighbor graph retains a subquadratic nature.

5 Illustrative Experiments

5.1 k-Means Convex Model

To illustrate visually the ability of our toolset to naturally handle clusters of arbitrary shapes, we visualize the process with the 2D "Two Moons" dataset generated by `sklearn` ($N = 10'000$, $\sigma_{\text{noise}} = 0.1$). Due to its convex Gaussian model, k-means cannot separate this data into 2 clusters (Fig. 1[top left]) whereas our procedure (e.g with $k = 10$ and $\tau_{\text{LOF}} = 1.1$, Fig. 1 [others]) robustly splits the dataset.

One can easily picture the underlying dendrogram from the evolution of the partition. At $\mathtt{k} = 1000$, connectivity overrules low density regions and the mkNN then shows a unique connected component, leading to a high FP rate.

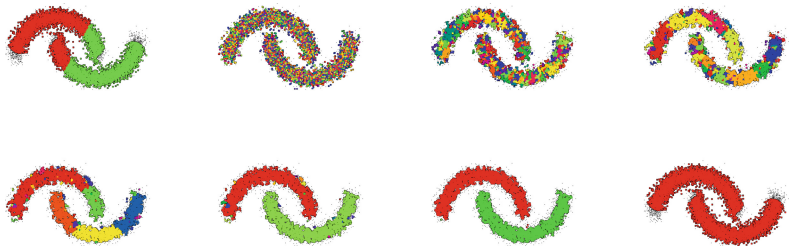

Fig. 1. Two Moons partitions. [top left] k-means partition. [others] our procedure fixing $\tau_{\text{LOF}} = 1.1$ and evolving $\mathtt{k} \in \{1, 4, 5, 6, 7, 10, 1000\}$ in reading order

Figure 2 pinpoints the fact that this behavior is prevented by the initial filtering. If we remove this step, a large connected component appears in the mkNN graph from $\mathtt{k} = 4$.

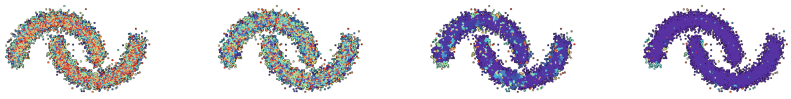

Fig. 2. Two Moons partitions without prior filtering and evolving $\mathtt{k} \in \{1, 2, 4, 5\}$ in reading order

5.2 Density-Based Clustering Model

Now, one strong bias brought by the `sklearn` "Two Moons" dataset is that the two clusters are of approximate equal densities. When modifying the generation to obtain contrasted densities, one falls in a setup known to be adverse to the density-based clustering algorithms, of which DBSCAN [7] is a major representative.

As shown in Fig. 3, k-means fails again to identify proper clusters due to its convex model. DBSCAN appears very difficult to tune for such data since the high density pushes towards a small `Eps` radius which creates an empty ball, not passing the `MinPts` threshold in low density regions. In contrast, such a setup is accepted by our mkNN procedure with the same parameters as before.

Fig. 3. Adapted Two Moons dataset with varying densities. [left] k-means partition. [center] DBSCAN partition. [right] mkNN-based partition

Discussion: The filtering step, to be useful, should retain most data from the original dataset. Finding the proper filtering value is a problem in itself, it should be related to *fords* detection, as discussed earlier. There are many hidden parameters in the clustering procedure we outline in Algorithm 4, derived from the theorems. This dependency goes also to the bounds of density function and the α value in the grid-compatibility constant in Theorem 1. Rather than giving a clustering algorithm, we have discussed the grounding of successful clustering strategies, proposing tools to build a dedicated connectivity-based clustering algorithm for datasets adhering to precise hypothesis.

We were able to replicate clustering experiments, as reported in [1] with a simpler procedure more amenable to analysis. Also following the conclusions of the cited research, the proposed tools are useful in low-dimensional datasets. They propose to use non-linear embeddings to two dimensions, using for example t-SNE [16]. As analyzed in [3], handling high dimensional datasets is riddle with problems derived from the phenomenon of concentration of measure which generates the so-called curse of dimensionality, one of its many forms is the phenomenon of hubness and the increase of the in-degree in metric graphs, with consequences in the connectivity of the mkNNG.

Further and maybe most importantly, no vector computation is required. Hence, this toolset may operate in metric spaces that are not necessarily vector spaces and where the data is given by similarity, known to respect the metric conditions. Again, this is to be contrasted with the popular k-means algorithm operating in vector spaces, as illustrated in Fig. 1.

Acknowledgments. This work is partly funded by the Swiss National Science Foundation under grant number 207509 "Structural Intrinsic Dimensionality".

References

1. Abbas, M., El-Zoghabi, A., Shoukry, A.: DenMune: density peak based clustering using mutual nearest neighbors. Pattern Recogn. **109**, 107589 (2021)
2. Abbas, M.A., Shoukry, A.A.: CMUNE: a clustering using mutual nearest neighbors algorithm. In: 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), pp. 1192–1197 (2012)
3. Angiulli, F.: On the behavior of intrinsically high-dimensional spaces: distances, direct and reverse nearest neighbors, and hubness. J. Mach. Learn. Res. **18**(170), 1–60 (2018)

4. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. SIGMOD Rec. **29**(2), 93–104 (2000)
5. Brito, M., Chávez, E., Quiroz, A., Yukich, J.: Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. Stat. Probabil. Lett. **35**(1), 33–42 (1997)
6. Elkin, Y.: A new compressed cover tree for k-nearest neighbour search and the stable-under-noise mergegram of a point cloud. The University of Liverpool, United Kingdom (2022)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD), pp. 226–231 (1996)
8. Gowda, K.C., Krishna, G.: Agglomerative clustering using the concept of mutual nearest neighbourhood. Pattern Recogn. **10**(2), 105–112 (1978)
9. Guyader, A., Hengartner, N.: On the mutual nearest neighbors estimate in regression. J. Mach. Learn. Res. **14**(37), 2361–2376 (2013)
10. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**, 2004 (2004)
11. Hu, Z., Bhatnagar, R.: Clustering algorithm based on mutual k-nearest neighbor relationships. Stat. Anal. Data Min. ASA Data Sci. J. **5**(2), 100–113 (2012)
12. Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. CoRR arXiv:1505.05770 (2016)
13. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
14. Ros, F., Guillaume, S.: Munec: a mutual neighbor-based clustering algorithm. Inf. Sci. **486**, 148–170 (2019)
15. Terrell, G.R., Scott, D.W.: Variable kernel density estimation. Ann. Stat. **20**(3), 1236–1265 (1992)
16. Van Der Maaten, L.: Accelerating t-SNE using tree-based algorithms. J. Mach. Learn. Res. **15**(1), 3221–3245 (2014)
17. Zhang, H., Kiranyaz, S., Gabbouj, M.: Data clustering based on community structure in mutual k-nearest neighbor graph. In: 41st International Conference on Telecommunications and Signal Processing (TSP), pp. 1–7 (2018)