

Delaunay simplices pruning based clustering

Octavio Razafindramanana and Gilles Venturini

Université François Rabelais de Tours - Laboratoire d'Informatique
64, avenue Jean Portalis - 37200 Tours, France
Email: { octavio.razafindramanana, gilles.venturini } @univ-tours.fr

Abstract. In this paper, we introduce a new clustering method using the Delaunay triangulation of a set of points as an input. The proposed method is based on pruning extra simplices of a triangulation according to a local heterogeneity measure, which we introduce here. This measure produces good clustering results as it yields to better inter-cluster simplices detection. The efficiency of the measure is evaluated on 2-D shape data set.

1 Introduction

Combination of interaction and visualization proposes a trustworthy paradigm for knowledge discovery. This combination allows both to grasp visually the conceptual meaning of the data and to interact with its user-dependent *interesting* parts. Therefore, to allow user to have a visual insight of this possibly existing conceptual information, the visualization has to respect the following two properties: being visually understandable and being *data-topology* preserving.

Our very basic objective is to construct qualitative information out of quantitative information by the means of analyzing distances and studying existence of connected components [1]. This means here constructing information about the way data points are related to each other in order to detect clusters of points. Graphs are a suitable tool to preserve the so-called *topology* as they may carry both local (neighbors) and high-level (clusters) information that exists among data points. Furthermore, they are well-suited to visualization. In particular, *proximity graphs*, also called *neighborhood graphs* [2], are a family of graphs for which closeness between pairs of vertices is symbolized by edges. Depending on how one might define *closeness*, some of the notable proximity graphs are *Delaunay*, *RNG*, *Gabriel* or *MST* ones [2]. In this paper, to embed this proximity information, we make use of the Delaunay triangulation (DT) as it is the dual of a structure which explicitly represents the topology among the data points: the Voronoi diagram.

The reasoning that follows the proposed method to cluster a given point cloud is straightforward. First, the topological relationships between input data points is computed and, secondly, this information is refined by detecting clusters. The proposed method constructs first the DT of the points and then computes local information to prune the possibly existing inter-cluster simplices, eventually revealing the intra-cluster ones. The underlying problem addressed throughout this paper is therefore revealing the possibly existing clusters among a set of points from their DT, by the means of pruning simplices.

2 Problem Statement and Related Work

2.1 Proximity graphs

Let us define here the proximity graph that describes a data points set $S \subset \mathbb{R}^d$. Let $G = (V, E)$ be an unoriented graph such that V and E are its vertices and edges sets respectively. A proximity graph of S is a graph G for which $V = S$ and $E = \{(p, p') \in S^2, close(p, p')\}$, with $close$ an indicator function which equals 1 if p is close to p' , 0 otherwise. Thus, the Delaunay graph of S , $DG(S)$, is a graph for which $E = \{(x_i, x_j) \in S^2, V_i \cap V_j \neq \emptyset\}$ with V_i and V_j the Voronoi cells of points x_i and x_j and $i, j \in [1..n]$. An edge is thus defined in case of existence of an adjacent Voronoi face between the Voronoi cells of two points of S .

2.2 Clustering using Delaunay diagram

Referred to as *graph theory-based* clustering methods [3], the algorithms that cluster data points by the means of proximity graphs construction are composed of two steps: constructing first the graph and then pruning irrelevant edges. A typical example is the well-known Zahn's clustering algorithm that seeks connected components as clusters by detecting and deleting inconsistent edges in the minimum spanning tree [4]. Introduced in the context of hierarchical clustering, AMOEBA algorithm leverages the density-preserving property of the DT of a set of points [5]. Indeed, it introduces a both local and global edge-removal criterion based on the edges length distribution: if the length of a given edge exceeds the global mean value plus a local tolerance value then it is pruned. It provides good results and detects nested clusters. AUTOCLUST algorithm [6] is based on both local and global edge-removal information as well. It detects points whose incident edges lengths have an unusually large standard deviation and eventually leads to deleting significantly long (and short) edges [6]. Algorithms such that in [7] or [8] also proceed to long edges deletion but only with a global criterion.

However, these algorithms do not take into account the topological information carried by the simplices themselves. To leverage the simplices emptiness information, the algorithm proposed in [9] derives the number of clusters from classifying triangles (2-simplices) as ones with high perimeter value and ones with low perimeter value, with respect to their distribution among the triangles. It then takes into account the fact that the constructed triangles do not contain any other point (as a consequence of being Delaunay ones) and therefore assumes that large triangles are prone to be inter-cluster ones. Following this reasoning, we introduce a new heterogeneity measure that provides a more precise indicator of being an inter-cluster simplex.

3 Clustering Approach

3.1 Delaunay simplices and heterogeneity measure

Our objective here is to evaluate the propensity of the measure we introduce to create an ordering over the constructed simplices: useless (inter-cluster) ones are expected to have significant higher values. In a given DT, three types of simplices can be enumerated: the inter-cluster, the intra-cluster and the noisy-points-involving ones. Inter-cluster simplices and those involving noisy points have significant larger perimeter: as clusters are denser parts of the data clouds, simplices composing them have lower perimeter. Indeed, following a *density-based* definition of clusters, typical edges linking two points of different clusters are longer as compared to intra-cluster ones. Moreover, typical inter-cluster simplices are composed of one face lying on the hull of one cluster and one $(d + 1)$ -th point lying on the hull of another cluster. The edges linking this last point and the d other ones of the simplex are then significantly longer than those on the hull of the other cluster. However, these remarks hold in case of clusters of same density. In case of clusters of different densities, the assumption of longer inter-cluster simplices may not hold, as clusters of lower density are composed of longer simplices and therefore may have the same perimeter as some inter-cluster ones. In this configuration, perimeter may fail to discriminate intra from inter-cluster simplices.

In order to enrich the perimeter indicator, we define an indicator value of a simplex to be lying between two clusters as the ratio of its longest and shortest radii. The higher this value is, the more the simplex is prone to be inter-cluster. Thus, this ratio may allow to differentiate simplices having equal perimeters by pruning first the ones presenting a higher value and therefore higher *heterogeneity*. The composite measure that we introduce here takes into account this ratio as along with the perimeter. Analogously to [6], we include as well the local information of the standard deviation of the heterogeneity among the adjacent simplices: the higher this value, the more the simplex is prone to be an inter-cluster one as the more the adjacent simplices are prone to belong to different clusters.

Thus, the introduced measure: 1) leverages the empty-sphere information using the perimeter, 2) evaluates the *heterogeneity* of one simplex (having at least one significantly larger edge than the other ones), as a large perimeter does not necessarily involves points of different clusters, 3) leverages the local information of a simplex by evaluating the propensity of the adjacent simplices to belong to the same cluster.

3.2 Clustering measure and algorithm

Given a simplex s belonging to the DT of S , its edges and adjacent simplices sets, E_s and $N(s)$, we define:

$$\mu_1(s) = \sum_{e \in E_s} |e| \quad (1)$$

$$\mu_2(s) = \frac{\max E_s}{\min E_s} \quad (2)$$

$$\mu_3(s) = \mu_1(s) * \mu_2(s) * LSD(\mu_2, s) \quad (3)$$

with, analogously to [6], $LSD(\mu, s)$ the local standard deviation of measure μ over set $s \cup N(s)$. Let μ_1 be the perimeter value of s (as experimented in [9]), μ_2 the heterogeneity indicator and μ_3 the composite measure we finally propose.

The measure value $\mu_3(s)$ takes into account local neighboring information and takes a higher value the more s is: 1) large, 2) composed of edges of heterogeneous length, 3) surrounded by simplices of different homogeneity. As mentioned earlier, weighting μ_1 by μ_2 allows to differentiate between two simplices having the same perimeter value by setting a higher value to the one presenting edges of heterogeneous length. As this remark may not hold in case of two clusters having different densities (the lower density intra-cluster simplex may be set to a higher value and thus be pruned away first), we weight $\mu = \mu_1 * \mu_2$ by $LSD(\mu_2, s)$ to leverage the local heterogeneity information around s . This weighting eventually put forward for deletion inter-cluster simplices and manages to differentiate between simplices with the same $\mu_1 * \mu_2$ value as it exhibits the ones with higher LSD values.

4 Experimental results

The proposed clustering algorithm is composed of the following three steps: 1) construction of $DT(S)$, 2) sorting of the simplices in increasing order with respect to the measure value, 3) pruning of the simplices presenting higher values. The method is tested on the well-known Zahn's compound data set [4] as it notably presents clusters of different densities and nested clusters. The data set is composed of 399 2-dimensional points defining 6 clusters. The cluster labels are indicated in Figure 1-(a). The original triangulation is composed of 399 points, 777 triangles and 1175 edges. Presented experiments rely on our implementation of the d -dimensional algorithm formulated in [10].

4.1 Analysis of the simplices deletion order

Let us study the ordering provided by measure $\mu_1 * \mu_2$. Figure 1-(b) illustrates that the first two exhibited connected components consist of the clusters $C_1 = \{1, 2\}$ and $C_2 = \{3, 4, 5, 6\}$. The first inter-cluster simplices that are pruned are the ones between C_1 and C_2 as they present the largest perimeters. Figure 1-(c) illustrates the exhibition of clusters $C_3 = \{3, 4\}$, $C_4 = \{5\}$ and nested cluster $C_5 = \{6\}$. The corresponding pruned simplices present high *heterogeneity*. Then, Figure 1-(d) reveals nested cluster $C_6 = \{2\}$. At this stage of pruning, we note that clusters 1 and C_3 are severely damaged. In particular, to reveal cluster 2, most of the edges of cluster 1 have been deleted as an expected consequence of: 1) weighting the measure by the perimeter and 2) cluster 2 being of lower density. We mention that, using measure μ_1 , μ_2 and $\mu_1 * \mu_2$, the presented experiments did not lead to actual exhibition of cluster 2 without severely

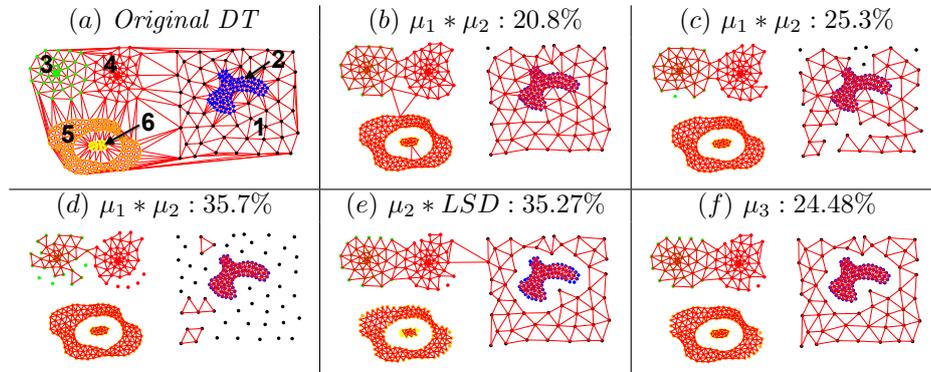


Fig. 1: **Measures and pruning.** (a) Original DT. (b)-(d) Impact of the pruning on the clusters exhibition for $\mu_1 * \mu_2$. (e)-(f) Nested cluster 2 exhibition for measure $\mu_2 * LSD$ and μ_3 . The associated proportion of pruned simplices is displayed.

deteriorating cluster 1. As illustrated in Figures 1-(e) and (f), $\mu_2 * LSD$ and μ_3 address this problem as they clearly reveal the inner hull of cluster 1. Besides, μ_3 prunes inter-cluster simplices for significant lower value and therefore better preserves the clusters quality, as showed in the following section.

4.2 Comparison of measures

Table 1 shows the influence of the pruning measure on the clustering quality. For one given cluster C , we evaluate indices $I_1(C) = 1 - \frac{|E_{inter}(C)|}{|E_{intra}(C)|}$ and $I_2(C) = 1 - \frac{|E_{inter}(C)|}{|E_{inter}^{ref}(C)|}$, with $E_{intra}(C)$ (resp. E_{inter}) the inner (resp. outward) edges of cluster C . $E_{inter}^{ref}(C)$ are the inter-cluster edges of C in the original triangulation. I_1 measures the propensity to reveal clusters without deteriorating them and I_2 the propensity to prune inter-cluster simplices. Both have to be maximized. Table 1 compares the effect of each presented measures on the clustering quality: simplices are sorted in increasing order with respect to the measure value and those with higher-values are pruned, yielding then values I_1 and I_2 . For demonstration, we arbitrarily pruned 30% of higher-valued simplices.

First, we note that μ_3 exhibits higher values for both I_1 and I_2 . For the given pruning value, μ_3 deteriorates less the clusters and put forward for deletion inter-cluster simplices efficiently. Compared to the other measures, μ_3 achieves better results as it yields the highest cumulated values for I_1 and I_2 . Compared to $\mu_2 * LSD$, which also detects cluster 2, μ_3 prunes simplices between clusters 1 and 2 for a lower pruning value as a consequence of leveraging the association of the perimeter and local heterogeneity information.

$I_1(I_2)$	no prun.	μ_1 [9]	μ_2	$\mu_2 * LSD$	$\mu_1 * \mu_2$	μ_3
1	0.34(0)	0(0.82)	0.66(0.55)	0.84(0.78)	0.83(0.90)	1 (1)
2	0.78(0)	0.94(0.74)	0.86(0.42)	0.93(0.72)	0.97(0.85)	1 (1)
3	0.71(0)	0.97(0.93)	0.89(0.66)	0.90(0.69)	0.93(0.79)	0.91(0.76)
4	0.68(0)	0.98(0.95)	0.91(0.73)	0.90(0.70)	0.95(0.84)	0.93(0.81)
5	0.81(0)	0.96(0.78)	0.98(0.90)	1 (1)	1 (1)	1 (1)
6	-0.06(0)	0.51(0.54)	1(1)	1 (1)	1 (1)	1 (1)
Sum	3.26(0)	4.37(4.76)	5.31(4.25)	5.58(4.89)	5.67(5.38)	5.84(5.57)

Table 1: **Comparison of the measures.** I_1 and I_2 values achieved for the 6 clusters after the pruning of 30% of the simplices (and with no pruning).

5 Conclusion

As shown in the experiments, the introduced method and measure provide promising results for data clustering as it creates an ordering which tends to exhibit efficiently inter-cluster simplices and notably performs better than perimeter measure. Moreover, the introduced measure is suitable for hierarchical clustering. Future works will include testing our measure on an AMOEBA-like hierarchical algorithm. Forthcoming works will focus on the extension to higher dimensions.

References

- [1] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, 2009.
- [2] Jerzy W. Jaromczyk and Godfried T. Toussaint. Relative neighborhood graphs and their relatives. In *Proceedings of the IEEE*, pages 1502–1517, 1992.
- [3] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–78, May 2005.
- [4] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 100(1):68–86, 1971.
- [5] V Estivill-Castro and I Lee. AMOEBA: Hierarchical clustering based on spatial proximity using Delaunay diagram. In *Proceedings of the 9th International Symposium on Spatial Data Handling*, pages 1–16, 2000.
- [6] V Estivill-Castro and I Lee. AUTOCLUST: Automatic clustering via boundary extraction for mining massive point-data sets. In *Proceedings of the 5th International Conference on Geocomputation*, 2000.
- [7] In-Soo Kang, Tae wan Kim, and Ki-Joune Li. A spatial data mining method by delaunay triangulation. In *Proceedings of the 5th International Workshop on Advances in Geographic Information Systems*, pages 35–39. ACM, 1997.
- [8] C. Eldershaw and M. Hegland. Cluster analysis using triangulation. *Computational Techniques and Applications: CTAC97*, pages 201–208, 1997.
- [9] V Estivill-Castro and M E Houle. Robust Clustering of Large Geo-referenced Data Sets. In *Methodologies for Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 327–338. 1999.
- [10] P Cignoni. DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed. *Computer-Aided Design*, 30(5):333–341, April 1998.