MustaCHE: A Multiple Clustering Hierarchies Explorer

Antonio Cavalcante Araujo Neto* Mario A. Nascimento*
Joerg Sander* Ricardo J. G. B. Campello**
*University of Alberta, Canada **James Cook University, Australia
{antonio.cavalcante, mario.nascimento, jsander}@ualberta.ca
ricardo.campello@jcu.edu.au

ABSTRACT

In this demonstration paper we introduce MustaCHE (*Multiple Clustering Hierarchies Explorer*), a tool that allows analysis and exploration of multiple clustering hierarchies in an interactive and visual manner. A known issue in the context of density-based clustering is how to set parameters. Typically one has to resort to trial-and-error, and its potential pitfalls, which may possibly include not finding existing clusters at all. In a previous work we have devised a very efficient technique to generate clustering hierarchies using HDBSCAN* *w.r.t.* a range of its clustering parameter, *mpts*. However, finding the "best" *mpts* value is still an open problem. In order to mitigate this issue we developed MustaCHE, a tool that allows a user to identify and visualize several different density-based cluster hierarchies of a dataset *w.r.t.* a large range of *mpts* values. The user can then explore hierarchies individually and, at the same time, see how they compare to the other hierarchies. The simultaneous visualization of multiple clustering hierarchies provided by MustaCHE makes it feasible (and easy) for a user to gain a deeper understanding of the data and how its cluster structures behave under different parameter settings.

PVLDB Reference Format:
Antonio Cavalcante Araujo Neto, Mario A. Nascimento, Joerg Sander and Ricardo J. G. B. Campello. MustaCHE: A Multiple Clustering Hierarchies Explorer. *PVLDB*, 11 (5): xxxx-yyyy, 2018.
DOI: https://doi.org/TBD

1. INTRODUCTION

Clustering algorithms are widely employed to find groups in datasets such that elements in the same *cluster* are more related to each other than to elements in other *clusters*. Among many different approaches, density-based clustering algorithms stand out for their ability to identify arbitrarily-shaped clusters and differentiate cluster elements from noise. For instance, DBSCAN [4], one of the pioneers and better known density-based clustering algorithms, is able to compute a data partitioning consisting of dense regions of points separated by low-density regions. However, DBSCAN requires the setting of two parameters, ε and *mpts* (a point p is considered dense whenever there are at least *mpts* other points in p's ε-neighborhood).

Proceedings of the VLDB Endowment, Vol. 11, No. 5

DOI: https://doi.org/TBD

In [3], the authors presented HDBSCAN*, the state-of-the-art density-based hierarchical clustering method, which produces a hierarchical organization of clusters in a dataset given *a single* parameter: *mpts*. While the performance of HDBSCAN* is robust *w.r.t. mpts*, in the sense that a small change in *mpts* typically leads to only a small or no change in the clustering structure, choosing a "good" *mpts* value can be challenging. In fact, certain data clusters may reveal themselves for different ranges of *mpts*, and without any means to choose a good value it is not hard for one to miss a valid hierarchical organization of the data. Normally, a user would resort to a trial-and-error approach and would explore several scenarios (*mpts* values) to ensure the selection of a value that is appropriate for the dataset or the application. Unfortunately, the exploration of large ranges of *mpts* values is not practically feasible due to the computational costs associated with the approach of running HDBSCAN* multiple times (once for each value of *mpts* in a given range).

In [6], the authors proposed RNG-HDBSCAN*, a strategy that is able to compute a set of HDBSCAN* hierarchies for a range of *mpts* values very efficiently. The replacement of HDBSCAN*'s complete graph with a much smaller graph makes it viable to compute several hierarchies with the computational cost equivalent to running HDBSCAN* just a few times. For instance, it has been shown in [6] that RNG-HDBSCAN* can generate the hierarchical organizations for a range of more than 100 values of *mpts* at the same time it would take the original HDBSCAN* algorithm to generate those for only 2-3 values of *mpts*. This allows one to potentially explore and analyze a wide range of *mpts* values.

However, analyzing a very large number of clustering hierarchies, and *learning* from them, is still practically challenging. While close values of *mpts* are likely to result in similar hierarchies, different *ranges* of *mpts* values may produce *significantly* different hierarchies. Therefore, we address in this work the following non-trivial questions: for a given dataset, (1) *how many of these ranges exist?*, (2) *how does one identify these ranges?* and (3) *how do the hierarchies in each of these ranges look like?*.

To address these questions, we propose MustaCHE, a tool that leverages the main results in [6] and allows a visual and interactive analysis and exploration of multiple clustering

hierarchies, thus helping users to better understand their data and its cluster structures.

Next we present the different visualizations available in MustaCHE and discuss what a user can learn from each of them, followed by a description of a demonstration scenario that illustrates MustaCHE's usability.

2. MustaCHE

Given a set of HDBSCAN* hierarchies of a given dataset for different *mpts* values, efficiently pre-computed using [6], MustaCHE offers a set of visualizations that simplify and aid in the analysis of those hierarchies. Its main overall goals are to assist the user to (1) (visually) find "good" values for *mpts* and (2) to understand which cluster structures are present *w.r.t.* different density parameters in the data. In this following, we discuss the motivations behind each of the visualizations available in MustaCHE.

2.1 Similarity Matrix

A common representation for a cluster hierarchy, which depends on a given value of *mpts*, is a dendrogram. However, in order to gain a broad understanding of how the parameter *mpts* affects the hierarchical organization of the data it is not necessary to look at actual dendrograms; but rather to identify the ranges of *mpts* values that produce similar hierarchies. In order to find those, one needs a way to measure similarity between hierarchies, which can be done, *e.g.*, using the Hierarchy Agreement Index (HAI) [5].

After computing the HAI values for every pair of hierarchies, one is able to represent the similarities in a symmetric matrix where a row index i and a column index j represent $mpts_i$ and $mpts_j$, respectively, from the given range of mpts values. A cell (i, j) contains the HAI value for the pair of hierarchies with respect to $mpts_i$ and $mpts_j$, respectively.

Plotting these values in a color scale, where lighter colors indicate a higher similarity, makes it possible to visually identify the *mpts* values that result in similar hierarchies. For instance, Figure 1 shows the pairwise HAI values for 50 hierarchies from a sample dataset *w.r.t.* $mpts \in [1, 50]$ Note that any two hierarchies *w.r.t.* $mpts \in [1, 15]$ have a high degree of similarity among themselves. Likewise, any two hierarchies *w.r.t.* $mpts \in [16, 50]$ are also similar among themselves, but to a lesser degree than in the previous case. Furthermore one can observe that within both of these ranges of *mpts* values, there are smaller sub-ranges for which the similarity is higher than in the larger one. Finally, any two hierarchies with *mpts* outside those those two ranges are dissimilar.

2.2 Meta-Clustering Dendrogram

While the similarity matrix plot presents an overview of the similarity among hierarchies for a range of *mpts* values, extracting the exact ranges that produce similar hierarchies only from this plot can still be difficult. For example, in cases where changing the value of *mpts* leads to a smooth decrease or increase of similarity, it is hard to draw boundaries that separate two ranges of values just by looking at the plot. Also, there might be cases where two hierarchies for non-consecutive values of *mpts* have a higher similarity than for consecutive values. In order to deal with such cases, we want to cluster these hierarchies while at the same time allowing the user to set the similarity threshold needed to consider two hierarchies as part of the same group.

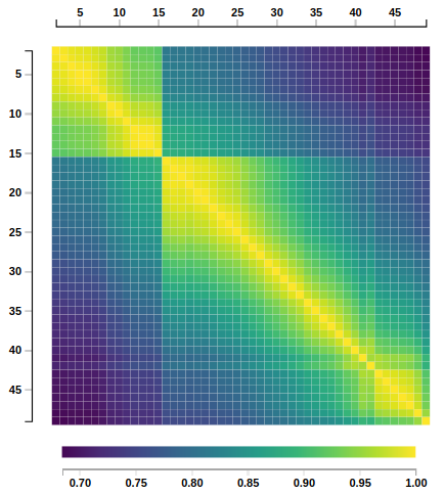

Figure 1: Pairwise HAI Similarity

To combine these two requirements, we do a meta-clustering process using the HAI values to construct a *clustering hierarchy of clustering hierarchies* with HDBSCAN*.[1] This meta-hierarchy can be visualized as a dendrogram, where the user can see similar hierarchies next to each other and is also able to distinguish similarity levels more clearly.

Figure 2 shows the dendrogram of the 50 hierarchies computed from the HAI values presented in the example in Figure 1. Note that it highlights five main (meta) clusters selected by the automatic extraction method (FOSC) [2] provided by HDBSCAN*, which is based on the stability of each cluster (we will discuss other cluster extraction methods also supported by MustaCHE in the next section). At this point, instead of having to examine 50 *"different"* hierarchies, the user knows that there are 5 main different hierarchical organizations of the data. Notice that there are also outliers (plotted in light gray) that might represent interesting results as well, since they are unique in the sense that they haven't been included into any of the found clusters. In Section 3 we discuss how these outliers can be interactively explored and how the user can select different partitionings from meta-hierarchies.

2.3 Reachability Plots

At this point, after perusing the visualizations previously described, the user has a better understanding of how the parameter *mpts* affects the similarity of the hierarchies and should be able to identify the ranges of values that produce *"significantly"* different hierarchies. Furthermore, deciding what is significant becomes more intuitive and more practical with the use of dendrograms. The next step now is to examine how the hierarchies from each meta-cluster look like. Since the hierarchies in each meta-cluster are similar to each other, there is no need to examine all of them. In this kind of visualization we select only the *medoid* hierarchy from each meta-cluster as its representative.

[1]Note that (1) as the HAI values express similarity between hierarchies, one has to convert them into dissimilarity before using them with HDBSCAN* , and (2) we use $mpts = 1$ to cluster the cluster hierarchies, which is equivalent to using Single Linkage clustering to cluster the cluster hierarchies.

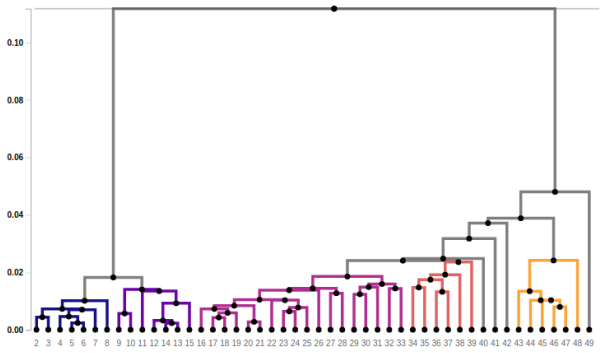

Figure 2: Hierarchy of hierarchies (meta-clustering)

Even though we choose to visualize our meta-hierarchy as a dendrogram, dendrograms are only suitable for visualization when the number of leaf nodes (*i.e.* elements being clustered) is "relatively small" to be displayed hierarchically as a tree. Fortunately, the number of leaf nodes in the meta-hierarchy corresponds to the number of clustering hierarchies (or, alternatively, the number of *mpts* values) under analysis, which in practice is rarely larger than a few tens in common real datasets. However, when we want to inspect individual clustering hierarchies, such as the meta-cluster medoids or outliers in the meta-hierarchy, the number of leaf nodes corresponds to the number of data points (*i.e.*, the size of the dataset), which may be too large to be properly visualized as a dendrogram.

When the number of points is large, the use of *reachability plots* [1] is more appropriate to visualize density-based clustering hierarchies. In a nutshell, reachability plots are bar plots where each bar corresponds to a point in the dataset, and they are sorted in such a way that points that belong to the same cluster at every density level are next to each other. The height of each bar is defined by the lowest density level that makes its corresponding point join the preceding points in the plot, so density-based clusters appear as "valleys" (or "dents") in the plot. Figure 3 displays five reachability plots, each of which corresponds to the medoid of one of the five meta-clusters found in Figure 2. For instance, the reachability plot for $mpts = 5$ indicates that the dataset might be decomposed into six main clusters, whereas the hierarchy for $mpts = 37$ shows only two main clusters.

The combined visualization of the reachability plots for the meta-cluster medoids allows the user to easily compare the different hierarchical organizations of the data across multiple *mpts* values. MustaCHE also allows the investigation of reachability plots in more details, by coloring the selected plot according to the FOSC (HDBSCAN* automatic cluster extraction) partitioning for that corresponding hierarchy. This type of visualization is illustrated in Figure 4, which shows the reachability plot for $mpts = 5$ with the clustering represented by different colors, and black representing noise. This visualization shows which points belong to which clusters in the partitioning performed by FOSC/HDBSCAN*, and which points are noise. We note, this visualization can be produced for any value of *mpts* selected by the user, i.e., not necessarily only the ones corresponding to meta-clusters (Figure 3), but also other elements of the meta-hierarchy (Figure 2), including outliers.

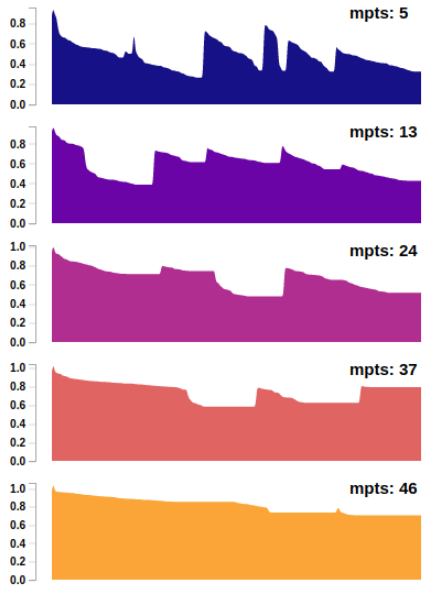

Figure 3: Reachability Plots

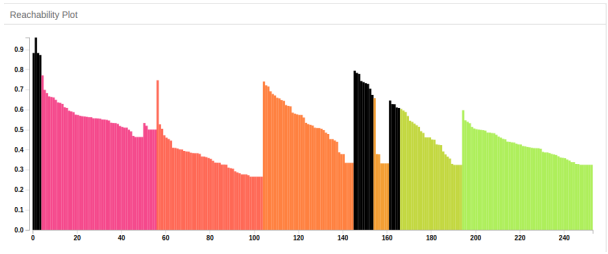

Figure 4: Reachability Plot for $mpts = 5$

At this stage, the user can see which points compose each of the clusters to understand the cluster structure in greater detail.

3. DEMO OVERVIEW

In this section, we describe how users can interact with and learn from MustaCHE. The first user interaction with MustaCHE is the selection of the dataset to be analyzed, and the specification of the range of *mpts* values to be considered in the analysis (Figure 5a). For this demonstration, users will have access to several datasets already preloaded into MustaCHE. (Even though we are able to compute hierarchies for an entire range of *mpts* values efficiently, MustaCHE's requires off-line preprocessing time for larger datasets before the (on-line) interactive visualizations can be performed.) After that, some metadata about the investigated dataset is displayed to the user (Figure 5b), and the plots described in Section 2 can be visualized.

The first visualization is the HAI matrix (Figure 5c). When positioning the cursor over a cell of the HAI matrix plot, MustaCHE shows which *mpts* values correspond to that cell and their similarity as computed by the HAI.

At this stage, the user may choose how the meta-clusters will be extracted from the dendrogram (Figure 5d). MustaCHE provides three main options. The first and default

Figure 5: MustaCHE

method consists of applying HDBSCAN*'s automatic cluster extraction method, namely FOSC as described earlier. In the second method, the meta-clusters are selected through a horizontal cut in the dendrogram. The interaction will occur through a slider that will move the threshold bar up and down along the y axis. The clusters selected with this method will be the ones that will appear below the threshold bar. Note that in Figure 5d, the threshold bar method is selected, and the bar is setting the threshold as 0.03, which results in three meta-clusters. The third and last method allows users to manually select the meta-clusters in the dendrogram, *i.e.*, they will be able to click the meta-clusters they want to investigate. This feature gives users flexibility to try different arbitrary selections. MustaCHE automatically reacts to the changes in the meta-clusters selection and updates the reachability plots accordingly (Figure 5e). One special aspect to note are the hierarchies that are labeled as outliers in the meta-cluster extraction. In fact, in order to inspect individual hierarchies, including outliers, in more details, users can select them either from the medoids (Figure 5e) or from the dendrogram (Figure 5d), and MustaCHE will show the detailed reachability plot (Figure 5f) colored according to its cluster partitioning (as extracted by FOSC).

We note that the "steps" above are not necessarily taken in such a linear fashion as discussed above. MustaCHEallows a (parameter-free) visual *and*, interactive exploration of the dataset hierarchical clustering.

4. CONCLUSION

We have presented MustaCHE, a visualization tool that allows the analysis of multiple HDBSCAN* density-based clustering hierarchies in a visual and interactive way. The use of MustaCHE makes it easier for a user to have a deeper understanding of how the cluster structures in the data behave under different density levels. A next step is to deploy this tool within an end-to-end web-based "Clustering as a Service" where users can upload, cluster, visualize, analyze, archive or share (if appropriate) their data.

5. ACKNOWLEDGMENTS

Research partially supported by NSERC, Canada, and by CNPq, under the Program Science without Borders, Brazil.

6. REFERENCES

[1] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander. OPTICS: ordering points to identify the clustering structure. In *ACM SIGMOD*, pages 49–60, 1999.

[2] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander. A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. *Data Min. Knowl. Discov.*, pages 344–371, 2013.

[3] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *IEEE TKDD*, pages 5:1–5:51, 2015.

[4] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM KDD*, pages 226–231, 1996.

[5] D. M. Johnson, C. Xiong, J. Gao, and J. J. Corso. Comprehensive cross-hierarchy cluster agreement evaluation. In *Late-Breaking Developments in the Field of Artificial Intelligence*, 2013.

[6] A. C. A. Neto, J. Sander, R. J. G. B. Campello, and M. A. Nascimento. Efficient computation of multiple density-based clustering hierarchies. In *IEEE ICDM*, pages 991–996, 2017.