Visualization of the OPTICS Algorithm

Sonja Biedermann* University of Vienna

Christian Permann† University of Vienna

ABSTRACT

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Treemaps; Human-centered computing—Visualization—Visualization design and evaluation methods

1 INTRODUCTION

The OPTICS algorithm is a density based clustering algorithm, that outputs a list of points, ordered according to the reachability of the points and the logic of the algorithm. This result is usually visualized by plotting this list as a bar chart with the output index as x-value and the computed reachability distance as y-value. This reachability plot then is used to make assumptions about the cluster-hierarchy in the data.

Our project idea is to visualize the resulting data from running the OPTICS algorithm on a given data set. OPTICS does not generate a simple mapping of points to a cluster ID, but rather outputs a list of reachability data - that is, the length (given by, e.g., the euclidean distance between those two points) that the algorithm had to jump from a given point to another. Short distances are preferred by the algorithm, so a series of short jumps likely marks a cluster.

However, in the end the user is looking at nothing but numbers and has to discern the patterns in the data himself. As such, the first step after running OPTICS is usually to draw a bar chart, which makes this task much easier.

As such, visualization is arguably already a core component of cluster analysis when using OPTICS. Why not allow for further manipulation of the algorithm, allowing to specify parameters such as the minimum point size to qualifiy as a cluster, or the cutoff distance that is applied onto the reachability data to define the actual clusters?

Furthermore, OPTICS is inherently capable of producing hierarchial clusterings, but this information is oftentimes discarded in favor of a simpler representation. Procuring an visualization method that is still simple, but allows for evaluation of hierarchial clusters is surely a worthwile undertaking, and one that we will strive for.

*e-mail:sonja.biedermann@univie.ac.at

†e-mail:a01463926@unet.univie.ac.at

2 MOTIVATION

Not many tools exist that take on this subject, although notable projects exist, especially Clustervision [2] and a DBSCAN visualization that we found [1], the latter of which only visualizes how DBSCAN goes about finding clusters (i.e. shows the epsilon neighborhoods). It is notable that DBSCAN results in a simple partition of points into clusters along with some metadata (e.g. core points versus edge points).

Clustervision is an especially powerful tool and takes on a multitude of algorithms (including OPTICS) and offers additional tooling like dimension reduction using TSNE. One of the core tasks of this tool is to validate the results of the algorithm - i.e. enable the user to check out the features of points in a cluster and see the relations between data that caused them to be classified into the same cluster.

This is something that we do not want to do, as we strongly feel that this is out of bounds for us. We want to make an example of simple data (i.e. low dimensional and spatial, a list of real points) and how the results given by OPTICS relate to this data set and the used settings, and show the partition derived from the reachability data.

2.1 Background information

2.1.1 Users

- Teachers and Students
- Researchers
- Anyone interested in the OPTICS algorithm

2.1.2 Tasks

The main task our implementation aims to help with, is to educate on how use OPTICS and to see if it fits given data and problem.

Teachers can either use our default data set or load a custom data set to explain how the algorithm works. They could use our implementation as a means of presentation and to show how parameter changes affect the output. The different views are also supposed to help understanding what the output actually means. Beeing able to play arround with parameters and having different usefull views could be beneficial for understanding the algorithm itself.

The second use-case in our mind was for researchers or anyone interested to find out if they want to do future work with this algorithm. For this they will probably just play around with our implementation to see if they want to put further work into OPTICS. Researchers may additionaly want to test their data set or part of it with our tool.

2.1.3 Data

We allow to select a preset data set or to load user defined data for analysis. Therefor we do not impose any restrictions on what data set can be visualized. The only requirements are that there have to be at least 2 dimensions, all points have to be of the same dimensionality and all values have to be numeric.

3 Related work

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.1 Other visualization solutions

See Motivation; I don't know if we need this section or if we should change Motivation

3.2 Previous visualization ideas

(-that you incorporated into your solution) Mockups on Website(some text also reusable?)

3.3 References

(- to both academic and commercial tools used) Maybe we don't need this section, since we talk about this elsewhere (e.g. implementation)

4 Approach

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.1 Description visualization design

Lots of images, describe each plot

4.2 Reasoning behind design choices

Why our chosen plots are better than possible other solutions (e.g. piechart for clustersizes) and heatmap stuff.

5 Implementation

5.1 toolkits, languages, platforms

For our implementation we used the Java Script with the d3 library. We also implemented the OPTICS algorithm ourselves to be able to extract additional information (in comparison to a premade solution) like the jump paths and custom cluster tagging logic.

We tested our implementation in Google Chrome, Microsoft Edge and Mozilla Firefox, whereas Firefox has some problems with the bigger data sets, we have not been able to combat.

5.2 Implementation challenges

- The OPTICS implementation isn't the most efficient one and it may be a good idea to have a backend for the calculations.
- We found it difficult to force the brush on top of the heat-map in a position where it makes sense, because since the x and y-Axis are mirrored, any non square selection doesn't make sense.
- We think the main problem that remains is how to make it behave well with larger inputs. Given a bad configuration (mainly a very large eps), OPTICS runs in quadratic time, and some components such as our scented widget will call the algorithm every time an input event is triggered, which may lead to an explosion of calls which all take n^2 time. This can't really be avoided (the output is needed to render the scented part), but maybe caching would cushion this effect. However, this depends on how often the same configurations would reappear, which may not be all that often.
- There seem to be performance issues with Firefox and large data sets, which we were not able to solve yet.

6 Results

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

6.1 Scenarios of use

A full iteration of how someone would teach the algorithm with our implementation + pictures (from webiste possibly + new tree view)

6.2 Performance of the system

Visual Performance (how it visualy scales with many points) vs Computational Performance (trivial implementation, which could be improved)

6.2.1 Computational Performance

Computationally our tool is currently not well optimized. Since we implemented OPTICS ourselves we took a trivial approach and the runtime complexity for bad input parameters may reach $O(n^2)$. For a more sophisticated implementation a server backend would be usefull as well as better optimized code.

A drawback of this is, that with current hardware working with a dataset larger than 400 points will result in stutter when interacting with different aspects of our tool.

6.2.2 Visual Performance

bla

6.3 Evaluation/Feedback

Mostly answer to the teachers feedback(Website), what we were told by testers(or what we think they would say...). Not just focus on bad feedback.

7 Discussion

teach/research/explore algorithm

7.1 Strengths and weaknesses

what we think is nice

7.2 Lessons we learned

d3, nice visualization, other nice stuff

8 Separation of Tasks

summery of the tables on the website i guess(+this/treeview/bugfixes)

9 Conclusion

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Acknowledgments

The authors wish to thank A, B, and C. This work was supported in part by a grant from XYZ. Probably not needed?

References

[1] N. Harris. Dbscan. https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/. Accessed: 2018-01-13.

[2] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. deFilippi, W. F. Stewart, and A. Perer. Clustervision. http://perer.org/papers/adamPerer-Clustervision-VAST2017.pdf. Accessed: 2018-01-13.