Graph mining project : Streaming Algorithms for k-center Clustering with Outliers

Garance Gourdel

February 2019

First, I would like to mention that I discussed of the project with Rémi Dupré and Mathieu Fehr, so it should be suprising if there are some similarities in our projects especially the approximation of the lower bound, but the implementation has been done entirely separately.

1 The problem

The problem those algorithm solve, is, given N points, to find the smallest radius r,(or whether a defined radius can cover them for the static algorithm) and the right k points to open facilities that with a radius r around them would cover all the N points except z outliers.

2 The implementation of the two algorithms

The main task was to implement two algorithms, first, the static algorithm described in section 3 of [1], then the streaming algorithm presented as algorithm 3.1 in [2]. The full implementation is available on github.

I enjoyed implementing the algorithm in C++, it was reasonably hard to implement, my biggest difficulty was handling the storing of free points and cluster center in the streaming algorithm. As I had created a special class for points and was using the key word *new*, I had to be careful to delete them carefully not to have any memory leak, but I was happy to succeed to manage without too much problem.

I chose to implement the algorithm with the euclidean distance as the README of the dataset mentioned that it was correct to do so but it may be interesting to test with the spheric distance to compare the result. As I had implemented a special class to store the points and compute the distance it would have been too much of a hassle but I didn't do it because It wasn't necessary and I had other project to work on.

On the other hand, I did implement a supplementary feature to improve the radius found by the algorithm. After the end of the algorithm, I go again through all the points in the dataset (except the outliers) and compute the radius to their closest cluster center, thus I get the real radius found by the chosen cluster centers. This new radius is called "improved r" in the result detailed later.

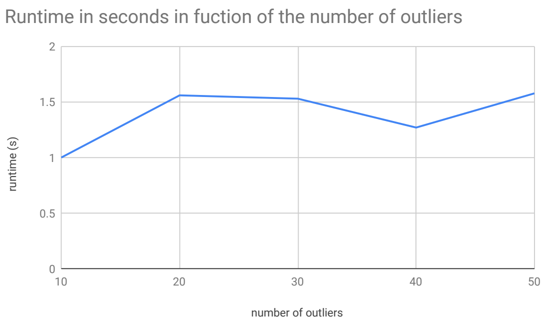
As a remark on the project : it might be because of my choice to use the euclidean distance but the algorithm was very fast to run and not very challenging which was a bit disappointing.

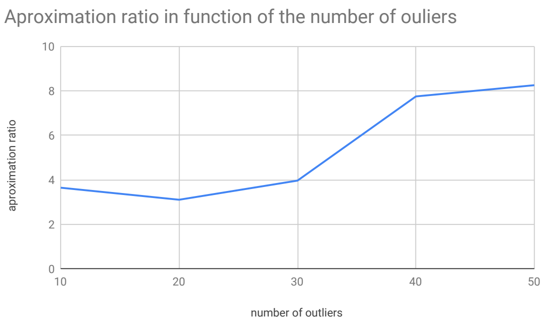
3 Computation of the lower bound

For the approximation of the lower bound, I chose to find the smallest r such that there was $k + z$ points at distance at least r from one another. To gain on efficiency I made a dichotomy, considering that if I didn't found $k + z$ points at least it meant that r was to big.

In terms of implementation, I chose not to implement the lower bound computation in a streaming setting as it was not part of the streaming algorithm.

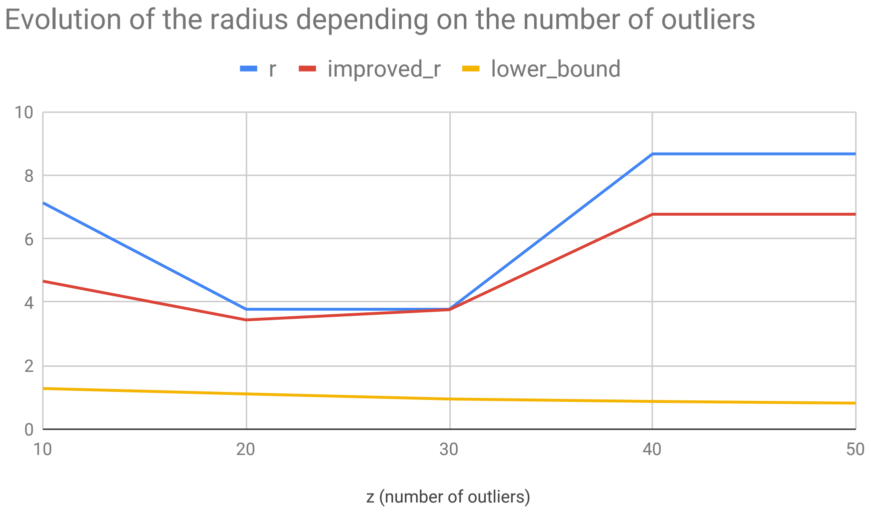
4 Analysis of the results


Runtime in seconds in fuction of the number of outliers


Aproximation ratio in function of the number of ouliers

z	10	20	30	40	50
r	7.12905	3.77602	3.77602	8.66942	8.66942
improved r	4.66063	3.44037	3.76176	6.76881	6.76881
lowerbound	1.27983	1.1097	0.949478	0.873184	0.819778
runtime	1	1.56	1.53	1.27	1.578
approximation ratio	3.6416	3.1002	3.9619	7.7518	8.2568

Table 1: Results in function of the number of outliers


Evolution of the radius depending on the number of outliers

My main observation on those result would be that there might be space for improvement on the initialization: intuitively, increasing the number of outliers should only decrease the resulted radius. But it does not because of the initialization, as we initialize r on the $k + z$ first points, the initialization changes and the algorithm highly depends on the initialization. The changes stay coherent with the approximation ratio of 8 but it's not very satisfying to get worst result when allowing for more exceptions...

Overall the approximation ratio found seems coherent with the theoretical result that the algorithm is an 8-approximation, and regarding the running time, It is too low for me to be able to have a critical analysis, It should be tested with a larger dataset or a more appropriate distance to be very challenging.

5 Bonus : plot of the result

I thought it was more fun to plot the dataset and the cluster center that the algorithm did output, it helps to visualize the problem, spot any mistake and see where there is space for improvement.

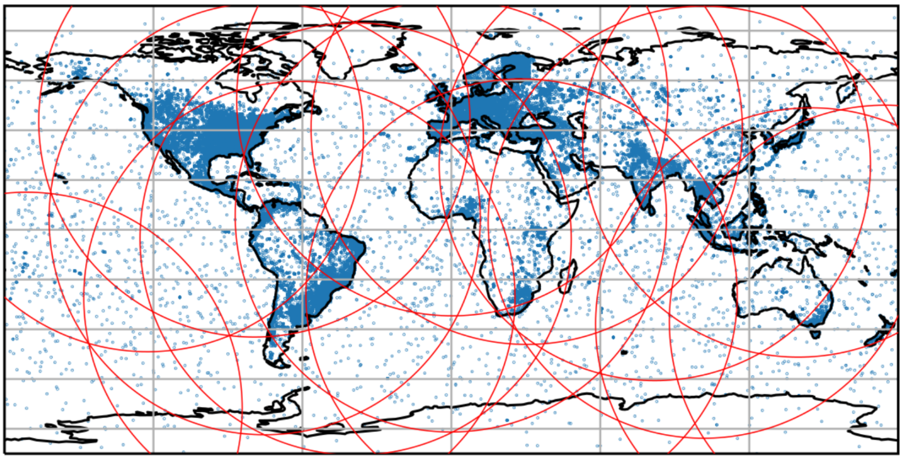


Figure 2: plot of the k clusters and the 10 allowed outliers

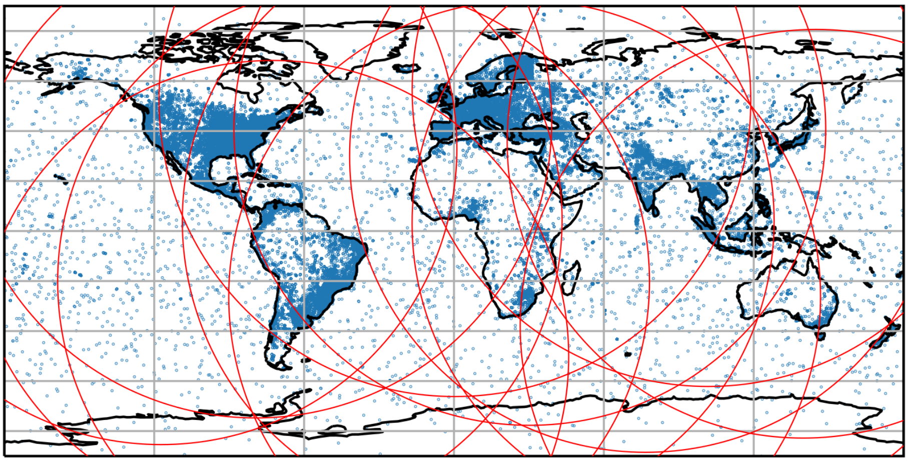


Figure 3: plot of the k clusters and the 50 allowed outliers

References

[1] Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 642–651, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[2] Richard Matthew McCutchen and Samir Khuller. Streaming algorithms for k-center clustering with outliers and with anonymity. In Ashish Goel, Klaus Jansen, José D. P. Rolim, and Ronitt Rubinfeld, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 165–178, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.