ORIE 4741 Midterm Report

Irina Wang, Kevin Ko, Camilo Cedeno-Tobon, Simran Shinh

1. Algorithm and Project Overview

The foundation of this algorithm is the k-clustering problem, in which there is a set of points D to be separated into k clusters. The clustering without over-representation algorithm introduces another parameter. Each point has a color, and the algorithm puts a constraint on the representation of that color in each cluster. In other words, it prevents any one color from dominating a cluster.

For the first step of the project, we sought to replicate the formulation of the original algorithm from the original paper. At a high level, this consisted of implementing a greedy clustering algorithm baseline, using Google OR tools to implement a linear program and maximum flow network, and preparing a large collection of text data for testing. Our plans for improving the algorithm are detailed later in the report.

The test data we used is called Reuters, and it is sourced from the UCI Machine Learning Repository. The data set consists of 50 texts from 50 authors (for a total of 2500 texts). Since the algorithm uses distances of points in its computations, we converted each text into a 10-dimensional vector using Gensim's Doc2Vec model. For the purposes of our algorithm, each text (now 10-dimensional vector) is a point, and the author of each text is the point's color.

2. Algorithm Analysis and Replication

2.1 Greedy K-Center (Baseline):

The first step in replicating the results of the paper is to implement the greedy k-center algorithm, which yields an upper bound for our eventual optimal solution. The greedy k-center algorithm is a simple method for solving the k-clustering problem, in which we begin with a set D of points and an integer bound k. The goal of the k-clustering problem is to cluster the points into a maximum of k clusters while also minimizing cost.

Although the k-clustering problem can be computationally complex, the greedy k-center algorithm provides a simple way to obtain a loose solution. The algorithm consists of 5 steps:

1. Pick an arbitrary initial point and make it a facility (i.e. cluster center)
2. For every point j in D, find the minimum distance to the nearest facility and record each distance. Find the maximum of the list of minimum distances, and make the corresponding point (the furthest client from the current set of facilities) a new facility
3. Repeat step 3 until the set of facilities has size k (there are k facilities)
4. For every point j in D, find the minimum distance to the nearest facility. The smallest minimum distance is λ, which is our solution
5. Return the set of facilities and λ

The caveat to our k-clustering problem is that each point has an extra parameter that differentiates our data set from the set of points in a classic k-clustering problem; this parameter takes the form of authors in our test data, and colors in the original paper.

The crux of the problem lies in this parameter: we are attempting to augment an algorithm that solves the k-clustering problem while also ensuring that each cluster is not overwhelmingly represented by one type of this parameter, such as a specific author or color.

2.2 α-capped k-center clustering problem (Improvement):

Since the greedy k-center algorithm does not account for the extra parameter, it can only give us a lower bound for our eventual optimal solution, and we must move on to a new algorithm to solve the α-capped k-center clustering problem. Our formulation of this algorithm contains 4 steps:

1. Solve a Linear Program Relaxation of the problem with a guess of the solution value λ' (experimentally chosen), to obtain a fractional solution and a set of facilities, F, associated with the fractional solution. In this LP, we only allow assignments to facilities if the client is less than λ' from the facility. This fractional solution will have “partial assignments”, such that each client may be partially assigned to multiple facilities, and the number of facilities chosen will be much greater than k, as we allowed fractional values.

2. Narrow down the set of facilities from F to F' by removing facilities that do not have a distance of at least 2λ” from all other facilities in the subset, for some experimentally chosen λ” value. This set must be maximized, such that there must not be any facility in F not chosen to be in F', but is at least 2λ” from all other facilities in F'. We choose the smallest λ” such that the number of facilities is less than or equal to k. *This step is different from the paper we reference's approach, which sets λ” = λ', and thus gives a bound of 3λ' for the cost of the integral assignment.

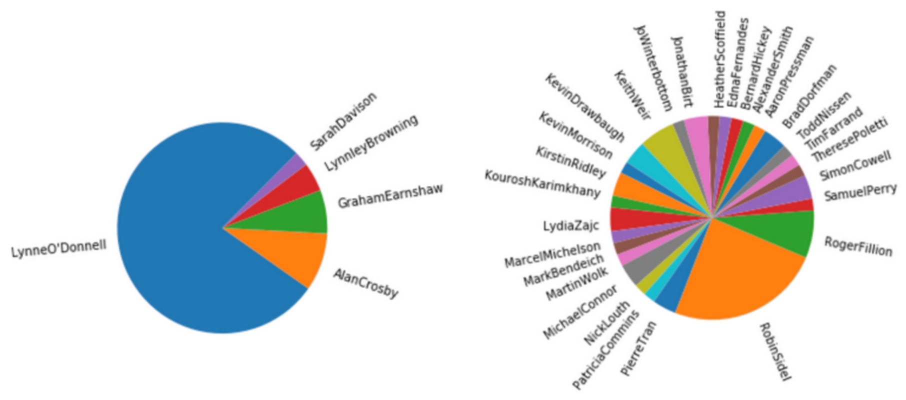
3. Re-assign clients to facilities in F': for clients assigned to facility f in F but not in F', we route the assignment to the nearest facility in F'. Notice that there must be a facility in F' within 2λ” from f, as F' is maximized. Since all clients are at most λ' from their initially assigned facilities, and the new facility is at most 2λ” away, the rerouted assignments must have cost within 2λ” + λ'.

4. Next, to obtain an integer assignment for every client, we construct a maximum flow network with integer edge capacities, and utilize the fact that feasible network flows with integral bounds on edges has an optimal integral solution. The specific construction of the graph is denoted in the paper we are referencing.

The solution to the maximum flow algorithm gives an integral solution to the assignment problem, with very limited violations of the α-cap constraint. The violations arise from the fact that we use maximum flow as an approximation algorithm, which may sacrifice the α-cap constraint when converting to an integer solution.

The LP is implemented using Google OR Tools' GLOP solver, and as it attempts to determine the values for $2500^2 + 2500$ variables, is the most time-expensive portion of the calculations. To determine an optimal guess of the cost λ', we began with $\lambda/2$, where λ is the solution value returned by the greedy algorithm. So long as the LP finds no solutions with the given λ' value, we slowly increment λ' and rerun the LP, until a feasible solution is returned. The maximum flow is implemented using Google OR Tools' Max-Flow solver.

3. Algorithm Results and Implications



(a) Greedy Algorithm (b) α-capped k-center Algorithm

Figure 1: Clustering Results Comparison

3.1 Greedy Algorithm Results:

For k = 25, the λ obtained is 19.6. From what we can observe in the above pie chart, unfairness in representation is indeed present in the clustering when only the greedy k-center algorithm is used. If the algorithm was more fair with regard to representation, then the sections of the pie chart would be much more even in size; however, we can see that there are a few authors that dominate this particular cluster.

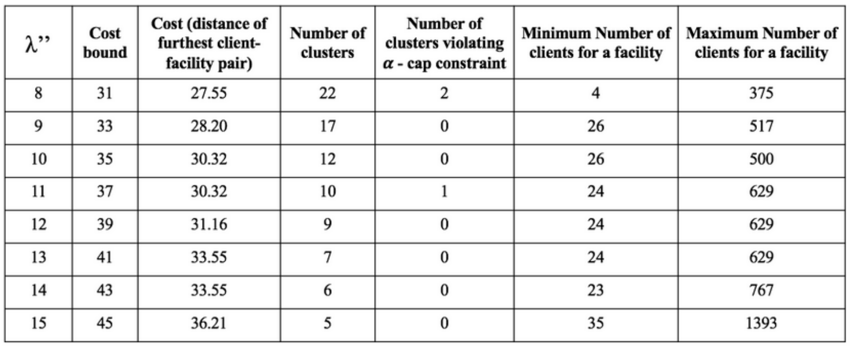
What does this mean for the greedy k-center algorithm? While it offers a viable solution, it is in no way even close to optimal, especially when considering our extra parameter. Therefore, we can use the greedy k-center algorithm to find a lower bound for our optimal λ , but because it does not actually calculate clusters with concern for the extra parameter (in this, authors), we must formulate a new algorithm. This leads us to the LP formulation.

3.2 α-capped k-center Clustering Results:

For the Reuters data, we set k = 25 and $\alpha = 0.5$, to first examine the significant case where no cluster has one significantly dominating author. As the LP is very time-consuming to solve, as of now we are unable to find an exact minimum λ' value where a feasible solution is possible - a working minimum we have is $\lambda' = 15$.

For this λ' value that produces a fractional assignment, we then experimented with λ" values for the optimal integral assignment. The algorithm in paper which we are referencing sets $\lambda" = \lambda'$, and thus gives a bound of $3\lambda"$ for the cost of the integral assignment. We noticed, however, that λ" values smaller than λ' is possible, and gives a better cost-bound and final cost.

The α constraint is mostly satisfied, with a few exceptions arising from the approximation step from maximum flow.



λ''	Cost bound	Cost (distance of furthest client-facility pair)	Number of clusters	Number of clusters violating α - cap constraint	Minimum Number of clients for a facility	Maximum Number of clients for a facility
8	31	27.55	22	2	4	375
9	33	28.20	17	0	26	517
10	35	30.32	12	0	26	500
11	37	30.32	10	1	24	629
12	39	31.16	9	0	24	629
13	41	33.55	7	0	24	629
14	43	33.55	6	0	23	767
15	45	36.21	5	0	35	1393

4. Next Steps

The observation we made regarding possible values for λ' and λ" requires further experimentation - perhaps we were able to find large discrepancies between λ' and possible λ" values due to the in-optimality of our λ' value, which could possibly be chosen to be smaller. This, however, opens up a new question - since testing λ" values for max flow is exponentially faster than testing λ' values for the LP, is there a worthy trade-off between the potential increase in cost for using in-optimal λ' values, and the decrease in time complexity? Regardless, we need to potentially find a method to speed up the LP computations such that more values may be tested, and repeat experimentation for other alpha values.

Even further explorations would be to define a similar algorithm for the k-means objective, which takes the cost as the average cost of client-facility assignments instead of the worst cost. The LP can be easily adjusted for this new objective - what follows is to construct a new facility and client rerouting scheme, similar to the above, that minimizes the new objective.

Finally, we can compare the results of this algorithm to unsupervised learning techniques covered in class, and we can test the algorithm other big data sets to further evaluate its robustness.

5. References

Ahmadian, Sara, et al. “Clustering without Over-Representation.” Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining - KDD 19, 2019, doi:10.1145/3292500.3330987.