Optimal Distance Measure for Agglomerative Hierarchical Clustering

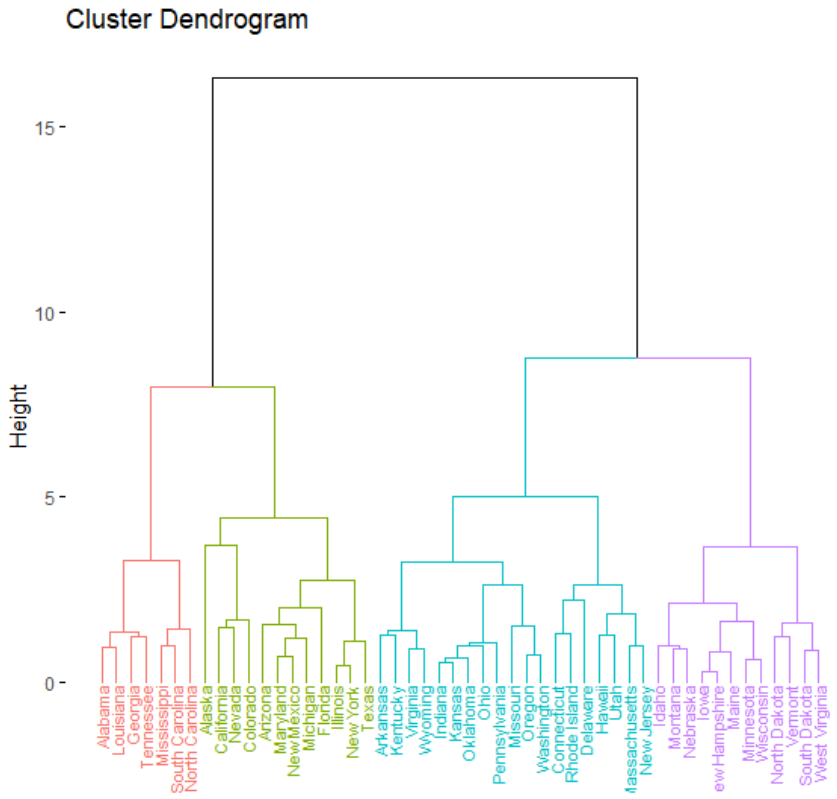
CS5100 Semester Project Proposal

Xichen Liu Jeff Turgeon

Problem Statement

Agglomerative Hierarchical Clustering is a bottom up approach to creating clusters within a given dataset. Rather than starting with one large cluster, and breaking the dataset into clusters based on specified criteria, the approach treats every data point as a cluster, and then starts merging clusters together based on the given algorithm. In hierarchical algorithms, agglomerative hierarchical clustering (ACH) has become one of the primary clustering algorithms due to less time complexity and better computational stability. [5]

The algorithm utilizes a linkage metric, and a distance metric to determine how the clusters are formed. Both metrics impact the how the clusters are built, and how clusters are joined, however, for distance-based hierarchical clustering methods, a poor choice of distance metrics may result in poor clusterings. [2] On the other hand, a few initially carefully chosen joins can greatly improve the accuracy of the entire dendrogram. [1] A dendrogram is a graphical representation of the resulting tree structure showing how the clusters are built form the base clusters up. A sample dendrogram, illustrating the clusters, and how they form a tree was generated in R following the steps outlined in [3]. This example utilized the USArrests built in dataset, and utilized the euclidean distance as the dissimilarity measure in creating the clusters. As you can see, the analysis resulted in four clusters, as designated by the different colors, that grouped the states based on the similarity of their arrest data.



Since the metrics supplied to the agglomerative hierarchical clustering algorithm can impact the output, it is possible to have multiple groups of clusters created from the same dataset. The clustering is the mathematical formation which discovers the particular patterns into the dataset which is inside every cluster discovers the similarity degree. [4] The distance metric is often considered a measure of similarity or dissimilarity between the data points in the initial dataset.

This project will aim to evaluate various distance metrics to determine a most optimal metric for a given dataset. The only input parameter that will vary between executions of the algorithm will be the distance metric, so we will be able to determine the impact that the metric has on the eventual clustering. While we will be able to see the differences in the output, we will need to determine a measure for optimization, and we will need to apply the measure against the various outputs to calculate a quantifiable difference. While our initial project will focus on the comparison of the metrics against a similar dataset, we are curious if we would see similar results across various datasets.

As part of our project we will investigate the following distance measures:

- Euclidean Distance
- Manhattan Distance
- Cosine Distance
- 2 Different Custom Heuristics

The custom heuristics will be developed once we complete some initial exploratory data analysis on the data set that we will work with.

Problem Analysis

The artificial intelligence components for our project consist of the agent that will execute the algorithm multiple times, storing the results for further analysis, and the heuristics that will be developed as distance measures. The agent will need to store the state following each call to the algorithm, so she results can be compared to determine the optimal value.

The state space the agent will need to store includes an array for each distance measure being evaluated. The array will consist of the cluster number that each item in the dataset belongs to.

StateSpace:

- Euclidean Distance Array
- Manhattan Distance Array
- Cosine Distance Array
- Heuristic 1 Array
- Heuristic 2 Array

The arrays will be used in the analysis phase when trying to determine which measure is the most optimal for the given dataset. The size of each array will be directly tied to the number of rows in each dataset, so the size of the state space will be equal to n X d where n represents the number of rows, and d represents the number of distance measures. The state transition will occur following the completing of each call to the clustering algorithm. The state transition is deterministic, in that there will be no randomness involved in the state transition.

Problem Representation

This problem environment can be described as follows:

- **Fully Observable**
  - All of the information in our environment will be observable since each clustering process will be a state, and there are no other moving parts in the environment.
- **Single Agent**
  - The agent that iterates through the distance measures and gathers the results will be the only agent in the environment.
- **Deterministic**
  - The states are based on the results from the clustering
- **Episodic**
  - Each state change adds the array for the latest algorithm execution. The arrays are independent and do not depend on each other.
- **Static**
  - The environment does not change, other than through the distance method parameters supplied by our agent.
- **Discrete**
  - There are a limited number of states that change in steps.

Data set or other source materials

We have not decided on a dataset yet. We are evaluating a few datasets from https://www.kaggle.com/, including some datasets that were built specifically for clustering. As part of our initial data processing and cleanup, we will need to put all of the columns in our dataset being processed into numeric format. This may require us to create some categorical representations that we need to replace fields with. We will complete some initial exploratory data analysis on the data to look for potential heuristics that we can use as distance measures in our analysis. Our intent is to complete the data clean up and programming using python, but we have gone through this process with the USArrests sample data within the R environment to gain an understanding of what we will need to do.

Deliverable and Demonstration

The main program to process the data set, the file of the original dataset, the statistics info of the clustered dataset, the comparison between the results of using different hyper-parameters.

The program will iterate through the various distance methods, such as euclidean, manhattan, or other heuristic based methods, that should be included in the comparison, will run the agglomerative heuristic clustering algorithm using each distance method, and will return an analysis comparing the methods. While the methods will produce similar results, the differences in the results will allow us to determine which approach is most optimal, and therefore would provide more accurate clusters for further analysis.

Evaluation of Results

We have a few ideas regarding how to evaluate the results of our distance measures. We could check how dense the inside of a cluster is, and how sparse it is between different clusters. We

could also apply a correlation calculation on each cluster to see how they compare to each other. Additionally, the ratio of the largest distance inside of the cluster and the smallest distance between the clusters can also evaluate the results.

We will use these metrics to compare the results from various distance measures to determine which measure is the most optimal for the given dataset.

Major components and schedule

First, we'll find the proper dataset to use; second, we need the dataset to be cleaned; third, we are going to determine which heuristic function to use to determine the actions of the agent which are the combinations; then we will run the clustering algorithm in different heuristic functions to get the results.

There will be 4 major components in the main program:

> The part that cleans the dataset, the part that implements the heuristic functions, , the part that implements the clustering algorithm, and analysis of the results.

The order to implement the major components will be from cleaning dataset to heuristic function implementations, then from heuristic function implementations to applying the dataset to clustering algorithm.

Each of us will respond to one major component, then do the third together.

Date	Xichen Liu	Jeff Turgeon
4/4	Clustering algorithm Program	Data Cleaning Routine
4/15	Half of heuristic functions	Half of heuristic functions
4/25	Analysis	

References

[1] Davidson, I., & Ravi, S. S. (2007). Using instance-level constraints in agglomerative hierarchical clustering: theoretical and empirical results. *Data Mining Knowledge Discovery*, 257-282.

[2] Heller, K. A., & Ghabramani, Z. (2005). Bayesian Hierarchical Clustering. *Proceedings of the 22nd International Conference on Machine Learning*, (pp. 297-304). Bonn, Germany.

[3] Kassambara, A. (n.d.). Agglomerative Hierarchical Clustering. Retrieved from Data Novia: https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/

[4] Pasupathi, S., Shanmuganathan, V., Madasamy, K., Yesudhas, H. R., & Kim, M. (2021). Trend Analysis using agglomerative hierarchical clusting approach for time series big data. *The Journal of Supercomputing*, 6505 - 6524.

[5] Zhou, S., Xu, Z., & Liu, F. (2017). Method for Determining the Optimal Number of Clusters Based on Agglomerative Hierarchical Clustering. *IEEE Transactions On Nueral Networks and Learning Systems, Vol. 28, No. 12*, 3007 - 3017.