The Effect of Dimensionality in Affinity Choice for Agglomerative Hierarchical Clustering

Xichen Liu
Jeff Turgeon

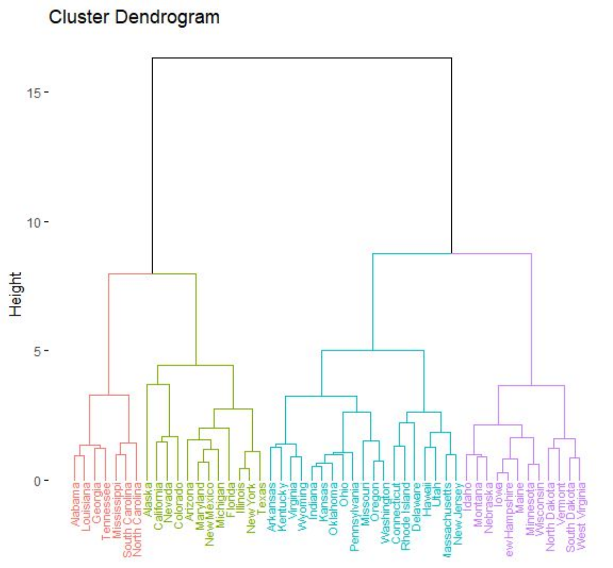
CS5100 - Introduction to Artificial Intelligence

Project Aim

Agglomerative Hierarchical Clustering is a bottom up approach to clustering data. There are two primary variables that can affect the clusters that are generated. Different combinations of these variables will generate different clusters.

- Linkage - A method for building the clusters based on how close the observations are.
- Affinity - The distance metric used to determine how close the elements are to each other.

The aim of this project is to see if the dimension of the dataset affects which affinity should be used in the clustering algorithm.



Project Objectives

- Are there any affinities consistently better as the dimensions increase or decrease?
- Are there any affinities that are relatively consistent across dimensions?
- Are there any combinations of linkage and affinity that perform better as the dimension increases?

We will use a combination of existing measures and observations to explore these questions.

Literature Review

- In 2010, Erisoglu and Sakallioglu explored different combinations of distance metrics and linkages in hierarchical clustering, and determined that the distance metric is impactful.
- In 2011, Rajalingam and Ranjini explored clustering with different data types, and determined that the run time for datasets of the same type of data, and with comparable record counts is relatively equal. The used a single distance metric for each data type.
- In 2014, Kumar, Chhabra, and Kumar explored 10 distance measures against 8 clustering algorithms, and determined that the performance and quality of the clusters varied based on the distance, clustering technique, and the nature of the data.
- In 2016, Irani, Pise, and Phatak explored distance metrics further, and determined that the metrics can tell you the distance between points, but do not consider the behavior of the data. They also note that negative data can be a challenge to clustering.
- In 2018, Ogbuabor, and Ugwoke explored a healthcare dataset using K-Means and DBSCAN with varying distance metrics, and evaluated the results using the Silhouette Score value.

Project Requirements

- Obtain Clean Datasets
- Vary the Dimensions of the Datasets
- Vary the Linkage Parameter
- Vary the Affinity Parameter (Including Standard and PreComputed)
- Calculate the Measures
- Analyze the Results

Datasets

Dataset	Description	Attributes	Records
CC General *	Credit Card Data	16	8,950
COVID *	COVID Statistics by Country	7	225
Credit Card Customer Data **	Credit Card Summary Data	6	660
Mall Customers *	Mall Customer Summary Data	5	200
Sales Transaction **	Sales Transaction Information	53	811
Wholesale **	Sales Summary Information	8	440

* https://www.kaggle.com/

** https://archive.ics.uci.edu/ml/index.php

Variables Part 1

Dimensions: 2, 3, 4

Linkage	Definition
Average	Average inter-cluster distance
Centroid	Distance between the two mean vectors of the clusters
Complete	The farthest distance between elements in each cluster
Single	The smallest distance between elements in each cluster
Ward	Creates new clusters that minimize the variance

Variables Part 2

Affinity	Definition	Formula *
Euclidean	Straight Line Distance	sqrt(sum((x-y)^2))
Manhattan	City Block Distance	sum(|x -y|)
Cosine	1 - Cosine Similarity	1 - (x * y)/(|x||y|)
Chebyshev	The maximum distance between vectors	max(|x - y|)
Minkowski	A generalization of the Euclidean and Manhattan distance	sum(w * |x-y|^p)^(1/p)
Mahalanobis	A measure of the number of standard deviations a point is from the mean distribution	sqrt((x-y)’ V^-1 (x-y)

* https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html

Measures

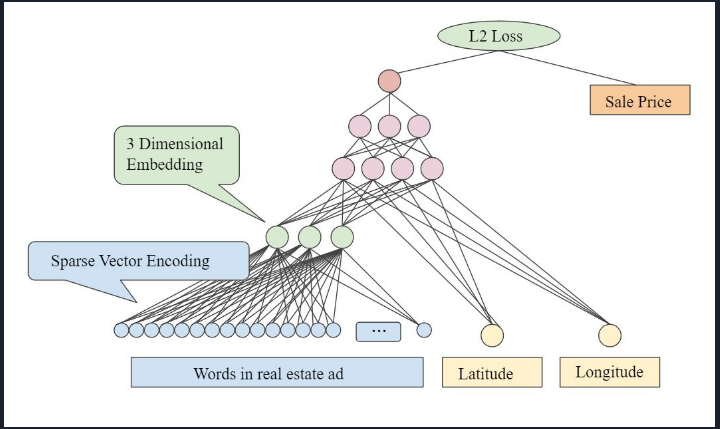
Measure	Definition	Interpretation
Silhouette Score	A measure of similarity between items in the same cluster	-1 to 1 0 means overlap Higher values are better
Calinski Harabasz Score	A measure of the variance in cluster dispersion	Higher values are better
Davies-Bouldin Index	A measure of the average similarity between clusters focusing on the size of the clusters and the distance between the clusters	0 and above Closer to 0 is better

- https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Prior Works

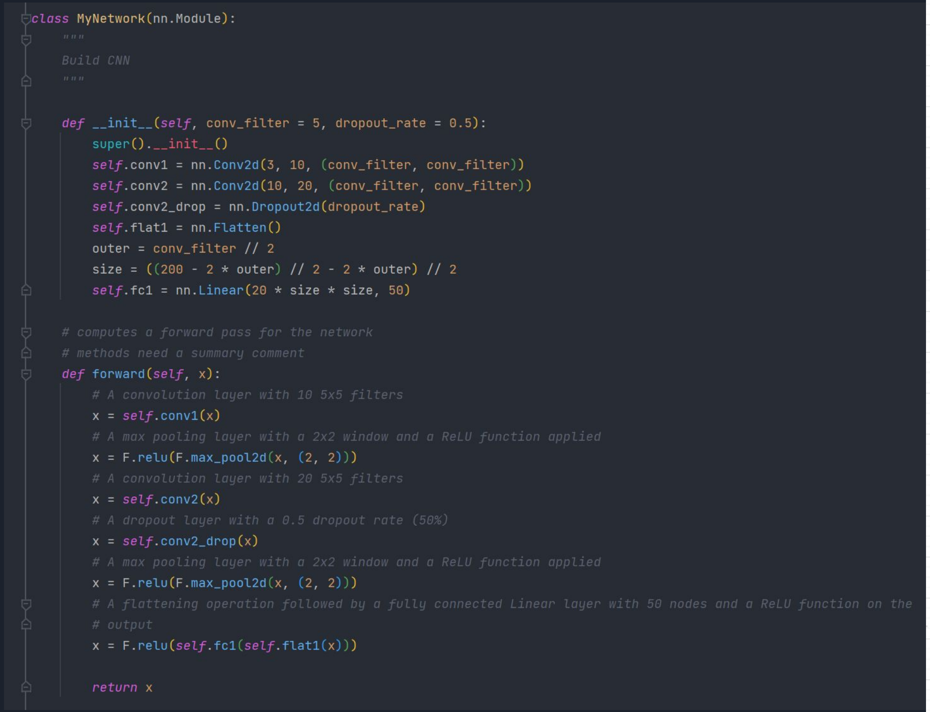
- Datasets
  - Numerical data
  - Clean datasets
  - Normalization
- Number of dimensions
  - Embedding space

Embedding space



- Embedding is a kind of dimension reduction technique
  - Neural network
  - Generally , it means to project the higher dimension values into lower dimension space
  - Proportionally

Demo of projection in to Embedding space

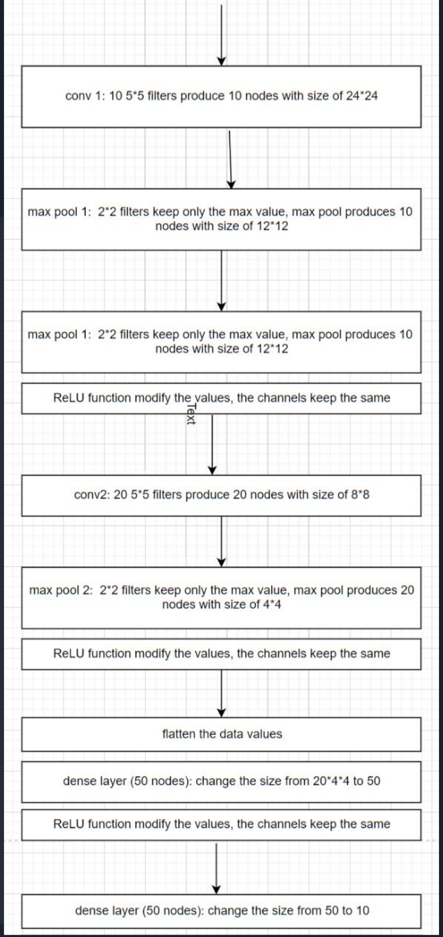


```python
class MyNetwork(nn.Module):
    """
    Build CNN
    """

    def __init__(self, conv_filter = 5, dropout_rate = 0.5):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 10, (conv_filter, conv_filter))
        self.conv2 = nn.Conv2d(10, 20, (conv_filter, conv_filter))
        self.conv2_drop = nn.Dropout2d(dropout_rate)
        self.flat1 = nn.Flatten()
        outer = conv_filter // 2
        size = ((200 - 2 * outer) // 2 - 2 * outer) // 2
        self.fc1 = nn.Linear(20 * size * size, 50)

    # computes a forward pass for the network
    # methods need a summary comment
    def forward(self, x):
        # A convolution layer with 10 5x5 filters
        x = self.conv1(x)
        # A max pooling layer with a 2x2 window and a ReLU function applied
        x = F.relu(F.max_pool2d(x, (2, 2)))
        # A convolution layer with 20 5x5 filters
        x = self.conv2(x)
        # A dropout layer with a 0.5 dropout rate (50%)
        x = self.conv2_drop(x)
        # A max pooling layer with a 2x2 window and a ReLU function applied
        x = F.relu(F.max_pool2d(x, (2, 2)))
        # A flattening operation followed by a fully connected Linear layer with 50 nodes and a ReLU function on the
        # output
        x = F.relu(self.fc1(self.flat1(x)))

        return x
```



conv 1: 10 5*5 filters produce 10 nodes with size of 24*24

max pool 1: 2*2 filters keep only the max value, max pool produces 10 nodes with size of 12*12

max pool 1: 2*2 filters keep only the max value, max pool produces 10 nodes with size of 12*12

ReLU function modify the values, the channels keep the same

conv2: 20 5*5 filters produce 20 nodes with size of 8*8

max pool 2: 2*2 filters keep only the max value, max pool produces 20 nodes with size of 4*4

ReLU function modify the values, the channels keep the same

flatten the data values

dense layer (50 nodes): change the size from 20*4*4 to 50

ReLU function modify the values, the channels keep the same

dense layer (50 nodes): change the size from 50 to 10

Code Demo

Analysis Approach

- Generated 342 records that included the dataset name, linkage, affinity, dimension, Silhouette Score, Calinkski Harabasz Score, Davies-Bouldin Index and the number of clusters
- Discarded any records with a single cluster
- Calculated the range between the fourth and second dimension for each measure for each linkage and affinity combination
- Averaged the results across the datasets
- Tabulated the data in separate tables for each measure for interpretation

Results - Change in Silhouette Score

Linkage	Chebyshev	Cosine	Euclidean	Mahalanobis	Manhattan	Minkowski
Average	0.07	0.18	0.31	-0.01	0.42	0.07
Centroid	-	-	-	-	-	-
Complete	0.30	0.08	0.36	-0.24	0.25	0.11
Single	-0.10	-0.17	0.91	-0.10	0.47	0.37
Ward	-	-	0.19	-	-	-

The dimension changed from 2 to 4

Average change in Silhouette Score was 0.18
Largest increase in Silhouette Score was 0.91
Largest decrease in Silhouette Score was -0.24

Results - Change in Calinski Harabasz Score

Linkage	Chebyshev	Cosine	Euclidean	Mahalanobis	Manhattan	Minkowski
Average	210.69	-362.5	109.18	15199.21	284.02	-5.77
Centroid	-	-	-	-	-	-
Complete	379.27	172.63	296.97	642.46	-3407.30	55.20
Single	-159.80	-102.41	90.21	7519.38	90.59	-10770.77
Ward	-	-	716.52	-	-	-

The dimension changed from 2 to 4

Average change in Calinski Harabasz Score was 576.73
Largest increase in Calinski Harabasz Score was 15199.21
Largest decrease in Calinski Harabasz Score was -10770.77

Results - Change in Davies-Bouldin Index

Linkage	Chebyshev	Cosine	Euclidean	Mahalanobis	Manhattan	Minkowski
Average	0.11	0.16	-0.19	-0.08	0.44	-0.05
Centroid	-	-	-	-	-	-
Complete	0.33	-0.27	0.51	-0.32	0.53	-0.10
Single	0.90	0	-0.91	-0.04	1.09	1.02
Ward	-	-	0.16	-	-	-

The dimension changed from 2 to 4

Average change in Davies-Bouldin Index was 0.17
Largest increase in Davies-Bouldin Index was 1.09
Largest decrease in Davies-Bouldin Index was -0.91

Results Summary

- Are there any affinities consistently better as the dimensions increase or decrease?
    - Euclidean, Manhattan, and Minkowski improved the Silhouette Score across linkages
    - Euclidean and Mahalanobis improved the Calinski Harabasz Score across linkages
    - Mahalanobis improved the Davies-Bouldin index across linkages

- Are there any affinities that are relatively consistent across dimensions?
    - We saw fluctuations in all measures as the dimension changed across all affinities

- Are there any combinations of linkage and affinity that perform better as the dimension increases?
    - Single Linkage with Euclidean improved by 0.91 in Silhouette Score
    - Average Linkage with Mahalanobis improved by 15199.21 in Calinski Harabasz Score
    - Since with Euclidean improved by -0.91 in Davies-Bouldin Index

Future Work

- Are there other measures that give additional insight?
- Evaluating the cluster statistics against a dataset Ground Truth
- Is there an upper threshold on the number of dimensions an affinity improves with?
- Do any of the newer affinities, like Geometric Distance, perform better at higher dimensions?

Conclusion

- There appears to be some affinities that perform better as the dataset dimension increases.
- Some affinities performed better with specific linkages, while others were more universal.
- Further research and findings in this area could be very impactful to future clustering efforts.

References

Bhasin, Arjun (2018). *Credit Card Dataset for Clustering* [Data set]. https://www.kaggle.com/datasets/arjunbhasin2013/ccdata

Bora, D. J., & Gupta, D. K. (2014). Effect of Different Distance Measures on the Performance of K-Means Algorithm: An Experimental Study in Matlab. International Journal of Computer Science and Information Technologies, 2501-2506.

Cardoso, Margarida G.M. S. (2014). *Wholesale customers* [Data set]. https://archive.ics.uci.edu/ml/datasets/Wholesale+customers

Chen, Dr. Daqing (2015). *Online Retail* [Data set]. https://archive.ics.uci.edu/ml/datasets/Online+Retail

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

Devakumar, K. P. (2020). *Covid-19 Dataset* [Data set]. https://www.kaggle.com/datasets/imdevskp/corona-virus-report

Erisoglu, M., & Sakallioglu, S. (2010). An Investigation of Effects On Hierarchical Clustering of Distance Measurements. Selcuk Journal of Applied Mathematics, 39-53.

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

Irani, J., Pise, N., & Phatak, M. (2016). Clustering Techniques and the Similarity Measures used in Clustering: A Survey. International Journal of Computer Applications, 9-14.

References 2

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

Kumar, V., Chhabra, J. K., & Kumar, D. (2014, June). Performance Evaluation of Distance Metrics in the Clustering Algorithms. INFOCOMP, 13(1), 38-51.

Ogbuabor, G., & Ugwoke, F. N. (2018). Clustering Algorithm for a Healthcare Dataset Using Silhouette Score Value. International Journal of Computer Science & Information Technology, 27-37.

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

Rajalingam, D., & Ranjini, K. (2011). Hierarchical Clustering Algorithm - A Comparative Study. International Journal of Computer Applications, 42-46.

Shwebabh (2018). *Mall Customers* [Data set]. https://www.kaggle.com/datasets/shwetabh123/mall-customers

Tan, James (2017). *Sales Transactions Dataset Weekly* [Data set]. https://archive.ics.uci.edu/ml/datasets/Sales_Transactions_Dataset_Weekly

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

Questions?

Thank you!