Investigation of Ohio Adjudicated Youth Through Cluster Analysis

Sam Hockenberry

March 7, 2019

Contents

1 Abstract

The Allegheny County Court System keeps records on all juvenile court cases, referrals, and accusations within Allegheny County. While the initial goals of this research were to identify common characteristics of dependent and adjudicated children within Allegheny County, the data requested from the Allegheny County Court System was denied. This denial lead into another interest—focusing on arrested and rearrested youth. In 2009, the University of Cincinnati Corrections Institute published a report on a new tool called the Ohio Youth Assessment System (OYAS) that identifies the needs and risks of juvenile offenders. This report describes five tools that make up the OYAS and their initial results on arrested and rearrested youth. This project uses the results of this report as a base to recreate representative data and discover new findings. Through cluster analysis we hope to identify common characteristics of arrested and rearrested youth to allow for more informed decisions concerning the health and safety of adjudicated youth.

2 Introduction

Cluster analysis uses grouping techniques to group data together to understand underlying associations within data. These techniques utilize different types of algorithms to cluster the data differently. For example, the k-means algorithm uses user-set information and the most center point in a cluster as reference while DBSCAN automatically calculates the amount of clusters in the data based on sample point density [5]. Cluster analysis will allow us to identify characteristics within the data that are not usually able to be seen.

Cluster analysis usually serves two purposes: clustering for "understanding" or clustering for "purpose" [5]. Clustering for understanding focuses on extracting specific pieces of information from the data of interest. For example, a business may be interested in clustering data to understand who is actually buying their product. In this case a business could cluster their customers by location of purchase, quantity of purchase, or even time of year. Clustering for purpose takes a much different route than clustering for understanding. This approach examines each cluster of data plotted and identifies the cluster prototypes, a data point that accurately represents the whole cluster of points. Since most analysis techniques have require exponential time as the data set gets large identifying the prototypes now allows researchers to easily apply different types of analysis techniques to the data without needing to use the entire sample. In this research we will be clustering to understand how dependent and juvenile children's characteristics mirror one another [5].

3 Cluster Analysis Clustering Methods

There are three main types of clustering methods: partitional, hierarchical, and density-based clustering. Within each method exists multiple types of clustering algorithms. Within this section we explain clustering algorithms that were thought to produce significant results prior to the acquisition of the data.

3.1 Partitional Clustering

Suppose a data set D contains n objects. Partitional methods are used when a user wants to separate a data set using some center representative node to identify a user-defined set of k clusters. These clusters, or partitions, are then further examined using a similarity function [2], a function selected to determine "likeness" between intracluster points, to identify underlying attributes within the data set. Partitional methods usually choose a representative to serve as the cluster's main center point, or centroid. The partitional methods that will be discussed in this paper are k-means and k-medoids.

3.1.1 K-Means Algorithm

As specified in the name, the k-means partitioning algorithm investigates the means of clusters as the centroid. Using this clustering technique allows for "intracluster similarity to be high [while] intercluster similarity is low" [2]. In order to increase intracluster similarity we must attempt to minimize some criterion function. In k-means, this criterion function is the sum of the distances between our mean and every other point in the cluster.

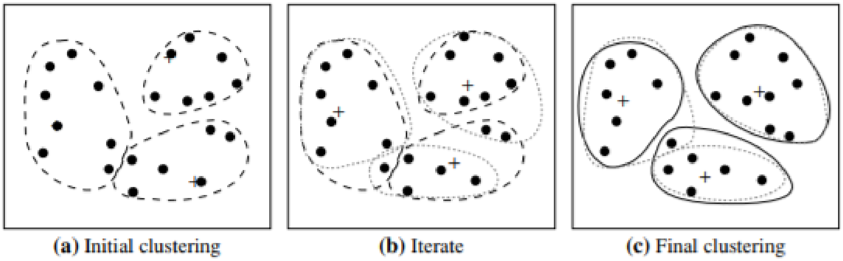


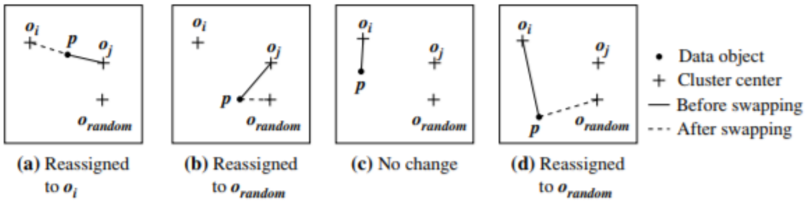
Figure 1: Illustrates k-means partitioning algorithm. (mean of each cluster denoted by +) [1]

Figure 1 depicts the k-means partitional algorithm. As shown in Figure 1a, k-means begins with random k centroids. These initial centroids are then used to determine the first k clusters of the algorithm by finding points within the minimal Euclidean distance from each given centroid. After these initial clusters are created, the algorithm iteratively evaluates each cluster. Figure 1b shows that each clusters' centroids have moved. With each iteration of k-means the algorithm calculates the mean of the points of the cluster and repositions the centroid of the cluster to this new mean. Following this calculation, all points are reevaluated using the criterion function and potentially assigned to new clusters. Once all centroids are fixed we are left with Figure 1c, the final clusterings.

While k-means can be used for many types of data sets, the k-means algorithm has a few potential flaws. The first flaw originates from our data set of interest. Supposed this data set contains 50 points. Within these 50 points there could exist 5 outliers. Since the mean is heavily influenced by outliers the clusters that are created may not accurately represent our data [1]. Furthermore, k-means can only be applied to a dataset that can produce a mean. For example, say database D only contains categorical data. Then k-means cannot be used since a mean cannot be calculated on categorical data. To solve some issues that can arise from the k-means algorithm, we next investigate the k-medoids partitional algorithm.

3.1.2 K-Medoids Algorithm

Similar to the k-means algorithm, k-medoids also selects a user-defined amount of clusters to partition the data set. The difference originates from the definition of the centroid. While k-means uses the mean of all points in a cluster, k-medoids makes the “most centered” point of the cluster as the centroid. By setting the most centered point as the centroid allows the clusters to potentially reflect the data more accurately [1]. The goal of this algorithm remains maximizing intracluster similarity by iteratively deciding the best k-medoids.



(a) Reassigned to o_i **(b)** Reassigned to o_{random} **(c)** No change **(d)** Reassigned to o_{random}

Figure 2: Illustrates four scenarios of k-medoids partitioning algorithm [1]

Similarly to k-means, k-medoids algorithms require a user-defined k to determine the amount of clusters to produce. K-medoids algorithms then randomly assign k points as the initial centroids to our clusters. Following these assignments, random points p and o_{random} are selected from our data. As shown in Figure 2, our goal now is to determine whether changing p's associated centroid to a new centroid would cause the quality of our clusters to increase. Given o_i,o_j, and o_{random} as our possible representative points, we compare p to each point. Figure 2a shows that the distance between o_i and p is less than that of o_j and p, so p's centroid is changed to o_i. Suppose p is closer to o_{random} than o_i

or o_j (Figure 2b & 2d), then o_{random} is defined as a new centroid for p and the connection to o_i, or o_j, is removed. The final possibility, shown in Figure 2c, is that no clusters benefit from p switching clusters, so the connection to centroid o_i remains.

While k-medoids and k-means both cluster data sets into k partitions, both methods rely on a user-defined k. This k can be calculated through many different type of heuristics, most common being cross validation. The cross validation method finds k-clusters by analyzing the cluster model created. Suppose we split data set D into k partitions and use $k-1$ partitions to create a cluster model. Once this model is complete, we can utilize the last cluster to assess the model we just created [1]. The major drawback to cross validation is needing to check all possible values of k until an optimal k is calculated.

The next type of clustering method investigated is Hierarchical clustering, where clusters are placed in a type of hierarchy, thus creating individual clusters.

3.2 Hierarchical Clustering

Hierarchical clustering methods examine and structure data into a type of hierarchy to help understand other properties about the data. For example, suppose our dataset of interest contains information about animals and their food chain relationship. This relationship serves as a hierarchy within our data which we can cluster using hierarchical techniques. In order to use hierarchical clustering techniques, we need to examine the two main ideologies of hierarchical clustering—agglomerative and divisive clustering.

3.2.1 Agglomerative and Divisive Hierarchical Clustering

Agglomerative and divisive clustering methods are two different techniques that hierarchical clustering methods use to structure data. These examples give an overview of hierarchical clustering, but serve as the backbone to all hierarchical clustering methods. To show the differences between the two, Figure 3 below shows the implementation of the basic hierarchical methods AGNES and DIANA.

The top of Figure 3 show the use of AGNES, the agglomerative hierarchical clustering method. The thought behind agglomerative clustering is starting all points as their own individual clusters. Then each cluster is grouped based on some criterion until we reach a user-specified k set of clusters or all objects are in the same cluster. Contrastingly, DIANA, our divisive clustering method, works in the opposite way. We first begin with all objects in the same cluster. We then divide the cluster into 2 separate clusters and continue this until we reach a user-specified k clusters or until every object represents its own cluster. While Figure 3 shows specifically how each method operates, we commonly use a tree-like structure called a dendrogram (Figure 4) to represent our data.

Similarly to Figure 3, Figure 4 shows how the set $\{a, b, c, d, e\}$ can be combined and divided based on similarities. For example, when using agglomerative

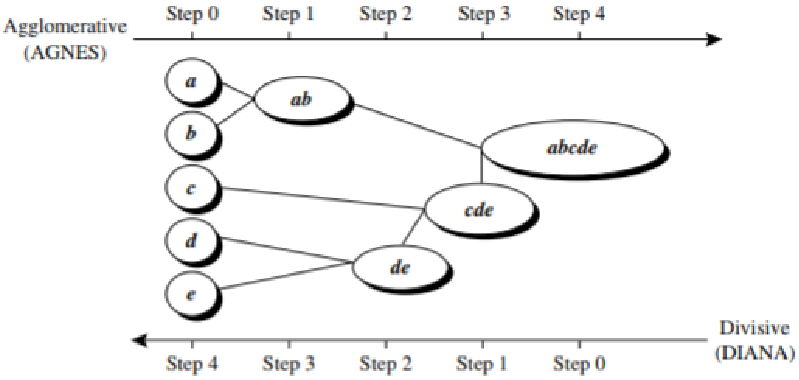


Figure 3: Comparison between agglomerative and divisive clustering techniques using the set of objects $\{a, b, c, d, e\}$ [1]

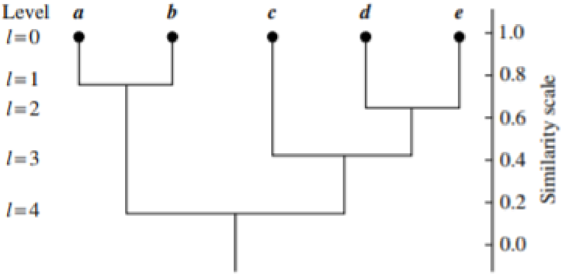


Figure 4: Dendrogram illustrating hierarchical clustering using the set of objects $\{a, b, c, d, e\}$ [1]

clustering, a and b are connected during the first iteration (Level $l = 1$ in Figure 4) due to their higher similarity (.8). Following down from the top of the dendrogram, c and d are connected next because c and d have the next highest similarity between the objects. After 4 iterations of this algorithm all objects are connected to the same cluster. Similarly, divisive clustering follows up the dendrogram to disconnect all objects from the same cluster.

While both methods above are used in many different hierarchical clustering methods, agglomerative clustering is preferred over divisive. An issue arises in divisive clustering when decisions must be made on how to divide the clusters. Since one incorrect division can affect all objects above in a dendrogram, agglomerative clustering is used more often. Since agglomerative clustering is preferred, we will investigate the agglomerative hierarchical clustering method called Chameleon.

3.2.2 Chameleon Hierarchical Clustering

The Chameleon hierarchical clustering technique combines dynamic modeling techniques with agglomerative clustering to identify similarity between pairs of clusters [1]. Similar to earlier methods discussed, Chameleon also determines similarities by a specific metrics—in Chameleon the metrics utilized are *interconnectivity* and *closeness* [1]. *Interconnectivity* is defined as how connected objects are in a cluster with other objects. For example, in Figure 5, graph A would have a higher interconnectivity than graph B due to the objects in graph A connecting to every other object. Furthermore, *closeness* is defined by how "close" objects are within a cluster. Graph A in Figure 5 would be considered to have a higher *closeness* than graph B due to the points having a closer Euclidean distance to one another.

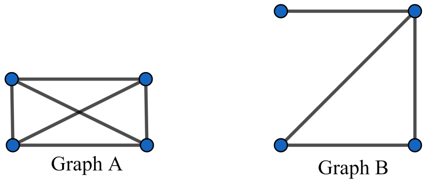


Figure 5: Graph A is considered to have higher interconnectivity and closeness than Graph B due to the connectedness of Graph A [1]

When compared to earlier clustering methods, Chameleon is vastly different when attempting to identify clusters. Chameleon begins by connecting all objects in the dataset together using a nearest-neighbor approach—where one begins with an arbitrary point and connects all closest points based on the distance function. This method then connects all points and assigns weights to edges based on each pair of points' similarity—which ultimately creates one cluster (Figure 6b).

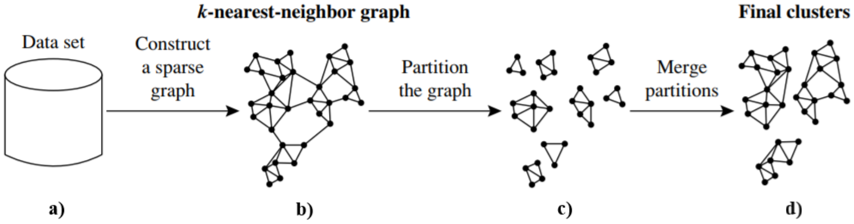


Figure 6: Adapted from Figure 10.10. Depicts steps of Chameleon clustering algorithm [1]

Once all objects are connected, this technique begins to analyze edge weights and removes edges with the smallest weights (Figure 6c), which leaves multiple subclusters. Finally, the Chameleon clustering algorithm combines these subclusters by comparing clusters' interconnectivity and closeness, and we are left with the final clusters in Figure 6.

The final type of clustering we will investigate in this paper is Density-based Clustering, where clustering techniques begin to analyze the density between objects, and resulting clusters in our data set.

3.3 Density-Based Clustering

Density-based clustering is used when trying to identify arbitrary shaped clusters within data [1]. In order to identify these arbitrarily shaped regions, we must now treat clusters as regions with a dense amount of points. Doing so allows us to identify these regions with ease. The density-based clustering methods that will be discussed are DBSCAN and OPTICS.

3.3.1 DBSCAN

Density-Based Spatial Clustering of Applications with Noise, or DBSCAN, is a type of density-based clustering algorithm that finds dense regions of points by analyzing ϵ-neighborhoods around each point. Given a user specified $\epsilon > 0$, an ϵ-neighborhood is all space that lies within a radius of ϵ around our point of interest. Furthermore, DBSCAN also utilizes a user-defined parameter s that represents a minimum amount of points that must be contained within an ϵ-neighborhood of our point of interest.

Due to the nature of density-based clustering, DBSCAN works differently than many of the algorithms discussed earlier. Suppose there exists n points in our data set. To start this method, DBSCAN first considers all points as "unvisited points" [1]. Next, this technique choses a random point p from our data set. Once this random point is selected, DBSCAN considers point p "visited" and analyzes the ϵ-neighborhood around point p. If this neighborhood contains at least s points, DBSCAN characterizes the ϵ-neighborhood around p as a cluster and further analyzes each point within the ϵ-neighborhood using the same technique. If the ϵ-neighborhood of p does not contain s points (i.e. point n in Figure 7), DBSCAN will consider point p as "noise" and pick another random point from our data set. In Figure 7, we see that eventually the path of ϵ-neighborhoods to point p stop. This is due to the user-defined constraint of s. Whenever a point is reached, for example point q in Figure 7, where the ϵ-neighborhood of a cluster point does not contain s points, the DBSCAN method will end that cluster and move on to another random point in our data set. This clustering method continues until all points are labeled "visited" [1].

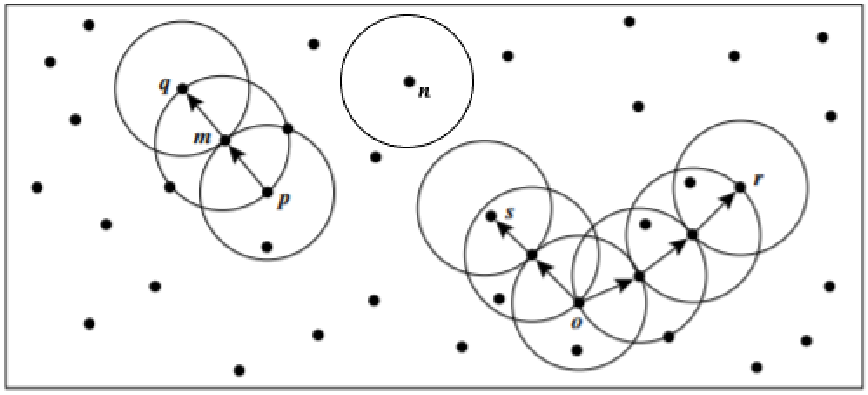
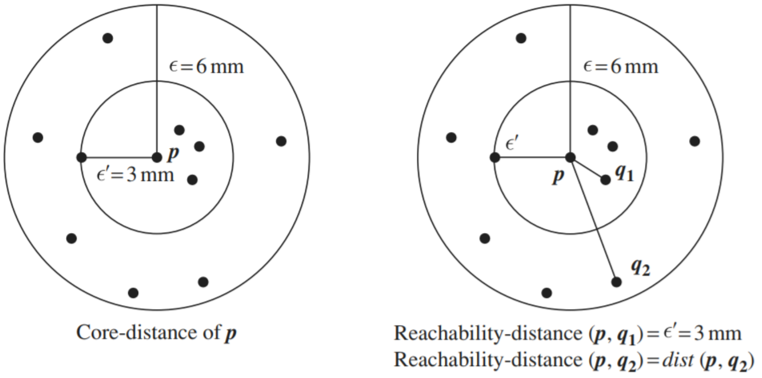


Figure 7: Adapted from Figure 10.14; depicts the DBSCAN algorithm [1]

3.3.2 OPTICS

The Ordering Points to Identify Clustering Structure algorithm, or OPTICS, takes a different approach to clustering objects. Unlike all other algorithms discussed previously, OPTICS does not require any global user-defined fields or return an explicit data clustering. Instead OPTICS returns a *cluster ordering* [1]. This ordering represents a "density-based clustering structure" of all objects in our data set. In order to understand the results of this algorithm, two new concepts must be introduced—core-distance and reachability-distance.



Core-distance of p

Reachability-distance $(p, q_1) = \epsilon' = 3$ mm
Reachability-distance $(p, q_2) = dist\ (p, q_2)$

Figure 8: Example of core-distance and reachability-distance in OPTICS [1]

As shown in Figure 8, the left most circle shows a point p with ϵ and ϵ' radii. The core-distance of p represents the smallest distance, denoted ϵ', required to reach a minimum amount of points m. In Figure 8 $m = 5$, so even though our original radius was $\epsilon = 6$mm the radius of $\epsilon' = 3$mm contains the m points. Furthermore, the right most circle in Figure 8 shows points q_1 and q_2 connected to point p. This distance, called the reachability-distance, is defined as the minimum required distance to reach a point within the ϵ-neighborhood of p. Since a point will either fall within outside of the ϵ' radius of p, the reachability-distance of any point is equivalent to $\max\{\textit{core-distance}(q_i), \textit{dist}(p, q_i)\}$, where q_i is the point of investigation, p is the clustering point, and $\textit{dist}(p, q_i)$ is the Euclidean distance between the points.

The OPTICS method begins by selecting some arbitrary point p, calculating the core-distance of p, and sets the reachability distance to "undefined" [1]. If point p does not have a core-distance (i.e. the ϵ-neighborhood of p does not contain a minimum points m), p is skipped for the rest of the algorithm. If p has a core-distance then each point within the ϵ-neighborhood is selected and assigned a reachability-distance. This process is then iterated until all points within our data set have been analyzed.

4 Ohio Youth Assessment System Data

As stated previously, the goal of our research is to investigate characteristics between singly arrested and rearrested children. Originally, we were to use data from the Allegheny Courthouse to research this topic, but after a few months our data request was denied. Following the denial, we were able to locate a study from the University of Cincinnati Center for Criminal Justice Research which describes a new assessment for juvenile delinquents living in Ohio. This assessment, Ohio Youth Assessment System (OYAS), was created to collect specific data to make better decisions with regards to juvenile justice. There are four main assessments in the OYAS. The first assessment is a staff-completed survey and a self-completed questionnaire when the child first enters the judicial system. The second assessment is a disposition questionnaire for youth who were placed on probation or received less than three months in a residential program. If a child received more than three months they would complete the residential questionnaire. Both assessments were completed just before release and have three parts: the first being a self-report questionnaire, second being a face-to-face interview, and the third being a file review of the youths official court record. Our data is based on the disposition (DIS) sample. The OYAS-DIS is used to assess youth at the time of disposition to determine the appropriate actions to be taken next [4].

4.1 Background of Significant Characteristics

As the OYAS-DIS was being created, the developers of this system sampled 594 children to investigate the advantages of alternative judicial decisions.

Throughout the OYAS-DIS, three types of surveys were used to assess the children, uncovering seven major characteristics found in most rearrested youth. Therefore, all questions in each survey were assigned a characteristic to keep track of how each child answers. Due to the scope of this research, some questions were deemed irrelevant, and therefore removed. The categories are identified and follow as such:

Juvenile Justice History (JJH) provides any information on the childs past involvement with the justice system including contact, adjudications and violations.

Substance Abuse, Mental Health and Personality (SAMHP) deals with drug, and alcohol usage with other high risk behaviors.

Peers and Social Support Network (PSSN) gains information on the type of people the child surrounds themself with, and how they interact with friends.

Values, Belief and Attitudes (VBA) represents the childs values beliefs and attitudes towards drugs, alcohol, criminal activity, and emotion towards others.

Pro Social Skill (PSS) questions provides information on how well the child can identify between good and bad behaviors and make good decisions based upon pros and cons.

Family and Living Arrangements (FLA) describes the childs family and living arrangements and how they view their family dynamics.

Education and Employment (EE) describes their relationship with authority such as teachers and bosses. It also gives insight to how they act under authority [4].

4.2 Simulation Creation

When creating the data, we needed to create a simulation to accurately represent the original data. Before data could be recreated, the questions used in the OYAS needed to be categorized and numerically represented. To do this, we assigned each question in the survey a characteristic value, and rescored the answers to either binary or ternary values. This allowed each question to be ranked on a yes or no basis, or a yes, no, indifferent basis. As each question is randomly answered, we also stored percentages listing the amount of certain types of answered questions. For example, if a child would answer yes for question 2, and this question was characterized as relating to Education, the education percent for that child would rise. Furthermore, this percentage also played a key role in determining how questions were answered. Suppose a child is struggling with their home-living situation and this child begins answering questions regarding this fact. It would be unlikely to find that a child states they have a difficult living situation, then answer no to the rest of our questions regarding this situation. Therefore, we use the percentages of questions answered to increase or decrease the likelihood of a question being answered.

Description of variables within data set are located in Appendix **A**.

5 Cluster Analysis of Simulation Data

For our research, cluster analysis was used to identify common characteristics between clusters of delinquent children. While clustering techniques are usually used with quantitative data, most of our variables are categorical in nature. In order to use cluster analysis on this data set, a numerical value needed to be calculated to represent the how cases differ from one another. This value was based on Hamming distance, a distance metric that sums the difference between each location in a vector. Hamming distance, H, is defined in general as:

$$H = \sum_{k=1}^{n} |x_k - y_k|$$

By using this metric, we then denote $D_{\cdot,m}$ as each column of the matrix D as:

$$D_{\cdot m} = \left(\sum_{k=1}^{258} |x_{jk} - x_{ik}| \colon 1 \leq i < j \right) \forall m = 1, 2, ..., 594$$

Finally, to achieve one value per case, we take the median of each column to achieve vector d, which is defined as follows:

$$d = \big(median \big\{ D_{\cdot m} \colon 1 \leq m \leq 594 \big\} \big)$$

Once the median Hamming distance is calculated, we can now continue with the use of cluster analysis. For the entirety of this research the variables of interest are age and median Hamming distance. These variables will be used to create clusters using several clustering algorithms. We will be investigating clusters through the distributions of ages, median Hamming distances, and most probable rearrest characteristic given a rearrest. These variables were chosen for investigation because prior background research suggested that age and cause of rearrest are the most significant variables within the data set, which allows the clustering algorithms to provide accurate results. Furthermore, we attempted to have each algorithm produce seven clusters due to the top seven characteristics identified in the OYAS report.

5.1 Partitional Clustering

5.1.1 K-Means Clustering

Through the use of the K-Means algorithm, we were able to create seven clusters of data. These clusters' median Hamming distance and age distributions are shown below.

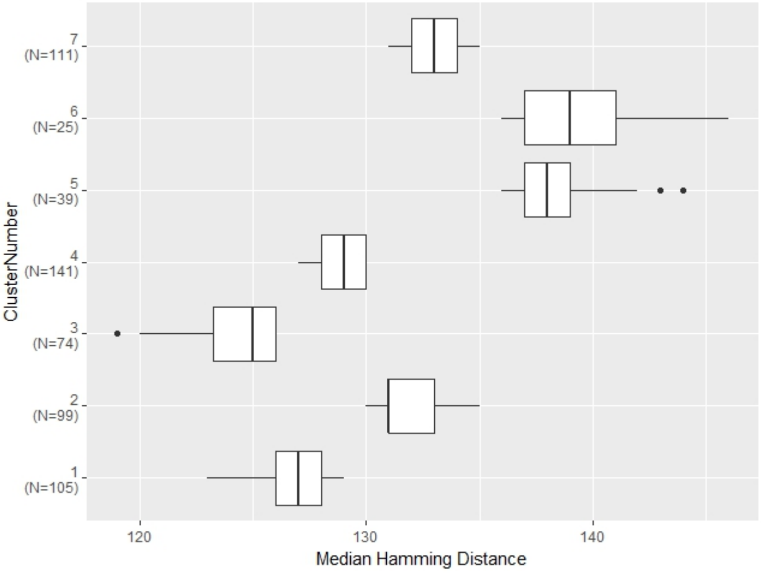


Figure 9: K-Means Median Hamming Distance Clustering

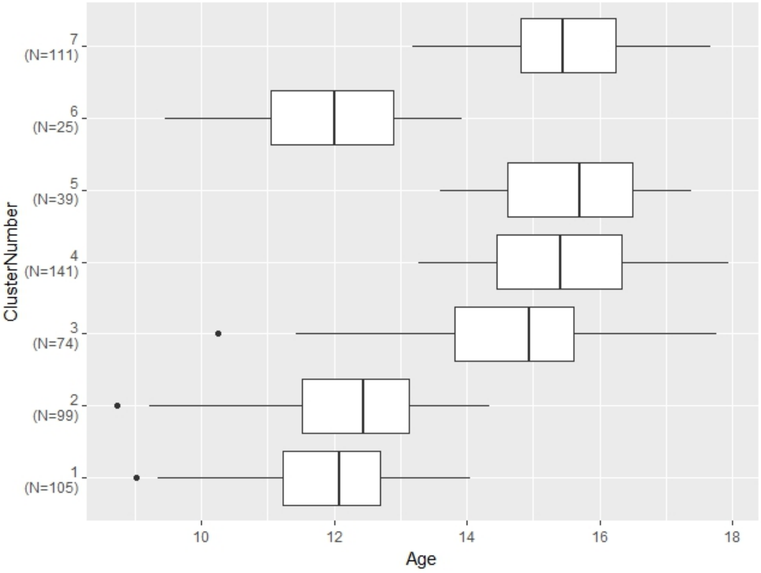


Figure 10: K-Means Age Clustering

Together these box-plots shown above illustrate the how median Hamming distance and age varies by cluster. For example, Figures 9 and 10 show that cluster 1 contains median Hamming distances that range from 123 to 128 and ages that range from 9 to 14. Now, when investigating these visualizations further, we find that most median Hamming distance distributions are not similar, except for clusters 5 and 6. These clusters are the only clusters that contain similar ranges of distances. When moving to their respective age distributions we find that the ages vastly differ between the two—cluster 5 ranging from 13.59 to 17.38 and cluster 6 ranging from 9.46 to 13.93. Due to these differences, the overall rearrest count was examined.

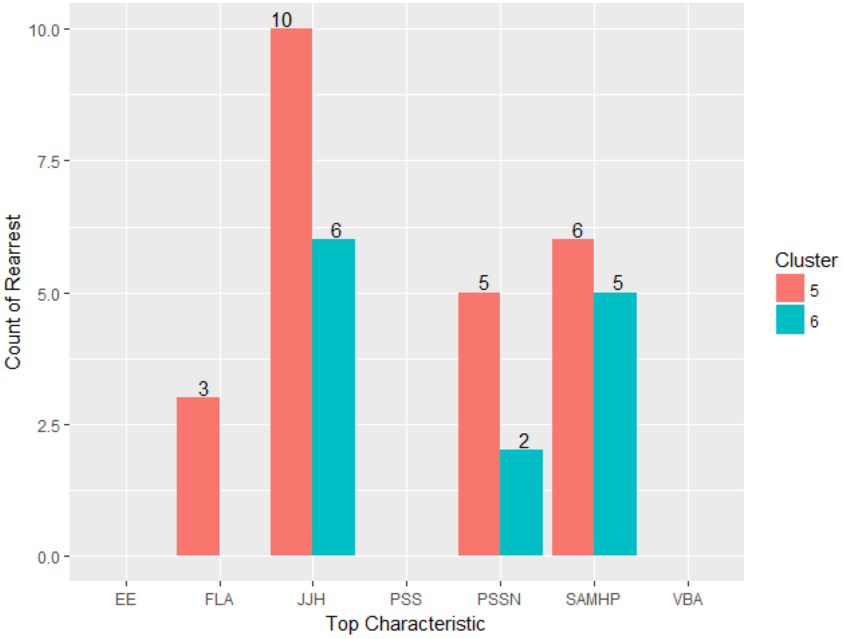


Figure 11: Clusters 5 and 6 Rearrest Counts Through K-Means Algorithm

Figure 11 shows the rearrest count for each of clusters 5 and 6. We see that cluster 5 has a larger amount of re-offending children than that of cluster 6. Specifically, the largest difference exists in the *JJH* characteristic. When taking cluster age into consideration, we find this result to be expected. Since cluster 5 age range includes that of pubescent ages for male and female, and children are more willing to rebel during pubescent ages, we would assume that the increase of rearrest, specifically with *JJH* characteristic, is due to the age difference between the two clusters.

5.1.2 K-Medoids Clustering

Through the use of the K-Medoids algorithm, we were also able to create seven clusters of data. These clusters' median Hamming distance and age distributions are shown below.

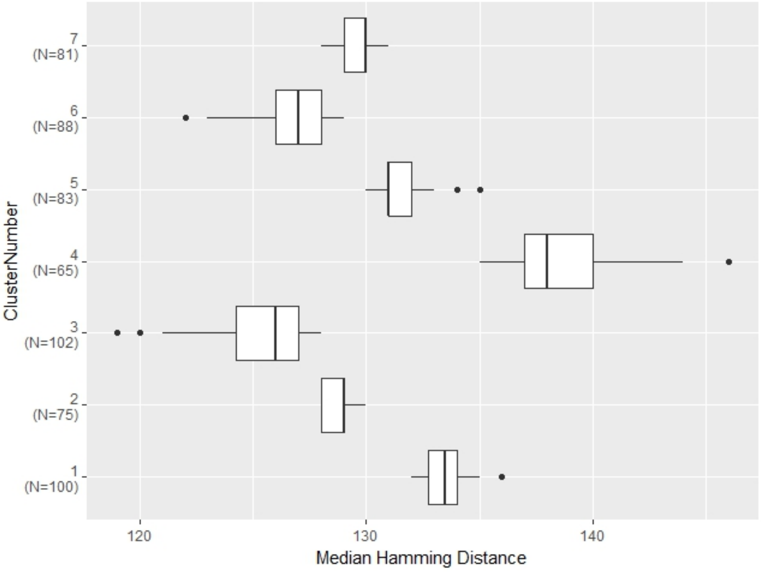


Figure 12: K-Medoids Median Hamming Distance Clustering

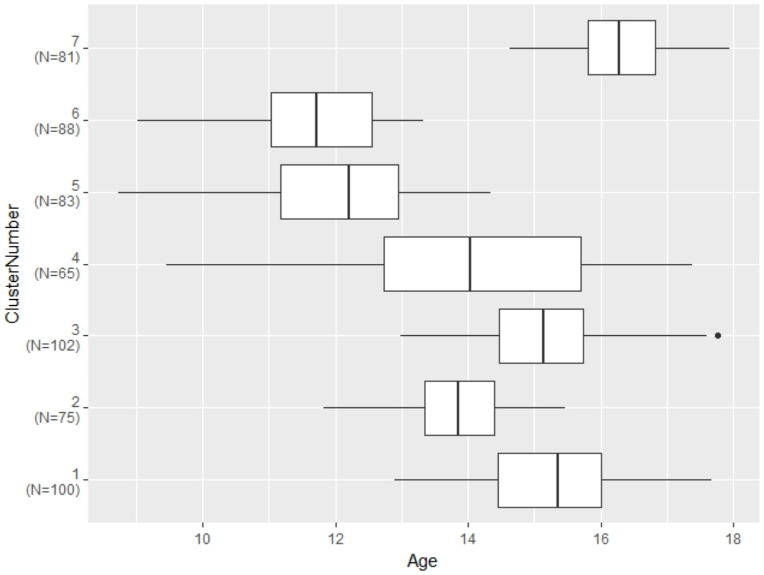


Figure 13: K-Medoids Age Clustering

Together, Figures 12 and 13 show the median Hamming distance and age distributions per cluster produced by the k-medoids algorithm. While the cluster distributions in Figure 12 have few similarities, we find in Figure 13 clusters 1 and 3 share very similar distributions. We would expect children of similar ages, especially pubescent ages, to answer similarly, but this is not the case. To further investigate this difference, Figure 14 shows the rearrest count for each cluster produced by the k-medoids algorithm.

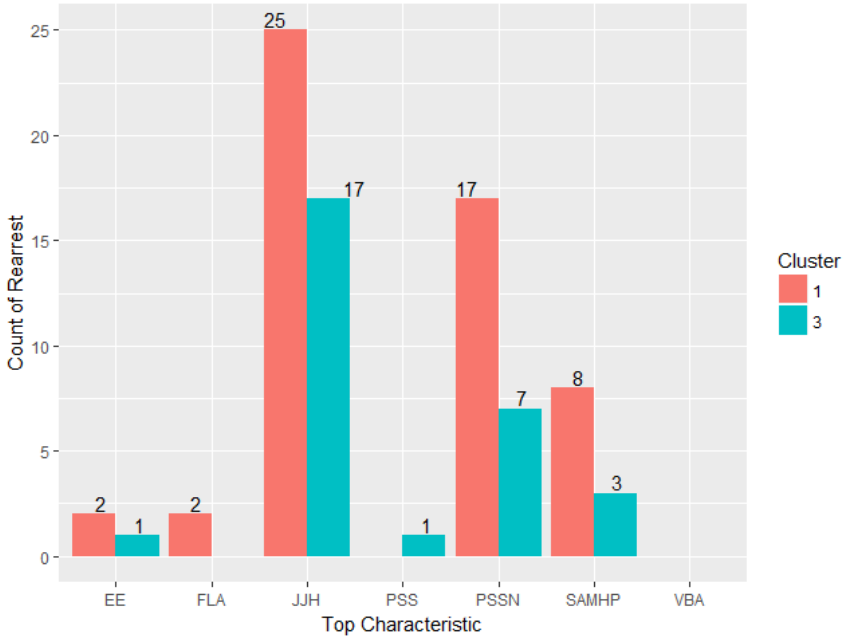


Figure 14: Clusters 1 and 3 Rearrest Counts Through K-Medoids Algorithm

We see in Figure 14 that cluster 1 has a much higher rearrest count than cluster 3. Since cluster 1 has the higher distribution of median Hamming distance, we would expect cluster 1 to have a much higher rearrest count than that of cluster 3. Similar to cluster 5 from the k-means algorithm, we also see that more children of pubescent ages are getting rearrested in regards to *JJH* characteristic. The occurrence of these similarities only further supports evidence that pubescent-aged children are more likely to be rearrested due to their judicial history.

5.2 Hierarchical Clustering

The hierarchical clustering method used in this research is called complete-linkage clustering—an agglomerative technique where all points begin as their own cluster and begin linking with other data points.

5.2.1 Complete-Linkage Hierarchical Clustering

Similar to the partitional clustering algorithms, the complete-linkage algorithm was able to create seven clusters of data. The median Hamming distance and age distributions are shown below.

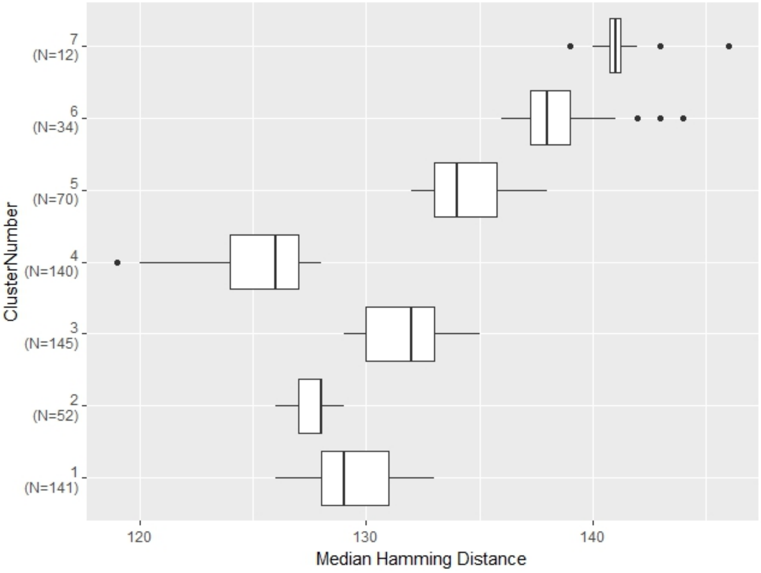


Figure 15: Complete-Linkage Median Hamming Distance Clustering

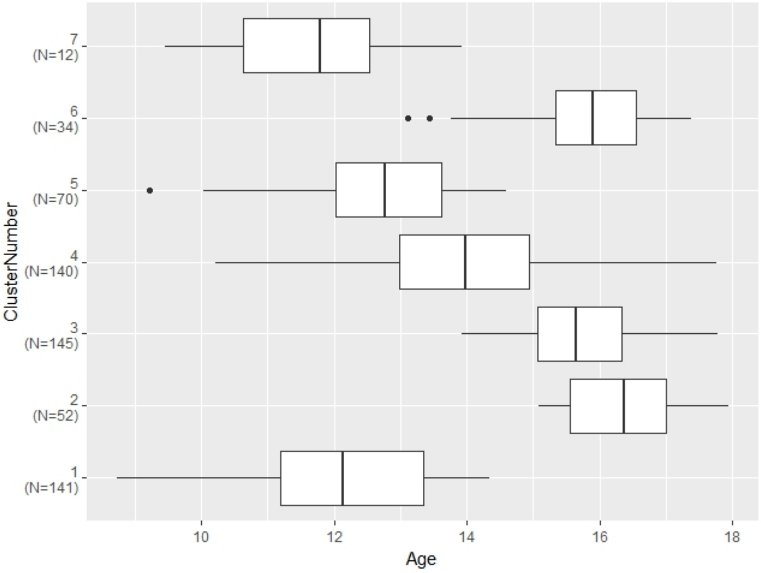


Figure 16: Complete-Linkage Age Clustering

Figures 15 and 16 show the distributions of median Hamming distance and age, respectively. While Figure 15 shows no similarity between median Hamming distance distributions, Figure 16 shows most clusters, specifically clusters 3 and 6, have high similarity in their age distributions. Comparable to clusters investigated from the k-medoids algorithm, clusters 3 and 6 found here have similar age distributions, but different median Hamming distance distributions. Furthermore, we see that cluster 3 contains over 100 more children than that of cluster 6. In order to examine this further, we look at the rearrest characteristic counts of these two clusters.

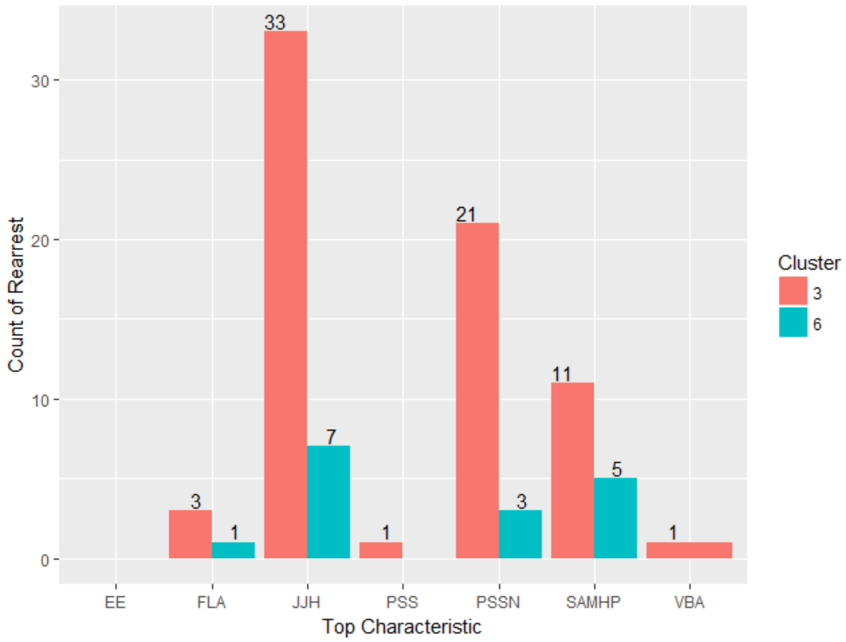


Figure 17: Clusters 3 and 6 Rearrest Counts Through Complete-Linkage Algorithm

As expected, Figure 17 shows that characteristic *JJH* has the highest count for rearrest between both clusters. However, while these clusters have vastly difference sizes, the relative rearrest count is proportional to each cluster. For example, there were 33 children in cluster 3 and 7 in cluster 6 that were rearrested relating to their judicial history. When examining these numbers compared to their overall size, we see that $\frac{33}{145} = .22$ and $\frac{7}{34} = .21$. Since these clusters have differing median Hamming distributions and sizes but similar relative rearrest percentages, we are likely to assume that age continues to play a vital role in determining child rearrest.

6 Conclusion

The research conducted in this paper, while it may have not presented new findings, does continue to verify facts we know about juvenile offenders. Firstly, we know that age plays a major factor in crime. As children age, the more independent they wish to become, but early independence can be dangerous for children without a support system. A lack of family or friend support can cause children to make poor decisions that can eventually cause issues with the law. Furthermore, we see that the judicial history characteristic appears more in clusters of children at older ages. This makes sense because if children are arrested once there should be a higher probability that the same child is arrested again in relation to the prior offense (given the child was not placed in a reform program).

Although this research did not shed light to new findings in children recidivism, we do hope that the judicial system as a whole can make more informed decisions regarding rearrested children.

A Simulated Data Variables

- *ID*: ID number for each child in the simulated data
- *Males*: Beta distribution value of male age if selected
- *Females*: Beta distribution value of female age if selected
- *Gender*: 0 for male, 1 for female
- *Age*: Age of child
- *Q1-Q206*: Answer for each question, values fall between 0 and 2
- *RA*: 0 for no rearrest, 1 for rearrest
- *TopCharacteristic*: Representing question type most answered by the child

Simulated dataset and commented code can be found at
https://github.com/SamHockenberry15/OYAS_Research

References

[1] Han, Jiawei, et al. *Data Mining: Concepts and Techniques.* Elsevier/Morgan Kaufmann, 2012.

[2] Han, Jiawei, et al. *Data Mining: Concepts and Techniques.* Elsevier/Morgan Kaufmann, 2001.

[3] Kassambara, Alboukadel. *Cluster Validation Essentials.* Datanovia, www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/#average-silhouette-method.

[4] Latessa, Edward, et al. (2009). *The Ohio Youth Assessment System: Final Report.* Unpublished manuscript.

[5] Tan, Pang-Ning, et al. *Introduction to Data Mining.* Pearson, 2019.