Territorial Analysis for Ratemaking

by

Philip Begher, Dario Biasini, Filip Branitchev, David Graham, Erik McCracken, Rachel Rogers and Alex Takacs

Department of Statistics and Applied Probability

University of California at Santa Barbara

Abstract

This report discusses clustering methods and their application to liability and collision insurance ratemaking for the state of Nevada. We first explore a variety of distance formulas based on exposure-adjusted pure premium and then proceed to describe the basics of the three clustering methods applied to the data. To conclude we introduce methods to validate and compare clustering results.

March 31, 2011, version 1.2

Key Words and Phrases. Data mining, territorial analysis, clustering, ratemaking, validation, K-Means, K-Medoids, hierarchical clustering, fuzzy clustering, R, PAM, CLARA, agglomerative clustering, FANNY.
# Table of Contents

1. INTRODUCTION  
2. DATA SETS  
  2.1 DEFINING DATA SETS  
  2.1.1 NOTATION  
  2.1.2 SUMMARY OF DATA SETS  
  2.1.3 DATA DISTRIBUTION  
  2.1.4 COMPARING SET 1 TO SET 2  
  2.1.5 DENSITY CONSIDERATIONS  
  2.1.6 INCONSISTENT OBJECTS  
  2.2 PREPARING DATA  
  2.2.1 ADJUSTING PURE PREMIUM CREDIBILITY  
  2.2.2 NORMALIZING DATA  
  2.2.3 ADJUSTED PURE PREMIUM-WEIGHTED EUCLIDEAN DISTANCE  
  2.2.4 NEIGHBORHOOD-BASED DISTANCE  
  2.3 DATA EVALUATION  
3. CLUSTER ANALYSIS: BASIC CONCEPTS AND ALGORITHMS  
  3.1 K-MEANS/PAM/CLARA  
  3.2 HIERARCHICAL CLUSTERING  
  3.3 FUZZY CLUSTERING  
4. COMPARING CLUSTERING TECHNIQUES  
  4.1 VALIDATION METHODS  
  4.1.1 MEAN SQUARED ERROR FOR PURPOSES OF COMPARISON  
  4.1.2 SILHOUETTES AND SILHOUETTE WIDTH VALUES  
  4.1.3 WITHIN PROXIMITY – BETWEEN PROXIMITY RATIOS  
  4.2 COMPARING CLUSTERING ALGORITHMS  
5. RESULTS  
  5.1 SELECTING A DISTANCE FUNCTION  
  5.2 OPTIMAL NUMBER OF CLUSTERS  
  5.3 BEST CLUSTERING TECHNIQUE  
  5.4 BEST DISTANCE FUNCTION  
  5.5 OVERALL RECOMMENDATION  
  5.6 FUTURE WORK  
6. REFERENCES
1. Introduction

Data mining provides a collection of tools increasingly recognized for their application to ratemaking. According to Jennings (2008), “Current actuarial ratemaking methodologies for the pricing of personal lines of automobile and homeowners insurance in the United States include a geographical component.” It is common to use zip-code boundaries to incorporate the geographic component. By grouping these regions territories may be defined that share similar rating traits. Cluster analysis is a data mining technique particularly well suited for grouping regions sharing similar definable traits, and thus shows potential as a ratemaking tool.

The goal of this project was, for rating purposes, to find the optimal way to divide a state into geographic regions. For this study, we use data sets containing two different periods of information on car insurance for zip-code defined regions in Nevada, and apply clustering algorithms based on K-Means, hierarchical, and fuzzy clustering methodology. We then compare the results obtained by different methods and interpret them through validation techniques.

Tan, Kumar, and Steinbach (2006), who served as our main reference for learning clustering methodology, provide an informative introduction to the topic. Cluster analysis is a data mining technique that partitions data into meaningful subgroups based only on information found in the data describing the objects and their relationships. The goal is to group data in such a way that objects within a cluster are similar in some sense to one another and different from objects within other groups. The greater the similarity within groups and dissimilarity with other groups, the better the clustering.

We utilized three clustering methods to identify contiguous geographic areas within the state of Nevada with similar ratemaking parameters. The creation of a proximity matrix is the first step to cluster data. The proximity matrix consists of the pairwise distances between all objects in a data set, computed by a distance function. The distance function will affect the output of the clustering, thus one must choose with care. The decisions should support the overall goals of the project.

Each clustering algorithm exhibits particular strengths and weaknesses. K-Means, for example, is suited for grouping compact clusters; however, outliers greatly affect its output. Hierarchical clustering provides a visual comparison of different numbers of clusters, but may be unsuited for large data sets. Interpreting the results and comparing the results of different algorithms is the most important part of the clustering analysis.

We chose the statistics software package R for our exploration of cluster analysis techniques. R includes functions for many clustering algorithms, provides widely available documentation, supports multiple platforms, and is free. For this analysis, we used the following R libraries: cluster, maps, maptools, fpc. R may be downloaded at: http://www.r-project.org/
2. Data Sets

This section introduces the data sets utilized in this exploration of cluster analysis. We discuss the characteristics of each object in the data set and notation used for the rest of the report, before turning to the decisions made through the creation of the distance function used for our proximity matrix. In our clustering process, we will look at information for Set 1 and Set 2 separately and then compare one to the other.

2.1 Defining Data Sets

2.1.1 Notation

Table 2-1 summarizes the main symbols used throughout the report. These symbols will use appropriate indices. For example, $e_{i,2}$ denotes earned exposure of the object $i$ in a data set corresponding to Set 2 such as CNVS2 (see Table 2-2).

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Number of objects in the data set</td>
</tr>
<tr>
<td>$n_k$</td>
<td>Number of objects in cluster $k$</td>
</tr>
<tr>
<td>$X$</td>
<td>Latitude</td>
</tr>
<tr>
<td>$Y$</td>
<td>Longitude</td>
</tr>
<tr>
<td>$\bar{p}$</td>
<td>Pure premium</td>
</tr>
<tr>
<td>$e$</td>
<td>Earned exposure</td>
</tr>
<tr>
<td>$\hat{p}$</td>
<td>Adjusted pure premium</td>
</tr>
<tr>
<td>$x$</td>
<td>Normalized latitude</td>
</tr>
<tr>
<td>$y$</td>
<td>Normalized longitude</td>
</tr>
<tr>
<td>$p$</td>
<td>Normalized adjusted pure premium</td>
</tr>
<tr>
<td>$o$</td>
<td>Object</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_x$</td>
<td>Standard deviation of $x$</td>
</tr>
<tr>
<td>$\bar{p}$</td>
<td>Mean pure premium</td>
</tr>
<tr>
<td>$E$</td>
<td>Exposure weight</td>
</tr>
<tr>
<td>$r$</td>
<td>Adjusted pure premium weight</td>
</tr>
<tr>
<td>$dist$</td>
<td>Distance between objects</td>
</tr>
<tr>
<td>$K$</td>
<td>Number of clusters</td>
</tr>
<tr>
<td>$c_k$</td>
<td>Cluster $k$</td>
</tr>
<tr>
<td>$c$</td>
<td>Centroid</td>
</tr>
<tr>
<td>$m$</td>
<td>Medoid</td>
</tr>
<tr>
<td>$w$</td>
<td>Membership coefficients weight</td>
</tr>
<tr>
<td>$q$</td>
<td>Fuzzifier</td>
</tr>
</tbody>
</table>

2.1.2 Summary of Data Sets

Our analysis began with four data sets consisting of two sets of years for Liability and Collision insurance in Nevada. We will refer to these data sets as CNVS1, CNVS2, LNVS1, and LNVS2. Table 2-2 summarizes the data sets.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Objects</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNVS1</td>
<td>Nevada Collision Insurance Set 1</td>
<td>148</td>
<td>2006-2008</td>
</tr>
<tr>
<td>CNVS2</td>
<td>Nevada Collision Insurance Set 2</td>
<td>146</td>
<td>2009-2010</td>
</tr>
<tr>
<td>LNVS1</td>
<td>Nevada Liability Insurance Set 1</td>
<td>148</td>
<td>2006-2008</td>
</tr>
<tr>
<td>LNVS2</td>
<td>Nevada Liability Insurance Set 2</td>
<td>148</td>
<td>2009-2010</td>
</tr>
</tbody>
</table>
The objects within the data sets contain information about set, zip code, earned exposure, pure premium, latitude, and longitude. The latitude and longitude values represent the center of each corresponding zip code region. Earned exposure represents the individual coverage in a zip code region for the given time span. Pure premium represents the average residual premium after removal of individual-policy factors. Table 2-3 presents a sample of the raw data, and Table 2-4 key statistics of attributes we will use for our computations.

<table>
<thead>
<tr>
<th>Set</th>
<th>Zip</th>
<th>Earned Exposure</th>
<th>Pure Premium</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89001</td>
<td>72.48</td>
<td>160.98</td>
<td>-115.49</td>
<td>37.36</td>
</tr>
<tr>
<td>1</td>
<td>89703</td>
<td>1514.02</td>
<td>161.89</td>
<td>-119.87</td>
<td>39.15</td>
</tr>
<tr>
<td>1</td>
<td>89406</td>
<td>1160.40</td>
<td>166.49</td>
<td>-118.33</td>
<td>39.58</td>
</tr>
<tr>
<td>1</td>
<td>89705</td>
<td>841.99</td>
<td>166.96</td>
<td>-119.83</td>
<td>39.09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Attribute</th>
<th>Max</th>
<th>Min</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNVS1</td>
<td>Pure Premium</td>
<td>4495.71</td>
<td>0.00</td>
<td>248.29</td>
<td>206.14</td>
</tr>
<tr>
<td></td>
<td>Earned Exposure</td>
<td>6911.43</td>
<td>0.70</td>
<td>1605.40</td>
<td>1129.02</td>
</tr>
<tr>
<td>LNVS2</td>
<td>Pure Premium</td>
<td>2300.73</td>
<td>0.00</td>
<td>221.02</td>
<td>183.69</td>
</tr>
<tr>
<td></td>
<td>Earned Exposure</td>
<td>4037.73</td>
<td>0.58</td>
<td>907.27</td>
<td>662.36</td>
</tr>
<tr>
<td>CNVS1</td>
<td>Pure Premium</td>
<td>2623.15</td>
<td>0.00</td>
<td>238.76</td>
<td>211.31</td>
</tr>
<tr>
<td></td>
<td>Earned Exposure</td>
<td>5894.94</td>
<td>0.70</td>
<td>1277.54</td>
<td>884.28</td>
</tr>
<tr>
<td>CNVS2</td>
<td>Pure Premium</td>
<td>2154.65</td>
<td>0.00</td>
<td>193.69</td>
<td>189.70</td>
</tr>
<tr>
<td></td>
<td>Earned Exposure</td>
<td>3416.40</td>
<td>0.33</td>
<td>721.71</td>
<td>492.54</td>
</tr>
</tbody>
</table>

2.1.3 Data Distribution

The objects contained in each data set are densely distributed at two locations, the Las Vegas, and Lake Tahoe regions, and less densely so throughout the rural portions of the state. For example, the CNVS1 data set is comprised of 57 Las Vegas objects, 45 Lake Tahoe objects, and 44 throughout the rest of the state. Figure 2-1 displays the distribution of the data within the CNVS1, with county lines displayed.
2.1.4 Comparing Set 1 to Set 2

Figure 2-2 plots the correlation between pure premium and exposure between objects contained in LNVS1 and LNVS2. We observed a strong correlation for exposure, and a weak correlation for pure premium. The weak correlation between pure premia made predictions for Set 2 based on Set 1 challenging.

2.1.5 Density Considerations

The distribution of the data may affect the quality of the clusters produced. In particular, differences in density between different regions may result in poor clusters for some clustering methods. As seen in Figure 2-1, more than one-third of the objects in the data set are contained...
within the Las Vegas region, 57 objects to be exact. The Lake Tahoe region is also very dense when compared to the remainder of the state. An option worth considering is clustering regions of similar density separately from one another. For example, one of the methods we explored was separating the Las Vegas objects from the rest of the data set between the longitude boundaries -115.40°, -114.91°, and latitude boundaries 35.95°, and 36.36°.

In later sections of this report, we explore clustering regions of different density separately. LV denotes Las Vegas data and NLV for all remaining data following the removal of Las Vegas. For example, LLVS2 would be Las Vegas Liability Insurance Set 2, and CNLVS1 would be No Las Vegas Collision Insurance Set 1. Table 2-3 defines these subsets.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Objects</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLVS1</td>
<td>Las Vegas Collision Insurance Set 1</td>
<td>57</td>
<td>2006-2008</td>
</tr>
<tr>
<td>CLVS2</td>
<td>Las Vegas Collision Insurance Set 2</td>
<td>57</td>
<td>2009-2010</td>
</tr>
<tr>
<td>LLVS1</td>
<td>Las Vegas Liability Insurance Set 1</td>
<td>57</td>
<td>2006-2008</td>
</tr>
<tr>
<td>LLVS2</td>
<td>Las Vegas Liability Insurance Set 2</td>
<td>57</td>
<td>2009-2010</td>
</tr>
<tr>
<td>CNLVS1</td>
<td>No Las Vegas Collision Insurance Set 1</td>
<td>91</td>
<td>2006-2008</td>
</tr>
<tr>
<td>CNLVS2</td>
<td>No Las Vegas Collision Insurance Set 2</td>
<td>89</td>
<td>2009-2010</td>
</tr>
<tr>
<td>LNLVS1</td>
<td>No Las Vegas Liability Insurance Set 1</td>
<td>91</td>
<td>2006-2008</td>
</tr>
<tr>
<td>LNLVS2</td>
<td>No Las Vegas Liability Insurance Set 2</td>
<td>91</td>
<td>2009-2010</td>
</tr>
</tbody>
</table>

2.1.6 Inconsistent Objects

Before we could make comparisons between Set 1 and Set 2, it was necessary to remove inconsistent objects. Specifically, a one-to-one match of zip codes was not present between the Set 1 and Set 2 data sets for either Liability or Collision. We removed inconsistent zip codes from each set to allow for comparison. This required the removal of 3 objects from CNVS1, 1 from CNVS2, 2 from LNVS1, and 2 from LNVS2. This change resulted in the clustering of 144 objects for each of these data sets.

2.2 Preparing Data

Our goal was to build territories based on geographic risk factors. We were primarily concerned with such indicators of risk as pure premium, latitude, and longitude. Based on the information observed in the data sets we first adjust objects with low exposure for credibility, standardize the data, and then select a distance formula to construct our proximity matrix.

2.2.1 Adjusting Pure Premium Credibility

As seen in Table 2-4, objects exist in the data set with low exposure values. Their respective pure premia are unlikely to provide credible information, as we considered exposure to be an indicator of the degree of confidence to which pure premium accurately describes the risk of a data object.
To account for objects with low exposure, we applied the square root rule for partial credibility to adjust the pure premium:

\[
\hat{p}_i = \min\left(\sqrt{\frac{e}{E}}, 1\right) \bar{p}_i + \left[1 - \min\left(\sqrt{\frac{e}{E}}, 1\right)\right] \bar{p},
\]  

(2.1)

where \(e\) is exposure, \(E\) is the minimum exposure that is assigned full credibility, \(\hat{p}_i\) is pure premium for object \(i\), and \(\bar{p}\) is the statewide mean average pure premium.

This formula adjusts pure premiums with low exposure values by taking its exposure-weighted average with the total average pure premium. We selected \(E = 1000\) based on the range seen in Table 2-4. Increasing the value of \(E\) will increase the amount of objects adjusted with the total average pure premium and vice versa.

### 2.2.2 Normalizing Data

We then normalize the latitude, longitude, and adjusted pure premium elements so that the scale of each value is comparable. The latitude, longitude, exposure, and adjusted pure premium all have different units of measurement, which cause serious problems in attempting to infer any sort of results. Therefore, the following three equations normalized the respective attributes of object \(i\):

\[
x_i = \frac{x_i - \bar{X}}{\sigma_x}, \quad y_i = \frac{y_i - \bar{Y}}{\sigma_y}, \quad p_i = \frac{\hat{p}_i - \bar{\hat{p}}}{\sigma_{\hat{p}}},
\]  

(2.2)

where \(X_i, Y_i,\) and \(\hat{p}_i\) are the latitude, longitude, and adjusted pure premium for their respective objects, \(\bar{X}, \bar{Y},\) and \(\bar{\hat{p}}\) are the statewide mean average latitude, longitude, and adjusted pure premium, and \(\sigma_x, \sigma_y,\) and \(\sigma_{\hat{p}}\) are the statewide standard deviation for latitude, longitude, and adjusted pure premium.

Each object \(o_i = (x_i, y_i, p_i, e_i)\) represents a vector containing these three normalized latitude, longitude, and adjusted pure premium values.

### 2.2.3 Adjusted Pure Premium-Weighted Euclidean Distance

Selection of a distance function to create a proximity matrix is the final step before clustering. Adjusted premium-weighted Euclidean distance is one possibility, although clustering algorithms accept other distance measures as well. Note that the measure uses adjusted pure premium and not the pure premium. The modified Euclidean distance formula used is

\[
dist(o_i, o_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + r(p_i - p_j)^2}.
\]  

(2.3)

where \(o_i = (x_i, y_i, p_i, e_i)\) are the objects for comparison, \(x_i, y_i,\) and \(p_i\) are normalized latitude, longitude, and adjusted pure premium values, and \(r\) is the weight given to adjusted pure premium.
In this study we have selected $r = 1.5$ to give an additional 50% weight to the adjusted pure premium. We wish to give greater emphasis to pure premium over geographic location in our clusters. The weight given to adjusted pure premium, $r$, controls a trade-off between geographically contiguous clusters and a strong clustering of risk factors. Weighting latitude, longitude, and adjusted pure premium equally with $r = 1.0$ results in compact contiguous clusters. Increasing $r$ places greater priority on adjusted pure premium, where similar values may be non-contiguous.

### 2.2.4 Neighborhood-Based Distance

To account for varying geographic density, we experimented with neighborhood-based distances as well. There are many definitions of neighborhoods and after trying out different approaches, we considered the following algorithm.

1. For every object $o_i$ in the data set:
2. Assign $o_i$ to neighborhood $N_0$.
3. Assign the 4 objects closest geographically to neighborhood $N_1$ (neighborhood-distance 1).
4. We now have two neighborhoods, $N_0$ and $N_1$ containing a total of 5 objects.
5. Find the closest object outside of $N_0$ and $N_1$ to each of the objects $o_j$ in $N_0$ and $N_1$. Assign these to neighborhood $N_2$ (neighborhood-distance 2).
6. Repeat until all objects $o_j$ are assigned to some neighborhood.
7. Let $dneigh(o_i, o_j)$ be the index of the neighborhood to which $o_j$ belongs.

The algorithm produces $dneigh(o_i, o_j)$ which we blended using weight $z$ with the pure premium-weighted Euclidean distance formula utilized above to create $d_{final}$:

\[
d_{final}(o_i, o_j) = \sqrt{dist(o_i, o_j)^2 + z \cdot dneigh(o_i, o_j)^2}.
\] (2.4)

### 2.3 Data Evaluation

After the division of the data set into territories, one way to evaluate the clustering is through an adjustment factor. Adjustment factors provide a quantitative value that describes the difference in objects. Section 4 of this report utilizes this value to make predictions for future sets of years based on all Set 1 data sets. The adjustment factor corresponding to $k^{th}$ cluster for data of Set 1 is:

\[
A_k^1 = \frac{\sum_{o_i \in C_k^1} \hat{p}_i e_i}{\sum_{i=1}^n \bar{p}_1 e_i},
\] (2.5)

where $o_i$ is an element of $C_k^1$, the $k^{th}$ cluster of Set 1, $\hat{p}_i$ and $e_i$ are the raw pure premium and exposure of object $i$ from set 1, and $\bar{p}_1$ is the average of all raw premia in Set 1.
3. Cluster Analysis: Basic Concepts and Algorithms

Cluster analysis partitions data into meaningful subgroups, known as clusters, when no information other than the observed value is available. Clustering uses similarities between objects based on characteristics found in the data. The proximity matrix measures the similarity or closeness of objects and therefore depends strongly on a choice of the distance function, as discussed in Section 2.

3.1 K-Means/PAM/CLARA

The K-means method is one of the more simple and intuitive ways to cluster objects. It partitions all objects (data points) in a data set into non-overlapping subsets (clusters) so that each object belongs to exactly one cluster. Each cluster contains a center point known as a centroid to assign points with close proximity. The K-Means algorithm, following Tan, et al. (2006), p. 497, is as follows:

1: Select $K$ objects as initial centroids.
2: repeat
   A: Form $K$ clusters by assigning each object to its closest centroid
   B: Recompute the centroid of each cluster.
   until Centroids do not change.

At Step 1, the algorithm randomly selects $K$ objects from the data set to serve as initial centroids; different methods of choosing initial centroids are also used. Each repetition of Step 2 assigns every object to a cluster with the nearest centroid, and recomputes the centroid based on the new clustering. The steps repeat until the algorithm produces the best centroid for each cluster. In K-Means algorithms, the centroid is the mean of the objects in the cluster.

Figure 3-1: Example of Choosing Centroids
Note that the cluster mean is the best centroid in the sense that it minimizes the sum of squared error (SSE) of the cluster. As usual, we interpret SSE as a measure of the squared distance from each point to the centroid for the associated cluster, written as

$$SSE = \sum_{k=1}^{K} \sum_{o_i \in C_k} dist(o_i, c_k)^2, \quad (3.1)$$

where $K$ is the predetermined number of clusters, and the second sum runs over $n_k$ objects $o_i$ in cluster $C_k$ with centroid $c_k$.

Although the K-Means algorithm may generate good results by creating well-defined clusters, it is extremely vulnerable to outliers, a common feature for mean-based statistical estimation methods. Also, because K-Means uses centroids that typically do not correspond to actual objects, the algorithm requires recalculations of proximity for each repetition of Step 2 and therefore is not time-efficient. Hence, noisy data and large data sets require usage of the K-Means algorithm.

The most popular K-Medoid method is Partitioning Around Medoids (PAM). Unlike K-Means, it defines the centroid of the cluster as the medoid, which is the most representative object for a cluster and must be an actual object in the data set. Thus, no recalculations of the distances are required and the efficiency of the algorithm is increased. In addition, the medoid approach handles outliers well. Similarly, to K-Means, PAM consists of two steps: the build step and the swap step. The build step consists of creating the clusters whereas the swap step updates the medoid of each cluster with a better fitting object. For further detail, see Dunham (2003), p. 143.

Because PAM works best for small data sets (roughly 100 data points or less), a more robust method is required for larger data sets. The Clustering Large Applications (CLARA) algorithm improves PAM by applying the sampling approach to handle large data sets. The idea of CLARA is to draw a small sample from the underlying data set and apply PAM to generate an optimal set of medoids $M$ for this sample. The resulting set $M$ is then used as the medoids for the original data set: each object in the original set $o_i$ is assigned to cluster $C_k$ with medoid $m_k$ to which it is the closest. Typically, the CLARA algorithm repeats the sampling and clustering process several times and selects the sample for final clustering based on the medoids with minimal cost, when the cost function is defined as

$$Cost = \frac{\sum_{o_i \in C_k} dist(o_i, m_k)}{n}, \quad (3.2)$$

where $m_k$ is the medoid for the $k^{th}$ cluster and $o_i$ is the $i^{th}$ object in the $k^{th}$ cluster, with $n$ being the total number of objects in cluster $k$ (Wei, Lee and Hsu, 2000).

Clustering using the CLARA algorithm can be computed in R using the function `clara` in the package `stats`. It is important to note that `clara` uses a matrix of adjusted and normalized variables and creates the distance matrix itself, therefore not requiring the calculation of the proximity matrix outlined in Section 2. The function `cluster.stats` in the package `fpc` creates validation statistics such as silhouette values and the within-between proximity ratio (will be covered in Section 4).
Although CLARA allows for larger data sets than PAM, its efficiency strongly depends on the sample size. In addition, a good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased: if certain objects are more likely to be selected than others, then the samples created to generated an optimal set of medoids are biased.

A weakness common to all K-Means based methods and their variants is that the user must specify $K$, the desired number of clusters, prior to running the algorithm. Since all procedures for determining $K$ are purely heuristic and inaccurate, it is easy to select an inappropriate amount of clusters.

### 3.2 Hierarchical Clustering

Hierarchical clustering provides an informative look into how the relationships between individual objects affect the way the clusters are structured. Unlike the K-Means method and its variants, hierarchical clustering does not require the pre-determined number of clusters $K$ as an input, and follows a step-by-step process creating a tree-like data structure, called a dendrogram. The vertical lines indicate leaves (objects) and the horizontal lines, branches, indicate each merging of leaves into new objects. All leaves together create a single tree (one cluster) and a horizontal “cut” across the dendrogram creates clusters comprised of leaves below the cut.

![Dendrogram Diagram](image)

Figure 3-2: Example of a Dendrogram with Cut Made (Jain, et al., 1999)

There are two main forms of hierarchical clustering: *agglomerative* and divisive. The agglomerative method was the technique we chose to focus on, as it is by far the most common. Agglomerative clustering starts with the proximity matrix, and individual objects as clusters, e.g. for a data set with four objects, there are four initial clusters. A distance function $\text{dist}(C_k, C_l)$ measuring proximity between the clusters is then defined, specific to each agglomerative algorithm. At each stage, the algorithm merges the two closest clusters according to this distance function and the procedure continues until only one cluster remains. Tan et al. (2006) writes a basic agglomerative algorithm as follows:
repeat:

Step 1: Merge the two most similar clusters.
Step 2: Update the proximity matrix to reflect the distances between the new cluster and the original clusters.

until Only one cluster remains. (p. 516)

Steps 1 and 2 require the use of a specific linking method to merge clusters. Ward’s method utilizes the error sum-of-squares (ESS) criterion function to define the distance between two clusters:

\[
\text{dist}_{\text{Ward}}(C_k, C_l) = ESS(C_k \cup C_l) - ESS(C_k) - ESS(C_l),
\]

where \( C_k \) is the chosen cluster and \( C_l \) are all other clusters. The ESS for cluster \( C_k \) (also true for \( C_l \)) is given by:

\[
ESS(C_k) = \sum_{o \in C_k} \text{dist}(o, c_k)^2,
\]

where \( c_k \) is the sample mean (centroid) of the objects in cluster \( C_k \). For details, see Tan et al., (2006), p.523, and Kamvar, Klein & Manning (2002).

The advantage of agglomerative clustering is that the dendrogram allows a user to visualize the steps of the algorithm and manually select an appropriate number of clusters. The algorithm, however, begins to get very complex in finding clustering solutions if the data set is very large (since it clusters step-by-step). Also, merging decisions, made locally at each stage, are final and cannot be reversed. This may create situations such as the one described in Tan, et al. (2006) where “clusters are not even stable, in the sense that a point (object) in one cluster may be closer to the centroid of some other cluster than it is to the centroid of its current cluster” (p. 526). Another weakness of the hierarchical clustering algorithms is their sensitivity to noise and outliers.

To perform clustering analysis using Ward’s method, we used the function \texttt{hclust} in the R package \texttt{stats}. The \texttt{cutree} function of the \texttt{stats} package cuts the dendrogram at the height that creates a user-defined cluster amount. For more information, see “Hierarchical Clustering” and “Cut a Tree into Groups of Data,” from the R Core Development Team Manuals.

We also attempted agglomerative algorithms with different definitions of the distance between clusters, namely, the single-link, the complete-link, and the group average methods. The preliminary results following the application of different linking methods on the Nevada data set were not satisfactory enough to warrant further research. The single-link and the group average methods produced non-globular shaped clusters for the Nevada data but the “chaining effect,” where clusters are poorly differentiated, affected the data. The complete-link method produced reasonable results in terms of cluster structures, but was similar to the clusters created by Ward’s method. Overall, the project’s objective to minimize variability within clusters, maximize variability between clusters, and Ward’s method making use of the SSE made Ward's the linking method of choice for the project (Equation 3.3).
3.3 Fuzzy Clustering

Fuzzy clustering is advantageous when the objects of the data set do not fall naturally into well-separated clusters. It assigns each object to several clusters and quantifies the degree of belonging to different clusters by weights, otherwise known as membership coefficients: \( w_{ik} \) is the weight assigned to object \( o_i \) and the cluster \( C_k \). The weights range from 0 to 1 and are subject to the constraints:

- All weights for a given object \( o_i \), add up to 1: \( \sum_{k=1}^{K} w_{ik} = 1 \), where \( K \) represents total number of clusters.
- Each cluster \( C_k \) contains, with non-zero weight, at least one object, but does not contain, with a weight of one, all of the objects.

The goal of the algorithm known as “FANNY” used in this project is to minimize the following objective function:

\[
\sum_{k=1}^{K} \sum_{i,j=1}^{n} w_{ik}^q w_{jk}^q \frac{dist(o_i, o_j)}{2 \sum_{j=1}^{n} w_{jk}^q},
\]

where \( dist(o_i, o_j) \) is the distance between any objects \( o_i \) and \( o_j \), \( q \) represents the fuzzifier of the weight (set to \( q=2 \) for our project) and \( n \) is the total amount of objects in cluster \( C_k \).

When each object has equal membership in all clusters, the clustering is entirely fuzzy. On the other hand, when each object has a membership of one in some cluster and zero membership in all other clusters the clustering is entirely hard. The computation Dunn's partition coefficient is calculated to have an idea how hard or fuzzy the clustering is:

\[
F_K = \frac{\sum_{k=1}^{K} \sum_{o_i \in C_k} w_{ik}^q}{n_k},
\]

where \( k \) indicates which cluster \( C \), and \( n_k \) = number of observations in cluster \( C_k \). This will always fall within the range \((\frac{1}{K}, 1]\), where \( F_K = \frac{1}{K} \) denotes an entirely fuzzy clustering and \( F_K = 1 \) denotes a hard clustering.

Similar to K-Means based algorithms, one of the difficulties in using fuzzy clustering algorithms is that the optimal number of clusters to be created, \( K \), has to be determined beforehand while no good method to do so exists. Also, the initial locations of the centroids of the clusters have to be guessed. On the other hand, unlike other types of clustering, fuzzy methods allow for some ambiguity in assigning the objects to clusters, which often occurs in practice.

Running the function \texttt{fanny} from the package \texttt{stats} in R on each Nevada data set, the fuzzy coefficient of clusters were considerably high (or close to one) in metropolitan areas, particularly Las Vegas. The fuzziest objects, or least well clustered, were the outlying objects of these metropolitan clusters. Once we separated the Las Vegas data for Sets 1 and 2 for both Liability and Collision insurance from the rest of Nevada, we saw complete geographic divides. A vertical axis separated an east and west cluster when two clusters were created. The creation of three
clusters split the eastern cluster into two separate clusters. With four clusters, the western half split, and so forth. This geographical trend made it difficult to interpret whether our clusters were accurate or not.
4. Comparing Clustering Techniques

4.1 Validation Methods

We applied validation methods to find which clustering algorithm and cluster amount worked best for each of our data sets. The first consisted of inspecting at the mean squared error (MSE) of each algorithm with different cluster amounts. The second method used silhouettes to determine how well objects were clustered. The final method involved within proximity – between proximity ratios (WB ratios) to analyze the differences between clusters.

4.1.1 Mean Squared Error for Purposes of Comparison

The Mean squared error (MSE) measures the effectiveness of a prediction or estimate. The definition of the MSE is the expected value of the squared difference between the estimate and the actual value. We used a modified mean squared error equation,

\[ MSE = \sqrt{\sum_{k=1}^{K} \sum_{o_i \in C_k^{-1}} e_i \left( A_k \bar{p}_2 - \hat{p}_i \right)^2}, \]

(4.1)

where \( \bar{p}_2 \) is the average raw pure premium in Set 2, \( A_k \bar{p}_2 \) is the predicted pure premium for object \( o_i \) in the \( k^{th} \) cluster of Set 1, and the actual Set 2 raw pure premium \( \hat{p}_i \) is weighed by its exposure \( e_i \) (from Set 2).

A small MSE indicates a better prediction and validity for the adjustment factors. Nonetheless, adjustment factors computed solely from one set of data are distorted when pure premiums vary greatly between different time periods. This was the case with Sets 1 and 2, signifying that the MSE may not be a good validation technique comparing the predicted pure premium to the Set 2 pure premium for each insurance policy.

4.1.2 Silhouettes and Silhouette Width Values

Silhouette values are computed for every object to determine whether objects were assigned to the appropriate cluster. We define \( a(o_i) \) as the average within proximity of \( o_i \) to all other objects belonging to the same cluster \( o_i, o_j \in C_k \):

\[ a(o_i) = \sum_{j=1, i=1}^{n} \frac{\text{dist}(o_i, o_j)}{(n_k-1)}. \]

(4.2)

A small \( a(o_i) \) is desired, indicating the cluster is relatively homogeneous. The average between proximity of \( o_i \) in cluster \( C_k \) to all objects of cluster \( C_l \) is \( d(o_i, C_l) \), where \( k \neq l \). A high \( d(o_i, C_l) \) implies a cluster is well differentiated from other clusters. Finally, we define

\[ b(o_i) = \min\{d(o_i, C_l)\}, \]

(4.3)

where \( b(o_i) \) is taken over all clusters when \( k \neq l \) for \( o_i, \in C_k \).
The following ratio computes the silhouette value for each $o_i$ (with the restriction $-1 < s(o_i) < 1$):

$$s(o_i) = \frac{b(o_i) - a(o_i)}{\max\{a(o_i), b(o_i)\}}. \quad (4.4)$$

A positive silhouette value close to 1 suggests objects are clustered together correctly. A silhouette value near 0 indicates the proximity of an object is about equal between two clusters. Finally, a negative value near -1 suggests the object was not clustered correctly or lacks any sort of similarity. We desire a large positive value and aim to maximize $b(o_i)$ and minimize $a(o_i)$.

Finally, the average of all individual silhouette values for the entire dataset is

$$\bar{s} = \left(\frac{1}{n}\right) \sum_{k=1}^{K} \sum_{o_i \in C_k} s(o_i). \quad (4.5)$$

The average silhouette compares silhouette values among different clustering algorithms and cluster amounts to find the best overall clustering. However, the use of $\bar{s}$ may not be realistic for our situation. For example, if the largest average silhouette value for the entire state of Nevada was for two clusters, then presumably this is the best solution. In spite of this, we opt for a higher number that maintains a relatively large silhouette value, as two clusters do not provide enough differentiation in pure premium amount. Other validation methods are required in conjunction with silhouettes to find the optimal clustering solution.

The `cluster.stats` function within the `FPC` package in R computes and plots the silhouette values easily.

### 4.1.3 Within Proximity – Between Proximity Ratios

In a similar vein to silhouettes, we calculated the ratio of between- and within-proximities to find the WB ratio,

$$WB = \frac{\sum_{k=1}^{K} \sum_{o_i \in C_k} a(o_i)}{\sum_{k=1}^{K} \sum_{o_i \in C_k} b(o_i)}. \quad (4.6)$$

A small WB ratio indicates a high average between-proximity $b(o_i)$ and a low average within-proximity $a(o_i)$. In addition, the WB ratio decreases as $K$ increases. By graphing WB against $K$ we can see where the WB-ratio naturally plateaus and thus decide on the optimal number of clusters $K$, see Figure 4-3 below. The WB ratio can be found using `cluster.stats` of the `FPC` package in R.

### 4.2 Comparing Clustering Algorithms

We utilized the three methods above to compare the results of each clustering algorithms applied to the Nevada data sets. In order to help guide the process for our final recommendations, the figures displayed below are from the results of data set CLVS1.
In Figure 4-1, FANNY maintained a low MSE when compared to the other algorithms with an optimal value at 4 clusters. Agglomerative clustering on the other hand, computed the MSE the worst and led us to opt for a different algorithm.

Figure 4-1: MSE Values for 1-12 Clusters

Figure 4-2: Average Silhouette Values for 1 – 12 Clusters

Figure 4-2 demonstrates the difficulty in choosing the algorithm and the amount of clusters necessary, as the average silhouette values tend to rise and fall. Four and five clusters had the maximum silhouette of all three methods while FANNY and hierarchical clustering were the best of the three.
Figure 4-3 displays how similar the WB ratios were for each algorithm. The optimal cluster amount is indicated when the slope of the graph stabilizes after several additional clusters. Specifically for CLVS1, we should cluster with about 8 territories.

![Collision LVS1 WB Ratio](image)

**Figure 4-3: WB Ratios for 1-12 Clusters**

We concluded FANNY clustering with four territories was the algorithm and cluster amount for CLVS1 using the three validation methods. Figure 4-4 shows the location of each cluster.

![Map of LVS1 FANNY, 4 Clusters](image)

**Figure 4-4: Map of LVS1 FANNY, 4 Clusters**
5. Results

5.1 Selecting a Distance Function

When creating the proximity matrix we considered density regions, adjusting for exposure, and the selection of the type of distance function. We separated the data set into two subsets due to differing density regions and clustered separately. Clustering the entire data set together resulted in non-congruent clusters within the highly dense Las Vegas region for all four data sets. Clustering Las Vegas separately resulted in more contiguous clusters.

The use of a credibility adjustment may be necessary to address objects with very low exposure. The full credibility standard $E$ in Equation (2.1) controls the level of credibility adjustment. Increasing $E$ decreases the influence of original raw premia.

The balance between the relative importance of premia and geographic distances controlled the weight $r$ in Equation (2.3) for the distance function. Smaller $r$-values created more geographic clusters (with little emphasis on pure premium) and larger $r$-values created non-congruent clusters (with more weight on pure premium). Finally, if the neighborhood-based distance function (2.4) is used, the user-input weight $z$ controls the weight placed on the neighborhood portion of the equation, and how much to the Euclidean distance portion.

All these weights are chosen by the user and have significant effect on the territories constructed.

5.2 Optimal Number of Clusters

The four clustering types that our group focused on were K-Means, K-Medoids, hierarchical clustering, and fuzzy clustering. The R package cluster contained the pam, clara, and fanny functions we used to compute the clustering solutions for K-medoids and fuzzy clustering, while hclust was the function for agglomerative clustering.

The K-Medoid algorithm utilized the medoid in place of the centroid, reducing the effect of outliers (higher/lower pure premiums and/or exposures) when creating clusters. PAM cannot operate properly using large data sets, hence the creation and our recommendation of the CLARA algorithm.

For hierarchical, we chose agglomerative clustering over divisive clustering. The algorithm created geographic clusters of objects, while still differentiating between different pure premium and exposure amounts. Of the different linking methods to combine clusters, Ward’s method and the complete-link method created defined cluster structures compared to single-link and the group average methods, which were subject to the “linking effect.” Therefore, we concluded that agglomerative clustering using Ward’s method clustered the best in urban areas of Nevada (such as the Las Vegas and Reno regions) where objects may be close in proximity to one another. As an example, Figure 5-1 shows the dendrogram for LLVS1 data set using 4 clusters (number of clusters chosen based on smallest MSE value). We see a well-defined tree structure with balanced branches and relatively balanced cluster sizes.
FANNY was the only type of clustering method considered for fuzzy clustering and performed similarly to PAM and CLARA. The membership coefficient weights produced by FANNY indicated hard clustering. This is illustrated in Figure 5-2 which shows membership weights of four objects from cluster $C_4$ based on clustering CNVS1 with $K = 6$ clusters. Figure 5-2 demonstrates how the objects had membership coefficient values close to 1.

We feel that FANNY and fuzzy clustering in general is a valuable tool for territorial analysis.

### 5.3 Best Clustering Technique

The three validation techniques used in choosing an appropriate clustering solution were the MSE, average silhouette values, and WB ratios. MSE scores are vital for comparison across Sets 1 and 2, while average silhouette values and WB ratios give guidance on selecting optimal number of clusters. All three validation methods concluded similarly on the optimal clustering
solution, but each have different information to show so it is important to look at all three when making a final decision.

We found that the Las Vegas optimal clustering was two or four clusters and the rest of Nevada was roughly six clusters. The validation techniques confirmed the conclusions made in Section 5.2; hierarchical (agglomerative) clustering works best for dense, urban areas (see Figure 5-3) and FANNY worked best for rural areas (see Figure 5-4). In Figure 5-4 we observe that FANNY put more emphasis on premia over geographic distances, making the clusters more spread out across Nevada. This may be useful in certain situations where areas are similar in terrain but not relatively near one another.

Figure 5-3: Hierarchical Clustering of CLVS1, 4 Clusters
5.4 Best Distance Function

We obtained better results for all validation methods using the neighborhood-based distance function when compared to the Euclidean distance function. Therefore, we opted to use the neighborhood-based distance function for clustering Nevada in place of the latter. Figure 5-5 displays the difference between MSE values for each of the two distance calculations for LNVS1.
The neighborhood-based distance function also has the ability to cluster Nevada as a whole. Splitting any state into separate data sets to cluster may lead to an increase in complications when attempting to decide how to make the split. These facts for the neighborhood-based function indicate an increase in effectiveness over the modified Euclidean distance.

Figures 5-6 and 5-7 show how geographically contiguous the clusters are when neighborhood-based distance is used.

![Hierarchical Clustering of LNVS1 Using Neighborhood Based Distance](image1.png)

**Figure 5-6: Hierarchical clustering of LNVS1 using neighborhood Based Distance**

![Las Vegas Region of Figure 5-5](image2.png)

**Figure 5-7: Las Vegas region of Figure 5-5**

### 5.5 Overall Recommendation

Given the low correlation in pure premia between the two data sets, predictive ratemaking based on territory analysis was difficult. Also, given the underlying structure of the clustering methods considered, we felt it was appropriate to treat the urban areas separately from the rural ones.

Agglomerative Clustering using Ward’s Method and the fuzzy clustering algorithm FANNY provided us with the best resulting clusters for Nevada. We suggest the usage of agglomerative clustering in the Las Vegas area with four clusters as it provided the perfect balance of a low MSE value with respect to the other algorithms, high silhouette values, and roughly equal amounts of exposure across all clusters created. Dendrograms also are an intuitive way to choose the number of clusters and visualize the clustered proximity matrix. FANNY, on the other hand, was able to cluster across longer distances, making it a strong candidate for the other areas of
Nevada, with a chosen cluster amount of six. In future territorial analysis, it may be important to see how well an object is related to its cluster.

**5.6 Future Work**

There is a variety of other clustering algorithms. In particular, it may be interesting to consider CHAMELEON agglomerative method and kernel-based density clustering, see Tan, et al. (2006), Section 9.4.4 and Section 9.3.3 respectively.

The available R packages provided the algorithms considered and used in this report. It would be useful to design custom algorithms for the particular problem of territory analysis. For example, one could design agglomerative clustering method constrained to create geographically contiguous territories.

The performance of the clustering methods across various data sets requires further study for analysis.
6. References


*Clustering Large Applications (CLARA).* (2010). Retrieved from the R Core Development Team Manuals web site:

*Compute or Extract Silhouette Information from Clustering.* (2010). Retrieved from the R Core Development Team Manuals web site:


*Hierarchical Clustering.* (2010). Retrieved from the R Core Development Team Manuals web site:


