Optimization of LaGrange College Admissions Counselors’ Paths Using Student Data

Sponsoring Faculty Member: Jon Ernstberger

Jared O. Lovin

LaGrange College attracts hundreds of freshman students across the Southeast each year. It would be beneficial to the college’s Office of Admissions to optimize admission counselors’ paths to reduce expenses on mileage and time. The $K$-means algorithm will be used to cluster students into groups based upon physical locations and to locate regional centers for the states of Alabama, Florida, and Georgia. Additionally, a solution to the Traveling Salesman algorithm will be implemented to find optimal routes to visit students in each cluster.

**$K$-means Algorithm**

The $K$-means clustering algorithm is a partition-based clustering method that partitions a set of observations into $K$ clusters where each cluster generates a mean. The overall purpose of the $K$-means algorithm is to minimize the total intra-cluster variance [2]. This is done by determining the clustering space $C$ that minimizes the Euclidean distance from each point in the cluster to the centroid (or weighted mean) of the cluster. This can be expressed as the minimization problem

$$\arg\min_C \sum_{k=1}^{K} \sum_{x \in C_k} \|x - r_k\|^2.$$  \hspace{1cm} (1)

In Expression 1, $K$ is the number of clusters, the $x$-values are the current data points in the iteration, and $r_k$ is the current centroid for the $k^{th}$ cluster, $C_k$, in the space of clusters $C$. [2].

This process is described in Algorithm 1. First, the data is grouped into $K$ clusters. Next, an iterative process of randomly selecting $K$ points as centroids for each cluster is initiated. Third, the algorithm assigns data points to their nearest cluster centroid determined by the objective function (1). Finally, the algorithm will calculate the new centroid for the data points in each cluster. This process continues iteratively until all data points are assigned to their nearest centroid.
for $k=1, \ldots, K$ let $r(k)$ be a randomly chosen point $D$

do
| while changes in clusters $C_k$ happen do
| form clusters:
| for $k=1, \ldots, K$ do
| \[ C_k = \{ x \in D | d(r_k, x) \leq \forall j = 1, \ldots, K, j \neq k \} \]
| end
| compute new cluster centers:
| for $k=1, \ldots, K$ do
| \[ r_k = \text{the vector mean of the points in } C_k \]
| end
end

Algorithm 1: Pseudocode for the $K$-means algorithm.[2] Here, $C_k$ is the $k^{\text{th}}$ cluster, $r_k$ is the mean of the values in the $k^{\text{th}}$ cluster, and the $x$-values are data values.

Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is defined as a method of finding the shortest route between each city in a given list. The fundamental idea is to choose pairs of cities that have the smallest distance between them. First, the variable $p_{ij}$ is defined as

\[ p_{ij} = \begin{cases} 1 & \text{the path goes from city } x_i \text{ to city } x_j \\ 0 & \text{otherwise.} \end{cases} \]

For $i = 0, \ldots, n$, let the $x_{ij}$ values be the distances from point $x_i$ to point $x_j$. From here, TSP can be written as

\[ \min \sum_{i=0}^{n} \sum_{\substack{j=0 \\neq i}}^{n} x_{ij} p_{ij} \quad (2) \]

subject to

\[ \sum_{i=0}^{n} p_{ij} = 1 \quad \text{and} \quad \sum_{j=0}^{n} p_{ij} = 1 \quad \text{for} \quad i, j = 0, \ldots, n. \]

In Expression 2, the problem requires that, at each city, the salesman arrives from only one other city and departs to exactly one other city [4]. The Traveling Salesman Problem can be solved by a variety of different methods. For
Let $s = s_0$;

for $k = 0$ through $k_{\text{max}}$;

\begin{verbatim}
   do
      \hfill T = temperature($\frac{k}{k_{\text{max}}}$);
      \hfill Pick a random neighbor, $s_{\text{new}} = \text{neighbor}(s)$;
      \hfill if $P(E(s), E(s_{\text{new}}), T) > \text{random}(0, 1)$ then
         \hfill move to the new state
         \hfill $s = s_{\text{new}}$
   end
\end{verbatim}

end

Output: the final state $s$

**Algorithm 2:** Pseudocode for simulated annealing. Here, $k_{\text{max}}$ is the maximum number of steps, $T$ is the temperature, and $s$ is the current state.

The case at hand, simulated annealing (a stochastic search) will be used to solve TSP.

The idea of simulated annealing originated from the physical annealing process. This process was a thermal process for obtaining a low-energy state for a solid in a heat bath. The annealing process consisted of two steps. The first step would be increase the temperature to maximum for the solid to melt. Second, the temperature would be meticulously decreased until the melted solid arranged its particles to be in a ground state of the solid. While in the melted state, the particles would arrange themselves randomly and in the ground state they would arrange in a structured lattice with the energy of the system at a minimal\cite{1}.

This process can be mimicked by software through an implementation of a series of steps. First, an initial "temperature", or energy state $s$, is declared. Then, the algorithm enters into an iterative process until a maximum number of steps has been reached. Inside this iterative process, a random state $s_{\text{new}}$ is chosen and an acceptance probability is calculated. If the acceptance probability is greater than a randomly chosen value between zero and one, $s$ takes on the value $s_{\text{new}}$. Otherwise, it remains the same. Finally, a terminal state is reached and delivered as output. This process has been generalized as the pseudocode of Algorithm 2. (See \cite{1, 3}.)
Blurred Data

Nine-digit zip code values, standard zip codes plus four location markers, were taken from data given by the college’s admissions department of all students through 2012 who had come to LaGrange College. Using the site geocommons.org, the zip values were converted to coordinate pairs in (latitude, longitude) form. Uniformly distributed random noise scaled to 0.1%, expressed as $U(0, 0.001)$, was added to each of these values to skew the exact locations of students for identity protection. This blurring process is described by the mathematical formula

$$\left(\bar{x}_i, \bar{y}_i\right) = \left([1 + U(0, 0.001)] x_i, [1 + U(0, 0.001)] y_i\right).$$

The $(\bar{x}_i, \bar{y}_i)$ coordinate pairs in Expression 3 are the values used in the following computations.

Results

The centroids indicated in Figure 1 serve as the average positions of student locations for each cluster. By using Google Maps, the approximate physical location of each centroid can be determined. It can be see that the centroid for Cluster One is approximately Lake County, Florida. Subsequently, the centroid for Cluster Two represents LaGrange, Georgia and for Cluster Three, Norcross, Georgia.

For each cluster, the optimal total distance (in miles) found by the Traveling Salesman implementation was compared to the average distance of all random walks through the points in each cluster. This comparison was then used to find a rate of improvement between the two distances. In Table 1, optimized distances, improvements (savings), and mileage savings for the three clusters have been computed. For example, as shown in Table 1, the optimal path for Cluster Two is 15,828.01 miles which is significantly shorter when compared to the randomized path of 56,578.32 miles (an improvement rate of 72%). Furthermore, by using the route found by the Traveling Salesman implementation, one will spend $8,072.29 as compared to $28,854.94 in mileage. On the whole, by calculating the optimal route for each cluster, the Office of Admissions can save in time and money.

Conclusions

The centroids generated by the $K$-means clustering method depict approximate locations for recruitment centers for students coming from the Alabama, Florida, and Georgia areas. These findings are graphically depicted in Figure 1 as the centroids fall into the areas of higher student concentration. By applying the Traveling Salesman algorithm (implemented using the method of simulated annealing) to the clustered student data and centroids,
Figure 1: A plot indicating clusters and centroids computed via the $K$-means algorithm overlaid onto a map of Alabama, Florida, and Georgia with latitude and longitude. The state border for Tennessee can be found along latitude $35^\circ$, the Alabama-Georgia border at longitude $-85^\circ$, and the Georgia-Florida border at latitude $30^\circ$.

it can be observed that the college's admissions counselors can reduce their mileage expenses by at least fifty-five percent. Furthermore, using the calculated routes provided, the recruiters will save significant quantities of time.

Future Work

It is known that it is not reasonable for the admissions counselors to travel to the location of each student. Consequently, future work will be directed at creating multiple small sub-clusters within each of the clusters found in this study. Then, the Traveling Salesman algorithm will be applied to the centroids of these sub-clusters which the admissions counselors can travel between as the most efficient route.

A potential (and realistic) inclusion to this project, explored but not implemented for this current study, is to input the locations of these centroids into the Google Maps API. This process would create truly minimized pathways with road directions to be followed.
<table>
<thead>
<tr>
<th>Cluster</th>
<th>Distance (Miles, Opt./Rand.)</th>
<th>Improv. (%)</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>15,828.01 / 56,578.32</td>
<td>72 %</td>
<td>$20,782.65</td>
</tr>
<tr>
<td>Two</td>
<td>36,593.24 / 86,770.78</td>
<td>58 %</td>
<td>$25,591.10</td>
</tr>
<tr>
<td>Three</td>
<td>27,364.46 / 60,434.65</td>
<td>55 %</td>
<td>$16,865.80</td>
</tr>
</tbody>
</table>

Table 1: Results indicating the optimal distance compared to the randomized distance, the improvement rate, and mileage savings for each cluster based upon the rate of $0.51/mile.

References


