Final Report
On
Advanced Artificial Intelligence

By
Student Name: Joel Medina

Table of contents
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Explain the Paper Adequately</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Explain the Paper Adequately</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Explain the Paper Adequately</td>
<td>6</td>
</tr>
<tr>
<td>4.1</td>
<td>Explain the Paper Adequately</td>
<td>7</td>
</tr>
<tr>
<td>5.1</td>
<td>Explain the Paper Adequately</td>
<td>8</td>
</tr>
<tr>
<td>6.1</td>
<td>Explain the Paper Adequately</td>
<td>10</td>
</tr>
<tr>
<td>7.1</td>
<td>Explain the Paper Adequately</td>
<td>11</td>
</tr>
<tr>
<td>8.1</td>
<td>Explain the Paper Adequately</td>
<td>12</td>
</tr>
<tr>
<td>9.1</td>
<td>Explain the Paper Adequately</td>
<td>13</td>
</tr>
<tr>
<td>10.1</td>
<td>Explain the Paper Adequately</td>
<td>14</td>
</tr>
<tr>
<td>11.1</td>
<td>Explain the Paper Adequately</td>
<td>16</td>
</tr>
<tr>
<td>12.1</td>
<td>Explain the Paper Adequately</td>
<td>17</td>
</tr>
<tr>
<td>13.1</td>
<td>Explain the Paper Adequately</td>
<td>18</td>
</tr>
<tr>
<td>14.1</td>
<td>Explain the Paper Adequately</td>
<td>19</td>
</tr>
<tr>
<td>15.1</td>
<td>Explain the Paper Adequately</td>
<td>20</td>
</tr>
<tr>
<td>16.1</td>
<td>Steps of Genetic Based Algorithm</td>
<td>21</td>
</tr>
<tr>
<td>16.2</td>
<td>Technical Details of Algorithm 1</td>
<td>22</td>
</tr>
<tr>
<td>16.3</td>
<td>Illustration of Algorithm 1</td>
<td>23</td>
</tr>
<tr>
<td>17.1</td>
<td>Steps of “A k-means algorithm”</td>
<td>27</td>
</tr>
<tr>
<td>17.2</td>
<td>Technical Details of Algorithm 2</td>
<td>27</td>
</tr>
<tr>
<td>17.3</td>
<td>Illustration of Algorithm 2</td>
<td>27</td>
</tr>
</tbody>
</table>
Clustering Algorithms

Clustering Algorithms are used to take a set of data and divide the data into separate groups of data called clusters. Data points which have been grouped in a cluster should have some kind similarity depending on which parameters where used to map the data. Applications of Clustering Algorithms include:

- Machine Learning
- Data Mining
- Pattern Recognition
- Image Analysis
- Information Retrieval
- Bioinformatics
Section 1: Clustering Paper 1

1.1 A K-Means Clustering Algorithm

Hartigan and Wong discuss the k-means clustering algorithm. They begin by explaining very briefly how the algorithm requires a matrix of M points in N dimensions and a matrix of K initial cluster centers. Also a well explained series of steps is given in the article in order to make use of the algorithm. They concluded with a section headed "Additional Comments" which explained an efficient way of selecting the initial cluster centers in which the points are first ordered by their distanced to the overall mean of the data sample. Then, for cluster L(L=1,2,3.. K), the \((1+(L-1)*[M/K])\)th point is chosen to be its initial cluster center.
Section 2: Clustering Paper 2

2.1 Suppressed fuzzy c-means clustering algorithm

Fan, Zhen and Xie discuss how the hard fuzzy c-means algorithm, Fuzzy C-means algorithm, rival checked fuzzy c-means algorithm and the suppressed fuzzy c-means algorithms are related and compares the performance of each algorithm with each other algorithm. They explain how the FCM algorithm improves partition performance but is much slower than the HCM algorithm. The RCFCM algorithm is only a modification of the FCM algorithm and was made to make the FCM faster. And finally the SFCM algorithm is was created to overcome some problems which the RCFCM algorithm. Every algorithm is explained and its faults are pointed out.
Section 3: Clustering Paper 3

3.1 A Monte Carlo Implementation of the EM Algorithm and the Poor Man’s Data Augmentation Algorithms

Wei and Tanner [3] explain the Expectation-Maximization algorithm and how it is implemented. Modifications were made to the algorithm in order to calculate the entire posterior, as well as starting points for the full data augmentation analysis. Examples are shown and explained to illustrate the methodology of the algorithm.
Section 4: Clustering Paper 4

4.1 Efficient clustering of high dimensional data sets with application to reference matching

McCallum, Nigam and Ungar explain how the usage of cheap measure techniques used in an algorithm which creates "canopies" on clusters is a very efficient way of clustering and time is saved by not having to use time expensive algorithms, such as k-means and EM algorithms. In the conclusion of their article they include various ways in which clustering algorithms can be applied which include: clustering stars astronomy, grouping DNA sequences based on their proteins and data mining.
Section 5: Clustering Paper 5

5.1 The global k-means clustering algorithm

Likas, Vlassis, and Verbeek discuss the global k-means clustering algorithm which adds one cluster center at a time. The algorithm does this through a deterministic global search procedure consisting of the k means algorithm executed a number of times equal to the size of the data set.
Genetic Algorithms

Genetic Algorithms is a method used to find optimal solutions to a certain problem. The method mimics the process of natural evolution by evolving using crossover and mutation to yield stronger offspring. Data is represented as chromosomes which are applied to a fitness function and given a fitness value. The chromosomes chance for crossover and mutation depends on their fitness values relative to other chromosomes. Some of its applications include:

- Artificial Creativity
- Economics
- Code-Breaking
- Bioinformatics in RNA structure prediction
- Multiple sequence alignment
Section 6: Genetic Algorithms Paper 1

6.1 Genetic algorithm-based clustering technique

Maulik and Bandyopadhyay discuss clustering technique based on genetic algorithms, these techniques are called GA-clustering. The searching capabilities of genetic algorithms is used to search and find the most accurate centers for clusters to be optimized. Chromosomes used in the genetic algorithm are made up of strings of real numbers which encode the centers of the fixed number of clusters. Four artificial data sets and three real-life ones are used to demonstrate the superiority of the GA-clustering algorithm over the k-means algorithm.
Section 7: Genetic Algorithms Paper 2

7.1 Genetic algorithms for clustering, feature selection and classification

Yang discusses how the genetic algorithm when used to cluster sets of data is much more effective and optimal when compared to the k-means algorithm. Further more he discusses how the genetic algorithm for clustering does not require the user to input an initial number clusters whereas the k-means clustering algorithm does.
Section 8: Genetic Algorithms Paper 3

8.1 Multi-Objective Data Clustering using Variable-Length Real Jumping Genes Genetic Algorithm and Local Search Method

Ripon and Chi-Ho discuss how a novel multi-objective evolutionary clustering algorithm using variable-length real jumping genes genetic algorithms evolve clustering solutions using multiple clustering criteria. Methods such as probabilistic cluster merging and splitting are introduced in the algorithm.
9.1 Optimizing the Performance of a Formula One Racing Car using a Genetic Algorithm

Wloch and Bentley discuss how a genetic algorithm will be used to optimize various variables such as gear ratios and tire pressure in a formula one car. The tests were done using a simulator software. Tests using various population sized were made and results showed that a population of 40 was the most appropriate.
10.1 An Image Segmentation of Fuzzy C-Means Clustering Based on the Combination of Improved Ant Colony Algorithm and Genetic Algorithm

Gong discusses an improved ant colony algorithm used as a method of dynamic fuzzy clustering analysis. They explain how the ant colony algorithm and the genetic algorithm are integrated and how doing this raises the ant group algorithm convergence rate and enhances the cluster precision using the ant colony algorithm’s parallelism.
Support Vector Machines

Support Vector Machine is a method used to analyze data and recognize patterns. SVMs categorizes a set of data into two categories by creating an optimal plane splitting the two categorized data sets and to have new additional data points fall into the appropriate category.
11.1 Least Squares Support Vector Machine Classifiers

Suykens and Vandewalle discuss a least squares version for support vector machine classifiers. The solution is done by solving a set of linear equations instead of quadratic programming due to the equality type constraints in the formulation.
Section 12: Support Vector Machines Paper 2

12.1 Optimal reduction of solutions for support vector machines

Hwei-Jen Lin and Jih Pin Yeh discuss how they integrate Support Vector Machines and a Genetic algorithm. The genetic algorithm used is used to optimally reduce solutions for a support vector machine by selecting vectors from the trained support vector solutions, such that the vectors best approximate the original discriminant function.
The support vector machine under test

David Meyer and Kurt Hornik discuss the effectiveness of support vector machines when compared to 16 classification methods and 9 regression methods. They state that the support vector machine showed mostly good performance, but that other methods proved to be very competitive.
Section 14: Support Vector Machines Paper 4

Support vector machine active learning with applications to text classification

Simon Tong and Daphne Koller discuss how they plan to use an active learning algorithm for choosing which instances to request next. They also show results that show that implementing their active learning method can significantly reduce the need for labeled training in both the standard inductive and transductive settings.
Sujun Hua and Zhirong Sun discuss how they will make use of support vector machines to predict sub-cellular localization of proteins from their amino acid compositions. They discuss how this new approach provides superior prediction performance when compared to other algorithms which have been used.
Section 16: Detailed Paper 1

Genetic Algorithm-Based Clustering Technique

16.1 Steps of Genetic Based Algorithm

The first paper which I chose is “Genetic Algorithm-Based Clustering Technique” by Ujjwal Maulik and Bandyopadhyay. I will first show how the technique is used in the article. I will then make some alterations to their algorithm and apply it to a small example.

In this article a technique for clustering using a genetic algorithm is proposed. It is compared to The k-means algorithm using several data sets. The following slides will explain how to convert a clustering problem into a genetic algorithm.

Genetic Algorithms are used to optimize a certain function by using data in the form of chromosomes made up of genes and mimicking natural selection and survival of the fittest. As in biology and evolution the operators crossover and mutation have also been applied to the chromosomes in order to yield new generations of chromosomes.

The concept of clustering is to take a large data set and cluster it into separate clusters, the number of clusters may be predefined, or they may depend on how the data is structured. In the k-means clustering algorithm the aim of the algorithm is to minimize the within cluster sum of squares:

\[
\arg \min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2
\]

We will see how this will be a critical formula when converting a clustering problem into a genetic algorithm.
16.2 Technical details of the Algorithm

Chromosomes:
In order to be a genetic algorithm, there must be chromosomes. The method used to represent the cluster centers as chromosomes is by simply ordering them in the chromosome. Each gene being a component of a cluster center. That means that each gene will be a floating point instead of the traditional 1 or 0.

Ex: Initial cluster centers (3,2) (4,1) and (5,6) are represented as the chromosome: [3 2 4 1 5 6]
This way any amount of centers in any amount of dimensions can be easily represented.

Initializing a population:

First one must determine the population size (number of chromosomes), this may depend on the data. To determine the initial chromosomes, first determine the number of clusters 'k' and then randomly take k number of points to be the initial centers of the data, then represent them as a chromosome. Do this for every chromosome in the population.

Ex:
X = (2,10) (2,5) (8,4) (5,8) (7,5) (6,4) (1,2) (4,9)
K = 3
Population = 4

Chromosomes:

[2 10 8 4 5 8]  
[2 5 8 4 4 9]  
[7 5 6 4 4 9]  
[2 5 1 2 4 9]

Fitness function:

From the function which we previously looked at we derive the clustering metric:

$$M = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - z_i\|$$
The k-means algorithm aims at minimizing this function, to get our fitness function we simply aim to maximize \(\frac{1}{M}\). The chromosomes with the highest fitness will be more likely to pass on its genes in crossover and mutation.

Crossover:

Crossover is probable during each run. In the genetic algorithm approach for clustering crossover is done by exchanging genes between the two selected parent chromosomes. The genes to be exchanged are chosen by generating a random integer called the crossover point in the range \([1,L-1]\), \(L\) being the length of the chromosome. Then all genes to the right of the position determined by the crossover point are exchanged.

Mutation:

Mutation is also probable during each generation. Since we are not working with binary representations of chromosomes, we cannot simply flip from a 1 to 0 or vice versa. If mutation was to occur, a random gene from the chromosome would be chosen and given a value \(v\), then a random number \(\delta\) in the range \([1,0]\) would be generated and the new value for that chromosome would be: \(v = v + \delta * v\) or \(v = v - \delta * v\) or – occur with an equal probability.

Termination Criterion:

The algorithm is done for a predefined number of iterations and the chromosome which yields the highest fitness is chosen and the centroids corresponding to that chromosome are chosen to be the centroids for the data set.

16.3 Illustration of the Algorithm

Short Example:

Given the following points
\[ A = (2,10) \ (2,5) \ (8,4) \ (5,8) \ (7,5) \ (6,4) \ (1,2) \ (4,9) \]
Use the GA clustering technique to find the optimal cluster centers. Cluster into 3 clusters and have an initial population of 4 chromosomes.

Initializing a population:
From the data points the following chromosomes are derived randomly using the method stated previously. And the fitness function $1/M$ will be used. $M$ represents the clustering metric.

Fitness function:

$$
\sum_{i = 1}^{K} \sum_{x \in C_i} ||x_j - z_i|| ^ {-1}
$$

X1 = [2 10 5 8 1 2]  
X2 = [2 5 4 9 1 2]  
X3 = [8 4 5 8 7 5]  
X4 = [6 4 1 2 4 9]

Computed fitnesses:

<table>
<thead>
<tr>
<th>Chromosome:</th>
<th>Fitness:</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1 = [2 10 4 9 1 2]</td>
<td>1/29</td>
<td>15.86%</td>
</tr>
<tr>
<td>X2 = [2 5 4 9 7 5]</td>
<td>1/13</td>
<td>35.38%</td>
</tr>
<tr>
<td>X3 = [8 4 5 8 7 5]</td>
<td>1/23</td>
<td>20.00%</td>
</tr>
<tr>
<td>X4 = [6 4 1 2 4 9]</td>
<td>1/16</td>
<td>28.75%</td>
</tr>
</tbody>
</table>

Crossover:

Due to the higher fitness of some of the chromosomes, those chromosomes are more likely to reproduce in the crossover stage. The following chromosomes have been chosen randomly for crossover.

X1 = [2 10 4 9 1 2]  
X2 = [2 5 4 9 7 5]  
X2 = [2 5 4 9 7 5]  
X4 = [6 4 1 2 4 9]

Crossover:

Due to the data being small, and the points close to each other I used a different method for crossover. First a random number between 1 and the number of genes in a chromosome is generated, then the point corresponding to the gene at that position in the chromosome will be chosen to be exchanged.

Ex: random number = 3

X=[2 10 4 9 1 2]

So then the point (4,9) will be exchanged

Offspring:

After crossover, the following offspring were yielded.
\[ X_1 = [2 \ 10 \ 4 \ 9 \ 1 \ 2] \quad X_1' = [2 \ 10 \ 4 \ 9 \ 1 \ 2] \]
\[ X_2 = [2 \ 5 \ 4 \ 9 \ 7 \ 5] \quad X_2' = [2 \ 5 \ 4 \ 9 \ 7 \ 5] \]
\[ X_2 = [2 \ 5 \ 4 \ 9 \ 7 \ 5] \quad X_2' = [6 \ 4 \ 4 \ 9 \ 7 \ 5] \]
\[ X_4 = [6 \ 4 \ 1 \ 2 \ 4 \ 9] \quad X_4' = [2 \ 5 \ 1 \ 2 \ 4 \ 9] \]

**Mutation:**

Due to the small data set and low values as points I have decided to change the mutation process for this example. An effective way would have been to choose a random gene from the chromosome to be mutated then generate a random number \( \delta \) in the range \([-1,1]\). Then add that value to the selected gene. For easy calculation I decided to simply randomly choose an integer, either -1 or 1 and add that to the selected gene.

Ex: \( X_1' = [2 \ 10 \ 4 \ 9 \ 1 \ 2] \)

\( \delta = 1 \quad X_1'' = [2 \ 10 \ 5 \ 9 \ 1 \ 2] \)

**Mutation:**

Mutation may either have a positive or negative effect on the mutated chromosomes. If it has a negative effect (the chromosome becomes weaker), it is less likely to be selected for crossover and may be discarded in later runs. The following chromosomes were yielded after mutation took place.

\[ X_1' = [2 \ 10 \ 4 \ 9 \ 1 \ 2] \quad X_1'' = [2 \ 10 \ 5 \ 9 \ 1 \ 2] \]
\[ X_2' = [2 \ 5 \ 4 \ 9 \ 7 \ 5] \quad X_2'' = [2 \ 4 \ 4 \ 9 \ 7 \ 5] \]
\[ X_2' = [6 \ 4 \ 4 \ 9 \ 7 \ 5] \quad X_2'' = [5 \ 4 \ 4 \ 9 \ 7 \ 5] \]
\[ X_4' = [2 \ 5 \ 1 \ 2 \ 4 \ 9] \quad X_4'' = [3 \ 5 \ 1 \ 2 \ 4 \ 9] \]

**New Fitnesses:**

<table>
<thead>
<tr>
<th>Chromosome:</th>
<th>Fitness:</th>
<th>Probability:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_1'' = [2 \ 10 \ 5 \ 9 \ 1 \ 2] )</td>
<td>1/26</td>
<td>17.41%</td>
</tr>
<tr>
<td>( X_2'' = [2 \ 4 \ 4 \ 9 \ 7 \ 5] )</td>
<td>1/13</td>
<td>34.81%</td>
</tr>
<tr>
<td>( X_2'' = [5 \ 4 \ 4 \ 9 \ 7 \ 5] )</td>
<td>1/18</td>
<td>25.14%</td>
</tr>
<tr>
<td>( X_4'' = [3 \ 5 \ 1 \ 2 \ 4 \ 9] )</td>
<td>1/20</td>
<td>22.63%</td>
</tr>
</tbody>
</table>

**Initial chromosomes:**

<table>
<thead>
<tr>
<th>Chromosome:</th>
<th>Fitness:</th>
<th>Probability:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_1 = [2 \ 10 \ 4 \ 9 \ 1 \ 2] )</td>
<td>1/29</td>
<td>15.86%</td>
</tr>
</tbody>
</table>
X2 = [2  5 4 9 7 5] 1/13  35.38%
X3 = [8  4 5 8 7 5] 1/23  20.00%
X4 = [6  4 1 2 4 9] 1/16  28.75%

Chromosomes after first run:

X1'' = [2 10 5 9 1 2] 1/26  17.41%
X2'' = [2 4 4 9 7 5] 1/13  34.81%
X2'' = [5 4 4 9 7 5] 1/18  25.14%
X4'' = [3 5 1 2 4 9] 1/20  22.63%

As you can see the chromosomes which have been yielded after the first run are relatively stronger with higher fitnesses overall than the initial chromosomes.
Section 17: Detailed Paper 2

The second paper which I chose is "A K-Means Clustering Algorithm" by J. A. Hartigan and M. A. Wong.

17.1 Steps of the Algorithm
The basic concept of the algorithm is to find a K-partition with locally optimal within-cluster sum of squares by moving points from one cluster to another. An effective method for choosing initial cluster centers is also suggested.

17.2 Technical details of the Algorithm
The method first orders the points in the data based on their distance to the mean. Then takes the \( \{1+(L-1)*[M/K]\} \)th element for each cluster L as its center.

17.3 Illustration of the Algorithm
Short example:

Given the data set A1(2,10) A2(2,5) A3(8,4) A4(5,8) A5(7,5) A6(6,4) A7(1,2) A8(4,9)

First we find the distance from each point to the mean then order the data set from least to greatest based on that distance.

Mean = (4.375,5.875)

<table>
<thead>
<tr>
<th>Point</th>
<th>Distance From mean</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1(2,10)</td>
<td>4.75986</td>
<td>7</td>
</tr>
<tr>
<td>A2(2,5)</td>
<td>2.53106</td>
<td>3</td>
</tr>
<tr>
<td>A3(8,4)</td>
<td>4.08121</td>
<td>6</td>
</tr>
<tr>
<td>A4(5,8)</td>
<td>2.21501</td>
<td>1</td>
</tr>
<tr>
<td>A5(7,5)</td>
<td>2.76699</td>
<td>4</td>
</tr>
<tr>
<td>A6(6,4)</td>
<td>2.48118</td>
<td>2</td>
</tr>
<tr>
<td>A7(1,2)</td>
<td>5.13870</td>
<td>8</td>
</tr>
<tr>
<td>A8(4,9)</td>
<td>3.14742</td>
<td>5</td>
</tr>
</tbody>
</table>

For cluster \( L = 1 \)
\( \{1+(1-1)*[8/3]\} \)rd point will be chosen as its center
3rd point is (2,5)

For cluster \( L = 2 \)
\( \{1+(2-1)*[8/3]\} \)th point will be chosen as its center
5th point is (4,9)

For cluster \( L = 3 \)
\( \{1+(3-1)*[8/3]\} \)th point will be chosen as its center
8th point is (1,2)

Results after the first iteration using the using the cluster centers \( C_1 = (2,5) \) \( C_2 = (4,9) \) \( C_3 = (1,2) \).
C1 - A2 A3 A5 A6
C2 - A1 A4 A8
C3 - A7

Results after the second iteration using the re computed cluster centers C1 = (5.75,4.75) C2 = (3.667,9) C3 = (1,2).

C1 - A3 A5 A6
C2 - A1 A4 A8
C3 - A2 A7

These final points are the optimal centers. When using this method to initialize the k means algorithm it only took two iterations to find the optimal points, whereas when using the random initial points in the k-modes it may take 3 or more iterations.