

Visualizing Clustering Results

Ian Davidson[†]

Introduction

Non-hierarchical clustering has a long history in numerical taxonomy [13] and machine learning [1] with many applications in fields such as data mining [2], statistical analysis [3] and information retrieval [17]. Clustering involves finding a specific number of sub-groups (k) within a set of s observations (data points/objects); each described by d attributes. A clustering algorithm generates cluster descriptions and assigns each observation to one cluster (exclusive assignment) or in part to many clusters (partial assignment). Throughout this paper, we shall refer to the output of a clustering algorithm as the clustering results, solution, or model.

The information in a clustering solution is extensive, a mixture model or K-Means model produces $k \cdot s$ conditional probabilities or distances. Visualizing the clustering results can help to quickly assimilate this information and provide insights that support and complement textual descriptions or statistical summaries. For example, we quickly wish to know how well defined are the clusters, how different are they from each other, what is their size, and do the observations belong strongly to the cluster or only marginally? Visualizing a clustering solution has many potential uses. The analyst user during the highly iterative model building process can quickly obtain insights from the visualization that suggest the adequacy of the solution and what further experiments to conduct. Alternatively, the business user can examine and query the final clustering solution using the visualization.

The interesting parts of a clustering solution will depend on the application. Database segmentation applications such as target marketing focus on the clusters and investigate which clusters are similar, which are autonomous and which have, for example, a high propensity to cross-sell. Anomaly detection applications attempt to identify those observations that do not “belong”, are interesting and require further investigation. The focus is the observations and we wish to know if they belong strongly or only marginally to their most likely cluster. Typical uses of anomaly detection are detecting money laundering, identifying network intrusion, and data cleaning [5].

In this paper, we describe a general particle framework to display the information in a clustering solution. Changes to the parameters of the framework can emphasize

[†] MineSet Group, SGI. Author’s current affiliation is the State University of New York, 1400 Western Avenue, Albany, New York, 12222, inpd@hotmail.com

information useful for a particular clustering application. Our three-dimensional information visualization represents the previously clustered observations as particles affected by gravitational forces. We map the cluster centers into a three-dimensional cube so that similar clusters are adjacent and dissimilar clusters are far apart. We subsequently place the particles amongst the centers according to the gravitational force exerted on the particles by the cluster centers. A particle's degree of membership to a cluster provides the magnitude of the gravitational force exerted.

The output of most clustering algorithms can be the input into our visualization framework. The inputs to the visualization are a k by k distance matrix containing the distance between the k cluster descriptions and for each observation, k columns containing the observation's degree of membership to every cluster that must sum to one. For instance, the distance matrix may contain the Kullback Leibler or Euclidean distances between the cluster descriptions. For a mixture model, the observation's degree of membership to a cluster could be the normalized likelihood, while for K-Means clustering it could be the observation's distance to the cluster normalized by the sum of its distance to all clusters. We focus on applications of clustering in data mining but believe our visualization approach is useful for other clustering applications.

Our approach to visualize clustering results is computationally very efficient. The calculations to visualize a clustering result of thousands of records take only a few seconds on desktop machines. The computational efficiency of the two steps in the approach is of order $O(k^2)+O(ks)$ where k is the number of clusters and s is the number of observations with $s \gg k$. The time to generate the visualization is linear with respect to the number of observations making it suitable to visualize large data sets.

We begin the paper by describing the particle visualization framework in detail. We define how to place the cluster centers in a three-dimensional space and how to place the observations amongst the cluster centers according to an attractive gravitational law. Next, we visualize the UCI [12] churn data set for the purpose of segmentation, discussing and verifying the insights and properties that the visualization provides. We then construct a special law suitable for anomaly detection applications whose usefulness we illustrate on the UCI cars data set. We conclude by describing related work, summarizing our approach and describing potential extensions to our framework.

Throughout this paper, our clustering results are from an EM [4] mixture modeler. An observation's degree of membership to a cluster is its normalized likelihood for that cluster. This paper builds upon our earlier work [5] by amongst other things: refining the basic framework, introducing the idea of density visualizations of the output and verification of the visualization.

Visualizing Cluster Solutions as Particles Affected by Gravitational Forces

Clustering is inherently density estimation in an instance space. The general aim of clustering is to find sub-regions of the instance space where many observations occur. The description of these sub-regions can vary depending on the clustering technique. If we consider the cluster centers as having a large mass and each observation a small mass then a natural graphical view of a clustering solution is as cluster centers pulling on the observations/particles. We wish our visualization to be a *snapshot* of the particle positions, at some instant in time, after the application of gravitational forces.

In our particle visualization approach to clustering results, we first place the cluster (density) centers in a three-dimensional space trying to preserve their spacing in the original d dimensional space. We then place the observations (particles) amongst the centers to reflect the gravitational pull on the particles as represented by the degree of membership that a particle has for each cluster. Throughout this paper, we describe our approach in two dimensions for clarity but in practice use three dimensions.

Placing the Cluster Centers

We first map the concentrated areas of mass (the cluster centers) as points onto the canvas while attempting to preserve the distance spacing that occurs in the original d dimensional space. In our experiments, we use the average Kullback-Leibler (KL) distances between two cluster centers to produce the distance matrix between the cluster centers (D_{Matrix}). We use Multi Dimensional Scaling (MDS) [6][9][7] to place the k cluster centers in a cube whose diagonal length is equal to one. Functionally MDS takes as input a k by k matrix (D_{Matrix}) that contains normalized cluster distances that sum to one. MDS attempts to create a layout of the points in the cube so that the calculated distances between the points (D_{Cube}) are close to those in D_{Matrix} . We initially randomly place the k points and move them while trying to minimize the objective function $|D_{Matrix} - D_{Cube}|^2$. We use a simulated annealing [8] approach with multiple random restarts to find a good local optimum. MDS is a powerful general technique, the type of MDS we implement uses Kruskal-Shepard and Metric scaling according to the classification scheme described by Buja et al [9]. Our MDS algorithm follows in pseudo code, variables are in *Italics*:

```

Place the  $k$  points randomly in the cube
Let the current set of points placements be  $P_{Cube}$ 
 $Err_{old} = \text{MAX\_FLOAT}$  (a very large positive real number)
 $Epsilon = \text{MIN\_FLOAT}$  (a very small positive real number)
While( $Err_{old} > Epsilon$  and Number_Of_Iterations < 100)
     $Err_{old} = \text{CalcError}(D_{Matrix}, P_{Cube})$ 
    Copy  $P_{Cube}$  to  $P'_{Cube}$ 
    Perturb the  $k$  point positions in  $P'_{Cube}$  (see Figure 1)
     $Err_{candidate} = \text{CalcError}(D_{Matrix}, P'_{Cube})$ 
    // Accept new position if new error <= old error
    If( $Err_{candidate} \leq Err_{old}$ ) then  $P_{Cube} = P'_{Cube}$ 
    //Perform a Metropolis test if new error > old error
    If( $Err_{candidate} > Err_{old}$ ) && ( $e^{(-Err_{candidate} + Err_{old})} < \text{rand}(0, 1)$ )
         $P_{Cube} = P'_{Cube}$ 
    Endif
End While
CalcError Function: CalcError(DistancesA, Points)
Calculate DistancesB From Points
Error = 0
For  $i = 1$  to  $k$ , For  $j = 1$  to  $k$ 
    Error += (DistancesA $i,j$  - DistancesB $i,j$ )2
EndFor
EndFor
Return Error

```

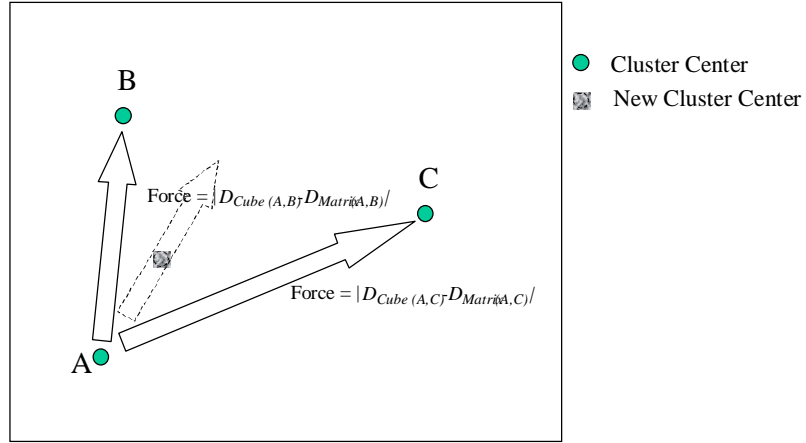


Figure 1: A perturbation of cluster center A where $(D_{Cube(A,B)} - D_{Matrix(A,B)}) > (D_{Cube(A,C)} - D_{Matrix(A,C)}) > 0$.

The MDS algorithm attempts to minimize the overall stress $(|D_{Matrix} - D_{Cube}|^2)$ by asynchronously moving each cluster center. Figure 1 illustrates the perturbation/movement of cluster center A. As the actual distance of A to B and C ($D_{Cube(A,B)}$ and $D_{Cube(A,C)}$) is greater than as expected ($D_{Matrix(A,B)}$, $D_{Matrix(A,C)}$) then A will be moved closer to B and C. Both B and C exert a force whose direction is given by a line connecting their centers to A, the size of the force is given by the difference between the expected (D_{Matrix}) versus actual (D_{Cube}) distances, therefore the force exerted by B is stronger. The resultant vector, which is the sum of the individual vectors, gives the direction of movement for A. The size of movement is a random number that can be as large as the magnitude of the resultant vector. The computational efficiency of this entire step is of order $O(k^2)$ as it is for most MDS algorithms [10]. However, k is typically less than 10.

Placing the Observations (Particles) Amongst the Cluster Centers

We need to place the observations to reflect the gravitational pull on the observations by the cluster centers. Each observation belongs to every cluster with some degree of membership. We now describe the observation placement shown in Figure 2. Firstly, we place the observation near its most likely cluster, cluster A, at a distance proportional to $(1 - \text{Pr}(x | \theta_A))$. $\text{Pr}(x | \theta_A)$ is observation x 's degree of membership in cluster A. This provides a circle on which to place the observation. Then every cluster other than A exerts a force on an observation equal to how likely the observation belongs to that cluster. The direction of the force is a straight line between the center of cluster A and the center of the other cluster. The intersection of the resultant vector (sum of all individual forces) and the circle gives the final observation placement. We repeat this procedure for each observation.

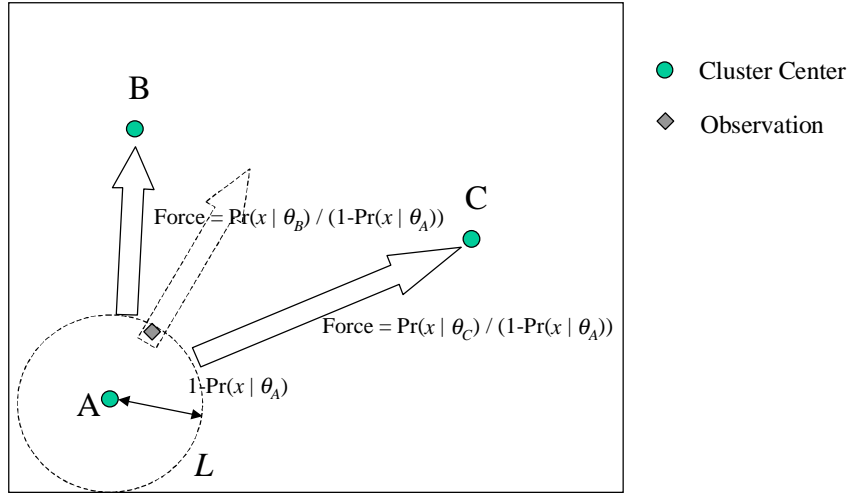


Figure 2: Placement of observation, x , where $\Pr(x | \theta_A) > \Pr(x | \theta_B) > \Pr(x | \theta_C)$. The direction of the resultant vector is the sum of the component vectors.

The computational efficiency of this step is of order $O(ks)$ where k is the number of clusters and s is the number of observations with $s \gg k$. The cost of this step scales linearly with the number of observations.

Properties of the Visualization

Our method of placing the cluster centers and particles produces a visualization with these properties:

- The distance between clusters is an indication of their similarity.
- The distance from an observation to a cluster reflects its degree of membership.
- A cluster's shape and opaqueness reflects the observation's degrees of membership to the cluster.
- The cluster center placement is stochastic, particle placements is deterministic.
- Adjacent observations have similar combinations of degrees of membership.

The last two points are worth further discussion. Point d) means that different seeds of the random number generator can produce different cluster center placements for the same clustering solution. However, for a given clustering solution and set of cluster center placements the positions of the particles amongst the cluster centers will always be the same. Point e) means that adjacent observations have very similar combinations of degrees of membership to the clusters. Strictly speaking, it does not mean that those observations' attribute values will be similar, but they usually are. We did not use the simpler approach of summing all component vectors to place the observations, as it would result in a visualization with different properties. In such an approach the position of particle x would be:

$$\sum_{j=1 \dots k} \Pr(x | \theta_j) C_j, \quad \text{Where } C_j \text{ is the location of the center of cluster } j. \quad (1)$$

Properties b), c) and e) would not hold if we used this approach and would result in the undesirable situation where adjacent observations have very little in common. Consider three cluster centers that are on a straight line of unit length. Their positions along the line are at 0, 0.5 and 1. Let the degree of membership to the three clusters be 0.34, 0.33 and 0.33 respectively, then the position of the observation using this simpler approach would be at approximately 0.5. However, if the degrees of membership became

0.5, 0, and 0.5 respectively, then the position of the observation would also be 0.5. This is a common problem/property with visualization techniques such as star-coordinates that use the sum of component vectors to position an observation [11]. By placing an observation always around its most likely cluster, we introduce a constraint that overcomes this problem.

Experimental Methodology and Results

How users will act upon and interpret the visualization will vary making verification of the usefulness of this approach both important and difficult. We begin our experiments by visualizing the churn data set using the previously described framework for the purpose of segmentation. We claim that the visualization can easily convey much of the information contained in a clustering solution. We verify this by showing that insights from the visualization reflect the quantitative properties of the clustering solution. We also hope that the visualization will be able to convey information that is not evident from the standard textual description or statistical summaries of the clustering results.

Next, we use the cars data set (with the *origin* variable removed) to illustrate how a small variation of the basic framework emphasizes outlier observations which is useful for anomaly detection. We show that although there are many anomalies, there are clearly different classes/types of anomalies some that are more interesting than others.

Our basic method produces a placement of particles amongst the cluster centers in a unit cube. We can display the results in a number of ways. We choose two popular methods to display the results: density visualization and scatter visualization. In the former, we visualize the density of the particles, in the later we represent each observation as a sphere. The density visualization breaks the unit cube into a series of very small regions. The number of observations located within a region determines its opaqueness. Which type of visualization is used depends on the purpose of the clustering exercise. For segmentation, where the major focus is the clusters, a density plot is more applicable. For anomaly detection, where the individual observations are of interest, a scatter visualization is more useful.

We demonstrate our general-purpose framework using the UCI churn data set [12] with the *state*, *churned* and *telephone area code* variables removed. Figure 3 and Figure 4 shows the density plot for the churn data set.

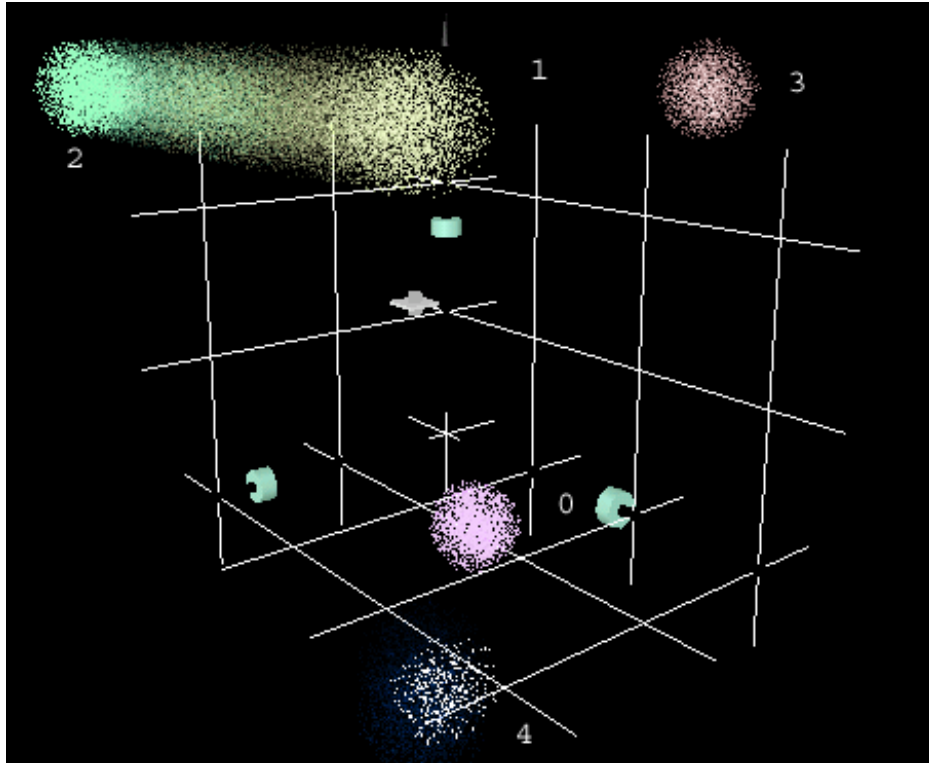


Figure 3: Density based visualization of churn data set with five clusters using the general framework. The Cluster IDs are next to the clusters.

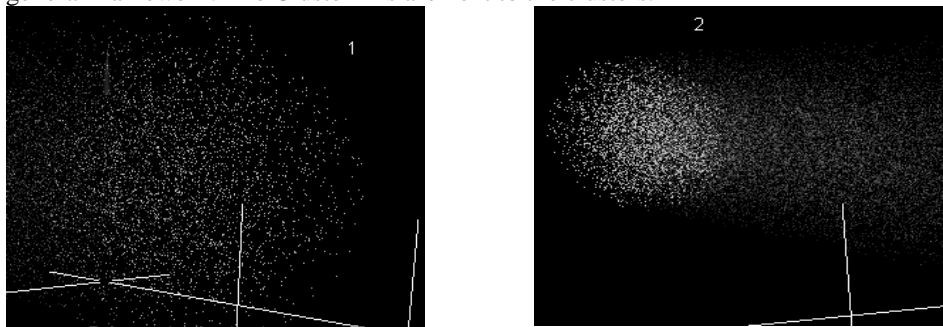


Figure 4: Zoom-in of clusters 1 and 2 centers

Interpretation, Verification and Uses

We can attempt to verify the visualization accurately represents the clustering solution by comparing important statistical metrics against insights derived from the visualization. From the visualization we can derive the following insights:

- a) Cluster 2 and Cluster 1 are more similar than Clusters 0 and 3 because they are closer together.
- b) Clusters 0, 3, and 4 are well-defined autonomous clusters due to their compact representation.

- c) Cluster 2 is the largest cluster as it occupies a large part of the canvas and is the most opaque.
- d) Clusters 0 and 4 are similar but do not share many observations.
- e) Clusters 1 and 2 are similar and share many observations that mesh together.
- f) Cluster 2 is more densely packed than cluster 1 as it is more opaque.

We calculate the KL distances in the higher dimensional instance/data space and the Euclidean distances between cluster centers in the three-dimensional space. We expect that the spacing between the cluster centers in the three-dimensional space should reflect the spacing in the higher dimensional space. We find:

$$EuclidDistance(0, 3)=0.69, MeanKLDistance(0, 3)=0.59$$

$$EuclidDistance(1, 2)=0.62, MeanKLDistance(1, 2)=0.54.$$

In both the higher and lower dimensional spaces, $D(\text{Cluster } 0, \text{Cluster } 3) > D(\text{Cluster } 1, \text{Cluster } 2)$ this illustrates that insight a) is correct.

From Table 1 we see observations whose most likely cluster is 0, 3 or 4 have degrees of membership to their most likely cluster that are on average very high with low standard deviations. These observations belong very strongly to their most likely cluster and hence insight b) is correct. Cluster 2 (2608 observations) contains the most observations, is the most “opaque” cluster and takes up a relatively large space in the visualization showing that insight c) is correct. Though cluster 1 occupies a large part of the canvas, it is not as densely packed. The textual descriptions of what differentiates clusters 0 and 4 illustrate that cluster 4 is a specialization of cluster 0 and hence they should be placed adjacent to each other showing that insight d) is correct.

| Clust. Id | Mean | Stddev | What Differentiates the Cluster From the Population | Churned Percentage | Size |
|-----------|------|--------|---|--------------------|------|
| 0 | 0.99 | 0.001 | voice_mail_plan:Different, | 5% | 996 |
| 1 | 0.90 | 0.14 | #vmail_messages:Very High, total_intl_minutes:Low, total_intl_charge:Low | 18% | 1070 |
| 2 | 0.95 | 0.110 | No Significant Difference | 17% | 2608 |
| 3 | 0.94 | 0.109 | Voice_mail_plan:Different, | 4% | 186 |
| 4 | 0.95 | 0.102 | number_vmail_messages:High, Voice_mail_plan:Different, #vmail_messages:Very High, total_eve_minutes:Low, | 7% | 140 |

Table 1: Summary statistics of the clusters. The mean and standard deviation are of the degree of membership to the observations’ most likely cluster, 14.1% of the entire population churned.

We can see from Table 2 that for observations whose most likely cluster is 1 or 2, the second most likely cluster is overwhelmingly cluster 2 and cluster 1 respectively, thereby showing that insight e) is correct. Observations whose most likely cluster is cluster 2 have a higher mean degree of membership and lower standard deviation than cluster 1 showing that insight f) is correct.

| | Mean DOM Cluster 0 | Mean DOM Cluster 1 | Mean DOM Cluster 2 | Mean DOM Cluster 3 | Mean DOM Cluster 4 |
|-----------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Cluster 1 | 0.0001 | 0.8989 | 0.1010 | 0.0000 | 0.0000 |
| Cluster 2 | 0.0000 | 0.0544 | 0.9456 | 0.0000 | 0.0000 |

Table 2: The mean degree of membership (DOM) to a cluster by observations whose most likely cluster is 1 or 2.

Using the Visualization to Go Beyond Verification

Cluster 2 and 4 look fundamentally different in the visualization, yet their summary statistics (mean and standard deviation) are very similar (see Table 1). If we only had these typical statistical summaries of the clusters, we would think the two clusters' distributions of degrees of membership are not different, but the visualization illustrates they are. This adds the following insight to investigate:

- g) Cluster 2 and Cluster 4 have very similar statistical summaries, yet are different, how?

Table 3 shows that observations whose most likely cluster is cluster 2 have a mean entropy amongst their degrees of membership that is less than those observations whose most likely cluster is cluster 4. The degree of belonging to clusters other than the most likely cluster is more uniformly distributed for cluster 4 than cluster 2. This means the shape of cluster 4 is more circular than the elongated cluster 2 as the forces on the particles is more uniform. We could infer this from the visualization, but at the very least the visualization tells us that the two clusters are somehow different and to perhaps investigate these differences.

| Cluster Id | 0 | 1 | 2 | 3 | 4 |
|--------------|------|------|------|------|------|
| Mean Entropy | 0.00 | 0.14 | 0.08 | 0.11 | 0.11 |

Table 3: Entropy of the degrees of membership by the observation's most likely cluster using base 10 logarithms

Uses of the Visualization

The visualization has potential uses in the different phases of a data mining project. In this example we illustrate how the model builder can use the visualization, in our next example we illustrate its use as a presentation tool to the end-user. Data mining is a highly iterative and time-consuming process that tries to produce a final useful model by conducting a series of experiments with slightly different parameters and/or variables. We can use the visualization as a first contact point to the experiment results to quickly and easily see interesting insights and phenomenon to investigate. We believe this will speed up the model building process. In this example, we see from Table 1 that over 630 of the 707 customers who churned are in clusters 1 and 2. From the visualization we note that these two clusters share many observations whilst the remaining clusters are quite autonomous. A valid next step is to create a filter expression that isolates the observations in clusters 1 and 2 and then perform more clustering experiments on these observations to determine if this sub-population can justify more than two clusters. After we divide this sub-population into autonomous clusters, we could then build predictive models for each segment.

A Specialization for Anomaly Detection

Anomaly detection has uses in many different applications: credit card fraud, data cleaning, and identifying material flaws [5]. In most applications the basic steps remain the same:

- 1) Identify normality by calculating some "signature" of the data.
- 2) Determine some metric to calculate an observation's degree of deviation from the signature.
- 3) Set a criterion, which if exceeded by an observation's degree of deviation makes the observation anomalous.

In clustering-based anomaly detection, the signature is the clusters found in the data. The measure of deviation from the signature is the degrees of membership of an observation to the clusters. A typical criterion is that if an observation does not belong to

any one cluster with a degree of membership greater than the *minimum degree of membership* it is anomalous.

Anomaly detection applications are typically user intensive particularly in applications like insurance fraud where the cost of incorrectly labeling a case anomalous is great. Our aim is to convey information that aids the analyst user in exploring anomalous observations. We need to convey what observations are anomalous and why. We explain our variation of the general framework that achieves this.

The first step of placing the cluster centers in the three-dimensional space is identical to our general framework. However, for particle placement we adopt a specialization of the general framework. We introduce the idea of the *radius of gravitational effect*. The radius of gravitational effect places a sphere around the cluster centers. The other clusters' gravitational pull does not affect those observations falling within the sphere, but affect those that fall outside the sphere. This has the desirable effect of clearly identifying those observations that belong very strongly to a cluster. Usually the radius of gravitational effect is equal to $(1 - \text{minimum degree of membership})$, so that gravitational forces only affect anomalous observations. We change the distance an observation is placed from its most likely cluster to be $(1 - \text{degree of membership})^2$. We randomly place particles that fall within the radius of gravitational effect on the surface of a sphere whose radius is proportional $(1 - \text{the degree of membership})^2$ we show this diagrammatically in Figure 5. With these changes to our visualization, we effectively hide the non-anomalous observations, clearly show the anomalies, and over-emphasize the gravitational effect from clusters other than the most likely.

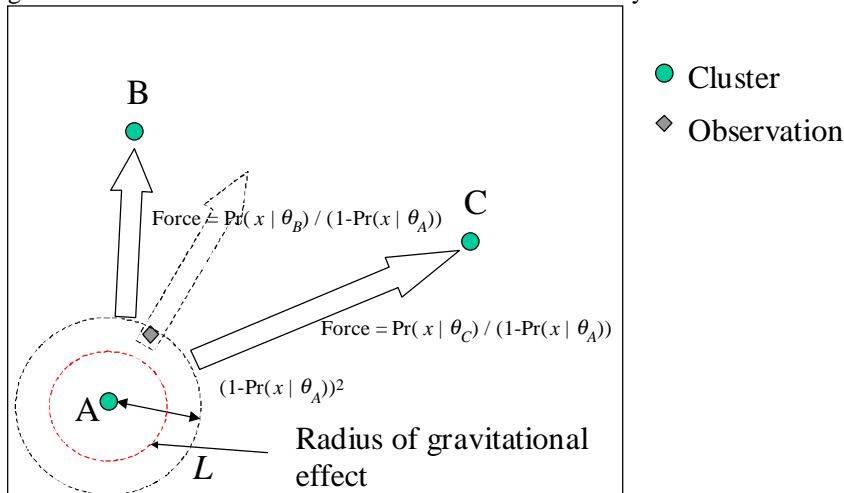


Figure 5: Placement of observation x for anomaly detection variation where $\Pr(x | \theta_A) > \Pr(x | \theta_B) > \Pr(x | \theta_C)$. If x had fallen within the radius of gravitational effect, we randomly place it on a circle of radius $(1 - \Pr(x | \theta_A))^2$.

Figure 6 shows our scatter visualization of anomalies for the UCI cars data set for four clusters that Table 4 describes. Note that the non-anomalous observations obscure each other due to the gravitational law in use. This is desirable in this application as our focus is the outliers. Observations that belong strongly to a cluster are near its central region. Anomalies do not belong strongly to any one cluster and tend to be *between* the cluster masses. The farther away an observation is from any of the cluster centers, the more anomalous it is.

We can see that there are many anomalies that lie between clusters 1 and 3 and clusters 0 and 3. There are two very interesting anomalies, one that lies in the center of the visualization and one anomaly that lies between clusters 2 and 3. We shall focus on these anomalies to see what makes them special.

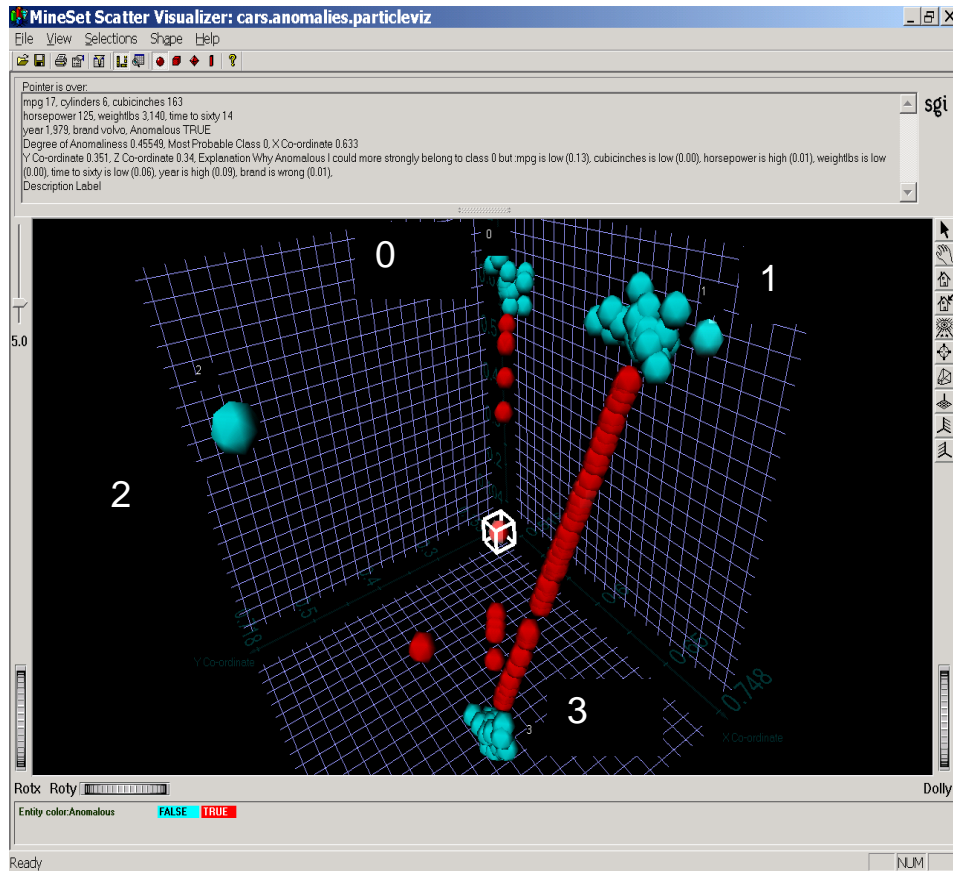


Figure 6: Visualization of anomalies generated from the UCI cars data. Anomalies are colored red (darker color). Cluster IDs are next to each cluster

| Cluster ID | Cluster Description | Size |
|------------|---|------|
| 0 | Larger, heavier and less fuel efficient six cylinder cars | 69 |
| 1 | Small four cylinder cars, contains many Japanese cars | 144 |
| 2 | Eight cylinder cars that have large engines and are very heavy | 102 |
| 3 | Larger four cylinder and smaller six cylinder cars, contains many European cars | 76 |

Table 4: A description of the typical observations found in each cluster for the UCI Cars data set.

The selected observation in Figure 6 is the most anomalous as it occupies the center of the visualization, it is also the observation with the greatest entropy amongst its degrees of membership. From the visualization we see it belongs most strongly to clusters 0 (six cylinder cars), and 3, which contains many European cars. From Figure 7 we can see that the anomaly is a Volvo that is different to all other Volvos as it has six cylinders **and** is quite fuel inefficient. It is also unusual because it is a European six-cylinder car but has many similar properties to the larger American six cylinder cars found in cluster 0.

| row # | mpg | cylinders | cubicinches | horsepower | weightlbs | time to sixty | year | brand | Anomalous | Degree of Anomaliness | Most Probable Class |
|-------|------|-----------|-------------|------------|-----------|---------------|-------|-------|-----------|-----------------------|---------------------|
| 274 | 17 | 6 | 163 | 125 | 3,140 | 14 | 1,979 | volvo | TRUE | 0.45549 | 0 |
| 120 | 19 | 4 | 121 | 112 | 2,868 | 16 | 1,974 | volvo | FALSE | 0 | 3 |
| 356 | 30.7 | 6 | 145 | 76 | 3,160 | 20 | 1,982 | volvo | FALSE | 0 | 3 |
| 206 | 20 | 4 | 130 | 102 | 3,150 | 16 | 1,977 | volvo | FALSE | 0 | 3 |
| 76 | 18 | 4 | 121 | 112 | 2,933 | 15 | 1,973 | volvo | FALSE | 0 | 3 |
| 178 | 22 | 4 | 121 | 98 | 2,945 | 15 | 1,976 | volvo | FALSE | 0 | 3 |

Figure 7: Description of Volvos in cars dataset. The anomaly selected in Figure 6 is in row 274.

By zooming-in and spinning the visualization, we can identify different anomalies and obtain an understanding why they are anomalous. The observation highlighted in Figure 8 belongs most strongly to cluster 3 but also has properties similar to observations in cluster 2.

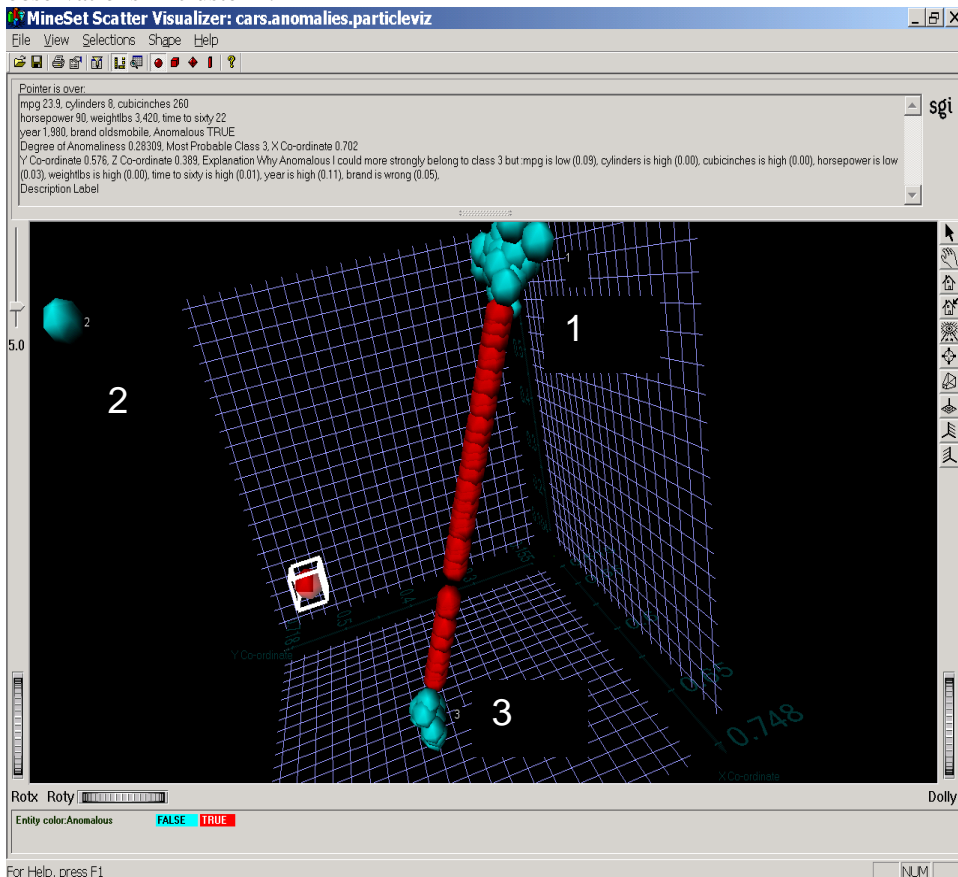


Figure 8: Visualization of anomalies generated from the UCI cars data. Anomalies are colored red (darker color). Cluster IDs are next to each cluster. This is the same visualization shown in Figure 6, but with the diagram spun around the z-axis 90 degrees to the left. Cluster 1 obscures Cluster 0

We can see from Figure 9 that the selected observation is the only Oldsmobile that is anomalous. The newer, less than eight cylinders and more fuel-efficient (higher mpg) Oldsmobiles typically belong to cluster 3. The Oldsmobiles assigned to cluster 2 have 8 cylinders with a very low mpg. This outlier Oldsmobile falls between the two types of Oldsmobile having the properties of both clusters. It is an eight-cylinder car (cluster 2), but is quite fuel efficient/high mpg (cluster 3). It has a large cubic inch engine (cluster 2) but has a horsepower similar to those Oldsmobiles found in cluster 3.

| row # | mpg | cylinders | cubicinches | horsepower | weight... | time to sixty | year | brand | Anomalous | Degree of Anomal... | Most Probable Class |
|-------|------|-----------|-------------|------------|-----------|---------------|-------|------------|-----------|---------------------|---------------------|
| 360 | 26.6 | 8 | 350 | 105 | 3,725 | 19 | 1,982 | oldsmobile | FALSE | 0 | 2 |
| 248 | 19.9 | 8 | 260 | 110 | 3,365 | 16 | 1,979 | oldsmobile | FALSE | 0 | 2 |
| 106 | 12 | 8 | 350 | 180 | 4,499 | 13 | 1,974 | oldsmobile | FALSE | 0 | 2 |
| 221 | 17 | 8 | 260 | 110 | 4,060 | 19 | 1,978 | oldsmobile | FALSE | 0 | 2 |
| 124 | 11 | 8 | 350 | 180 | 3,664 | 11 | 1,974 | oldsmobile | FALSE | 0 | 2 |
| 69 | 12 | 8 | 350 | 160 | 4,456 | 14 | 1,973 | oldsmobile | FALSE | 0 | 2 |
| 299 | 23.9 | 8 | 260 | 90 | 3,420 | 22 | 1,980 | oldsmobile | TRUE | 0.28309 | 3 |
| 382 | 38 | 6 | 262 | 85 | 3,015 | 17 | 1,983 | oldsmobile | FALSE | 0 | 3 |
| 271 | 23.8 | 4 | 151 | 85 | 2,855 | 18 | 1,979 | oldsmobile | FALSE | 0 | 3 |
| 306 | 26.8 | 6 | 173 | 115 | 2,700 | 13 | 1,980 | oldsmobile | FALSE | 0 | 3 |

Figure 9: Description of Oldsmobiles in cars dataset. The anomaly selected in Figure 8 is in row 299.

Related Work

In this section we discuss relevant work in the field of visualizing clustering results. We begin by highlighting what differentiates our work and then summarize relevant previous work. We conclude the section by describing the unsuitability, in their current form, of general-purpose approaches like force directed graphs and parallel coordinates to visualize large non-hierarchical clustering results.

Much of the work in the cluster visualization field has been for visualizing *hierarchical* (often-called agglomerative) clustering [13]. Hierarchical clustering is typically bottom-up clustering using exclusive assignment. Since our visualization is for non-hierarchical clustering with multiple degrees of membership, it is not directly comparable to previous hierarchical clustering visualizations using exclusive assignment. Our approach could visualize a set of hierarchically clustered observations if fractional assignments to each cluster were somehow calculated, which to our knowledge has not been achieved.

However, several pieces of prior work use the idea of placing observations around the cluster centers at a distance equal to the degree of membership to only one cluster. What differentiates our work is:

- 1) We attempt to visualize an already established clustering solution, rather than providing decision support to help in forming the clustering solution as others have [14].
- 2) Our visualization is specifically for non-hierarchical clustering and we place our observations subject to the constraint of multiple degrees of membership, not just one.

- 3) The placement of cluster centers so that similar clusters are adjacent and different clusters are far away
- 4) Our principled use of attractive laws to place observations around the cluster centers.
- 5) The scalability of our approach to visualize large data sets.

We now survey related work. There are many examples of work that use visualization to provide decision support for creating clustering solutions. In [15] the authors use a scatterplot view of cluster centers that they obtain by hierarchical clustering, with interactive control of the splitting criteria to increase or decrease the number of visible clusters. There has been some work in visualizing non-hierarchical *exclusive* assignment clustering solutions in the text-processing field [16]. In this work each cluster occupies a fixed size region and the documents, represented as points, are placed at a distance proportional to their similarity to only a single cluster center. Leuski and Allan [17] propose value adding to the results of an information retrieval query by placing the documents, represented as spheres, in two or three-dimensional space according to their degree of similarity. This is in principle similar to the way we place cluster centers. The application of this idea to all observations in the data set would ignore the results found by the clustering algorithms and would be computationally very expensive (in the order of $O(s^2)$) where s is the number of observations.

We could visualize non-hierarchical clustering results using general-purpose visualization approaches such as parallel coordinates [18], force directed graphs [19] and MDS [9] (to position the observations, not just the cluster centers). In force directed graph techniques the nodes are physical objects that are subject to various forces. The aim is to re-position the nodes to represent best these forces. While we could use these three techniques, they suffer from three problems that make them (in their current form) unsuitable for large-scale data mining problems.

- 1) Scaleable calculations for many observation positions.
- 2) Scaleable visualization of many observations.
- 3) Ignoring the model based nature of K-Means clustering and mixture modeling.

Most work in metric scaling MDS and force directed graphs are limited to at most one hundred objects of interest. Above this limit, the combinatorial nature of these techniques and the calculation of the “error” mean they become computationally very expensive. Calculating the position for an observation in parallel coordinates is computationally very efficient, however visualizing thousands of observations becomes difficult as the lines merge into a blob. Finally and most importantly, our aim is to visualize the clustering model found. To use force directed graphs and MDS techniques to position all observations would mean ignoring the model, as we would need to translate an observation’s degree of membership to the cluster to a measure of similarity to all other observations.

Conclusion and Further work

We have presented a general framework to visualize clustering results where we represent the observations as particles affected by gravitational forces. To our knowledge, visualizing clustering results as particles affected by many gravitational forces is unique and naturally fits well with the non-hierarchical clustering philosophy. In our framework, we place the cluster centers in a three-dimensional cube such that similar clusters are adjacent and dissimilar clusters are far apart. We then place the observations amongst these centers to reflect the degree of membership that each observation has for the clusters. Our approach is computationally efficient; calculations for a 5000-observation data set took less than ten seconds on a desktop machine

(Pentium II 500MHz). The computational efficiency of the approach is of order $O(k^2)+O(ks)$ where k is the number of clusters and s is the number of observations with $s \gg k$. The time to generate the visualization is linear with respect to the number of observations for a fixed number of clusters making it amenable to application to large data sets.

We verified that the insights found from visualization are consistent with statistical summaries of the clusters and that the visualization has desirable properties. Our information visualization can present a lot of information and we propose a specialization for anomaly detection that focuses on information useful for that purpose. We believe that specializations of our framework can be created for other data-mining uses of clustering and other uses of clustering such as information retrieval.

Our framework to place cluster centers and observations has many potential uses and we generally describe two. Firstly, when displayed as a density visualization it aids the model builder to quickly form insights into a clustering result. Data mining is typically an iterative process of conducting experiments on a data set by applying a technique (like clustering) with different parameters and variables/columns. Interpreting the result of each experiment determines the next experiment to conduct. We believe use of our visualization may shorten the data mining process as the analyst can quickly interpret the clustering solution. The anomaly detection variation displayed as a scatter visualization presents the results to the end user and allows them to see the different types of anomalies and focus on those that are most interesting.

A natural extension to our ideas would be to incorporate brushing between other visualization tools such as parallel coordinates. We believe that linking our visualization and parallel coordinates holds particular promise in the field of anomaly detection. A further extension to our work would be to include the motion of the particles around the cluster centers into the visualization. This could better convey the information in a clustering solution. Our idea is to essentially model the particles as being affected by the gravitational laws of attraction, one could modify this idea to follow other attraction/repulsion laws such as those found in the fields of electricity and magnetism.

Acknowledgements

Thanks for Marc Hansen and Jose Renteria of the MineSet Visualization team for their thoughts on this work and comments on the paper. Thanks to Professor Matthew Ward of WPI for suggesting using MDS and suggesting related work.

References

- [1] R.S. Michalski, R. Stepp, "Learning from Observation: Conceptual Clustering," in R.S. Michalski, J.G. Carbonell, T.M. Mitchell editors, *Machine Learning: An Artificial Intelligence Approach*, C A: Morgan Kaufmann, 1983.
- [2] M.J. Berry and G. Linoff, *Data Mining Techniques For Marketing, Sales and Customer Support*, Wiley Computer Publishing, 1997.
- [3] J. G. Campbell, C. Fraley, F. Murtagh and A. E. Raftery, "Linear Flaw Detection in Woven Textiles using Model-Based Clustering", *Pattern Recognition Letters*: 18(1997):1539-1548
- [4] P. Dempster et al, Maximum Likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society B*, Vol 39 pages 1-39, 1977.
- [5] I. Davidson and M. Ward, "A Particle Visualization Framework for Clustering and Anomaly Detection", ACM KDD 2001 Workshop on Visual Data Mining, San Francisco, 2001.
- [6] Bentley and Ward, "Animating Multidimensional scaling to visualize N-Dimensional data sets", *Proceedings of Information Visualization 1996*, pages 72-73, 1996.
- [7] J. Kruskal and M. Wish, *Multidimensional Scaling*, Sage Publications, London, 1978
- [8] E. Aarts, and J. Korst, *Simulated Annealing and Boltzmann machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing*, Anchor Press, 1989.
- [9] D. Buja, M. Swayne, N. Littman, vis: Interactive Data Visualization with Multidimensional Scaling, to appear in the *Journal of Computational and Graphical Statistics*
- [10] B.Lerner, H.Guterman, M. Aladjem, I.Dinstein and Y.Romem, "On pattern classification with Sammon's nonlinear mapping- An experimental study", *Pattern Recognition*, Vol.31, No 4, pp.371-381, 1998.
- [11] E. Kandogan, Visualizing Multi-dimensional Clusters, Trends, and Outliers Using Star Coordinates, KDD 2001, pp. 107-116, 2001.
- [12] C.J. Merz and P.M. Murphy, UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
- [13] T. Clifford, W. Stephenson, An introduction to numerical classification, Academic Press 1975
- [14] T. Sprenger, R. Brunella, and M. Gross, "H-BLOB: a hierarchical visual clustering method using implicit surfaces," *Proc. Visualization 2000*, pp. 61-68.
- [15] G. J. Wills, "An interactive view for hierarchical clustering", *Proceeding of Information Visualization '98*, pp. 26 - 31, 1998.
- [16] M. Song, "BiblioMapper: a cluster-based information visualization technique," *Proc. Information Visualization '98*, pp. 130 - 136, 1998.
- [17] J. Leuski, J. Allen, "Lighthouse: Showing the Way to Relevant Information", *Proceedings of Information Visualization 2000*, pages 125-129, 2000.
- [18] Y. Fua, M. Ward, and E. Rundensteiner, "Hierarchical parallel coordinates for exploration of large datasets," *Proc. Visualization '99*, pp. 43-50, 1999
- [19] T. Fruchterman and E. Reingold, *Graph Drawing by Force-Directed Placement*, *Software - Practices and Experience*, 21(11):1129-1164, 1991.