ClusterGram – A Visualization Tool to Aid Data Clustering and Entity Resolution

Ren Bauer

University of North Carolina at Chapel Hill

Abstract When performing entity resolution and clustering large datasets, it can be difficult to get a feel for where to begin. Even after the creation of a distance matrix, determining parameters for generating optimal clusters can be a daunting challenge. ClusterGram is a light java application that parses distance matrices in an attempt to provide a concise visualization from which useful information can be obtained. In specific, the application strives to provide approximate distance measure cutoffs which can be used as guidelines when clustering data and performing entity resolution.

1 Introduction

In many fields, the process of using a distance metric to establish relationships between entities is an invaluable step towards separating data into meaningful clusters. However, even after the application of such a metric, it can be difficult to determine which data points should be considered in the same class. Additionally, data entry errors can often lead to the same entity being represented by multiple data points. Ideally, these data points will be separated by a particularly small distance as calculated by the metric. Entity resolution takes advantage of this to resolve nearby data points that likely represent the same entity. Once again, determining the ideal distance at which entities should be resolved is often a non-trivial problem. As data sets grow larger and larger, there is an increasing need for tools to assist researchers in these areas.

ClusterGram attempts to provide a lightweight application to aid in these processes. To accomplish this, it employs a visualization technique akin to a circular dendrogram. However, the representations of distances between entities both along the perimeter and radially from the center allow this unique approach to offer assistance beyond that of other methods. This paper provides an overview of the features provided by the software, followed by a description of the data mining techniques used to generate the display. Following that, it will offer examples that attest to ClusterGram’s effectiveness, and finally suggestions for future work.
2 Software Features

The features provided in ClusterGram are designed to provide rapid, intuitive assistance to researchers who are searching for guidance in performing clustering and entity resolution. The following is a description of eight major features which help make ClusterGram a powerful tool.

1. Input
As input, ClusterGram natively parses data matrices in the form of comma separated value files. The first row and column are automatically used to name the entities. This format requires little if any preprocessing, and allows data directly output from distance generating algorithms to be fed into the visualization tool. This provides an advantage over other tools, which requires the tree structure to be provided in the input.

2. Data Metric
Because the input provides no information on tree structure, ClusterGram generates its own distance trees. In order to do so, it employs a nearest neighbor algorithm. Within this process, there are multiple options for calculating the distance between clusters of data points. ClusterGram provides support for three of the most commonly used metrics: min, mean, and max distance (excluding mode). The ability to select the best metric for a
given data set is an additional problem for which this software can provide support. The effects of using these different metrics on a single data set can be seen in section i of appendix A.

3. Zoom and Pan

When viewing large datasets, the initial view will likely be insufficient for determining optimal values for clustering and resolution distances. With the ability to zoom up to 27x natively, and pan in any direction (via mouse dragging), it even a very large dataset can be viewed at a per-entity scale. An example of this feature can be viewed in section ii of appendix A.

4. Rotate

After zooming in to view visualizations in more detail, panning may be a taxing method by which to manipulate the graph. Rotation gives the ability to zoom in on one section of the perimeter, and rotate the tree through 720° in order to view each entity in detail.

5. Font Size

ClusterGram provides the ability to increase or decrease the font size of data points displayed on the tree. This feature may seem trivial, but in certain cases it is necessary to improve the readability of entities.

6. Guide Clustering

One of the integral features of ClusterGram, the ability to provide aid in determining an ideal cluster distance is achieved through the employ of an expandable circular filter. When the perimeter of this circle intersects with a leg of the tree, that branch is cut off from the surrounding nodes in order to form a new cluster. These clusters are displayed visually by coloring the entities according to the clusters in which they are included. From the user’s perspective, the ClusterGram provides a two-fold output in order to provide optimal information. First, it provides a count of the number of clusters displayed at the current setting. This is useful as a ballpark for the number of clusters a researcher would like to see output from a clustering algorithm. Additionally, the approximate distance between clusters is provided, a useful metric to use as input for clustering algorithms which use that information to define clusters. An example of this feature in use can be seen in section iii of appendix A.

7. Guide Entity Resolution

Another integral feature of ClusterGram is the ability to assist researchers in the resolution of entities. Graphically, entities that are within the resolution threshold are combined to the same point, and tree branches beyond their mutual node are hidden from the visualization. This allows users to visually recognize the best threshold at which to perform entity resolution. This distance is then displayed to the user, so it can be employed in an entity resolution algorithm. An example of this feature in use can be seen in section iv of appendix A. Note: Although entities in this example are represented by integers, any string given in the initial distance matrix can be used to identify entities.

8. Visualize Distances

Unlike other software that offers data visualization in radial patterns, ClusterGram strives to provide visual representations of the distances between data points. Whereas many methods display end nodes as equidistant along the perimeter of the
display, ClusterGram provides a clear distinction between data points that are near to each other and those with more distance between them. Although it is impossible to provide an accurate representation of the distance from every point to every other point in a single dimension, the employed metric is quite effective locally. Visually, a solid colored wedge is used to separate the far left of the tree from the right, in an effort to signify that the distance across that gap does not represent the distance between the two data points. Without this wedge, users might assume that the graph is entirely cyclic, and get the wrong impression of distance between the far sides of the display. Additionally, distance information is employed when determining radial distance of the arcs connecting branches. In other visualizations, these arcs are equidistant based on the radius of the visual and the maximum depth of the tree. Without these built in distance visualizations, the function of the clustering and resolution tools would be extremely limited.

3. Evaluation

In order to appropriately evaluate ClusterGram’s effectiveness, one must view its ability to guide both clustering and entity resolution. In order to evaluate clustering, a distance matrix generator was employed to create both data sets with obvious nodes and random data, and the visualizations created by ClusterGram were analyzed. In the case of entity resolution, a real world data set with multiple known disjoint entities (as judged by a human) was analyzed. In non-related experiments, one researcher used ClusterGram to determine the optimal resolution threshold, and another used more standard methods. On top of these evaluations, one must consider runtime a factor, as when viewing large datasets a poor runtime can pose a significant barrier to utility.

1. Clustering

First, to ensure the clustering algorithm employed in ClusterGram was able to provide basic performance, data points with obvious clusters were split into groups using solely the software’s circular delimiter. Data was created using a two dimensional point generator, using basic Euclidean distance as a distance metric. Once these data sets had been visually split into clusters using ClusterGram, the colors were transferred to the original data so the results could be evaluated. Note all shown evaluations are performed using the ‘mean’ distance metric.

In this example, ClusterGram’s ability to recognize obvious clusters is apparent (Note: the two blue clusters are recognized as distinct). However, it is also important for the algorithm to be effective on data in which the clusters are non-trivial. In order to evaluate this functionality, both random data sets and data sets with inherently difficult to cluster were visualized in ClusterGram.
In this example, ClusterGram effectively clusters a random data set of two hundred points.

Here, a data set that exhibits snaking, a line of points that are extremely close together and sometimes cause abnormal clustering results, is visually split via ClusterGram. In this case, one can see the effectiveness of the software’s visual clustering even very difficult cases.

2. Entity Resolution

In order to test entity resolution, ClusterGram was used to view a hand-picked data set of twenty nine entities. This set consisted of a sample of real life data which had been distanced on a scale of 0 (definitely the same entity) to 1 (no chance of being the same entity) using an experimental method. More specifically, the initial data consisted of first name, last name, date of birth and encrypted social security number for several subjects. However, the set included multiple data entry errors which, as judged by a human, resulted in multiple data points representing the same entity. After applying the distance metric, defining an entity resolution threshold was necessary in order to rectify these errors. In analyzing this data, one researcher utilized ClusterGram to find an optimal threshold of around 0.11. Using alternative methods, including trial and error and analyzing the differential relationship between threshold value and entities resolved, another researcher similarly theorized an optimal value between 0.10 and 0.13. This example, the graph for which was provided on page two, shows the real life applicability of ClusterGram’s entity resolution assistance.

3. Runtime

When performing any computation, a tradeoff between runtime and effectiveness must be considered. Because ClusterGram takes distance matrices as input instead of pre-built trees, as many other visualization tools, the software must use significant resources to create a tree before it can be displayed. Currently, ClusterGram employs a nearest neighbor algorithm to turn the array of distances to a distance sensitive tree. To judge runtime, data sets of multiple sizes were displayed using ClusterGram. To avoid external factors, these tests were run on the same machine, using the previously explained matrix generator and random data. Performance for data sets on the order of several hundred points was promising, on the scale of many seconds. However, a larger data
set with around 2300 data points took much longer, at around eight to ten minutes. This provides an issue for the use of ClusterGram to view very large data sets, on the order of tens of thousands, which is not uncommon for large data mining projects. However, it is likely a random sample of such data sets could be visualized using ClusterGram, and useful data could still be extracted using a multiplier. Additionally, once the tree was generated there was little to no speed decrease in the manipulation of the visualization. The resulting output of the large data set can be viewed in section ii of appendix A.

4. Future Work

Based on these evaluations, one can conclude that ClusterGram has effectively achieved its initial goals. This paper offers three suggestions for future work which would expand ClusterGram’s functionality to provide additional use. First is the ability to read in data that already includes detailed entity information, and display this alongside attributes generated by the software. Second, the ability to output the result of clustering and entity resolution being performed on the input. Finally, runtime optimization could increase the applicability of ClusterGram to larger data sets.

1. Additional Input

If it were possible to provide ClusterGram with attribute-level information on each entity, the software could display this back to the user at runtime. This input could take the form of another comma separated value sheet which listed entities and their attributes that would be provided alongside the distance matrix. This additional information could be useful in two ways. Upon disabling the circle which splits the data into clusters, the clusters which were provided with the data (assuming there were any) could be displayed to the user. If the data had been clustered using the same algorithm and metrics employed in ClusterGram, this could be useful in realizing how the user must modify his or her clustering process in order to achieve optimal output.

Additionally, when viewing entities at a close-up level in order to determine optimal resolution threshold, one could view the details of nearby data points in order to better decide whether or not they actually represent the same entity. For example, in the scenario used for evaluation, one could scroll over or select two nearby entities in order to view their first name, last name, birthday and encrypted social security number in order to view the differences that caused them to be separated by the distance metric, and make a more educated decision on whether or not they should be resolved to the same point.

2. Output Results

Because ClusterGram was initially designed to be an aid to clustering and entity resolution, the software contains no utility to output the results of actions performed on the visualization. This feature could be included via the creation of comma separated value files that listed the name of each resolved entity and their cluster based on their color on the visualization. Additionally this report could include attribute level information if this had been input, including merged information for resolved entities. However, before this output functionality is implemented, the effectiveness of ClusterGram’s clustering and entity resolution algorithms must be more thoroughly evaluated.
3. Optimize Runtimes

Because the main innovation of ClusterGram is its unique visualization, clustering is currently performed with a very basic algorithm, which results in poor runtime scaling. It is possible this could be updated to a more efficient algorithm, or further optimization could be performed on the current implementation. Additionally, for extremely large data sets, a built in functionality to cluster only a random sampling of the input could be implemented, while automatically employing a multiplier to display correct data to the user.

5. Conclusion

To summarize, ClusterGram is a lightweight application designed to provide researchers with guidance when performing clustering and entity resolution on data to which a distance metric has been applied. Taking a distance matrix as input, and making use of a nearest neighbor search engine with a selection of cluster distance metrics, a radial visualization of the data set is provided to the user. Next, through manipulation of the interface, the user can be provided with useful statistics on the optimal resolution and cluster threshold for his or her data. Based on the evaluation of intentionally clustered, random, and non-trivial data sets, and through a real world example, the effectiveness of ClusterGram’s assistance has been established. However, the runtime of ClusterGram may make it difficult to use for extremely large data sets.

Appendix A

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th></th>
<th></th>
<th>ii</th>
<th></th>
<th></th>
<th>iii</th>
<th></th>
<th></th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

i ii

Zoom: 1.0x 
Zoom: 8.4x

ili

Clusters: 8
Distance: 639.79

iv

Resolve: 0
Resolve: 0.11