ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 6 (2011) 141–146

Procedia
Computer Science

Complex Adaptive Systems, Volume 1
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2011- Chicago, IL

Evaluation of classification quality and comparative analysis of clustering and self-organization

Aaron Larocque, Iren Valova

University of Massachusetts at Dartmouth, 285 Old Westport Rd., North Dartmouth 02747, USA

Abstract

Clustering is a way of classifying a multi-dimensional dataset by the similarities of its dimensions. The results from clustering must be analyzed to test the accuracy of the algorithm and its implementation. This analysis is sometimes done by a visual representation of the clustered dataset. However, it is impossible to visually represent a dataset with more than four dimensions. Statistical analysis makes this feasible. The analysis performed on the output calculates the centroid of each cluster and the cluster's relation to that centroid. We have investigated two modes of hierarchical clustering and spectral clustering. The standard deviation of each dimension from the centroid, the maximum Euclidean distance from the centroid, and the dimensions that elements of each cluster have in common are also computed. The performed experiments demonstrate which clustering algorithm presents most accurate results under certain circumstances through the use of a synthesis of visual representation and the statistical analysis proposed above.

Keywords: Statistical analysis; Hierarchical clustering algorithms;

1. Introduction

Clustering is an important form of data mining today and it is required to be as efficient as possible. Many different algorithms are continually improved or created to increase effectiveness. The question remains: How do we measure effectiveness on data for which we do not have category labels? Using statistical analysis we propose that clustering algorithms can be defined by mathematical accuracies of each dimension. Statistically analyzing each dimension we continue to see how accurate a cluster is beyond the 3 visual dimensions (assuming we do not utilize Einstein's space-time as a fourth). If we can provide analysis of which clustering algorithms are effective and under what conditions, we have a better chance at accurate results. Also, if we know which algorithms are more effective then future work can be placed into these to increase accuracy. Standard deviations allow us to create a cut-off to determine whether a dimension of a cluster is similar or dissimilar. In our previous work with clustering [3], we developed a clustering algorithm, which depends on basic self-organization. With this work, we are investigating the best self-organization, unsupervised method to serve our purposes. While there is a multitude of clustering algorithms, we have excluded the basic k-means and nearest neighbor due to the simplicity of the cluster shapes they

doi:10.1016/j.procs.2011.08.028

create [4]. We have also excluded very complex algorithms, such as DENCLUE [5] or K-Means+ [6]. The goal of this work is to compare the capabilities of hierarchical and spectral clustering and analyze statistically the performance and the reasons for it in the algorithms. Based on this work, the RADDACL [3] algorithm can be further refined and statistically tested. First, we discuss each algorithm's implementation and the datasets used during clustering. Next, we discuss in detail the math used and how it is implemented. Finally, the results will be presented both visually and statistically for maximum coverage.

2. Algorithm Implementation

All three clustering algorithms start with an I/O portion. In this initialization phase, data is first imported into a list readable to that language. Utilizing the Java frame work the data is inserted in an *ArrayList<ArrayList<Tuple>*, where a *Tuple* is a class representing a single multi-dimensional element. Once the data has been imported an adjacency matrix is created, where each index is simply the Euclidean distance between two corresponding elements. The Euclidean formula that I used for the matrix is as follows:

$$A_{ij} = \sqrt{(i_1 - j_1)^2 + (i_2 - j_2)^2 + \cdots + (i_\mathrm{n} - j_\mathrm{n})^2}$$

After the import functions have completed, an *output.txt* file is created, so that as each algorithm completes the clustering the results are immediately sent to this file in a secure location. The act of clustering done by the three algorithms can be quite similar and dissimilar at the same time. Single-Link and Complete-Link essentially work the same way except for the distance that is selected, which I will go into detail soon. However, Spectral clustering varies from these two substantially.

2.1. Single-Link Clustering

Single-Link Clustering (SLC) initializes with zero clusters and creates them sequentially as the algorithm progresses through the data set. Starting at the beginning of the ArrayList, a series of nested loops begin to check the values of the adjacency matrix. If possible, it can be effective to skip the diagonal of this matrix as the values are all 0.0. The main feature of SLC is that, as the algorithm scans the adjacency matrix, the minimum distance between the two clusters under consideration is evaluated against the threshold. If it is within the threshold, the two corresponding elements are merged into a single cluster. At this point the ArrayList containing the Tuples is updated and then the next iteration of merging begins. SLC will continue to run until no merges are made during one iteration, and at this point the output file is closed.

2.2. Complete-Link Clustering

Complete-Link Clustering (CLC) initializes each element as a separate cluster, rather than starting with zero clusters. This cluster initialization allows merging at each iteration to reduce the number of clusters by one. CLC operates similarly to SLC except for the cluster merging. Where the SLC algorithm evaluates the minimum distance between two clusters, CLC evaluates the maximum distance against the threshold. After no merges are made, the program completes the same way that SLC does.

2.3. Spectral Clustering

Spectral Clustering (SPC) operates very differently from SLC and CLC. SPC is centered on matrix manipulation and eigenvector mathematics. The SPC that I used was built upon the research done in [1]. The first operation of this algorithm is to generate an affinity (similarity) matrix. This is very similar to the Euclidean distance matrix mentioned above except that once we calculate the distance the affinity matrix uses a different equation that utilizes the use of an exponential equation to force all distance approach 1.

$$\mathrm{Affinity}_{x,y} = e^{\frac{-Euclidean}{2\sigma^2}}, \qquad where\ \sigma = \frac{maximum\ Euclidean\ distance}{m^{\frac{1}{n}}}$$

Next, a degree matrix is formed where each index of the diagonal is the sum of the corresponding row in the affinity matrix. The purpose of this is that in the next step we need to normalize the affinity matrix. The result is called the Normalized Laplacian Matrix where:

Once the Laplacian matrix is compiled, we then perform Eigen value decomposition on it so that we can select a user specified number of largest Eigen vectors to construct the final normalized matrix 'U'. Essentially 'U' renormalizes the data to have the number of dimensions equal to the number of Eigen vectors selected previously. Finally, we have a matrix representing the similarities that each element shares with other elements features. With this new matrix, we perform K-means clustering and "assign" a number to each element that represents which cluster it belongs to. So that we can later analyze the results, SPC required one additional step forming the *output.txt* file. There are many available algorithms for spectral clustering with "minor changes" [2], but all seem to function based off these principles.

2.4. Datasets

The data that was used for the clustering analysis were generated beforehand. The purpose of this step was so that we can create two-dimensional datasets in which we know what the clustering results should be in an optimal situation. Since they are two-dimensional, we can easily map them using the Cartesian coordinates, and differentiate clusters by colors and symbols for a visual representation of cluster analysis, in addition to the statistical analysis that to be discussed next. Figure 1 presents the datasets generated for our cluster analysis.

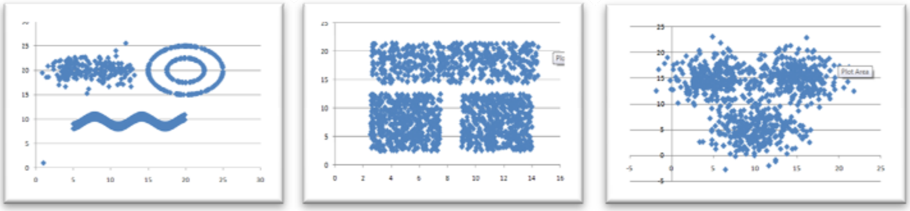

Figure 1 Three data sets used in the experiments

3. Cluster Analysis

Cluster analysis was implemented in its own java class allowing for modularity amongst all three algorithms. Since the data located in the output file is formatted in layers we can analyze each cluster individually. First, the centroid, a.k.a. geometric center, of each cluster is found. Once we find the centroid of each cluster we can then find the standard deviation of each cluster's dimensions by substituting the average with the centroid's dimensional values. This is essentially how we determine how similar the cluster's elements are per each dimension. We assume that if 65% of the cluster elements were within the standard deviation of centroid, then the cluster exhibited similar features. Since the results can vary drastically by changing the threshold for SLC and CLC, in talking about them we discuss the optimal scenario.

3.1. Single-Link analysis

With SLC, all elements “next” to each other are classified into the same cluster. Visually, in Figure 2 the results are presented, without looking at the statistical information. The third dataset is classified as a single cluster since SLC cannot associate difference between Gaussian distributions, due to the overlapping elements, as seen in Figure 1. The mathematical analysis of SLC is less effective on 2D data sets due to merging of points when their Cartesian coordinates are close by. The only clusters that show dissimilarity in SLC seem to be the sinusoidal wave and the rectangle, because they span a great range in one dimension. Both of these shapes have around 50% similarity therefore classifying them as dissimilar.

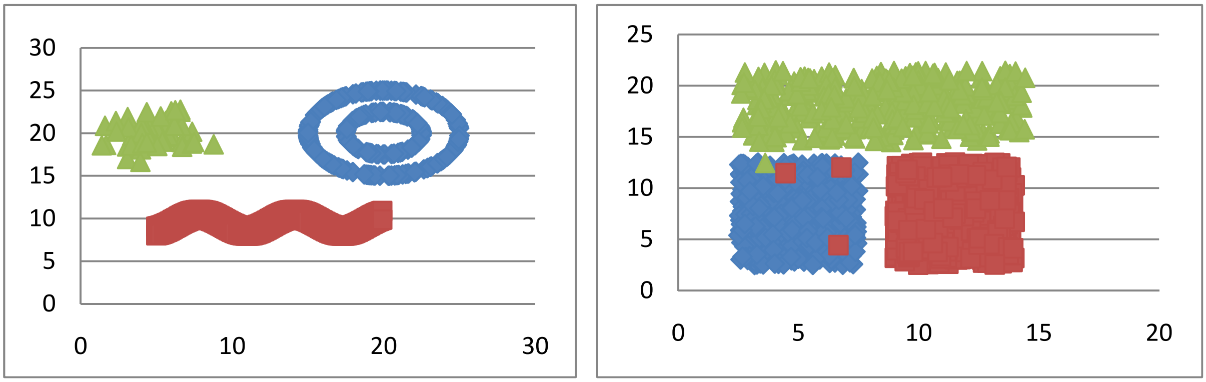

Figure 2 Single-link clustering results

3.2. Complete-Link analysis

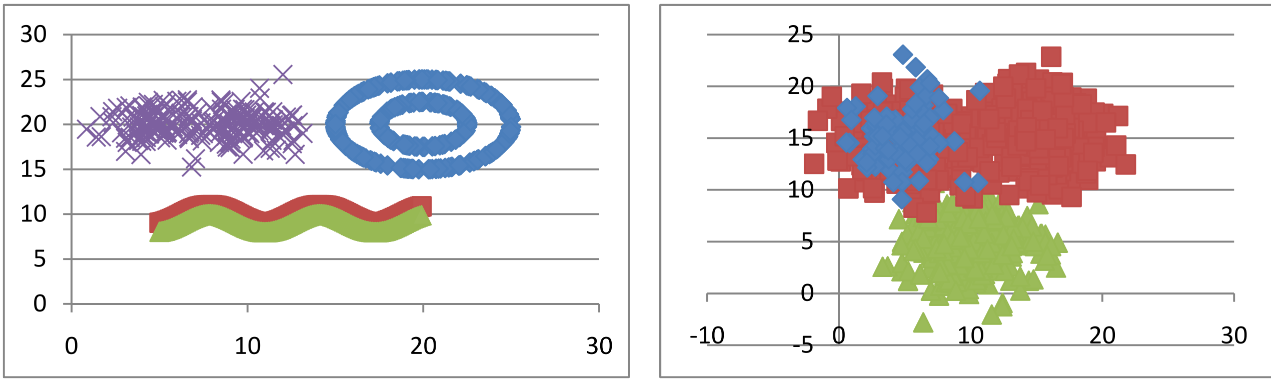

Figure 3 Complete link results

The subtle differences between SLC and CLC are in the sinusoid and the three overlapping Gaussians. Since the maximum distance between clusters is tested and each element starts in its own cluster, we see a random association between the two shapes depending on the order of their analysis. Statistically, the sinusoidal wave has similar results to SLC, except that CLC presents two clusters within the wave. The CLC determines the bottom part of the wave to be a separate cluster because the farthest distance exceeds the threshold. While the threshold is modeled after the data set, in our future work we are developing a genetic algorithm to automate the process. A step up for CLC is that it can determine the dissimilarities between the overlapping Gaussian distributions. While we do not have complete separation of the three clusters, CLC is able to distinguish them as being present. Also, mathematically, all three clusters prove to be similar over the span of the two dimensions.

3.3. Spectral analysis

SPC proves, visually and statistically, to be the most efficient algorithm of the three. The results are presented in Figure 4.

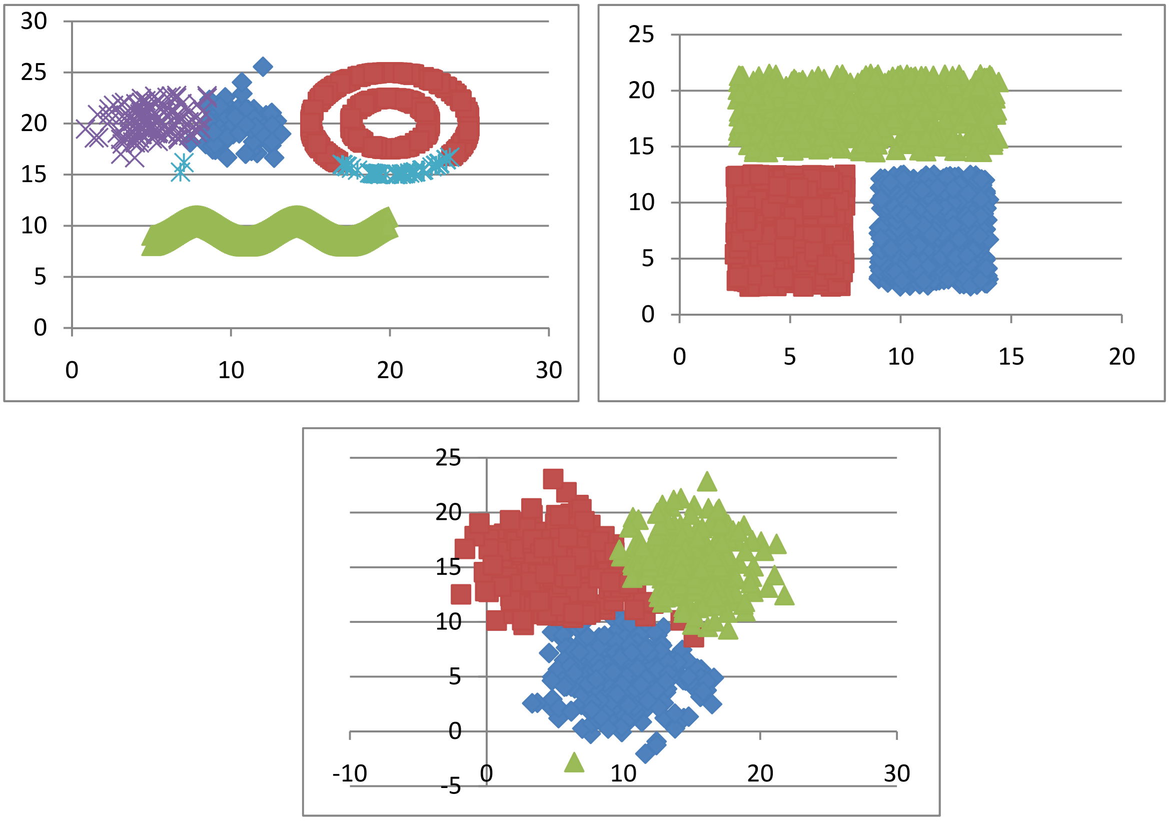

Figure 4 Spectral clustering results

Figure 4 clearly shows that SPC handled the Gaussian distributions without any issues, but there is a problem with the circles. The reason for this is in the shapes in the first dataset being too similar, thus making clustering much more difficult. One considerable benefit of SPC is that the clusters that are similar are much more statistically similar than in previous algorithms, most of which exhibit nearly 100% similarity.

4. Discussion

Using statistics to analyze clustering is something that is increasingly vital as the number of dimensions increases. One thing to remember is that the similarities between the two-dimensional datasets are so high because the Cartesian system takes over for the dimensions. Anything over three dimensions, possibly four, is nearly impossible to visualize and this is where the statistics come into play. By demonstrating that they are in fact an effective way to test results of clustering smaller dimensioned datasets, then we can assume that the statistics will scale to the larger ones since the math does not need to adapt to the dimensions. Using math to analyze remains to be one of the most effective tools we have and this is demonstrated via spectral clustering, which is almost solely based on math. For future work we will be looking to optimize spectral clustering using different computer science techniques for maximum efficiency, and then testing the algorithms on 10+ dimensioned data. In this research we will see the statistical analysis prove most effective.

Starting with SLC we see in Figure 2 that any non-complex shape of elements is correctly classified. Any complex shape, however, demonstrates the limitations of this algorithm. One such shape is the Gaussian distributions because the overlapping elements make it impossible to distinguish the three shapes. CLC proves to be just as effective with the simple shapes as SLC, but a difference can be seen in the complex ones. Figure 3 illustrates the sinusoidal wave is correctly classified into two distinct clusters. While SLC would classify them as a single cluster based on Euclidean distance, CLC is able to distinguish them because the farthest elements of the two clusters, when analyzed in order, extend longer then the threshold. The best results are with SPC. Most importantly, a predetermined threshold does not limit the SPC algorithm in the merging of clusters. Having this threshold removes the unsupervised elements of the algorithm, since for optimal results, the threshold must be tailored to the dataset. Figure 4 demonstrates that SPC is able to correctly classify the most complex shape used, i.e. the Gaussian distributions. By selecting the k-largest eigen vectors and the performing k-means clustering on the normalized Laplacian matrix, SPC is able to use the density of similar dimensions to classify elements into correct clusters. The first dataset in Figure 4, however, does show the limitations of SPC. While SPC can correctly classify most complex shapes independently, making shapes too similar within a single dataset will throw off the effectiveness of the clustering process as evidenced in Figure 4 first set with sinusoid and circles.

Using statistics to analyze clustering is something that is increasingly vital as the number of dimensions increases. Mathematical similarities between two-dimensional datasets are high because the Cartesian system takes over for the dimensions. Anything over three dimensions is nearly impossible to represent visually and this is where the statistics of each cluster become effective means of analysis. By demonstrating that this is an effective way to test the results of clustering with smaller dimensioned data, we can assume that the effectiveness will scale to a larger scale of dimensionality. For future work with improvements on the algorithms we will be using this statistical analysis to demonstrate the effectiveness of the algorithm on more dimensional data.

5. References

[1] Ng A., Jordan M.I., Weis Y., 2001, “On Spectral Clustering: Analysis and an algorithm”, In Advances in Neural Information Processing Systems, pp 849-856, MIT Press.

[2] Yu S., Shi J., 2003, “Multiclass Spectral Clustering”, In Proceedings Computer Vision Conference, Vol.1, pp 313-319

[3] Beaton D., Valova I., 2007, “RADDACL: A Recursive Algorithm for Clustering and Density Discovery on Non-linearly Separable Data”, International Joint Conference on Neural Networks (IJCNN), pp 1633-1638.

[4] Dunham M., “Data Mining: Introductory and Advanced Topics, Prentice Hall.

[5] Keim D., 1999, “Tutorial 3: Clustering Techniques for Large Data Sets – From the Past to the Future.” ACM-Knowledge Discovery and Data Mining.

[6] Huang H., 2005, “K-means+ Method for Improving Gene Selection for Classification of Microarray Data”, IEEE Computational Systems Bioinformatics Conference.