National Aeronautics and Space Administration

Determining the Number of Clusters In a Data Set Without Graphical Interpretation

Nathan Aguirre[1,2,3], Dr. Misty Davies[3]

1-New Mexico Institute of Mining and Technology, Socorro, NM 87801
2-MUST Program, Hispanic College Fund, Washington, D.C., 20005
3-NASA Ames Research Center, Moffett Field, CA 94035

Introduction

Clustering analysis is a data mining technique that is meant to simplify the process of classifying data points.

The basic clustering process requires an input of data points and the number of clusters wanted.

The clustering algorithm will then pick starting *C* points for the clusters, which can be either random spatial points or random data points. It then assigns each data point to the nearest *C* point, where "nearest" usually means Euclidean distance, but some algorithms use another criterion.

The next step is determining whether the clustering arrangement this found is within a certain tolerance. If it falls within this tolerance, the process ends. Otherwise, the *C* points are adjusted based on how many data points are in each cluster, and the steps repeat until the algorithm converges.

The Problem

Clustering can be a useful means to classify data, find correlations, etc. but there are several issues that make it troublesome to implement.

1) The biggest issue is that the number of clusters is an input, which means that the person using the algorithm must have some idea regarding the number of clusters. In cases where the data is two-dimensional and well separated, like the Simple Data Set pictured at right (Fig. 1), this isn't a major problem. However, most real-life data is not as well segregated, and commonly based on at least several different dimensional variables, it can be very difficult to know how many clusters to expect.

2) Another issue is that clustering will not always return the same results, even if done multiple times using the same number of clusters input. This occurs due to the fact that the *C* points are randomly selected, and causes the clusters to converge differently, and sometimes not at all.

Types of Clustering Algorithms

Clustering algorithms vary depending on how much the person needs to know.

K-means clustering is the most basic, which simply returns *k* number of clusters. Fuzzy *c*-means clustering operates like *k*-means, but instead returns the probability for each data point to belong to any one of the clusters. There are many variations of these codes that are meant to deal with non-circular clusters, clusters within clusters, etc.

The Expectation-Maximization algorithm (EM) also requires a *k* input, and is called the best clustering algorithm thus far. However, it is computationally expensive, requires several iterations to find a reasonable answer, and typically expects the data to be Gaussian. There are versions that are able to handle other data types, but it is difficult to switch between several types.

Idea

In order to deal with the two issues presented, we decided that instead of trying to create the "perfect" clustering algorithm, we would implement a function that would return k. This way, it would vastly improve any clustering algorithm.

In the case of *k*-based algorithms, instead of running the function 10 times for every possible *k* value, it would be much more efficient to run *k*-means for only the "correct" number of clusters.

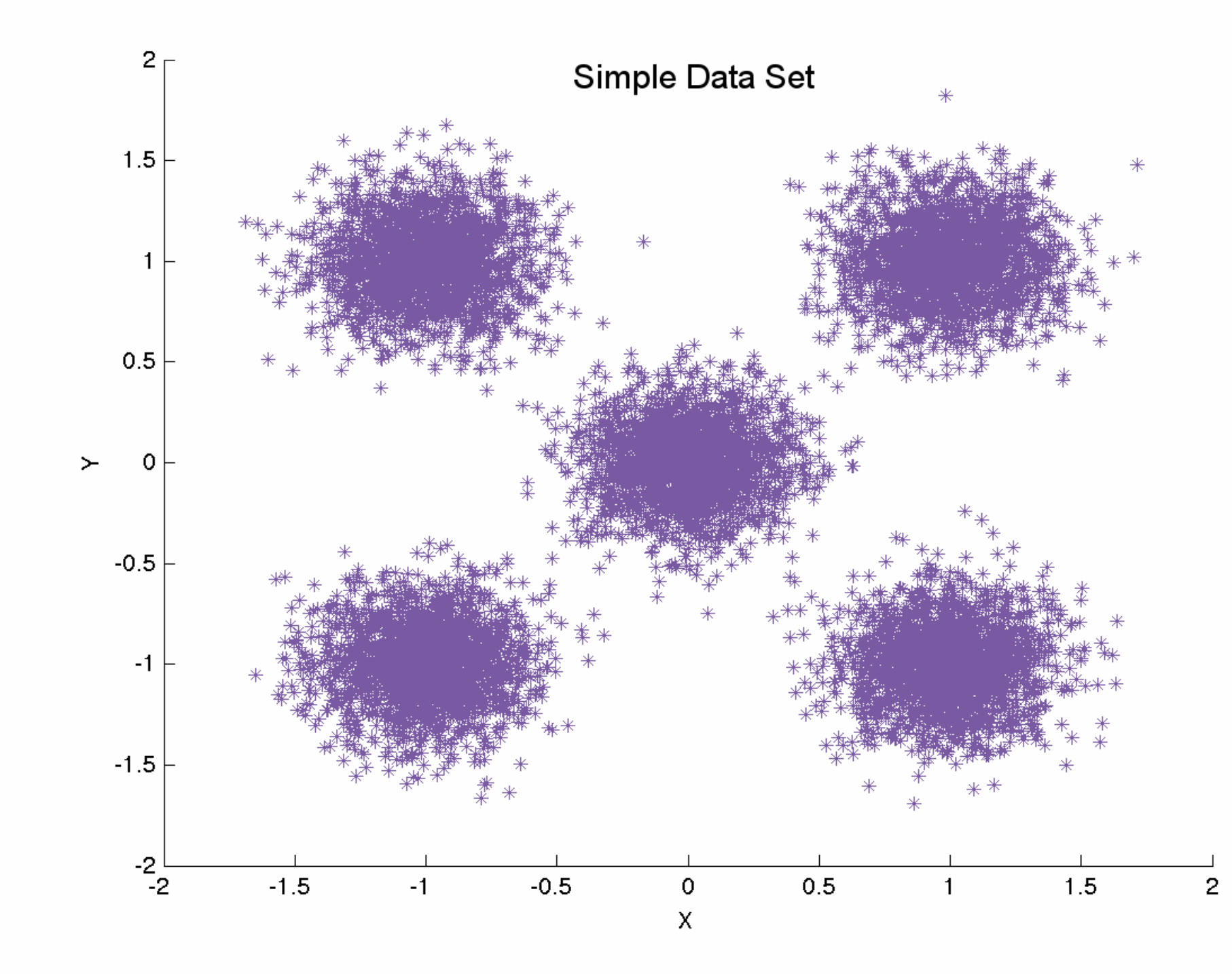

Fig. 1

Explanation

In the Simple Data Set, the human eye can clearly pick out that there are five clusters. And so, if we were to run a clustering algorithm through this data set, we would expect the code to return the good clustering (Fig. 2).

However, it is not uncommon for a clustering algorithm to completely get it wrong. In Fig. 3, this is indeed a clustering using *k*=5, but the algorithm decided to overlap one of the clusters with another.

This behavior is typical of many algorithms, but it also provided a clue to finding the solution. After reaching the ideal value of *k*, the clusters are much more prone to overlap. Based on this, and the original work of Dr. Matthias Schonlau (1), we are implementing a method that can resolve *k* for a given data set.

Our Algorithm

Our code applies any given clustering algorithm to a data set. It then performs clustering for *x* values of *k*, where *x* indicates to what extent the clustering is carried out.

It applies the chosen clustering algorithm, picking increasing *k* values for the number of clusters. It then counts the number of data points in each cluster as well as the distance between clusters. It then produces a graph like that shown at right (Fig. 4).

The wider the line, the more data points in that cluster. Thinner lines tend to indicate a small number of data points being reassigned to another cluster.

In this example, there were five clusters. As can be seen, even though the clustering at five clusters was particularly bad, we can still return the correct number of clusters by looking further along the graph.

This example in particular demonstrates how this algorithm works because the clusters are quite well-defined. It also shows how this process can be expanded in order to work with more complex data sets.

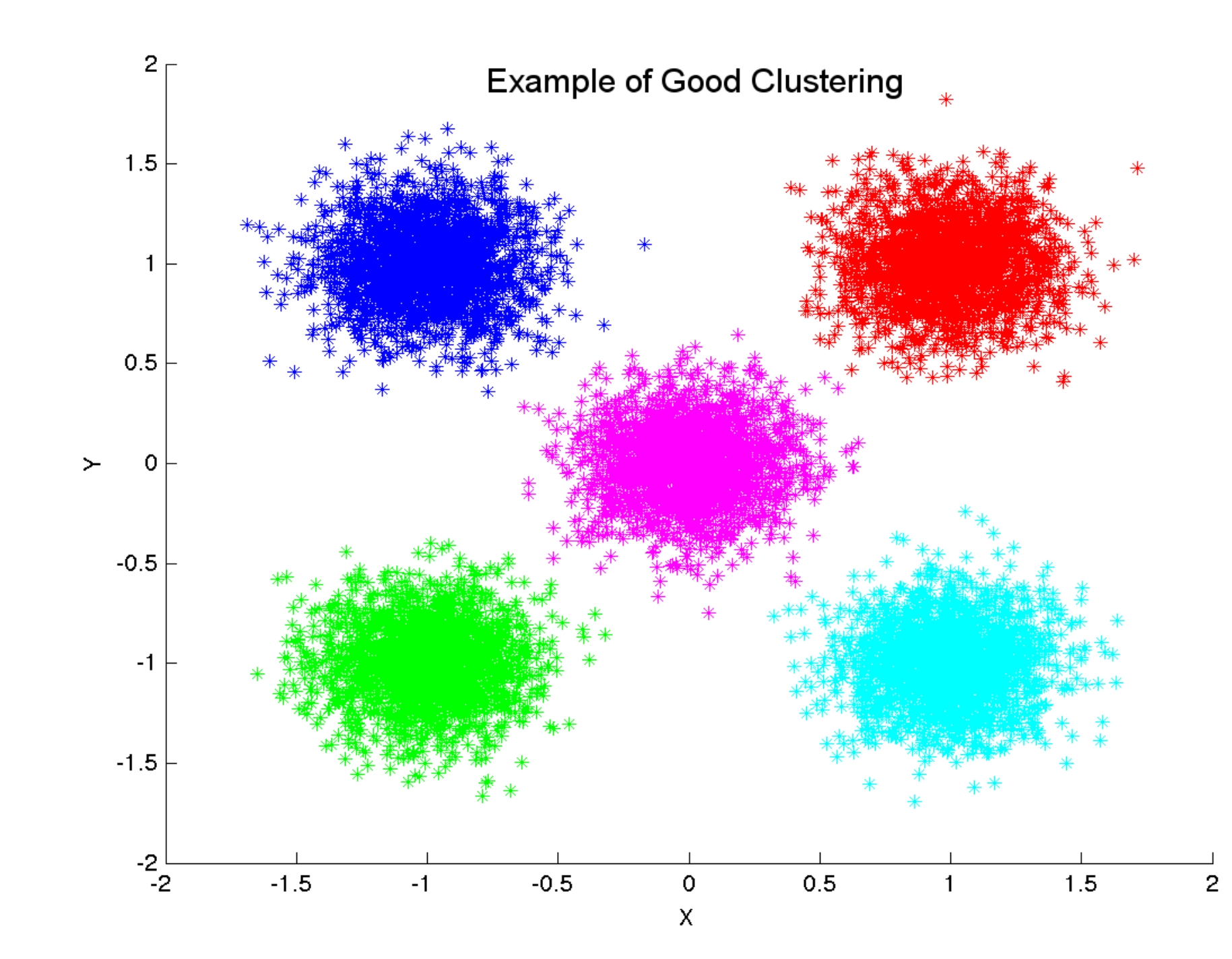

Fig. 2

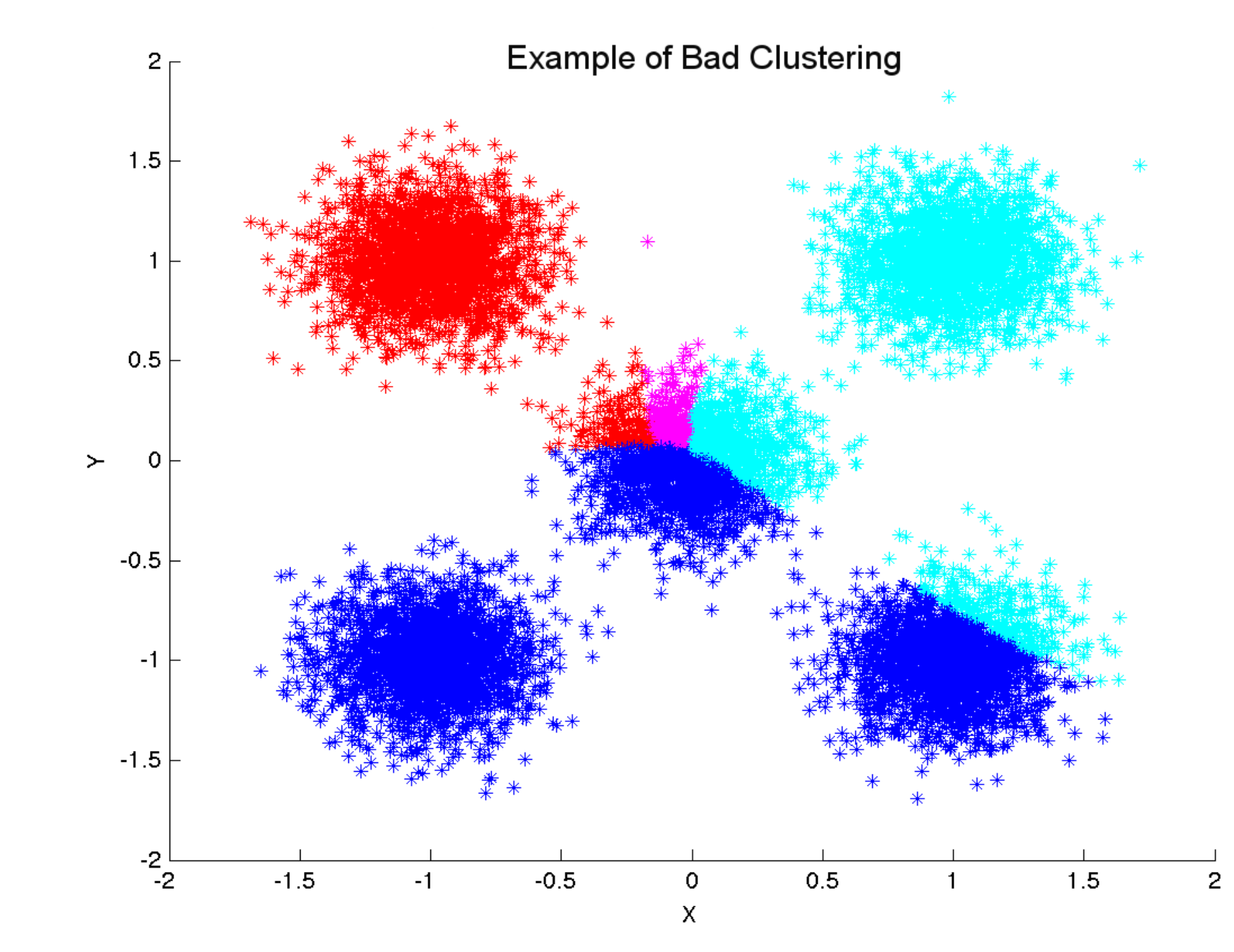

Fig. 3

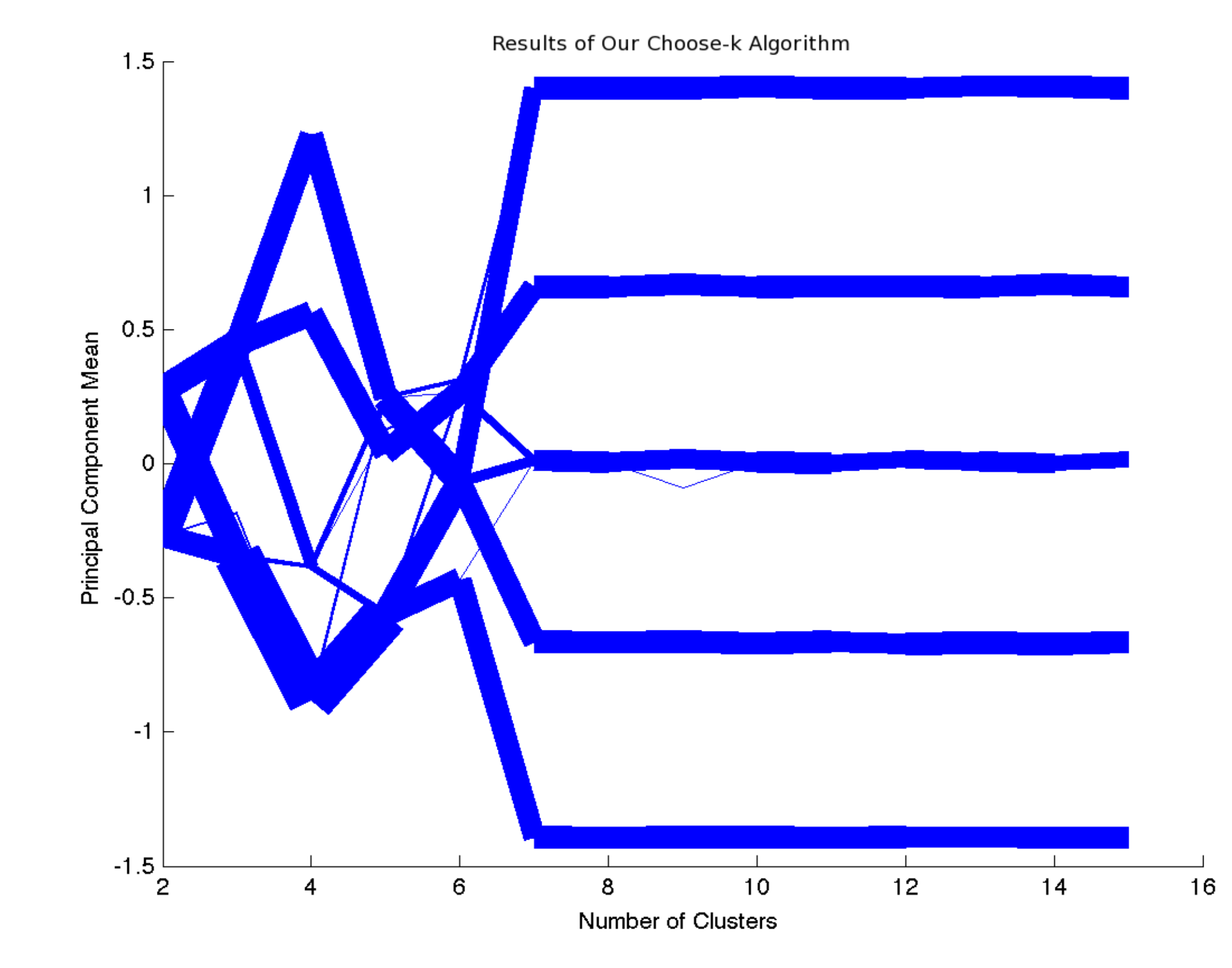

Fig. 4

Future Work

The simple example presented here shows a great potential to apply it to larger, more complex data sets.

We are also looking at using larger *x* intervals with the hopes of still being able to pick out an accurate result, while reducing the time dramatically. Another goal is to possibly find a way to discern whether there are smaller or less noticeable clusters within clusters and giving the option to show those as well.

One final goal that we want to accomplish is to be able to code a method that will return all the data we get from a graph like Fig. 4, but without having to visually inspect it (2), and to do this with larger data sets.

References

1. Schonlau, Matthias. The Clustergram: A graph for visualizing hierarchical and non-hierarchical cluster analyses http://www.schonlau.net/publication/02stata_clustergram.pdf. 06 July 2011.

2.Pascual, D., F. Pla, and J. Sanchez. "Cluster Validation Using Information Stability Measures." *Pattern Recognition Letters* 31 (2010). Web. 03 July 2011.

Background image from www.stockvault.net.

www.nasa.gov