Clustering Algorithms - A New Look at Mass Spectra -

By Anna Ritz '06, Ben Anderson '05, Kate Nelson '04
With Dave Musicant, Assistant Professor of Computer Science
Carleton College

The Concept of Clustering

In the most general sense, clustering means finding trends (or relationships) in data without using any prior knowledge of what those trends might be. Clustering is usually done on large sets of data, where the trends might not be easily seen. Although clustering can be useful, there are a few subjective decisions one must make in order to produce meaningful results:

- How do we define whether two elements are related or not? What is the measurement?
- How many clusters do we see? Subtleties might cause one person to see a certain number of clusters, while someone else sees a different number.
- How do we represent the clusters we find? What is the most meaningful way?

Clustering Mass Spectra

In our research, we have been focusing on clustering the spectra produced by Gromit to find relationships between groups of particles. In addition to the above obstacles, the nature of the data causes more. Since each spectrum has 30,000 points on it and we are comparing spectra, we are in effect clustering a 30,000-dimensional graph. Add to this elements such as particle size and other measurements (like mercury levels, for example), and the problem has become quite complex.

The Distance Metric Problem

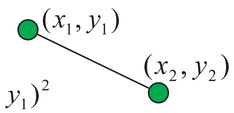
The two distance metrics below are commonly used in clustering, and we have implemented both of them in our algorithms. These metrics also work regardless of our multi-dimensional data, making them easy to use. The examples below, however are in a 2-D plane for simplicity.

Euclidean-Squared Distance

This gives us the length between two points "as the crow flies," which is what most people intuitively think of as the length.

Given: (x_1, y_1) and (x_2, y_2)

Euclidean^2= $(x_2 - x_1)^2 + (y_2 - y_1)^2$

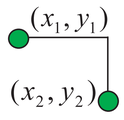


Manhattan (or City-Block) Distance

This gives us the length between two points by only traveling along the x or y-axis. It is similar to navigating around blocks in Manhattan - hence the name.

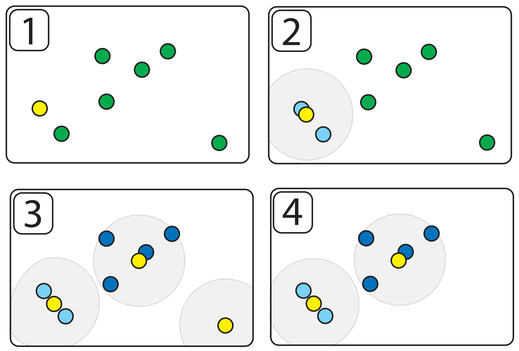
Given: (x_1, y_1) and (x_2, y_2)

Manhattan = $| x_2 - x_1 | + | y_2 - y_1 |$



Clustering Algorithms

Art2a

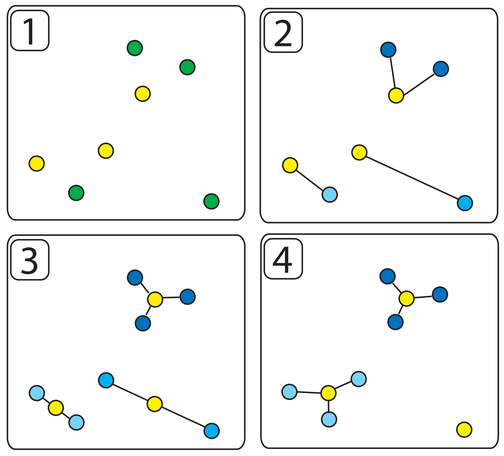


Step 0: To run Art2a, the algorithm needs to know three things: the data, the vigilance, and the learning rate.
Step 1: Choose an arbitrary point P_1 and call it the first centroid C_1.
Step 2: Choose another point P_2 and measure the distance from all centroids $\{C_1, C_2, C_3 \ldots\}$ using the specified distance metric. If the distance is smaller than the vigilance, then P_2 is added to the cluster containing C_1 and C_1 will move the amount of the learning rate towards the new point P_2. If the distance is larger than the vigilance, then P_2 becomes a new centroid.
Step 3: Repeat step 2 until all of the data points are in a cluster. Discard all the clusters where there are an insignificant amount of points.
Step 4: Report the rest of the clusters with their centroids.

K-Means

Step 0: To run K-Means, the algorithm needs to know two things: the data and the number of clusters "k" that are desired in the end.
Step 1: Choose k arbitrary points $\{P_1, P_2 \ldots P_k\}$ and call them the centroids $\{C_1, C_2 \ldots C_k\}$.
Step 2: Choose another point P_i and measure the distance from all centroids $\{C_1, C_2 \ldots C_k\}$ using the Euclidean Squared distance metric. Add P_i to the centroid with the smallest distance. Repeat until all points have been added to a cluster.
Step 3: Re-evaluate the centroids $\{C_1, C_2 \ldots C_k\}$ by averaging the distances from all points assigned to each centroid.
Step 4: Repeat steps 2 and 3 until clusters are stable (when re-evaluated, the centroids don't change much). Report these clusters and their centroids as they stand.

"k" = 3 :



K-Medians

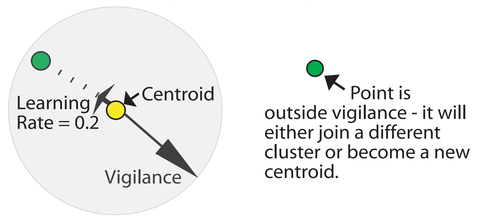
K-Medians is exactly like the K-Means algorithm with three significant exceptions:

- K-Medians uses the Manhattan distance metric instead of the Euclidean Squared distance metric.
- When K-Medians re-evaluates the centroids, it takes the median of the distances from all points to their respective centroid instead of the mean.
- To report the centroids, the points from the clusters need to be averaged in order for them to be meaningful. Otherwise, the spectra will be so sparse that it will be impossible to analyze them.

Glossary

centroid - Cluster center; the best evaulation of a cluster.
distance metric - Way of measuring the distance between two data points.
"k" - Number of clusters.
learning rate - Percentage of distance from the centroid to a new point P that the centroid will move when P is added.
vigilance - Largest radius a cluster can have.

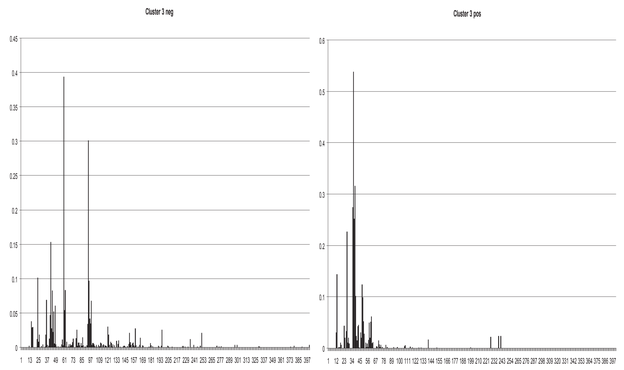
Art2a Learning Rate and Vigilance:



Preliminary Results

Even though we are in the preliminary stages of testing and comparing these different algorithms, we have already found some interesting tendencies in the clustering. When we were testing the relationships between the distance metrics and the algorithms, we found that, though Art2a can handle either distance metric, it is not guaranteed to produce stable clusters. K-Means, however, is guaranteed to produce stable clusters ONLY IF it implements the Euclidean-Squared distance metric. Similarly, K-Medians is guaranteed to produce stable clusters ONLY IF it implements the Manhattan distance metric. These results are significant because the chemistry community has been using Art2a for clustering, when it might not produce accurate results.

We have found that Art2a, K-Means and K-Medians generally find similar groups of clusters. However, there are often clusters that overlap within the three algorithms. Currently, we are trying to find which algorithm produces the most distinct results on a dataset. We will need to work closely with the Chemistry research group to establish which algorithm makes the most sense with our data.



Sample centroid (positive & negative spectrum) from Art2a using Euclidean-Squared distance metric

Future Plans

One of the more specific clustering obstacles is the fact that data might be clustered compared to a local trend in the data instead of a global trend. We are currently improving these algorithms by making them pick more intelligent starting points, which will help this issue. In other words, we are *refining the initial centroids* so the algorithm will cluster more efficiently, if not more accurately. Generally this involves clustering a small, random sample of the data and using these centroids as starting centroids for the clustering algorithm.