Clustering

Jeremy Teitelbaum
November 7, 2018
UConn Math Club

Clustering: The basic problem

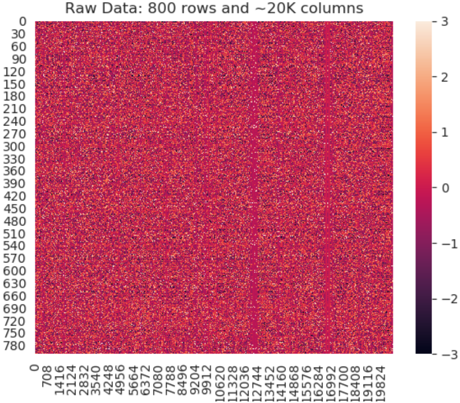
Results of an experiment:

Clustering: The basic problem

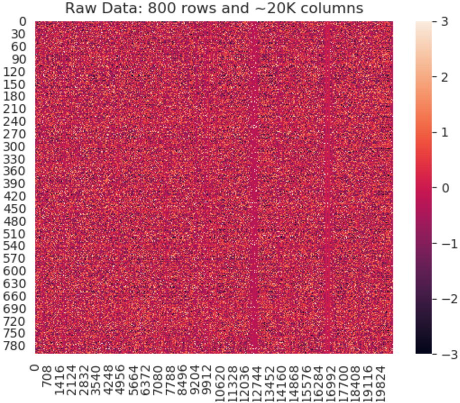
Looks like three things of interest:

How to pick them out?

Not so easy in high dimensions



Not so easy in high dimensions



Is there structure here?

Clustering

- Clustering is a problem in *unsupervised machine learning*.

Clustering

- Clustering is a problem in *unsupervised machine learning*.
- Strategy 1: work from the bottom up.

Clustering

- Clustering is a problem in *unsupervised machine learning*.
- Strategy 1: work from the bottom up.
- Strategy 2: work from the top down.

Bottom-Up (Hierarchical/Agglomerative) Clustering

General Strategy

- Group a few points that are very close together into clusters.

Bottom-Up (Hierarchical/Agglomerative) Clustering

General Strategy

- Group a few points that are very close together into clusters.
- Find the point not yet in a cluster, but closest to one of the existing clusters, and add it to that closest cluster.

Bottom-Up (Hierarchical/Agglomerative) Clustering

General Strategy

- Group a few points that are very close together into clusters.
- Find the point not yet in a cluster, but closest to one of the existing clusters, and add it to that closest cluster.
- Repeat step 2 until every point is in a cluster.

Bottom-Up (Hierarchical/Agglomerative) Clustering

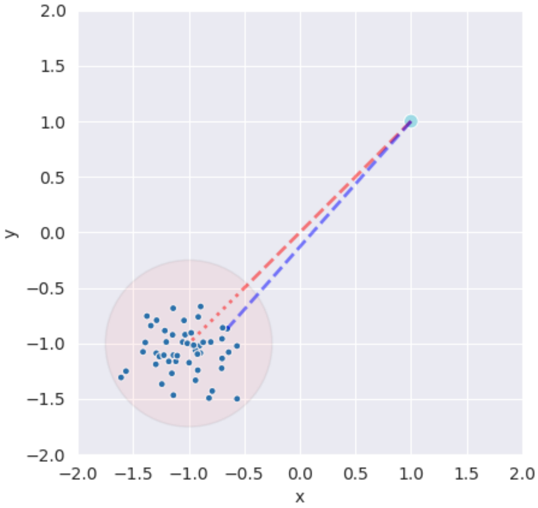
General Strategy

- Group a few points that are very close together into clusters.
- Find the point not yet in a cluster, but closest to one of the existing clusters, and add it to that closest cluster.
- Repeat step 2 until every point is in a cluster.

The Catch

What is the distance between a point and a cluster?

What is the distance to a cluster?



Different Notions of Distance

- Distance between closest points.

$$d(X, Y) = \inf_{(x,y) \in X \times Y} d(x, y)$$

Different Notions of Distance

- Distance between closest points.

$$d(X, Y) = \inf_{(x,y)\in X\times Y} d(x, y)$$

- Distance between farthest points.

$$d(X, Y) = \max_{(x,y)\in X\times Y} d(x, y)$$

Different Notions of Distance

- Distance between closest points.

$$d(X, Y) = \inf_{(x,y)\in X\times Y} d(x, y)$$

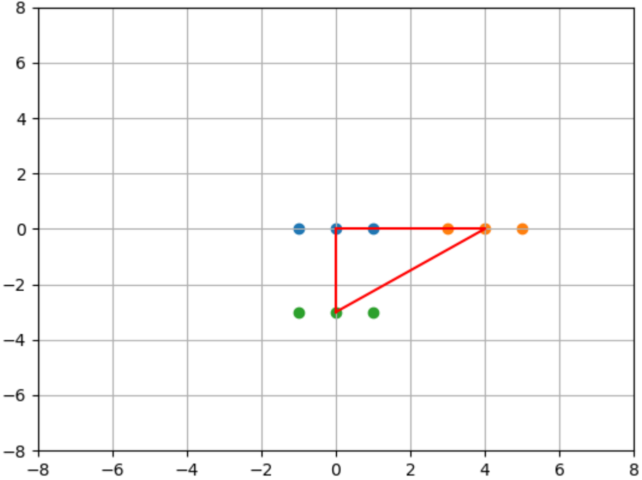
- Distance between farthest points.

$$d(X, Y) = \max_{(x,y)\in X\times Y} d(x, y)$$

- Distance between centroids.

$$d(X, Y) = d(\overline{X}, \overline{Y})$$

Some Test Cases



Algorithmic Considerations

Definition

Let X_i be a set of k clusters. The "dissimilarity matrix" is the symmetric $k \times k$ matrix whose (i, j)-entry is $d(X_i, X_j)$.

Algorithm

1. Given n points y_i to start, construct the $n \times n$ symmetric matrix D_0 whose entries are the $d(y_i, y_j)$. Set $N = 0$.
2. Find the two closest points (later clusters) by finding i', j' where $D_N(i', j')$ is minimal.
3. Combine the two points into a cluster c_0. Update D_N by removing the two points $y_{i'}$ and $y_{j'}$ and adding a row and column for the distances between the remaining points and c_0, yielding D_{N+1}.
4. Repeat steps 2 and 3 until you have only one cluster.

Algorithmic Considerations (cont'd)

To make this approach efficient, we want to be able to update the dissimilarity matrix without recomputing all of the distances from scratch.

When the distance between clusters is the distance between their closest points, or the distance between their farthest points, this is straightforward.

- (closest): If X and Y are merged, and Z is another cluster, then the distance $d(Z, X \cup Y)$ is $\min(d(X, Z), d(Y, Z))$ and this can be computed directly from the dissimilarity matrix D.
- (farthest): If X and Y are merged, and Z is another cluster, then the distance $d(Z, X \cup Y)$ is $\max(d(X, Z), d(Y, Z))$ and this can be computed directly from the dissimilarity matrix D.
- What about centroids?

A geometry problem

Problem

Let $X = \{x_1, \ldots, x_{n_x}\}$, $Y = \{y_1, \ldots, y_{n_y}\}$ *and* $Z = \{z_1, \ldots, z_{n_z}\}$ *be three sets of points in* $\mathbf{R}^m$ *and let* $\overline{x}$, $\overline{y}$, *and* $\overline{z}$ *be their respective centroids. Find the centroid of the merged set* $X \cup Y$ *and the distances between that centroid and* $\overline{z}$ *as efficiently as you can.*

A geometry problem

Problem

Let $X = \{x_1, \ldots, x_{n_x}\}$, $Y = \{y_1, \ldots, y_{n_y}\}$ and $Z = \{z_1, \ldots, z_{n_z}\}$ be three sets of points in $\mathbf{R}^m$ and let $\overline{x}$, $\overline{y}$, and $\overline{z}$ be their respective centroids. Find the centroid of the merged set $X \cup Y$ and the distances between that centroid and $\overline{z}$ as efficiently as you can.

Recall that the centroid of a set of points is their (vector) average:

$$\overline{x} = \frac{1}{n_x} \sum_{i=1}^{n_x} x.$$

Let $A = X \cup Y$ and $\overline{a}$ be the centroid of A. The centroid of the merged set could be computed from the original points, but a more efficient approach is to observe that

$$\overline{a} = \frac{n_x \overline{x} + n_y \overline{y}}{n_x + n_y}$$

Since knowing the sizes of the clusters is going to be helpful, lets assume we keep track not only of the dissimilarity matrix D but also the sizes n_X for each cluster X. Initially, all $n_X = 1$.
The remaining piece of our geometry problem is:

Problem

Write $d(\bar{a}, \bar{z}) = |\bar{a} - \bar{z}|$ in terms of n_x, n_y, n_z and $\bar{x}$, $\bar{y}$, and $\bar{z}$.

Proposition

We have

$$|\bar{a} - \bar{z}|^2 = \frac{n_x}{n_x + n_y}|\bar{x} - \bar{z}|^2 + \frac{n_y}{n_x + n_y}|\bar{y} - \bar{z}|^2 - \frac{n_x n_y}{(n_x + n_y)^2}|\bar{x} - \bar{y}|^2$$

Remark

It's much easier to work directly with the squared Euclidean distance than the usual one when making and updating the dissimilarity matrix.

Ward's Criterion

Ward's criterion is an additional way to decide which clusters are closest and should be merged next.

Definition

If X is a cluster, the 'within cluster' sum-of-squared error is

$$s(X) = \sum_{i=1}^{n_x} (x - \overline{x})^2$$

The error S is the sum of this over all clusters:

$$S = \sum_{X} s(X).$$

Notice that at the beginning of the clustering process, when all the clusters have only one point, $S = 0$. Ward's criterion says that *when merging clusters, always choose the two that increase S by the least amount.*

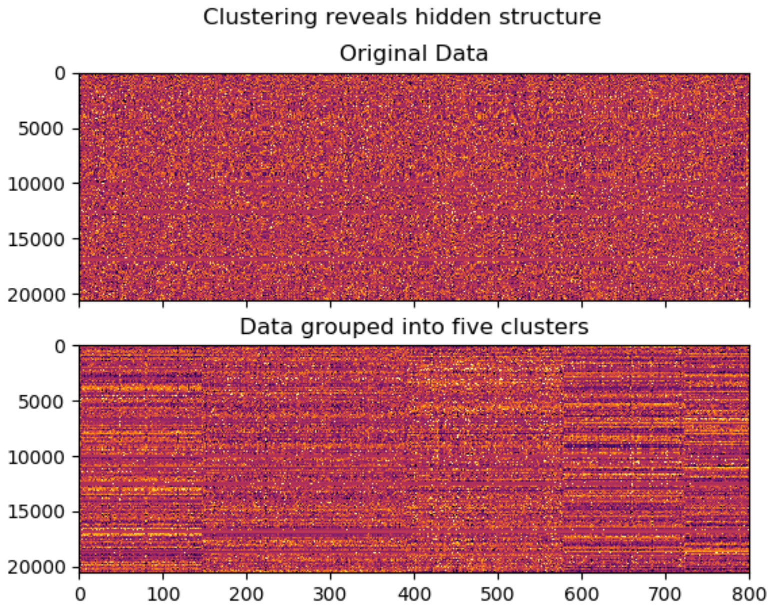
Ward's Criterion (cont'd)

Proposition

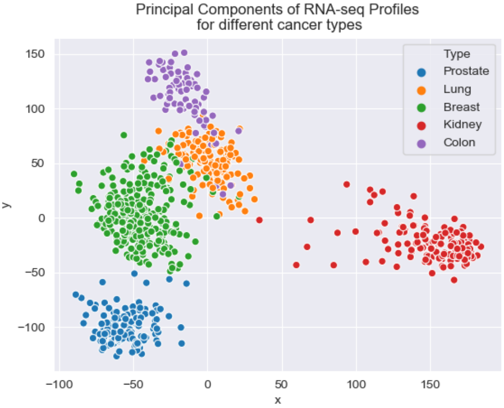
When two clusters with sizes n_x and n_y, and centroids $\overline{x}$ and $\overline{y}$, are merged, the increase in S is given by

$$\Delta S = \frac{n_x n_y}{(n_x + n_y)}(|\overline{x} - \overline{y}|^2)$$

So one way to look at Ward's method is that it combines the closest clusters by their centroid distances, but it weights those distances by the sizes of the clusters. It prefers to merge smaller clusters.


Clustering reveals hidden structure
Original Data
0
5000
10000
15000
20000
Data grouped into five clusters
0
5000
10000
15000
20000
0
100
200
300
400
500
600
700
800

Principal Component Analysis

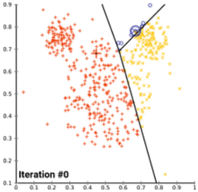


This slide is for those who saw the PCA talk a few weeks ago.

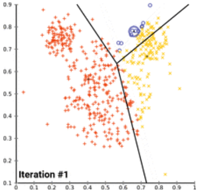
Top Down Clustering

Hierarchical or Agglomerative Clustering works from the bottom up. The most common top-down algorithm is called "k-means clustering."

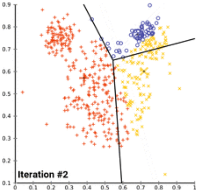
1. Decide in advance how many clusters (say, k) that you want to find. (How? good question!)
2. Pick k points at random in the space of data. Call these points $m_1, \ldots, m_k$.
3. For each data point, find the closest of the m_i and put your point in that "cluster."
4. For each "cluster", compute the centroid, yielding new means $m'_1, \ldots, m'_k$.
5. Repeat until the m_i stop moving.



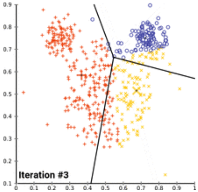
With thanks to Wikipedia.



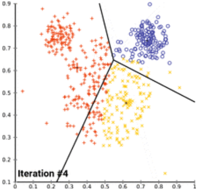
With thanks to Wikipedia.



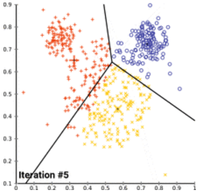
With thanks to Wikipedia.



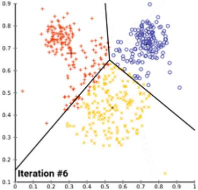
With thanks to Wikipedia.



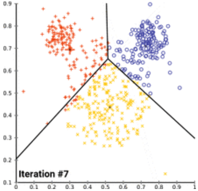
With thanks to Wikipedia.



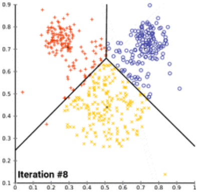
With thanks to Wikipedia.



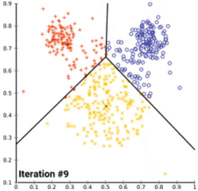
With thanks to Wikipedia.



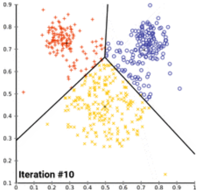
With thanks to Wikipedia.



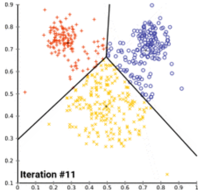
With thanks to Wikipedia.



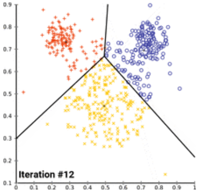
With thanks to Wikipedia.



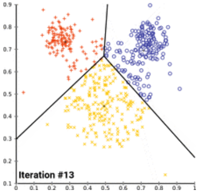
With thanks to Wikipedia.



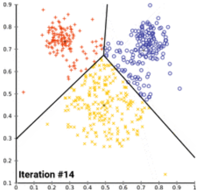
With thanks to Wikipedia.



With thanks to Wikipedia.



With thanks to Wikipedia.



With thanks to Wikipedia.

Further Reading

For those interested in applying clustering algorithms, there is a powerful set of tools in the Python `scikit-learn` library and in many `R` packages.