Single Linkage Clustering

John Burkardt (ARC/ICAM)
Virginia Tech
....
Math/CS 4414:
"Single Linkage Clustering"
https://people.sc.fsu.edu/~jburkardt/presentations/
clustering_linkage.pdf

....
ARC: Advanced Research Computing
ICAM: Interdisciplinary Center for Applied Mathematics

VirginiaTech

WirginiaTech

This is the first of several talks on clustering.

Clustering is a method of trying to find order in a set of data.
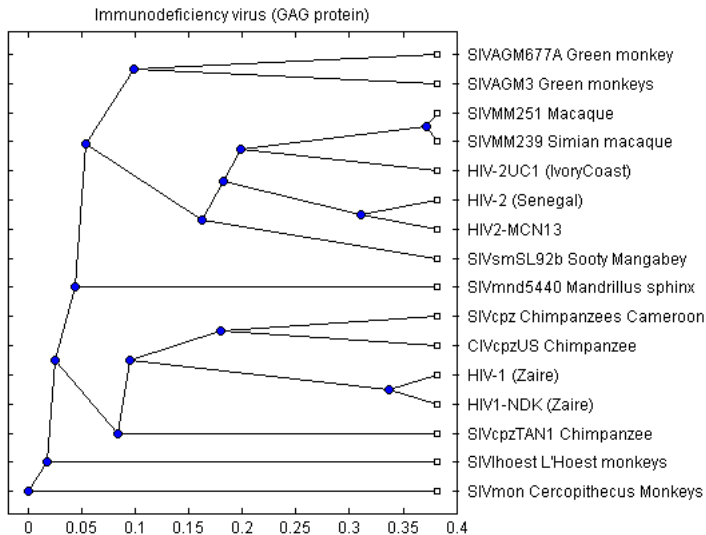
The data may be discrete or continuous.

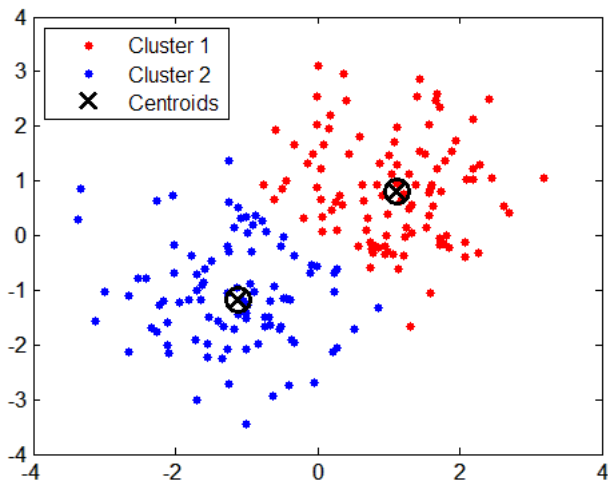If the data is discrete, it might be 15 objects or 1,000,000.

The data may be numerical, or it may be initially described by words or symbols (genetic data).
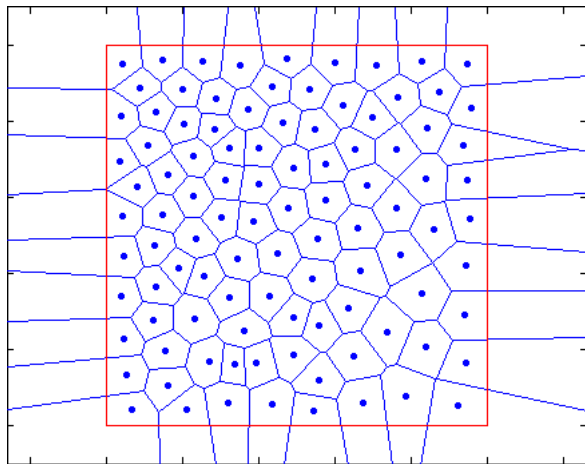
We will look at three methods of clustering:

VirginiaTech

Immunodeficiency virus (GAG protein)

Clustering: K-Means Clustering

WirginiaTech

The average of a set of numbers has many nice properties.

It is common to take the average as a **representative** of the set. Our sense is that with the average you "can't be too far off".

Let's take all the fun out of that idea and prove it.

THEOREM: Let $\bar{x}$ be the average of a set $\mathbf{X} = \{ x_i : i = 1 \ldots n \}$. Then $\bar{x}$ is the unique number which minimizes the *energy* function:

$$\mathcal{E}(c, X) = \sum_{i=1}^{n} (c - x_i)^2$$

PROOF: Solve $\frac{\partial \mathcal{E}(c,X)}{\partial c} = 0$.

WirginiaTech

The energy function measured the squared distance of every point in X to the single center point c. Trying to minimize that value leads us to the average.

But we can generalize the energy function, by allowing c to be a set of centers C, that is, more than one "average". Then the energy function measures, for each point in X, the distance to the closest value in C.

$$\mathcal{E}(C, X) = \sum_{i=1}^{n} \min_{c_j \in C} \ (c_j - x_i)^2$$

Then we can try to choose the points C to minimize this generalized energy. We will come back to this idea when we look at K-Means and CVT clustering.

VirginiaTech

The average is our first example of clustering. Clustering is a way to try to organize data, or to detect patterns in data, by breaking the data up into subsets called clusters.

Often, there is a single special value associated with each cluster, called the **cluster center**. The average of a set of numbers played this role in our previous discussion. The cluster center can be useful as a representative of all the elements in the cluster.

If we can assume that all the points in the cluster are close to the center, then we may be justified in compressing the data by replacing all the clustered values by the center value.

We could have asked MATLAB to find the value of **c** that minimized the energy function. Since MATLAB doesn't (usually) work with symbolic formulas, we have to have a particular set of data in mind.

MATLAB has a function **fminunc** which takes a starting estimate for the minimizer, and the name of an M-file that evaluates the function to be minimized, and returns the minimizer.

```
x0 = 14;
x = fminunc ( @energy, x0 );
```

VirginiaTech

The FMINUNC Example

```
function fminunc_example ( )

%% FMINUNC_EXAMPLE shows an example of unconstrained minimization.

  x0 = 14;
  x = fminunc ( @energy, x0 );
  fx = energy ( x );

  fprintf ( 1, '\n' );
  fprintf ( 1, '__The_minimizer_X__=_%f\n', x )
  fprintf ( 1, '__and_F(X)____=_%f\n', fx );

  return
end
function value = energy ( x )

%% ENERGY evaluates the energy function.

  x1 = 1;
  x2 = 2;
  x3 = 10;
  x4 = 27;
  x5 = 20;

  value = ( x - x1 ).^2 ...
        + ( x - x2 ).^2 ...
        + ( x - x3 ).^2 ...
        + ( x - x4 ).^2 ...
        + ( x - x5 ).^2;

  return
end
```

WIIIVirginiaTech

The "unc" in the name **fminunc** stands for **unconstrained**, that is, we don't make any extra conditions on the minimizer. Now the average value is often not actually a value in the set **X**, which is why we say that the average family has 2.3 children.

We could constrain our minimizer, that is, add an extra condition, by requiring that it be one of the values in the set $\underline{X}$, or that it satisfy some other condition. Then we would be talking about a **constrained** optimization, and MATLAB has different procedures for that problem.

- Overview
- The Average of a Set of Data
- **Distance Functions**
- Single Linkage Clustering
- Chain Letters

Single linkage clustering is appropriate for a relatively small amount of data (10 to 1,000 objects, say).

The objects do not need to be numeric. They could be oil paintings, new cars, jpeg images, or a sample pizza from every restaurant.

However, the important thing is that, given any two objects, we must be able to measure a kind of "distance" between them. We are free to define this distance so that "close" things are the most similar.
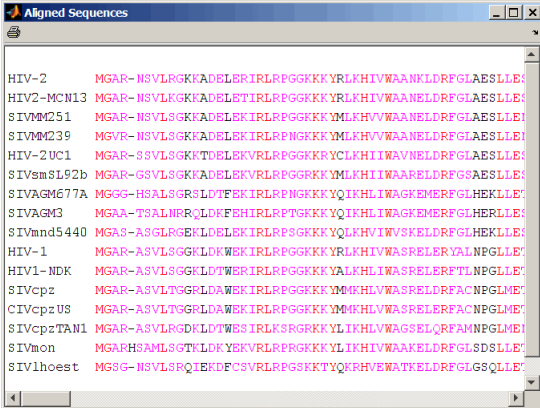
Properties of a Distance Function

We will represent the distance between objects **A** and **B** by a function **d(A,B)**. We will expect it to have most of the properties of a geometric distance function:

1. $d(A,A) = 0$;
2. but $d(A,B) = 0$ does not necessarily mean that $A = B$;
3. $0 \leq d(A,B)$;
4. $d(A,C) \leq d(A,B) + d(B,C)$ *triangle inequality*.

WirginiaTech

Sequence Analysis Distance Function

In Bioinformatics, researchers compare chemical sequences that correspond to various proteins. Their data looks like this:

You can see that, at least for this portion, the various sequences are very similar. A place where a sequence has a gap is marked by a "-" sign. The distance between two sequences is determined by lining them up as well as possible, and then counting the changes in going from sequence A to sequence B:

1. *mutations*, changing one letter, cost looked up in a table;
2. *deletion of K letters* adds a cost of $2 + K/2$;
3. *insertion of K letters* adds a cost of $2 + K/2$.

WORD GOLF Distance Function

A distance function can be given for two English words of the same length. We define a path as a sequence of words which differ only by a single letter. The distance from one word to another is the length of the shortest path between them.

The distance from WORD to GOLF is 4:

GOLF
GOL**D**
COLD
CO**R**D
WORD

You can get from MAN to APE in 5 steps.
The distance from SWORD to PEACE is probably 11.
Some words are "infinitely" far apart.

Hamming Distance Function

Richard Hamming invented a code to try to detect and correct errors in transmission of a signal.

Given two strings S1 and S2 of the same length, d(S1,S2) is equal to the number of positions at which the two strings differ.

Using this simple idea. Hamming was able to create a set of "words" with the property that if a single letter was changed, you could aways tell what letter was changed and correct it.

WirginiaTech

Distance Tables Can Replace a Distance Function

In some cases, we don't have a distance function. Instead, we have some sort of table that measures the pairwise similarity of the objects. We might simply have asked people to rate the similarity of cars on a scale of 0 to 100. (We may need to flip the data, to ensure that low scores indicate high similarity.)

```
 0   8  50  31  12  48  36   2   5  39  10
 8   0  38   9  33  37  22   6   4  14  32
50  38   0  11  55   1  23  46  41  17  52
31   9  11   0  44  13  16  19  25  18  42
12  33  55  44   0  54  53  30  28  45   7
48  37   1  13  54   0  26  47  40  24  51
36  22  23  16  53  26   0  29  35  34  49
 2   6  46  19  30  47  29   0   3  27  15
 5   4  41  25  28  40  35   3   0  20  21
39  14  17  18  45  24  34  27  20   0  43
10  32  52  42   7  51  49  15  21  43   0
```

- Overview
- The Average of a Set of Data
- Distance Functions
- **Single Linkage Clustering**
- Chain Letters

WirginiaTech

Single Linkage Clustering: The Idea

Our goal is to gradually group the objects into fewer and fewer clusters until they are all together. The way in which the objects join should tell us something about their relatedness.

Once we have chosen a distance function, we can assign a distance between every pair of our objects. We now begin the process of clustering. We start by considering each of our **N** objects to be a cluster by itself.

Then, one step at a time, we find the two "closest" clusters, and merge them into a new cluster.

After N-1 steps, there is a single cluster. The relatedness information is recorded in the history of which clusters were merged at each step.

On the very first step, each cluster is a single object, so it makes sense to talk about the distance between clusters.

But once a cluster has more than one object in it, what does the distance mean?

We define the distance between two clusters **A** and **B** to be the smallest distance between any object $a \in$ **A** and any object $b \in$ **B**.

We have extended our idea of *distances between objects* to *distances between sets of objects*.

If we examine the output from a single linkage clustering, we can see that it is telling us about the relatedness of the data.

The very first pair of items merged together are the closest. The next item might join that cluster, or merge with another to make a different pair.

There's a lot of information in the procedure, and it's hard to print out. The best way to see it is through a **dendrogram** or "tree diagram". Even then, the data has to be rearranged in a different order so that the tree prints out nicely.

WirginiaTech

MATLAB's Statistical Toolbox includes several functions that are very useful for single linkage clustering. We suppose we are given an N x 2 array XY of data. Then

- **sv = pdist (xy)** computes the distance vector of a set of points;
- **sl = linkage (sv, 'single')** returns the single linkage information;
- **dendrogram (sl)** plots the single linkage information as a tree.
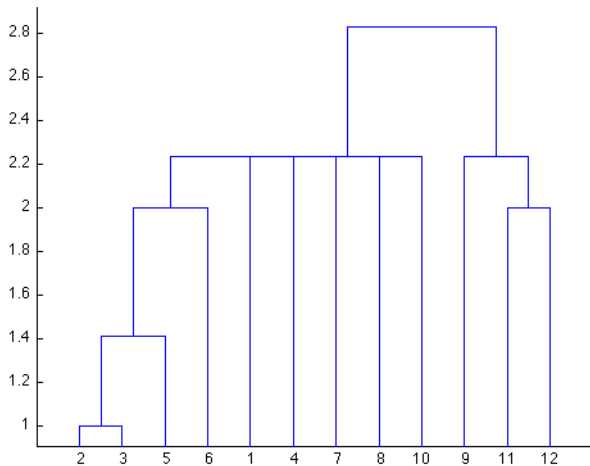
WirginiaTech

Here is some simple point data to analyze:
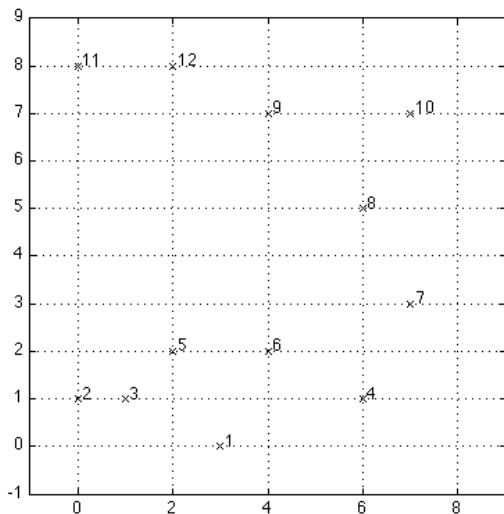
Single Linkage Clustering: Applied to Simple Data

Here is the resulting tree diagram:

Our car data was **not** derived by a distance function, so we can't call **pdist**. But actually, we already have the distance table that **pdist** would have computed for us.

Our only task is to convert the **NxN** distance matrix **ct** to an **N*(N-1)/2** vector that lists the entries in the upper part of the matrix. The MATLAB command **squareform** will do this for us.

- **pv = squareform (pt)**: distance matrix to distance vector.
- **pt = squareform (pv)**: distance vector to distance matrix.

```
cv = squareform ( ct );
cl = linkage ( cv, 'single' )
dendrogram ( cl )
```

Single Linkage Clustering: Applied to Car Data

Here is the resulting tree diagram:

Single linkage clustering turns out to be equivalent to the "greedy" algorithm for finding the minimum spanning tree of a graph.
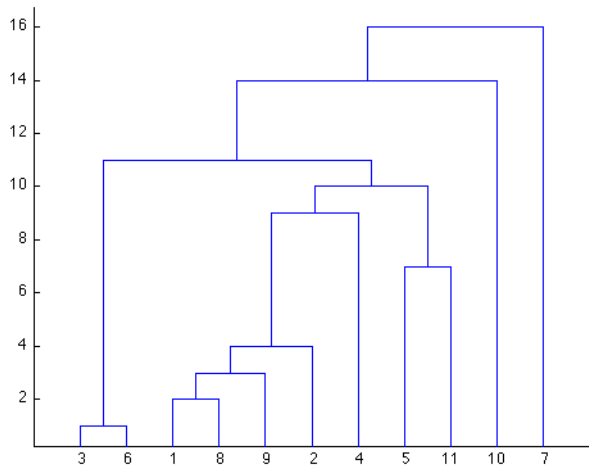
This means that the clustering produced by single linkage clustering minimizes the sum of the "linkages", that is, the distance between pairs of clusters that are merged.
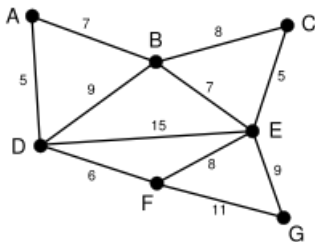
One version of the greedy algorithm starts by "selecting" a node. It then considers all links between the selected node and the unselected nodes, and takes the shortest link, and transfers the unselected node at the end of this link to the "selected list". After N-1 choices, we've selected all nodes.

Another version of the greedy algorithm starts by "selecting" the shortest link. Then it selects the next shortest link, as long as that link does not create a circuit. After N-1 choices, we will have connected all the nodes.

- Overview
- The Average of a Set of Data
- Distance Functions
- Single Linkage Clustering
- **Chain Letters**

WirginiaTech

In 1952, a letter was mailed anonymously from Tennessee:

This Prayer has been sent to you and has been around the world four times. The one who breaks it will have bad luck.

The Prayer. Trust in the Lord with all thy heart and lean not on thy own understandance in all thy ways acknowledge him and he will direct thy path.

Please copy this and see what happens in four days after receiving it. Send this copy and four to someone you wish good luck. It must leave in 24 hours. Don't send any money and don't keep this copy. Gen Patton received $1,600 after receiving it. Gen Allen received $1,600 and lost it because he broke the chain.

You are to have good luck in 4 days. This is not a joke and you will receive by mail.

VirginiaTech

Copies of this letter were still being received 40 years later. Especially in the beginning (before Xerox copiers, and then email) a person had to write the copies by hand, or typewrite them. A copy received after 40 years had probably been copied 4,000 times!

All copies of the letter can be traced back to one person, the original sender. But any two copies of the letter may come from a single sender much more recently.

The version that was seen in the 1980's was very interesting, because it mentioned several people by name. However, because of carelessness in copying, these names varied from copy to copy. The copying errors make it possible to try to cluster the chain letters. Letters in which the names do not differ much probably came from the same person not too long ago. If letters have many differences, their common ancestor must be further back.

VirginiaTech

The names in the chain letter are useful because...they are not important! They can change without the letter losing its meaning.

There is a similar situation in DNA. Substantial amounts of DNA are the same for all animals because it contains vital information that can't easily be changed without harming the animal.

But there is much genetic material that doesn't do anything. It's called **junk DNA**. Since it's useless, it can change and the animal will pass on the mutation. Thus, junk DNA can be used to track relatedness of individuals.

The names in the chain letters are like junk DNA. As we watch them change, we can try to guess the pattern of ancestry!

Here are some key points in the chain letter:

1. The original is in **New England**.
2. An **R.A.F.** Officer received $470,000.
3. In the Philippines, **Gene Welch** lost his wife...
4. The chain...was written by **Saul Anthony de Groda**...
5. Do note the following: **Constantine Dias**...
6. **Carlos Daddit**, an office employee...
7. **Dalan Fairchild** received the message...
8. In 1987, the message received by **a young woman in California**...

WirginiaTech

To compute the distance between two chain letters, first locate the 8 names that show up in the positions described on the previous slide.

We are going to compute the distance between the two chain letters based on their agreements on these 8 points.
We will assign a score for agreement or disagreement on each point, and then sum to get the distance.

When we say **New England**, we don't mean the chain letters must both have that particular word. We are instead trying to suggest the position in the letter that must be checked. So we mean, do the chain letters both have the same word there (perhaps **New Zealand!** or not.

WirginiaTech

1. **New England**, 0 if the same, 1 otherwise
2. **R.A.F.Officer**, 0 if the same, 1 otherwise.
3. **Gene Welch**, 0 if the same, 1 if one name different, 2 if both different.
4. **Saul de Groda**, 0 if the same, 1, 2 or 3 if 1, 2 or 3 names different or added
5. **Constantine Dias**, 0 if the same, 1 or 2 if 1 or 2 names different
6. **Carlos Daddit**, 0 if the same, 1 or 2 if 1 or 2 names different
7. **Dalan Fairchild** 0 if the same, 1 or 2 if 1 or 2 names different
8. **a young woman in California**, 0 if both letters have or don't have this part, 5 if one does and one doesn't

WirginiaTech

You will be assigned one of the 11 chain letters, which are labeled **A** through **K**.

Compute the distance of your chain letter to all the other chain letters. Do this by computing the pairwise score of your chain letter against the others.

When you are done, you will have computed 11 numbers, which represent one row of the distance matrix. When I have collected all the rows from you, I can do a single linkage clustering.

This assignment is due by Monday, 21 September. You can submit it to Scholar or turn it in on Monday.

VirginiaTech