Cluster Analysis for Survey Exploration [Draft 2009 - 05 - 25]

If you can distinguish birds from snakes, or potential partners from mating disasters, then you are already doing *cluster analysis*. This is what the dating services do when they match you up with a *compatible cluster of* persons, pick one! Personality profiles do this too, when they tell you what cluster of others you are most like (or unlike). This note will simply help you put a little math behind these ideas. (The math needed here is very modest, consisting only of vectors and distances between them. For a refresher you might want to take a look at my web site *milagrosoft.com* and check out the 4 page tutorial called *Vector Operations Quick Look)*.

The technical term for what we are about to learn is called *cluster analysis.* The purpose of cluster analysis is to distinguish similar groupings of entities based on their attributes, features, factors, form, or characteristics. This technology could be useful for you in a survey context since it could help you to group your survey respondents on the basis of several criteria such as annual income, number of children, home owner, age, or years of schooling. Or, suppose you are looking at U.S. educational data. What if you clustered the United States *states* on the basis of their per-capita educational budgets? What states would be most similar under that criterion? California closest to New Hampshire and Kentucky, or maybe Arizona, New Mexico, and Utah would cluster together?

Cluster analysis is the simplest of the *multivariate techniques* and is a good first choice when confronting a confusing collection of entities either extracted from surveys or other sources. I will explain the most common clustering algorithm that uses triangular distances (Euclidean) between entities, based on their attributes. You can go further and visualize these clusterings and their separations by constructing a 'tree', called a *dendrogram*, that gives you a graphical view of the connections between clusters and some numeric measures of how close each cluster is to the others. You will see a dendrogram for the *Cut to the Chase* example.

In this document, I use the *Mathematica* package for everything: math calculations, graphics, as well as typesetting this document. As I explain the ideas behind the most common technique of cluster analysis, I use the *Mathematica* package to do the calculations and graphics. You will find many other packages that will do Cluster Analysis and I recommend using one of them as soon as you get an idea of what is going on behind the curtain!

Cut to the Chase

This section can be read quickly to see if it is worth checking out the rest of the note. Consider the table below showing 5 countries labeled 1 through 5, each being measured on 2 attributes/characteristics: X1= current per capita spending on education, and X2= current country GDP. How to break up these 5 countries into similar clusters based on their characteristics? That's the task. The solution is to calculate distances between entities, represented by their attributes, and then sort the distances, from smallest to largest. Smaller is closer and is the basis for deciding which clusters ought to be merged into new, larger clusters.

To start off, it is helpful to think of *each* entity as its own 'cluster'. So, in this case I start with 5 clusters. To find the next cluster, I calculate the distance between every pair of entities. For example, the distance between entity 3 (cluster <3>) and entity 4 (cluster<4>), is the distance between the two points (vectors) in space with X-Y coordinates of {30, 10} and (30, 15}. That distance would be:

$\sqrt{(30-30)^2+(10-15)^2} = 5$

Doing this for every pair gives me a 'similarity' array of numbers. Take the smallest number and that pair of entities becomes the next 'cluster'.

It turns out that entity 3 and entity 4 *are* the closest, at a distance of '5', so they are merged to become my next cluster, denoted <3, 4>. That is, entities 3 and 4 are no longer distinct but are replaced by a new entity the 'cluster' <3, 4>. For this new <3 , 4> cluster I replace the individual entity locations by a location calculated as its center of mass (taking each entity as having a unit mass). In this case that is the point (vector) ({30,10} + {30,15})/2 = {30, 12.5}. In other words, the individuals, entity 3 and entity 4, are now represented by a single new point (vector) that is their center of mass. All calculations now use this location in place of the individual entities 3 and 4.
To find the next cluster I again calculate the distance between each cluster using the 'merged' cluster <3, 4>'s center of mass. That will give me an array of numbers from which I again pick the smallest to be my next cluster and so on. In general, each discovered cluster is replaced by its center of mass for use in subsequent calculations. I will show this below if you would like to continue.

End Cut to the Chase

- **A few math preliminaries for setting up the cluster analysis task**

I have shown the actual *Mathematica* steps needed for this task, as a possible insight for those wishing to use a similar package. For many statistical/math packages, the reader may find graphical inputs that make this Cluster Analysis task much smoother. I prefer this step by step approach though.

```
mat = {{"Country  ID ", "Per-capita spending\non education\n  X1", "GDP\n  X2"},
       {1, 10, 5},
       {2, 20, 20},
       {3, 30, 10},
       {4, 30, 15},
       {5, 5, 10}};
```

- **Think of the entity locations in space in terms of vectors of their attributes**

A natural way to think of these entities is in terms of their attributes, which are *vectors* located in 2-dimensions having unit mass at that point (I think of each point as having unit mass so that I can use the physical idea of *center of mass* in subsequent calculations, as you will see):

entity 1 represented by the vector **r1** = { 10, 5}, and having unit mass at that point
entity 2 represented by the vector **r2** = { 20, 20}, and having unit mass at that point
entity 3 by **r3** = {30, 10}, and having unit mass at that point
entity 4 by **r4** = { 30, 15}, and having unit mass at that point
entity 5 by **r5** = {5, 10}, and having unit mass at that point

```
x1 = { 10, 20, 30, 30, 5}; x2 = { 5, 20, 10, 15, 10}; pairs = Thread[List[x1, x2]];
```

```
Grid[mat, Frame → All]
```

Country ID	Per-capita spending on education X1	GDP X2
1	10	5
2	20	20
3	30	10
4	30	15
5	5	10

Graphic view of the entities in 2 - dimensions

It' s pretty clear from the diagram below how these countries cluster, but if you had to deal with maybe 50 or 100 then you definitely need some computer package help!

- **Starting the configuration with 5 clusters (the individual entities are taken to be the starting clusters)**

First I compute the distance between each pair of entities, to find the pair with the smallest distance. For example, the distance between entity 3 and 4, in terms of their attributes is:

$d34 = \sqrt{(30 - 30)^2 + (10 - 15)^2} = 5,$

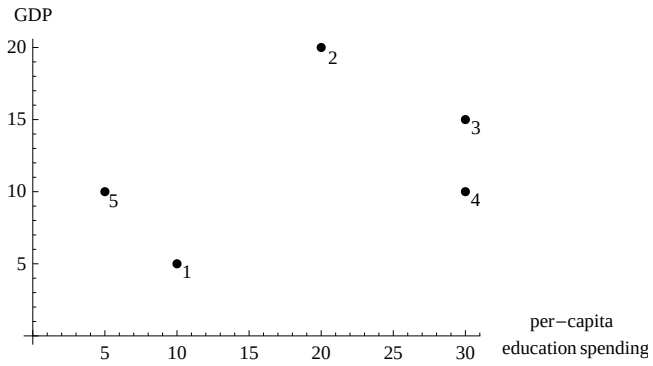
while the distance between entity 3 and 1 is:

$d31 = \sqrt{(30 - 10)^2 + (10 - 5)^2} = 20.6$

Doing this for every pair shows that d34 is actually the smallest and therefore <3> and <4> are most similar and so will be merged to constitute my first cluster denoted by <3, 4>. By merging I simply mean that the <3, 4> cluster is characterized by its *location* given by its center of mass.

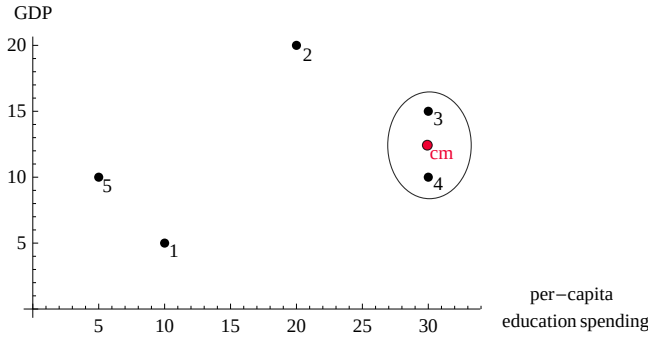
- **Plotting and labeling the entities using their attributes as coordinates**

```
Graphics[ {PointSize[0.02], Point[pairs ]}, Axes → True, AxesOrigin → {0, 0},
 AxesLabel → {"per-capita\n education spending", GDP}]
```



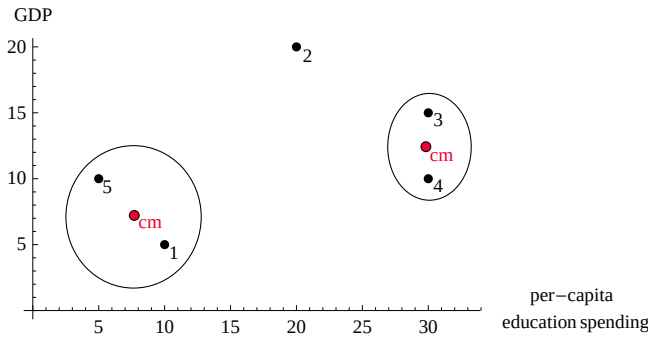
- **Second cluster configuration, characterizing the cluster consisting of <3, 4>**

Entity 3 and 4 are now considered a single cluster which I will denote as <3, 4>. I need to use a single number to describe that cluster's location. Among many variants, the idea of using the *center of mass* is most appealing to me. Recall that statistics as well as physical descriptions use the idea of center of mass to describe a *central location*. For the pair of vectors that represent entities 3 and 4 we calculate their center of mass:
r3 = { 30, 10}, **r4** = {30, 15}. Thinking of a *unit mass* being at each location, then the center of mass is the midpoint of the line joining **r3** and **r4**. (or more intuitively, the vector to that midpoint). In more detail,
$r_{\textbf{34 centerof mass}}$ = (1* **r3** + 1* **r4**)/ (1+1) = (**r3** + **r4**)/2 = (30 + 30, 10+15)/2 = {30, 12.5}



- **Third configuration, with <1>, <5> being clustered**

Using the center of mass for <3, 4> and recalculating distances between all clusters, shows that the smallest distance is between entities 1 and 5, so that becomes the next cluster, <1, 5>. The center of mass for that cluster is shown below.



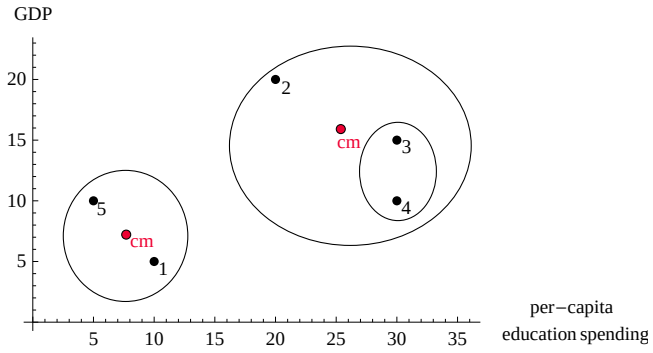
- **Fourth configuration with <3 , 4> merged with <2>**

After calculating the distances between the merged clusters <3, 4> and <1, 5> and the singleton <2>, I find that <2> is closer to the <3, 4> cluster than the <1, 5> cluster. So, <2> is merged with the <3,4> cluster to yield a new cluster with a center of mass location of

$r_{342\ \text{centermass}}$ =(**r2** + **r3** + **r4**)/3 = ({20,20} + {30,10} + {30,15})/3 = {80/3, 45/3} = {26.7, 15}.

Note: The student of physics may recognize that multiple centers of mass may be combined to yield a new center of mass instead of going back to individual masses.

That new center of mass is shown in the diagram below.



Finding clusters the Mathematica way

```
Needs["HierarchicalClustering`"]
```

Below I simply invoke the Mathematica built in command to find clusters. The package uses a variant of what I have presented but the clusters are the same. The output shows that the entities 1 and 5 form a reasonable cluster as contrasted with entities 2,3,4 as we discovered in the discussion above.

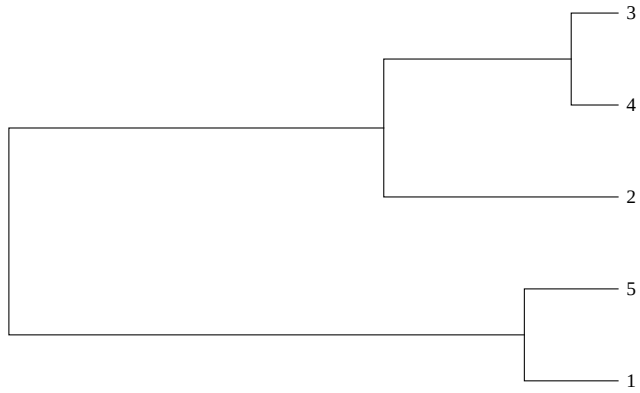
```
FindClusters[pairs → Range[Length[pairs]]]
```

```
{{1, 5}, {2, 3, 4}}
```

Final configuration of the country clusters showing a dendrogram

This diagram shows the clusters at the tips of the ' dendrogram' tree. Country 3 and 4 are the closest while the cluster of << 3, 4>,2> is distinguished from the cluster of <1, 5>. This diagram shows how entities 3 and 4 are linked up (actually at a distance of '5' as we calculated earlier). These two entities are then linked up with entity 2 and that triple being distinguished from the cluster consisting of entities 1 and 5.

```
DendrogramPlot[pairs, LeafLabels → Automatic, Orientation → Left]
```



A little more realistic example

Here is some made up data that might have been obtained from a survey where the first attribute could be weight and the second a measure of bone density identified by respondent.

```
records = {{"Edward", "Dijkstra", 150, 50.4}, {"Mary", "Shaw", 140, 64.4},
   {"Bob", "Martin", 130, 88}, {"Martin", "Fowler", 235, 71.1},
   {"Rebecca", "Wirfs-Brock", 225, 71.4}, {"Grace", "Hopper", 168, 62.},
   {"Tom", "DeMarco", 243, 70.9}, {"Thomas", "Erl", 225, 71.4}};
```

- **Since I only want to cluster on the numerics, I will drop the identifying names but will pick them up for the final display.**

```
data = Drop[records, None, {1, 2}]
```

```
{{150, 50.4}, {140, 64.4}, {130, 88},
 {235, 71.1}, {225, 71.4}, {168, 62.}, {243, 70.9}, {225, 71.4}}
```

- **This next step clusters the data and then labels the data entries with the original identifying names**

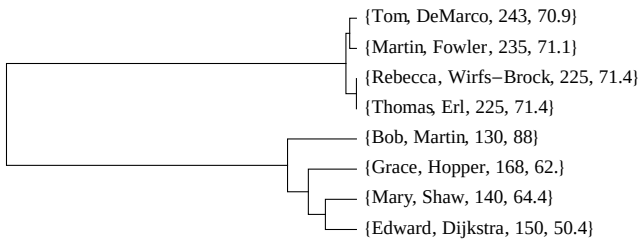
So, Edward, Mary, Bob and Grace fall into one cluster while Martin, Rebecca, Tom and Thomas fall into the other.

```
FindClusters[data -> records]
```

```
{{{Edward, Dijkstra, 150, 50.4}, {Mary, Shaw, 140, 64.4},
  {Bob, Martin, 130, 88}, {Grace, Hopper, 168, 62.}}, {{Martin, Fowler, 235, 71.1},
  {Rebecca, Wirfs-Brock, 225, 71.4}, {Tom, DeMarco, 243, 70.9}, {Thomas, Erl, 225, 71.4}}}
```

- **A tree view of the clusters**

```
DendrogramPlot[data, LeafLabels → records, Orientation → Left]
```



Summary

As you can gather, clustering is a very human capability and is essential for our survival. Cluster Analysis is not essential for our survival but is a useful tool when you want to organize your entities on the basis of their attributes. There is an extensive literature about this technology with bewildering options as to distance measures and number of cluster to consider. Note that there is a large measure of your judgment in these analyses since all the math can do is calculate distances, you have to decide what distances are significant.