Audio transcription of the clustering course

Part 1. Hierarchical clustering

(Slides 1 - 13)

Slide 1

This week, we're going to look at clustering methods, including hierarchical clustering, and a partitioning method called k-means.

Slide 1b (outline)

The course videos for this week get into the following things: After a brief introduction on data of interest for clustering, and the goals of clustering, we are going to have a look at some general principles of clustering, and in particular, hierarchical clustering. We'll have questions like: what criteria to use? Which algorithm to use? Then, we'll take a close look at a partitioning method, the well-known k-means algorithm. Following this, we'll get into how we can use hierarchical clustering and k-means at the same time, and how to do clustering with high-dimensional data, and also qualitative data. To finish, we'll consider ways to characterize individuals found in the same classes.

Slide 2 (outline continued)

Let's begin this video with a few definitions, followed by a first look at hierarchical clustering.

Slide 3

What does the word clustering mean to us? Here, it means putting together, or building, classes, clusters, groups, categories. Classes are sets of individuals or objects which have some common features. What do we mean by common features? Well, traits, characteristics, resemblances, from the set of characteristics defining each individual or object. Anyway, the whole idea of clustering should come as no shock to you! You already know plenty of examples. For instance, the animal kingdom is simply a particular hierarchical clustering of species. The hard drive of your computer, organized into folders, subfolders, and files, is another.

Classes, you already know a lot about them too. Social classes, job categories, political classes. Pretty much, we're grouping together individuals with shared characteristics. Which brings us to two different types of clustering. The first, hierarchical, involves building a tree-like structure to see how the individuals or objects are organized with respect to each other. For obvious reasons, this is called hierarchical clustering. And then there are partitioning-type methods, where we simply try to divide, i.e., partition, all of the individuals or objects into groups, where the members of each group have certain similarities to each other.

Slide 4

Here is an example of a hierarchical tree, the animal kingdom. Look at how it divides into branches. There are branches for arthropods, annelida, Mollusca, chordata. Then for the chordata there are two branches: vertebrates and invertebrates. For the vertebrates, there are new branches: bony fish, amphibians, reptiles, birds, mammals.

Within the mammals, it branches yet again and so on. And as we go deeper and deeper into the tree, we find more and more similar types of animals in each smaller and smaller branch. Deep into the tree, animals in the same branch become very alike. In contrast, as we move in the other direction, classes or branches join together, and on average, the animals in the bigger class resemble each other a little less. And so on, and so on, up the tree.

Slide 5 (outline)

Let's now look at the general principles of hierarchical clustering. We're going to see what criteria to use, as well as algorithms helpful for building hierarchical trees. We'll also look at how to quantify the quality of a partition, and take a closer look at a method for Euclidean data, called Ward's method.

Slide 6

So, what types of data are we going to do hierarchical clustering with? Well, it's the same types of data we had for principle component analysis, that is, data tables with individual as rows, and quantitative variables as columns. Even though the variables are quantitative, we'll see at the end of the video what to do when we have qualitative ones. The goal of our clustering is to build a tree structure with hierarchical links between individuals, or groups of individuals.

For example, for the hierarchical tree here on the right, individuals A and C are very close to each other, and also fairly close to individual B. These three individuals form a class. Similarly, individuals D, E, F, G and H are similar to each other, and within this group, F and G are even more similar to each other. This tree-like representation also helps us to find the "natural" number of classes in a population. For example, here it would make sense to split the set of individuals into two classes: A B C, and D E F G H.

Slide 7

Well then, what is our clustering criteria going to be? To perform clustering, we are going to have to define a similarity measure between individuals. When are two individuals close to each other? When are we going to put them in the same class? A standard and natural distance measure that is often used to visualize the data is the Euclidean distance. We've already used this measure in PCA, and we're going to see that, in clustering, on tables with individuals as rows and variables as columns, that it's also a natural measure to use.

There are also many other similarity measures that could be used, which are sometimes associated with specific subjects. In ecology, for instance, the Jaccard index is often used. Indeed, there are a huge number of ways to define similarity between individuals. In this course, we are mostly going to focus on the Euclidean distance, which will help us to make a link with factor analysis methods and representations, which are also Euclidean. So, that was the similarity between pairs of individuals. Then how does similarity work between groups of individuals?

In the little drawing on the right, we can see two groups of individuals, and a first idea for measuring the similarity between the groups: the single linkage. It's the red line in the picture. In this measure, the minimal distance between two groups is equal to the smallest distance between one individual of one group, and one individual of the other. Another distance measure between two groups is the complete linkage which is the largest distance between an individual of the first group, and one of the second.

Later in the course, we'll have a look at another similarity measure, called the Ward criterion. As you can see, there are many similarity measures possible between individuals and between groups. The choice of similarity measure is going to change the clustering result we get. Therefore, depending on the data, we're going to use certain similarity measures between individuals, and others between groups of individuals.

Slide 8

Starting from simple example, we're going to build, by hand, a hierarchical tree, in order to understand how the algorithm works. Here, we consider eight points: A B C D E F G and H, and their coordinates in two dimensions. We can therefore plot these points and visualize the distances between them.

To start with, each point represents a class made up of just one individual, itself! That's why each point is surrounded by a little ellipse containing only that point.

The first thing to do is then to calculate the distances between the points. Here, we'll use the Euclidean distance. We end up with this distance matrix. For example, between points A and B, we have a distance of 0.5, between A and C, a distance of 0.25, and between B and C, 0.56, etc.

What's the next thing we do? Go and find the smallest distance in this matrix. It turns out to be 0.25, corresponding to the points A and C.

We thus start to build our hierarchical tree by regrouping the points A and C. This happens at a height of 0.25, that is, the distance between these two points. So, now we have groups of individuals made up of just one point each, and one group with two points: A and C.

Next, we have to calculate the distance between each individual and the group A C. To do so, let's use the minimum distance criterion. For the distance between A C, and B, for the minimum distance, it's going to be 0.5. That's because the distance between A and B is 0.5, and the distance between C and B is 0.56. So, the smallest of these two distances is 0.5. We then repeat the same process between the A C group and each individual. This gives us a new distance matrix.

Then, we look for the smallest distance in this new matrix. It turns out to be the distance between the group A C, and the individual B.

Therefore, we're going to join together these two groups, at a height of 0.5 on the tree. We now are left with a group of A B C, and five groups of one individual each.

We now calculate the distance between the A B C group, and each of the individuals, to get a new distance matrix. In this matrix, we see that the smallest distance is between F and G, so they must be put together to form a new group.

These two points are joined together at the height of 0.61

We then calculate the new distance matrix, which now contains the group F G. This time, the smallest distance is between D and E, equal to 1.

We join D and E at a height of 1. So far, we therefore have groups A B C, D E, F G, and H.

Time to calculate the next distance matrix. The smallest distance is between F G, and the point H.

So we join F G with H, at a height of 1.12.

So now we have only three groups left. In the next round, the smallest distance is between the D E group, and the F G H group.

We merge them at a height of 1.81.

Only two groups are left now. And they join together at a height of 4.07.

The tree is now finished! Therefore, iteratively, we've managed to build a hierarchical tree by joining together, step by step, the closest groups. Obviously, building a hierarchical tree is pretty simple here, because we have only a few individuals. But as soon as we have more, it becomes harder and harder to build a tree by hand, and we soon must switch to doing it with a computer.

Slide 9

Once we've constructed a tree, a hierarchical tree, we can do what happens to many real trees in the end: cut it! Here, we are going to cut the tree in order to get classes.

By defining a certain height at which to cut through the tree, we end up creating a partition. In the next tree, the cutting height, represented by the black line, defines a partition into four classes. Therefore, it's clear that defining the cutting height corresponds to defining the number of classes.

Clearly, having seen how we build trees, this particular partition is not necessarily "optimal". Indeed, building the tree involved hierarchical constraints between individuals and groups of individuals, which is not particularly helpful when it comes to defining partitions. By removing this hierarchical constraint, it's possible to improve the partitioning, as we'll see at the end of this section. Though, even if the partition obtained by cutting a hierarchical tree is not necessarily "optimal", it's nevertheless often a quite good quality one.

Slide 10

So what does it mean to have a good-quality partitioning? Does it mean that individuals in the same class are very close to each other? That they have similar features? Of course, a partitioning will also be good if individuals from different classes are far from each other, with few features in common. How can we translate these ideas into mathematics?

Well, individuals in the same class are close to each other if the within-class variability is small. That is, inside the class, there is very little variability, and the individuals have a strong resemblance to each other. And, conversely, individuals from different classes will be far from each other if, from one class to the next, there is great variability. That is, there is a large between-class variability. Overall, we thus want to have a small within-class variability, and a large between-class one.

So, which of these two criteria should we focus on? It's always a bit delicate to choose one or the other, but thankfully, here, they both basically represent the same thing.

Slide 11

What do we mean by this? Well, the total inertia, the total variability, represented in blue, can be broken down into a within-class variability (shown in black), plus a between-class variability (shown in red). Therefore, here, Xiqk is the value taken by the i-th individual of the class q for the variable k. X bar k is the mean of the k-th variable. X bar qk is the mean of k in class q, for the individuals in class q. The "within" inertia, it's the variability inside the class, which corresponds to the sum of the squared differences between the Xiqk and the X bar qk.

The "between" inertia is the sum of the squared differences between the means of each class, X bar qk, and the means of each variable, X bar k. Thus, thanks to Huygens' theorem, we know that the total inertia is equal to the "between" inertia, plus the "within" inertia. This kind of reasoning can be done variable by variable, considering the sum over the set of variables, to help understand this total inertia equation, equal to "between" plus "within" inertia. As a consequence, minimizing the "within" inertia is equivalent to maximizing the "between" inertia, because the total inertia is constant.

Which means, surprise! That really we have just one criteria in the end. We can either concentrate on minimizing the "within" inertia, or maximizing the "between" inertia.

Slide 12

This all leads us to put forward an index to measure the quality of a partition: the ratio of the "between" inertia over the total inertia. This ratio gives a value between 0 and 1, and the closer it is to 1, the better the partition is.

The "between" inertia over the total inertia equals 0 when, for all variables k, the X bar qk are equal to the X bar k. This means that all classes have the same mean, for each variable. Well, clearly if all classes have the same mean, we have a partition which doesn't separate the classes, and doesn't help with clustering.

If the "between" inertia over the total inertia equals 1, that means that the "within" inertia is zero. This means that inside each class, the individuals are identical. This means that we have extremely homogenous classes, which is ideal for clustering.

But be careful! Don't overinterpret this criterion. In reality, it also depends on the number of individuals and the number of classes. If we increase the number of classes, it's easier to have homogenous ones. In the other extreme, if we have a small number of classes, the variability inside each will be larger. We should therefore take into account the number of individuals and number of classes before making strong conclusions based on this criterion.

Slide 13

This criterion for the quality of a partition suggests a new way to do hierarchical clustering, which is now known as Ward's method. It works like this: we start from the clustering where each individual is its own class. This mean of course that inside each class, there is no within-class variability, so the between-class variability must be equal to the total inertia. Thus, in a way, it's the perfect partition.

The aim is then to choose two classes, a and b, so that their aggregation minimizes the decrease of the between-class inertia. Essentially, the between-class inertia can only decrease when we aggregate two classes. And we want to minimize this decrease.

Let's have a look at how we write the sum of the inertias of a class "a" and a class "b" as a function of the inertia of their aggregation. The inertia of "a" plus "b" is equal to the inertia of the aggregation of these two classes, minus a certain quantity: (m a time m b), divided by (m a plus m b), multiplied by "d a b" squared. Here, "m a" is the number of individuals in the class "a", "m b" the number in the class "b", and "d a b" squared the squared distance between the centers of gravity of classes "a" and "b". As we want the inertia of their union to be as close as possible to the inertia of "a", plus the inertia of "b", we need to minimize the second part of the formula. Which contains two things: weights, and a squared distance.

First let's look at this term: "m a" times "m b", divided by "m a" plus "m b". This term pushes us to group together objects with small weights, and avoid what are known as chain effects. Here's a little plot with two classes: blue and red, and hierarchical trees, the one on the left using the minimum distance criterion, the one on the right using Ward's method. We can see that when the two classes are well-separated, we get similar trees with both methods, clearly separating the two classes. However, with the same two classes, and some extra points that stretch from one class to the other, the minimal distance criteria is overtaken by a chain effect, bringing individuals into the cluster one by one. The consequence of this is that you can no longer see the two original classes in the hierarchical tree. With Ward's method, on the other hand, due to this weight in the formula, the red and blue classes remain separated in the hierarchical tree.

The second part of the term to be minimized is "d a b" squared. This is the distance between the barycenters of classes "a" and "b". It makes a lot of sense to group together classes whose centers of gravity are close together. In terms of clustering, it's obvious: we end up grouping together classes that are close to each other.

So: so far, we've seen how hierarchical clustering works. In the next videos, we're going to run it on an example, and see how to use it to choose the number of classes, and thus a partitioning of the individuals. Later, in the last video, we'll see how to characterize individuals found in the same class. Don't forget to do the quiz, to be sure that you've understood all of the things we've shown you so far!

Part 2. An example, and choosing the number of classes (Slides 14 - 20)

In the previous video, we saw how to build a hierarchical tree. Now we're going to apply this to a real example.

Slide 14 (plan)

Our illustration involves a data set of meteorological data.

Slide 15

The rows, or statistical individuals, here, are 23 European capitals. The columns are quantitative variables, giving the mean monthly temperatures. These have been calculated and averaged over thirty years. So, for example, in Amsterdam, in January, on average it's 2.9 degrees Celsius. This value of 2.9 is the mean over all the days of January and over 30 years. We have therefore twelve variables, one for each month of the year. And added to this, one qualitative variable, corresponding to the region of Europe the capital is in: north, south, east or west.

We are going to construct a hierarchical tree using only the temperature data. Here constructing a hierarchical clustering means looking to group together capitals with similar temperature data, and then trying to find characterizing features of each group. A little note before we continue: we are going to work with the centered and standardized versions of the variables rather than the raw data, to help give the same importance to each variable when building the hierarchical tree.

Slide 16

Here is the hierarchical tree we get using the Euclidean distance and Ward's criterion. We see for example that Sofia and Sarajevo have similar weather profiles. The temperatures in these two towns are quite similar throughout the year. Another fairly homogenous group is visible: Athens, Lisbon, Madrid and Rome. Going further in, we see that Madrid and Rome are the most similar to each other in this group. We can also get an idea of the distance between cities, and groups of cities. In the bar plot at the top-right, we show the evolution of the inertia for different partitions.

Slide 17

Let’s have a closer look at this plot. It shows the loss of between-group inertia when we group together two classes. More precisely, it shows the loss of inertia when moving from 23 to 22 classes, from 22 to 21, and so on, right down to moving from 2 classes to 1.

If we calculate the sum of the losses of inertia, we get the value … 12. 12 corresponds to the sum of the variances of the variables in the data set, because here we have 12 variables and they have been centered and standardized. Thus, by taking the sum of the losses of between-class inertia, we indeed find the total inertia, which is 12. Let's now try to understand what each bar in the bar plot is telling us. The big bar, on the left, shows the loss of between-class inertia when moving from 2 classes to 1. The loss of inertia is 6.76, which is rather large.

This means that this attempt at grouping together two classes brings together quite different individuals. We really don't want to do this. The red bar shows the loss of inertia when moving from 3 to 2 classes. This loss of between-class inertia is 2.36. This is still quite large, and again, we'd rather not group together the 3 classes into 2. Looking at the plot from the other end of the scale, we see that moving from 23 to 22 classes, from 22 to 21, etc., involves very small losses of between-class inertia, so these groupings are quite acceptable.

So, the question hanging over us is: how far should we go when grouping together classes, and when should we stop? Another way of saying this is: how many classes should we have?

Slide 18

In our example, how many groups should we retain? 2? 3? 4? This is a very good question!

If we choose the cut here in orange, we end up with two groups.

The between-class inertia, as we've seen, was 6.76, as compared with the total inertia, 12. This means a ratio of between-class inertia over total inertia of 56 percent. Thus, by separating the cities into two groups, those in green (Athens, Lisbon, Madrid and Rome), from those in orange (Reykjavik, Moscow, etc.), we recover 56 percent of the information contained in the data set. We are tempted to call this a rough but relatively useful two-class summary of the data.

But what should we compare this 56 percent WITH?

Slide 19

Well, we could compare it with the percentage of variability found in the first PCA dimension in this plot. This represents around 83 percent of the information in the data set. From the clustering, separating just the green cities from the orange ones, we get 56 percent of the information, so rather less than the PCA's first dimension. Indeed, the first dimension gives more detailed information: it separates Athens from Lisbon and Madrid, with Athens pushed further out from the center. Similarly, Helsinki is pushed out much further than Paris, whereas in the classifier, these two cities are in the same class. Clearly, the classifier gives a less-refined summary of the data than the first PCA dimension.

Slide 19 (continued)

Now, if we separate the orange cities, the ones that are cold, into two groups, we will have a clustering with 3 groups in total.

Moving to 3 classes helps us not to lose the inertia lost when moving from 2 to 3 classes. We recover a between-class inertia of 2.36, which is about 20 percent of the information. Separating the cold cities into two groups gives us back 20 percent of the total information.

Slide 20

So, with a 3-class classifier, we recover 56 plus 20, that is, 76 percent of variation in the data. Three classes give us 76 percent of the total variation. Note that with PCA and two axes, we recover about 98 percent of the information.

Slide 20 (continued)

The choice of the number of classes is important, because partitioning with too few classes risks leaving classes which are not at all homogenous. On the other hand, partitioning with too many classes risks creating classes that are not very different from each other. Clearly, the big question is: how many classes should we stop at? One possible strategy is to choose the number of classes by looking at the tree. We might consider cutting the tree somewhere where the branches are quite long. We can also look at the plot with the bars, and choose to stop the grouping together at the point where the jump from one bar to the next becomes small, meaning that little information is gained and it's no longer useful to group together any more classes. Or, looked at from the other end of the tree, there's no point continuing to cut classes in two when the jump from one bar to the next gets small.

Choosing a good number of classes also depends on the data itself, and the number of individuals. If we have a survey with thousands of individuals, it may be useful to have 5, 6, or even 10 or more classes. And if we have much less individuals, it's probably better to have fewer classes. As for the data itself, if we have a huge variety of very different individuals, it's better to have more classes. When all is said and done, the most important criterion is perhaps: after building the classes, can we interpret what they mean, or not? Can we understand and characterize each class? It's useless to divide a class in two if we don't really understand what it is that makes each class different from the other and similar within its own members.

In this example, we've seen how hierarchical clustering helps us to suggest the number of classes to keep. In the next video, we'll look at how to build a partitioning method when the number of classes is fixed.

Part 3. Partitioning methods and other details

(Slides 21 - 28)

Slide 21 (outline)

We've seen in the previous video how to do a hierarchical clustering, with a tree-like structure. Now we're going to look at a direct partitioning method, not requiring a hierarchical tree. This partitioning method is called k-means.

Slide 22

We are going to introduce k-means using a 2-d example. The basic k-means algorithm requires as input the number of classes Q that we want to have at the end. Here, we're going to take Q = 3, and therefore partition the data into three classes.

To begin, we choose three points, or centers, at random: Moscow, Berlin and London. This is the first step of the algorithm, called: initialization.

Next, we associate each individual, each point, with the closest of the three centers. Here, all of the red individuals are closer to Berlin than to Moscow or London. All those in black are closest to Moscow, and all those in green, closest to London. Therefore, we have now divided the individuals into three classes, but it's not all over.

Next, we calculate the center of gravity of each of the three classes. Here they are. They are no longer located on top of individuals, they are just points in the plane. Maybe you can see where we're heading with this. We are going to iterate now, starting from the beginning again, with these three points.

Again, we associate each individual with the closest of the three points. We calculate the center of gravity of each of the three new classes. And we iterate. Again and again. As time goes on, the centers of gravity move, and move again.

Again we calculate the centers of gravity. And this time, we see that the classes haven't changed, and the centers of gravity have stopped moving. This means that the algorithm has converged. We therefore have found our three classes! That's how k-means works. It's a very fast algorithm, it can be run with lots of individuals, BUT, it has a couple of drawbacks. The first, is that you have to know the number of classes before you begin. And the second, is that the final partition depends on the places you put the centers in the initialization step, which were chosen randomly. For different initializations, the partitions we get at the end can be quite different. To overcome this problem, one thing we can do is to run the algorithm many times, and keep the "best" partition, that is, the one with the smallest within-group inertia.

Slide 23 (plan)

So far, we've seen how to do a hierarchical clustering and how to partition the individuals. Here, we're going to look at a few more issues related to clustering, notably how to make a partition robust, how to do partitioning with high-dimensional data, how to do it with qualitative data, and last but not least, what's the interest in combining principal component method and clustering?

Slide 24

Sometimes, we can try to reinforce the partition obtained by hierarchical clustering. We've seen that the partition obtained by cutting a hierarchical tree isn't optimal, because it can be cut anywhere, giving a whole hierarchy of individuals and groups of individuals. Well, we can reinforce a partition found in this way, by using k-means. What we do is start from the partition obtained by a certain cut of the hierarchical tree, and use this as the initialization step in the k-means algorithm, and then iterate until convergence. This can only improve the partitioning, because, by construction, at each step, the k-means criteria decreases. On the other hand, the inconvenience that appears is that we lose the hierarchical links between individuals. Certain individuals are going to change groups with respect to the original tree-based partition, and the hierarchical structure will disappear.

Slide 25

As we saw earlier in the example, constructing a hierarchical tree involves calculating, at each step, lots of distances between individuals and groups of individuals. For high-dimensional data, this algorithm takes too long and doesn't work very well. So how are we supposed to deal with high-dimensional data? Well, if there are lots of variables, we could start by doing a PCA and only keeping the first dimension, or the first few. As we know, PCA concentrates the information in the first dimensions. Then, we can move on and do clustering on a data table with the individuals as rows, and the first few factor dimensions as columns. Pretty much, this means transforming the data to a smaller and much more manageable size.

How about if we have a huge number of individuals?

Here, the clustering algorithm takes too long in practice, and we struggle to build a useful hierarchical tree. One possible strategy is to work in steps. First, we build a rough partition using k-means, leading to a hundred classes or so. This step, due to the large number of classes, ensures that for the most part, individuals close to each other end up in the same class. Then, we can simply extract the set of centers of the classes, and use these to represent each class, and build a hierarchical classifier, treating them as individuals. We'll also use the number of individuals in each class as a factor in the tree's construction. If we think about it visually, the tree that we obtain should resemble the upper part of the tree we would have had if we'd started with all the individuals. We "lose", in a way, the bottom of the tree. Perhaps this isn't such a bad thing if we have a million of individuals! The bottom of the tree would be quite hard to look at anyway, and not of great use, as we are usually most interested in commenting on the top.

Here's an example with 300 individuals. On the left, it's the tree built using all of them, and on the right, the one built after first partitioning them into 50 classes. We see that the tops of the trees are very alike, so our interpretation of them will be similar.

Slide 26

Up till now, we have only treated the case where the variables are quantitative. What can we do if we have qualitative variables? Well, there are two well-known strategies. In the first, we can transform them into quantitative variable using multiple correspondence analysis. MCA can help us move from a table of qualitative variables to one of coordinates of individuals over several dimensions.

These coordinates are indeed quantitative variables. We can keep the first few MCA axes, or even all of them. Once we have rebuilt our table with individuals as rows, and quantitative variables as columns, we can run one of our clustering algorithms, as before.

A second strategy consists of using distance measures specifically adapted to qualitative data. Many such indices exist: similarity measures, dissimilarity measures, and so on, such as for example, the Jaccard index. These types of measures can help us to create a distance matrix or dissimilarity matrix between individuals. Then, using this matrix, we can construct a hierarchical tree, as before.

Slide 27

We've just seen how if we have qualitative data, MCA can give us quantitative principal components, which we can directly use to do clustering with the same algorithms. But these principal component methods have other advantages too. We'd especially like to highlight this general strategy of using principal component method followed by clustering. So, why do you think this might be a good thing?

Well, principal component method, whether it be PCA, or correspondence analysis or multiple correspondence analysis, tends to concentrate the useful information in the first principal components, leaving just noise in the remaining ones, random noise. Doing factor analysis before clustering is therefore a way to try and remove noise before doing clustering. We get rid of the random, that is, the noise, before classifying, which gives, in practice, a more stable classifier. When we say "more stable", we mean in the sense that adding or removing a few individuals doesn't perturb the class structure too much.

Slide 28

Another positive when using principal component methods along with clustering, is the possibility to be able to represent the hierarchical tree, as well as the classes, on the first dimensions. If we manage to put the points, the hierarchical tree, and the individuals colored in terms of their class, on the same factor plane, we get three types of information for the price of one. In particular, we get a continuous vision of how things are, due to the factor analysis, and a discontinuous one using hierarchical clustering, which also gives us information about what is happening in the 3rd and 4th dimensions, for example.

In our example, the individuals are perfectly represented in the first plane, so their proximities in this plane are the same as their proximities in the higher dimensional space. But, in general, it's possible that individuals can seem close in the first plane, but actually be far apart in subsequent dimensions, which the hierarchical clustering is able to show in this kind of plot. In other examples, it may be that three or four dimensions are needed to interpret the PCA well. Overall, this kind of plot is very useful, bringing together class information, and more subtle information from the factor plane.

That's all for this video, where we've seen how to construct classes in the data. In the next video, we'll take a closer look at how to describe and characterize groups of individuals found in the same class.

Part 4. Characterizing the classes

(Slides 29 - 42)

Slide 29 (outline)

Now that we've seen how to construct classes of individuals, let's take a closer look at detecting and describing common features of individuals put in the same class.

Slide 30

To start with, we can look for the "model individual", that is, the one closest to the center of the class. Why are we particularly interested in this one? Well, because the actual center of gravity of the class is likely to be a point in space rather than exactly on top of one individual, it's clearly a better idea to use a real individual to understand the "average" features of a class. Here, for example, we see that in class 1, it's Oslo that's the closest to the center. It's at a distance of 0.34 from that class' center of gravity, so to look at how the class is on average, we can take a closer look at Oslo. In class 2, it's Berlin that's the closest to the center. And for class 3, Rome. People who have done a lot of data analysis already know that these "model individuals" will be quite useful to us.

Here, we show how we can put these model individuals on the factor plane. Here is the center of gravity of class 3, and here, the model individual, Rome.

Slide 31

This characterization of each class using the closest individuals to the center, is helpful but not really enough to say a lot. We'd also like to describe classes using the variables. The goal is to find the variables that most influenced, overall, how the partition was formed, or the variables that best represent a given class. In terms of questions, we therefore have: What variables most influence the partition? What variables best describe or represent individuals in class 1?

Slide 32

The partition, that is, the division of the individuals into the various classes, can be seen as a qualitative variable with as many categories as there are classes. Thus, looking for the variables that best characterize a partition, really just means looking for the variables that best characterize a qualitative variable. For each quantitative variable, we can build an analysis of variance model for it, which will play the role of response variable, as a function of the class variable, which will play the role of covariate. In summary, we do an analysis of variance of the quantitative variable as a function of the class variable, and run a Fisher test to see if the class variable has an effect on the quantitative one.

One way to proceed is then to keep variables with p-values below 5 percent, and sort them by increasing p-value. In our example, we see that the variable that best characterizes the classes is October; it has the smallest p-value, and best separates the classes.

But, be careful! Remember that, like when we described the dimensions in PCA, we have to use these tests wisely, because the October variable plays an active role in the construction of the classes, as it's part of the distance calculations between individuals. To be rigorous, only the p-values associated with the supplementary variables of north/south/east/west can be "legally" interpreted in the usual way. But the p-values for the internal variables are still useful, even if they can't be used like in classical tests. An F-test just helps us find variables that more-or-less influence the partition.

Next, let's try to understand what each class can be defined or described by. Which variables are most important for each class?

Slide 33

This plot helps us to visualize the data. Each row corresponds to a variable, each point to a city. The black points represent Helsinki to Stockholm, the red points Amsterdam to Sofia, and the green ones, Athens to Rome. Now, what if we want to see if there are certain variables, that is, months, that describe effectively a certain class? Are the values of a certain variable particularly outlying for the individuals from one of the classes?

For example, we have that for January, the black cities take smaller values that all the others, so it's clear that the January variable can be used to precisely define the black class. Similarly, we have the impression that the green cities take larger values than the others. Obviously, it's too much work to go through each variable and each class, one by one, with a plot like this.

What we really need is an automated procedure.

Slide 34

The basic idea we're going to use to decide whether a variable describes a class well, is the following: if the values of a quantitative variable X for the individuals of class q seem to be randomly drawn from all possible values of X, then X does not characterize that class well, because those individuals don't take any particular precise set of values of X. Inversely, if the values of X for the individuals of class q are found in a particular region of the possible values of X, then it’s doubtful that they were drawn randomly from across X, and we can state that the variable X in some way describes that class.

Here the values of two variables are shown. One variable with random values, the other showing the values for the January variable. We can see that for the randomly drawn variable, we have black, green and red points mixed up everywhere, whereas for the January variable, there seems to be much more structure in the location of points. For instance, January seems to characterize well the green class, by the fact that its points take all the most extreme values to the right.

We can also consider that the more are random draw appears unlikely, the more X characterizes the class q. So, how to we go from this intuitive idea to a real statistical test?

Slide 35

The underlying strategy is to compare the actual data with n q randomly drawn values from N. The class q has n q values, and we want to know if we can reject the possibility that these seem to be randomly drawn from the N values in the overall population. To study this, we need to know what values X bar q can take, the mean value of that class. So, is there any way to know what the distribution of X bar q looks like?

Well, the expected value of X bar q, under the hypothesis that this class' values were randomly drawn, is simply the population mean value, that is, X bar. In order to calculate the variance of X bar q, we have to draw without replacement n q values from within a population of size N. Therefore, it's the standard error of the population, here, s squared, divided by the number of individuals drawn, n q. Though, because we are dealing with a finite population, there is a correction factor of the square root of (N minus nq) over (N minus 1). As for the distribution of X bar q, since it's a mean, we can suppose that it follows the Normal distribution, thanks to the Central Limit Theorem.

We can therefore now calculate the standardized and centered X bar q minus X bar, over the standard error of X bar q, that is, the square root of the variance of X bar q. This quantity, under the hypothesis of random draws, follows a standardized Gaussian distribution. If the test statistic is between -1.96 and 1.96, we can't reject the hypothesis that the data could have come from a standardized Gaussian distribution.

However, if the absolute value of the test statistic IS above 1.96, then this test statistic is uncommon, so it is unlikely that the data came from a standardized Gaussian distribution. This would lead us to question the hypothesis that these n q points were randomly drawn from N points. That is, the values of X for class q show some structure, so, to some degree, X characterizes class q. Taken to extremes, the more the absolute value of the test statistic is large, the more the random draw hypothesis is doubtful, and the more X can be though of as characterizing class q.

So, what we can do is rank the variables in terms of increasing absolute value of the test statistic.

Slide 36

Here are the results for the black class. The test statistic has the largest absolute value for the month of March. We need to read this table from the bottom up, because the actual test statistic values are negative here, so the most extreme ones are at the bottom. The mean for the individuals in the black class for March is -1.14 degrees, whereas the mean for all individuals, including this class, is 4.06. We see that the standard error in this class is 1.1, and the population standard error is 4.39. The last column shows the p-value associated with the test for whether the data follows a Gaussian distribution. We see that there are lots of very small p-values, so this class of cities is characterized by many of the "month" variables. All of the test statistics are negative, signalling that the values taken by individuals in this class are smaller than those of individuals in general. And obviously this is true, these are cities where it's cold most of the year!

Slide 37

As for the second, red class, none of the variables help us describe it well, in the sense that these cities don't really take extreme values at one end of the spectrum or the other. They are all in the middle, spread out perhaps, but not far out to the sides.

And for the third green class, we have positive test statistics. The variable that best-describes this class is September, with the most extreme test statistic of 3.81. The mean temperature of the third class is 21.2 degrees for September, whereas it's 14.7 for all individuals. This class of cities are where it's hot most of the year.

Slide 38

Can we also describe the classes using qualitative variables? To do so, we have to again consider that a given partition is a qualitative variable, and look for the links between pairs of qualitative variables, between the partition of the class variables and other qualitative variables. For each qualitative variable, we can do a chi square test between it and the class variable. Then, we can sort the qualitative variables by increasing test statistic. In our example, we only have one qualitative variable, dealing with the location in Europe of each city. We see indeed that this variable is connected with the partition of the cities.

Slide 39

Can we also characterize each class in terms of the categories of our qualitative geographic variable? Take for example the third class, and let's see if the category South describes it well. The basic idea is to compare the proportion of South in the third class with the proportion of South in the whole set of individuals, that is, cities. We can fill in a simple table with just the category South, and all others joined together in another category, called Other. In the columns, we have the third class, and then all the other classes joined together.

Let's now construct a test to compare the proportion of South in the third class with its proportion in the whole population. The null hypothesis H_0 is that the two proportions are the same, and the alternative hypothesis is that South is over-represented or under-represented in the third class.

Under the null hypothesis, the random variable N_mc, which represents the number of individuals with category m in class c, follows a hypergeometric distribution with parameters n, n_m over n, and n_c.

We can then calculate the probability to have an even more extreme value than the observed one, that is, four.

We obtain this table. In the rows, we have the categories. Here, we have just one row, corresponding to South. The first column of the table gives the proportion of towns from the south that are in the third class. The calculation to be done is therefore: n_mc over n_m, giving 4 over 5, or 80 percent. The second column gives the percentage of towns from the third class that are in the south. Here, four towns out of four in the third class are in the south: one hundred percent.

The third column gives the proportion of towns that are in the south, out of all towns, which is n_m over n, giving 5 over 23, or 21.74 percent. Now we can calculate the p-value, which will indicate to us whether this category, South, can be said to significantly characterize the third class. We get a value of 0.000564, which is less than 5 percent. Therefore, we can reject the null hypothesis, and consider that South indeed does characterize this third class.

So, in summary, it is useful to comment on p-values, and compare proportions in a given class with the general population, to see if categories are over or under-represented. The test statistic itself, which is just a way of converting the p-value into a quantile of a normal distribution, also provides this information. If its absolute value is greater than 1.96, we can say that the category of interest characterizes the class of interest. Furthermore, if the sign is negative, the category is under-represented, and if it's positive, the category is over-represented. Here for example, the South category is over-represented in the third class.

And again, like for quantitative variables, all of the categories of qualitative variables used to characterize the classes, can be sorted with respect to increasing p-value.

Slide 40

We can also try to characterize classes using dimensions. These dimensions are quantitative variables, so we can use exactly the same approach that we used a little bit earlier for quantitative variables. We can see that the first class is well characterized by the first dimension. As the test value is negative, this means that the individuals in this class have quite extreme and negative values in the first dimension.

The mean in this class is -3.37, while the overall mean is 0. For the dimensions, the overall mean is always 0, because the coordinates on a principal component are centered. We see that the second class is characterized by the third dimension, with quite negative values for the individuals in this second class, while the third class is characterized by the first dimension, with large and positive values for the individuals in this class.

Slide 41

To conclude, lets quickly recall all that we've seen in this part of the course. First, clustering methods can be used on data which pairs individuals with sets of quantitative descriptive variables. However, if variables are qualitative, we can instead use MCA to transform them into dimensions, which are quantitative variables.

Hierarchical clustering gives us hierarchical trees, which show distances between individuals and groups of individuals. Hierarchical clustering also gives us an idea of the number of classes there are in the data.

Another strategy to build classes is to use partitioning methods like k-means, to support the conclusions obtained from the hierarchical tree, or to improve the class assignation. This can lead to more stable classes, but also to a loss of the hierarchical information from the tree. Remember also that we can use partitioning methods as a pre-treatment, simplification step, when we have high-dimensional data.

And, as we've just seen, we can try to describe and characterize the classes we get, using quantitative and qualitative variables. These can be variables that were already used to calculate the distance between individuals, or supplementary ones. In other words, active variables or illustrative ones.

Slide 42

Remember that a course on clustering with the same methodology of our video can be found in this book. Also, don't hesitate to try and plot the same graphs and get the same outputs seen in the videos using R functions from the FactoMineR package!

That's it for the clustering videos! We recommend going and looking at the FactoMineR video now, to see how to put clustering into practice, and then characterize the classes you've found. Also, don't forget to do the quiz and try the course exercises!