Explanation:

I have sent you a custom\_python\_clustering.ipynb notebook.

I have imported and prepared all the necessary variables you need. The only one you are concerned with should be microstates.

A black screen with green text

Description automatically generated

After running this cell we should have a clean microstates.

Let me explain what each element inside the first 10 elements is:

A black and white table with numbers

Description automatically generated

First I need to explain digitized:

A grid with lines and numbers

Description automatically generated

Imagine this data where x-axis is PCA1 and y-axis is PCA2. I grid each dimension into 10 bins. Of course in the real data I use 7 PCs, but this here is just a visualization to explain to you.

So vector [5,4] represents the data inside this bin. We calculate its probability by counting how many different data points are in this bin and divide by total number of points we have. For example, if this bin [5,4] has 20 points then its probability is 20/2635.

State index is not important for you. Do not use it.

Pk is the cluster label my clustering algorithm gave each bin. So all vectors that have the Pk value are in the same cluster.

A screen shot of a computer code

Description automatically generated

This code here labels each of the 2638 points from the original PCA space with its Pk value.

Pks\_for\_data is a list that tells us the cluster label for each of these points here below.

A screenshot of a computer

Description automatically generated

Why is that useful?

Because if we have one cluster and is equal distance between 4 clusters and we want to decide where to merge it to, then we need to compute pearson correlation coefficient using these PCA values. Every data point with Pk1 vs Pk2 vs Pk3 vs Pk4 will be compared to each other. And we know which Pk each PCs has from pks\_for\_data list we computed above.

So what we should do is group each PCA point in this cluster, then compute pearson between different clusters. Some clusters have 50 points and some only have 3 and some just 1. We need to be careful about all these.

The important variables in the notebook to keep track of are the pks\_fpr\_data which tells us for every data point in our pca space what is the cluster label.

In total we have 455 clusters. 420 of these clusters have less than 5 data points in them. Most of them have just one data point. The goal of this clustering is to achieve a final clustering result where every cluster has a minimum of 5 points (or probability = 5/2638).

We only focus clustering on the keys\_less\_than

A screen shot of a computer

Description automatically generated

LOGIC FOR CLUSTERING:

1. Compute distance using the VECTORS between all clusters (including keys\_less\_than). I showed an example of this in the notebook I sent you before.
2. Select any point inside keys\_less\_than that has the smallest probability (most likely has a probability = 1/2638).
3. Find the smallest distance to this cluster or Pk. Most likely the first smallest distance will be = 1 if we use the Manhattan (city block distance)
4. If there are more than one cluster equally distant from this cluster then merge it with the cluster that has the smallest probability WITH THE CONDITION THAT its probability is less than 5/2638.

What does this mean?

For example, if I have clusters A, B, C, and D. And cluster A is the cluster of interest. It has equal distance to clusters B, C, and D. AND ALL of clusters B, C, and D have probability less than 5/2638: P(B) = 2/2638, P(C) = 1/2638, P(D) = 4/2638, then we choose to merge cluster cluster A with cluster C and relabel them both to be under a new name or old name of both are now cluster A.

1. Update the clusters and their probabilities and their elements.
2. Update distances between new clusters and these clusters. There is an efficient way to do this without having to recompute the distances every time by taking the minimum between these new merged clusters and the other unmerged ones. But we can talk about this later. You can now choose to recompute distances however you like.
3. The next cluster we choose should also have smallest probability. Remember. We are only interested in clusters with probability less than 5/2638, so we should should choose one with the lowest probability (same is in point 2).
4. Find smallest distances.
5. Choose what clusters to merge.
6. If it is equally distant from many clusters and those clusters all have probability GREATER than 5/2638, then we need to use pearson correlation to determine to which of these clusters we merge it to.

EXAMPLE 1:

Cluster A is my cluster I want to merge. It has probability = 4/2638. Its nearest clusters are B, C, and D where P(B) = 20/2638, and P(C) is 50/2638, and P(D) is 12/2638, then we choose to merge with the cluster that cluster A has highest pearson correlation to.

EXAMPLE 2:

Cluster A is my cluster I want to merge. It has probability = 4/2638. Its nearest clusters are B, C, and D where P(B) = 3/2638, and P(C) is 50/2638, and P(D) is 12/2638, then we choose to merge cluster A with cluster B because it has a probability less than 5/2638 and is the smallest one.

EXAMPLE 3:

Cluster A is my cluster I want to merge. It has probability = 4/2638. Its nearest clusters are B, C, and D where P(B) = 3/2638, and P(C) is 2/2638, and P(D) is 12/2638, then we choose to merge cluster A with cluster C because it has a probability less than 5/2638 and is the smallest one. Both clusters B and C have probability less than 5/2638 but C has smaller probability. So we merge cluster A with cluster C.

1. We continue this process until there is no more any cluster with a probability of less than 5/2638.

In total, we should have at least 35 final clusters. I think we should have more.