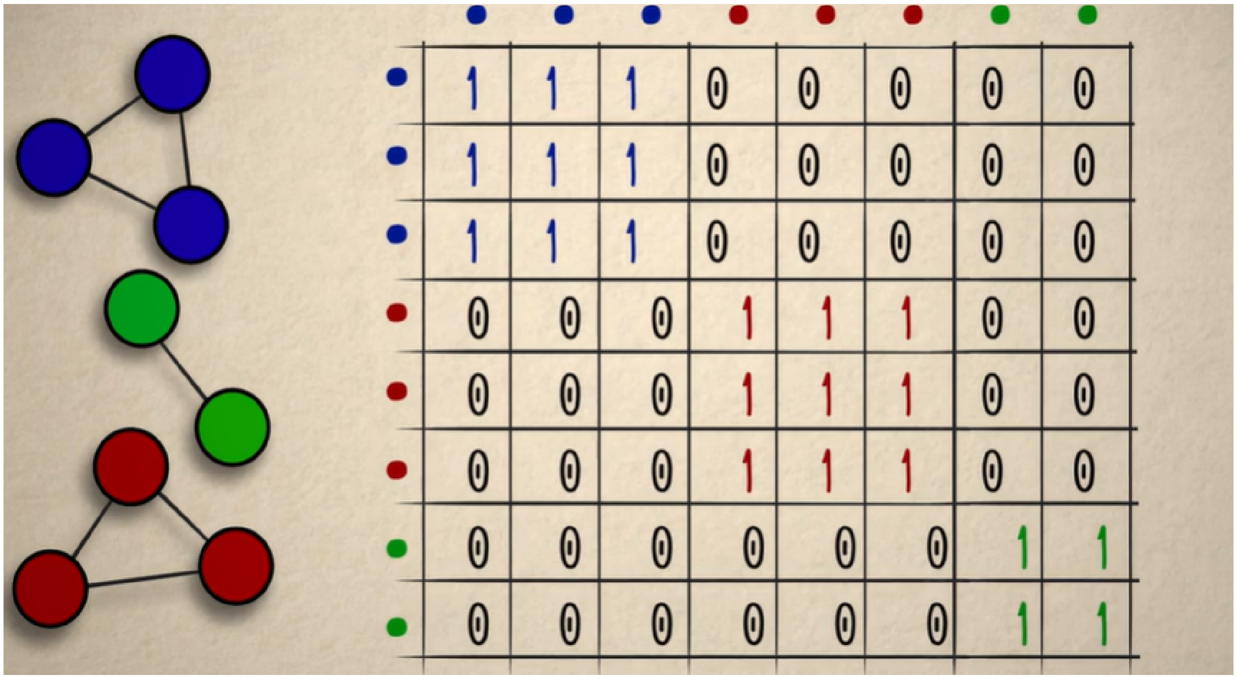
How to use Eigenvectors?

In the previous section, we looked at different criteria to find community in the graph structure and saw how eigenvectors capture important information about the network. In this section, we will put everything together.

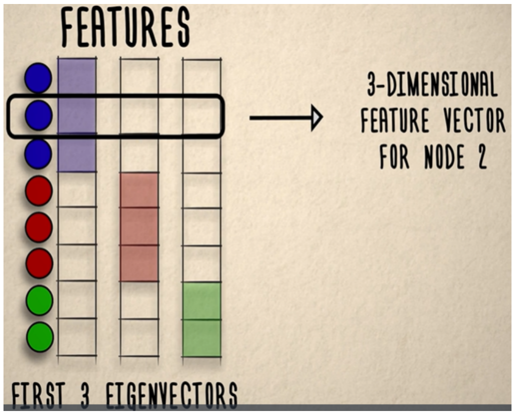
Let’s start with the graph when there is no link between the communities, and build the adjacency matrix for it.

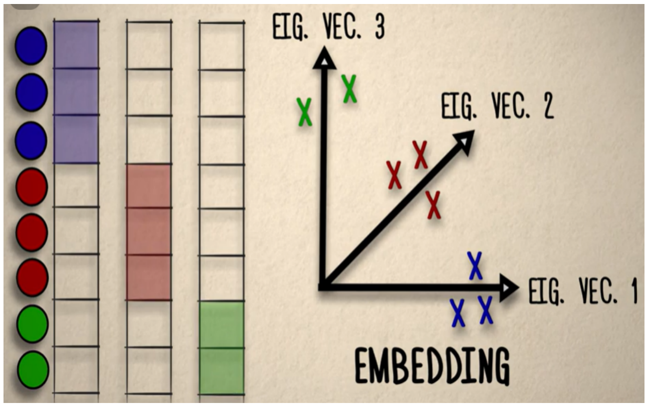


	🔵	🔵	🔵	🔴	🔴	🔴	🟢	🟢
🔵	1	1	1	0	0	0	0	0
🔵	1	1	1	0	0	0	0	0
🔵	1	1	1	0	0	0	0	0
🔴	0	0	0	1	1	1	0	0
🔴	0	0	0	1	1	1	0	0
🔴	0	0	0	1	1	1	0	0
🟢	0	0	0	0	0	0	1	1
🟢	0	0	0	0	0	0	1	1

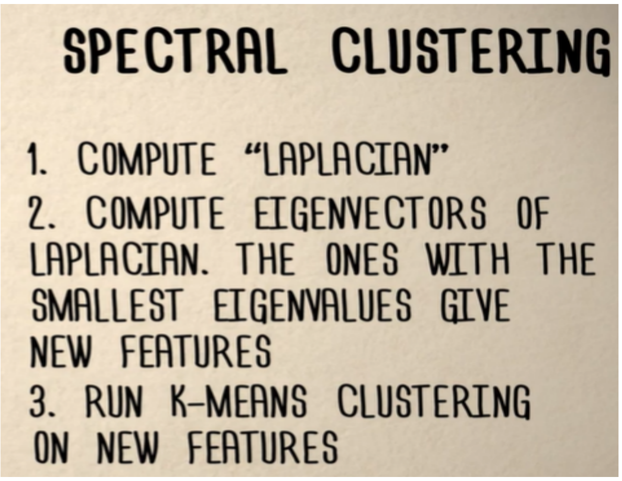
We can clearly observe a block structure in the Image given above, but in real life, this structure is somewhat arbitrary and we don't know how to make this block pattern. How do we solve this problem for a real graph where the structure is somewhat arbitrary?

Let’s look at the eigenvectors, The bottom 3 vectors, exactly indicated the block structure and each indicates the nodes in the long community. The remaining eigenvectors are less important, and this method using eigenvectors to find structure works even if the nodes are scrambled. For each node, we create a feature vector, as we saw in the previous module. We take one entry from each eigenvector to create a feature vector which is also called embeddings. These features directly indicate which community a particular node belongs to. In reality, communities are not separated so clearly. There are edges between communities, and there are edges within communities that are missing. But there are not too many of these extra edges, the eigenvector does not change much and they can still do a decent performance.





After creating these features, we can use them as embedding as we used in the previous module. Here we have 3 features for each node, hence 3D, and we can also plot these embeddings in 3D space and can see clusters emerging.
The following approach is called **spectral clustering.**



SPECTRAL CLUSTERING

1. COMPUTE "LAPLACIAN"
2. COMPUTE EIGENVECTORS OF LAPLACIAN. THE ONES WITH THE SMALLEST EIGENVALUES GIVE NEW FEATURES
3. RUN K-MEANS CLUSTERING ON NEW FEATURES

The steps for spectral clustering are simple as we did previously:

1. Compute Laplacian
2. Compute Eigenvectors of Laplacian and create embeddings

3. Run K-Means clustering on these embeddings.

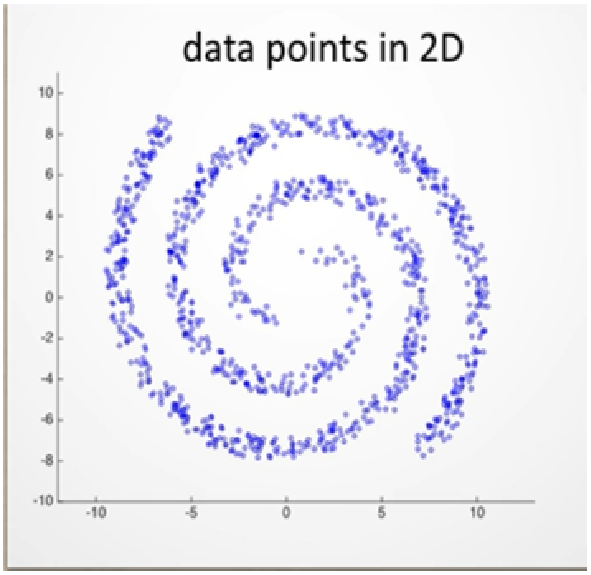
These steps worked well and because of the magic of eigenvectors, they can separate the community very well.

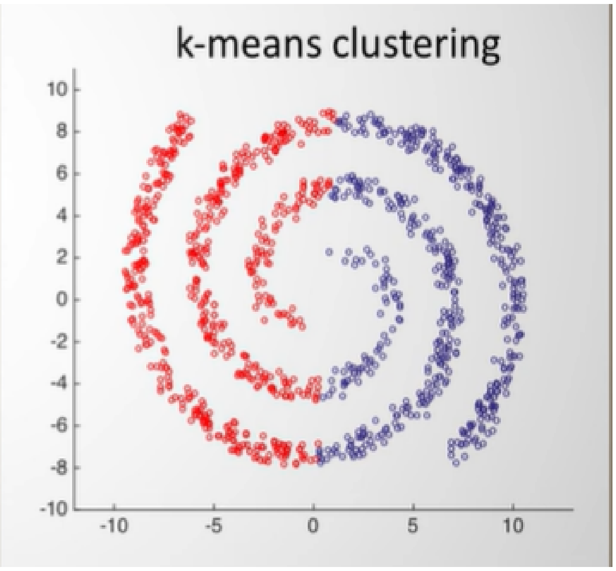
Now, let's recall the first criteria. Where we assumed to have clusters of decent size if we have a small cut value between them. They may sound different from eigenvectors here, but there is a closed connection.

We can give labels to the nodes to indicate the optimal clusters for the criterion. For example, a special positive number for cluster 1 and a negative number for cluster 2. The eigenvector approximates those levels.
In fact, spectral clustering is used in places where the data is not in the form of a network.

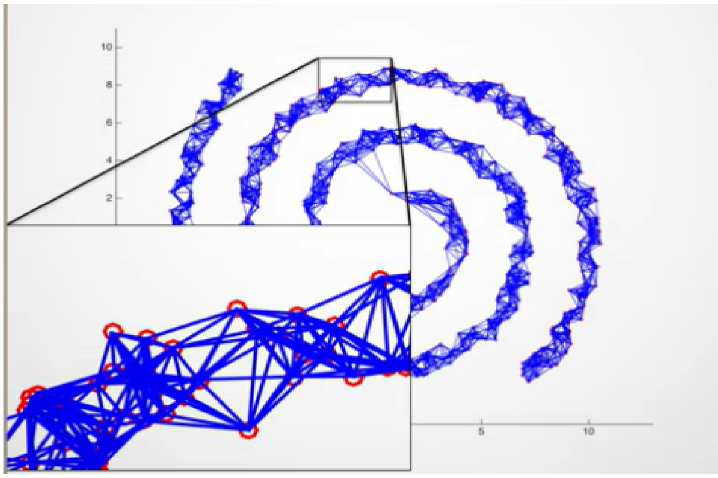
For example, in K-Means clustering, if we plug the data points the cluster we found are round. Now, let's see what happens when the data is not round.

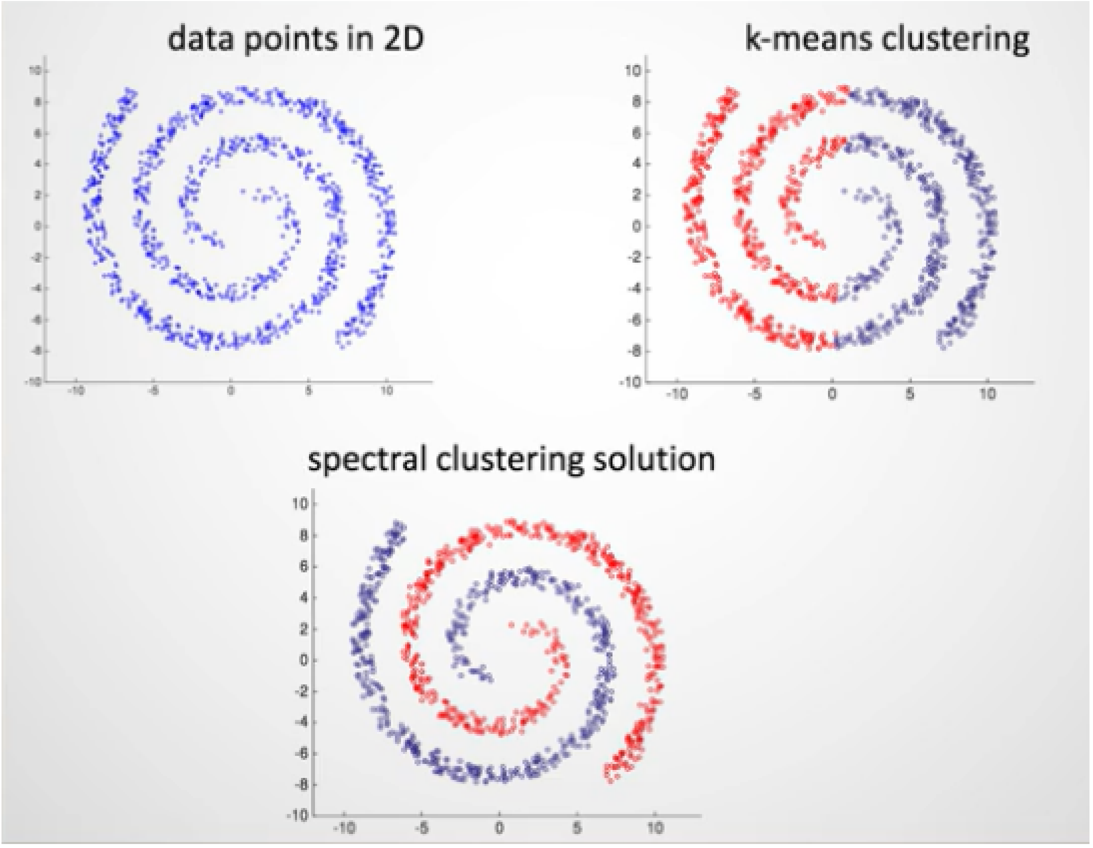
Consider the images: The data here is curved and intuitively we want to capture this curvy structure. If we run K -Means clustering in this data the result is the image on the right.





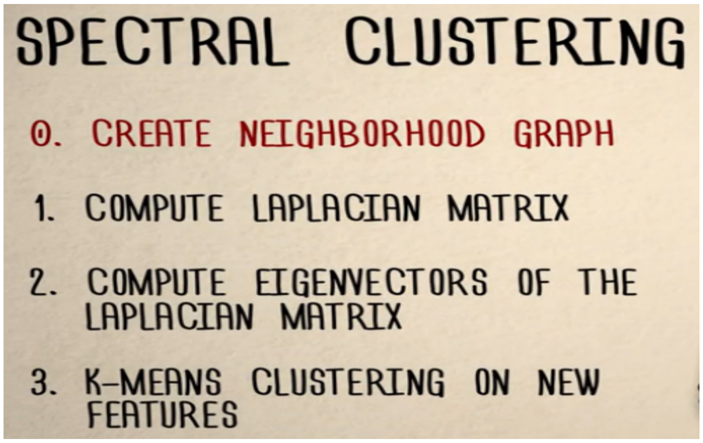
The clusters look odd and they disregard the structure of the data. The Spectral clustering can help in this. We can translate the data into the form of a graph and build a graph from this data. For each data point, we create a node and connect it with the nearest point. We get clusters of the points that are densely connected in the graph. These from the below, unlike K-Means clusters, do follow the shape. They look very different and more meaningful.





So, if we need to do spectral analysis on some other data which is not graphical, we have to just make it in a graphical representation.

So our final step will be:



SPECTRAL CLUSTERING

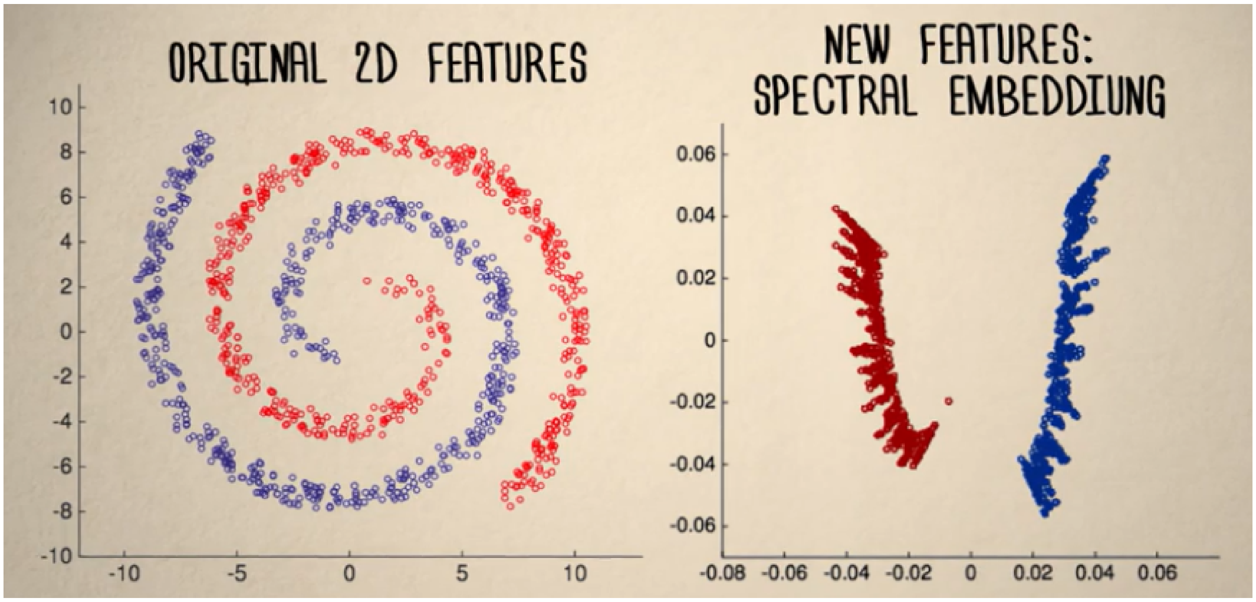
0. CREATE NEIGHBORHOOD GRAPH
1. COMPUTE LAPLACIAN MATRIX
2. COMPUTE EIGENVECTORS OF THE LAPLACIAN MATRIX
3. K-MEANS CLUSTERING ON NEW FEATURES

Cautions before making this graphical representation:

1. The edge is weighted and is determined by the similarity of the points. A common similarity function is the gaussian similarity function, which decays exponentially with the distance.
2. We connect a point to a fixed number of closest points. Typically it should not be too many, and the resulting graph has very few disconnected components.

Difference between K-Means and spectral clustering?

1. The distance measure in K-Means does not capture the complicated shape of the data.
2. Spectral clustering also uses K-Means clustering, but with an important difference. We created new feature vectors and use K-means with those new features. Then the clusters follow the inherent shape of the data.


ORIGINAL 2D FEATURES
10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-5
0
5
10
NEW FEATURES:
SPECTRAL EMBEDDIUNG
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06