Magic of Eigenvectors - Spectral Clustering

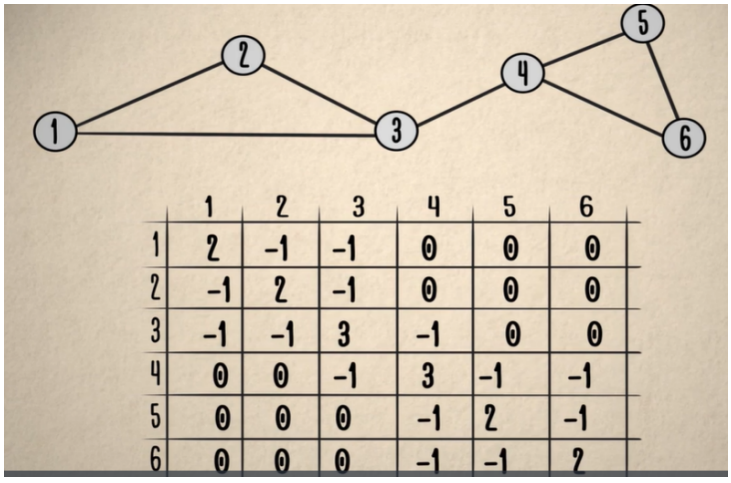
In the previous module, we learned about how clustering in the graph can be helpful and its different criteria.

Next question is, how to compute the similarity or the criteria?
The first important thing is the global connectivity structure of the graph, and for that, we will again use eigenvectors to compute it as they help us encode information about the structure of the graph.

These methods are called **spectral methods or spectral clustering.** The spectrum of a matrix is the set of its eigenvalues, this is because as we know, eigenvalues help in finding the inherent directions in the dataset.

We saw in PCA, that we calculate eigenvectors and eigenvalues from the Covariance matrix and the eigenvectors help us to find the major direction of covariance in the data, but what matrix we will use here?

The matrix is called the **Laplacian of a graph.** Let's see how we construct it, see the image attached below.



	1	2	3	4	5	6
1	2	-1	-1	0	0	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	0	0	0	-1	2	-1
6	0	0	0	-1	-1	2

If there is an edge between 2 nodes we represent it by -1 else 0. On the diagonal, we have, the degree of the node.

The Degree of a node is the number of edges it has. For example, the degree of node 1 is 2 as it is connected with node 2 and node 3 and has 2 edges.

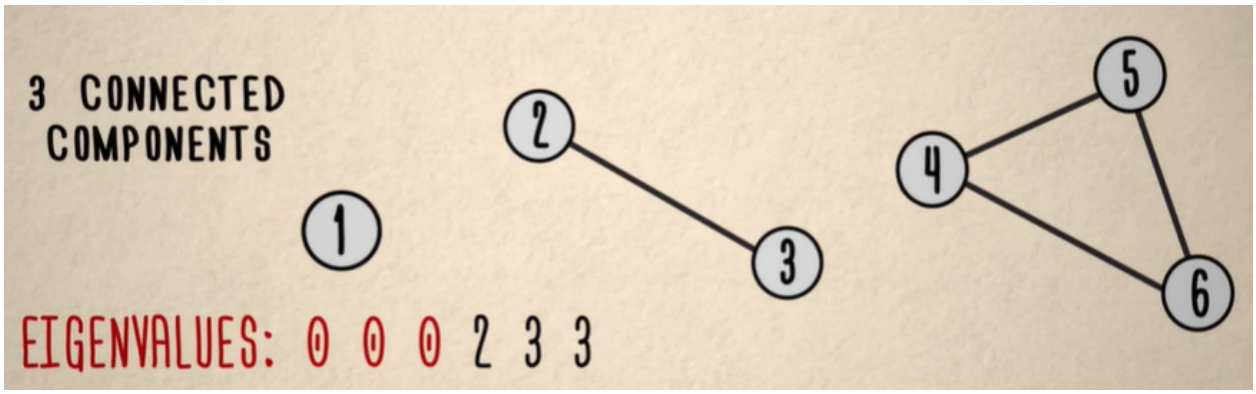
Let's construct eigenvectors for this Laplacian matrix.
Note that each eigenvector has a corresponding eigenvalue and we have sorted them in ascending order of their eigenvalues.

We are interested in the **smallest and largest eigenvalues**. In this case, the smallest eigenvalue is always 0.

EIGENVALUES	0	0.4	3	3	3	4.6	
CORRESPONDING EIGENVECTORS	0.41	-.46	-.05	0.71	0.28	0.18	1
	0.41	-.46	-.05	-.71	0.28	0.18	2
	0.41	-.26	0.10	0.00	-.57	-.66	3
	0.41	0.26	0.10	0.00	-.57	0.66	4
	0.41	0.46	-.75	0.00	0.16	-.18	5
	0.41	0.46	0.65	0.00	0.41	-.18	6

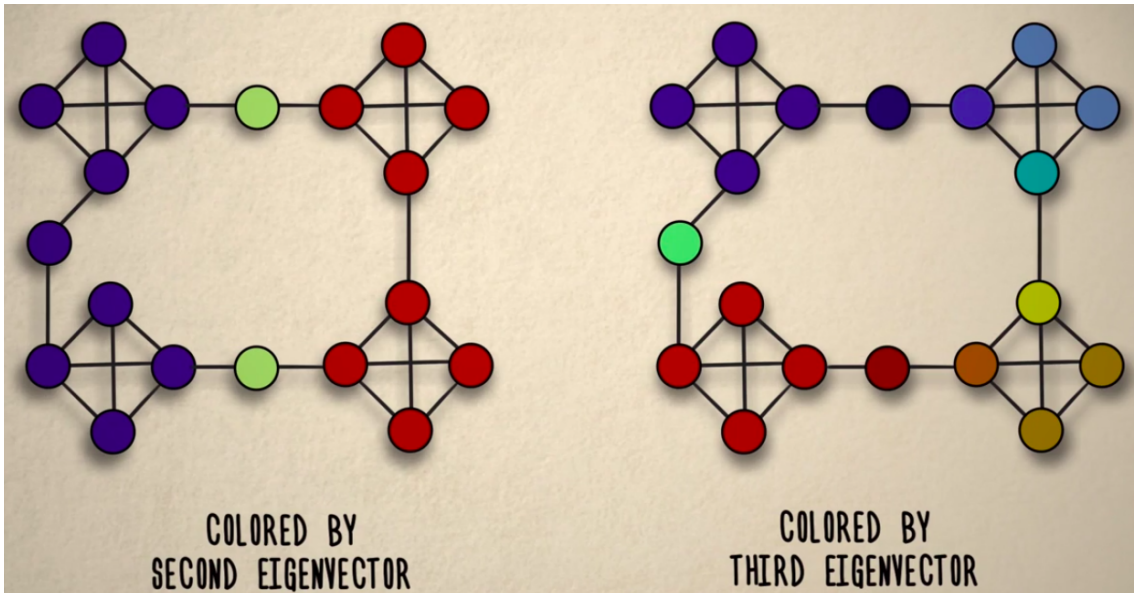
We can also define the first structural relationship using eigenvectors:
The number of 0 eigenvalues represents the number of connected components in the graph.

For example, In the graph given below, there are 3 connected components.

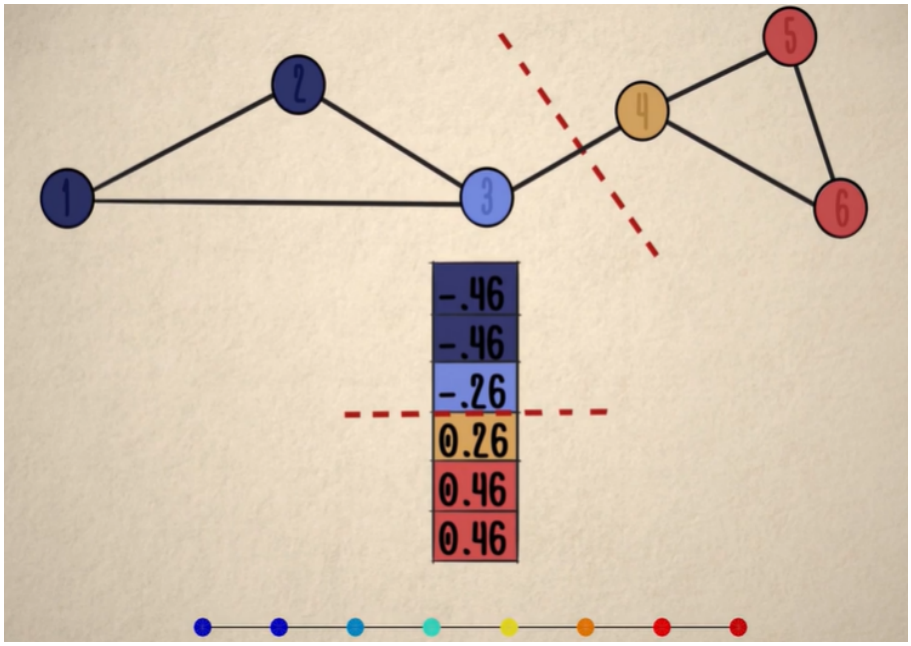


Now, let's find other relationships using eigenvectors. Let’s consider the second eigenvector.

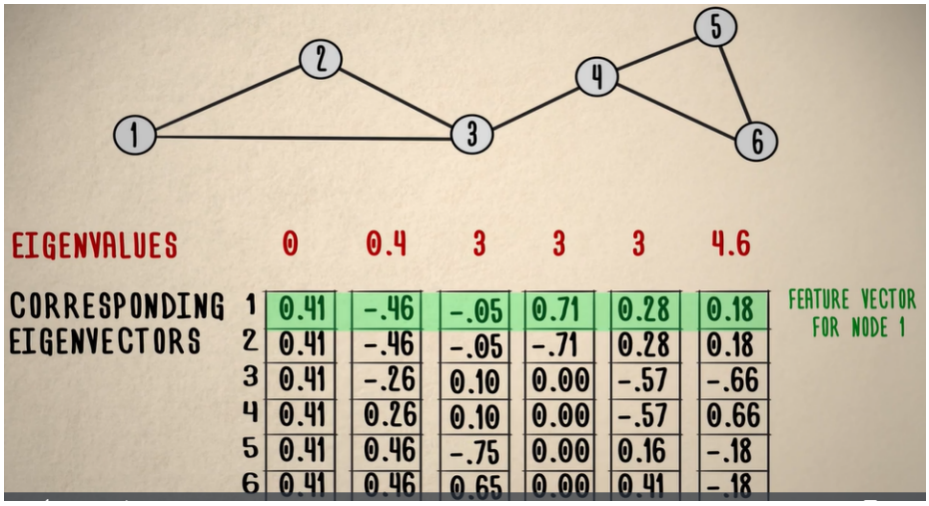
Each eigenvector has 6 entries, 1 for each node. We can color code the numbers based on the values in the eigenvector. Going from blue to red for lower to higher values respectively.



If you observe, the value in the eigenvector reflects the structure in the graph.
Closer nodes have similar colors, and we can even find the partition, which can be used to separate groups. Look at the red dotted line.

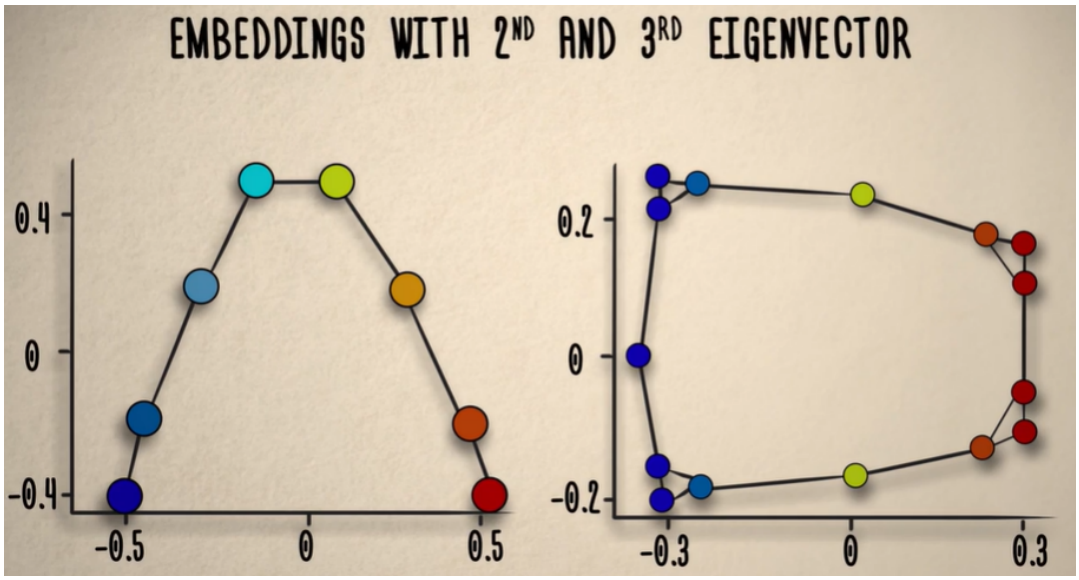


Remember, our nodes do not have any feature vector associated with them. We did all this using the Laplacian matrix only. But using eigenvectors, we can assign features to each node. Now, we each assign 1 feature value from each eigenvector and we have a feature vector for each node. These new features are called **embedding.** PCA also gave us an embedding, for PCA we used an eigenvector with the largest eigenvalues, here the smallest ones are important.

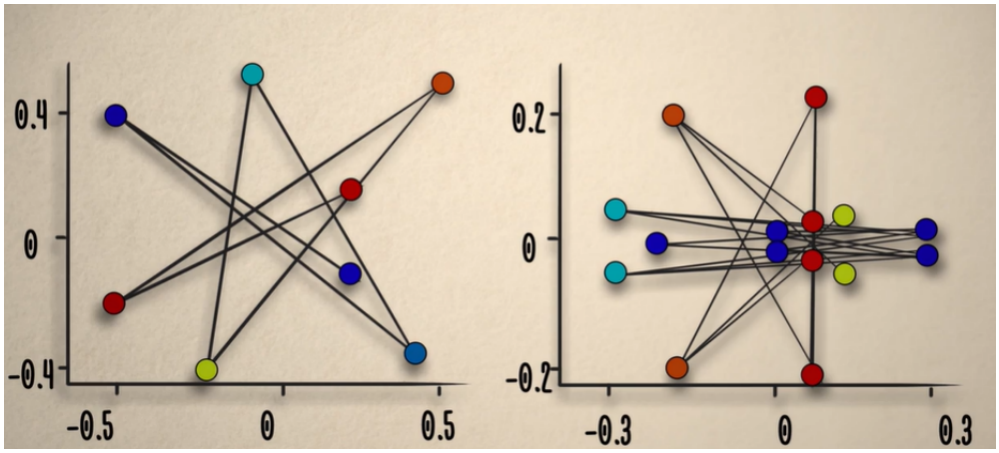


Just like we plotted features from PCA, let's take the second and third smallest eigenvectors and use them as coordinates for the nodes.

Here it looks like, the nodes that are connected are positioned close by, and our embedding here wants to **minimize the stretch of the edges, and clusters in the graph become clusters in 2D.**

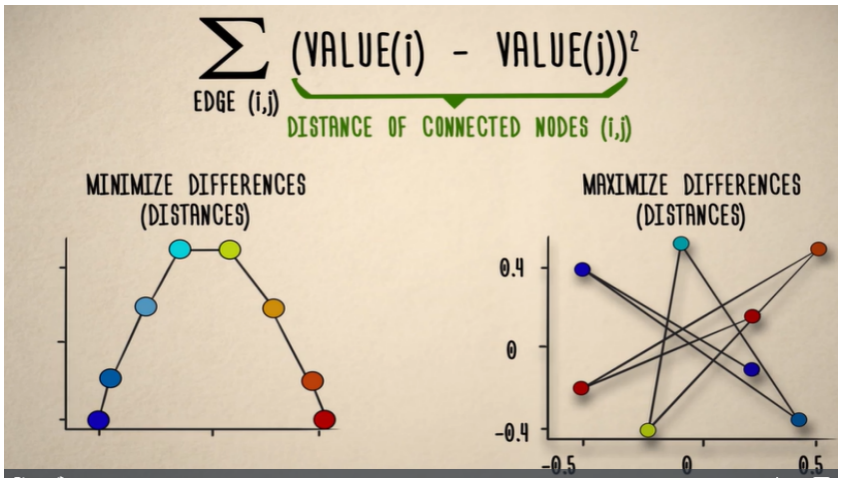


This is what happens when we use eigenvectors with lower eigenvalues, and what happens when we use vectors with the largest eigenvalues.
Here is what happens when we use vectors with the largest eigenvalues. They try to **maximize the stretch of edges.** The nodes that are connected are placed far away on the plane.



In fact, this is the same graph, but looks so different, is there any explanation for this? **The eigenvectors minimize or maximize the different scores.** Remember, each node gets a value when we create features for the nodes. For each edge, we take the difference of the adjacent node values and square that, we do this for all edges and sum it up. The eigenvectors with small eigenvalues minimize these differences and give similar values to connected nodes and this difference can be seen as the stretch of the nodes.

The eigenvector with large eigenvalues maximizes this difference. You can see this in the eigenvectors we created.



Overall, we found out that eigenvectors and values have a lot of information associated with them. They can help us in determining the graph structure which thus helps in clustering.