Diffusion Maps

The attached file contains the 2-dimensional coordinates of $n = 500$ points. Denote the points corresponding to the first 250 rows by *inner* points, and the points corresponding to the last 250 rows by *outer* points.

(a) Plot the point cloud, and mark the two different kind of plots with two different symbols. Here is how you can do this in R `http://www.endmemo.com/program/R/pchsymbols.php`

(b) Perform PCA on this data set, and show a 2-D plot of your data points (using the 2 principal component scores). Again plot the inner and outer points with distinct symbols.

(c) Implement the diffusion maps algorithm we covered in class. (Note that a simple way to compute the matrix of all pairwise distances in one shot is

$$DIST = as.matrix(dist(A,\ method = "euclidean",\ diag = FALSE,\ upper = TRUE,\ p = 2));$$

Experiment with different values of the parameter ϵ in diffusion maps, and include the embeddings you obtained via the first two non-trivial eigenvectors, for two different values of $\epsilon = \{0.75, 1\}$, and plot the inner and outer points with distinct symbols. Which of these two representations is easier to cluster/separate? Compare the results with those from (b).

(d) You could make the problem harder, by artificially increasing the dimensionality of the data, for example, by appending extra columns to the initial 2D coordinates:

$$A = cbind(A,\ z = s*\ runif(n),\ t = s*\ runif(n)\ )$$

where s is a scalar. For large enough value of s (for example $s = 5$), it would be hard to linearly separate the inner cluster of red points from the blue points.

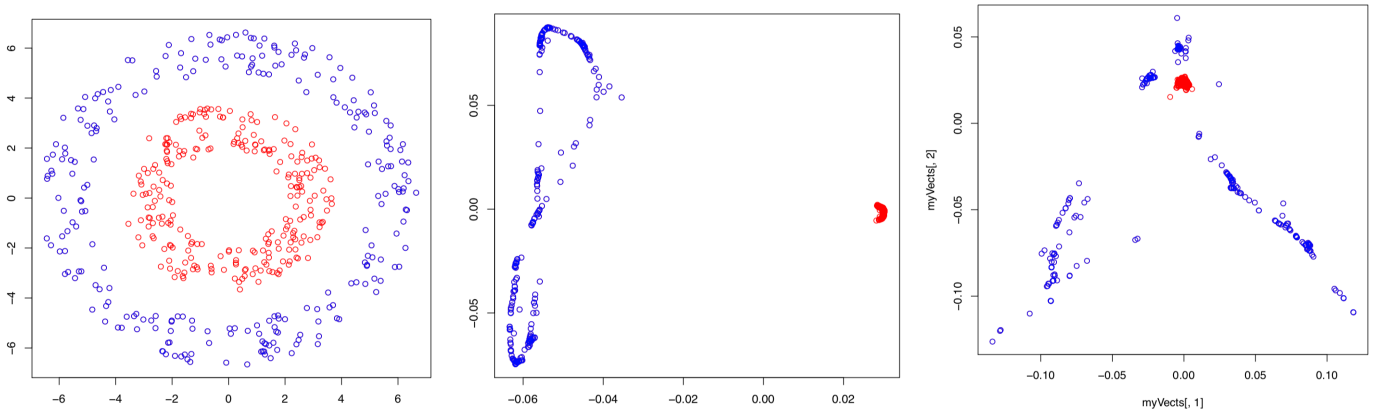

Figure 1: Left: Ground truth 2D embedding. Middle: 2D diffusion map embedding for $s = 0$. Right: 2D diffusion map embedding for $s = 5$