

Homework 5

November 19, 2006

1 Manifold learning

Unfortunately, there is not any good R packages for manifold learning algorithms. So in this homework, we will have to write a few fragments of codes to experiment the algorithm(s) in for simplest cases.

For most people who have access to Matlab, there are plenty of Matlab demos. See for example <http://www.math.umn.edu/wittman/mani/>

1.1 Review

In the class, I showed an example of Swiss roll and suggested that PCA cannot reveal the object's intrinsic 2-D structure. In this exercise, we will use a simpler toy example to demonstrate it algorithmically.

We will use Isomap as a nonlinear dimensionality reduction algorithm for this exercise. First, recall the 3 steps of the algorithm:

- Compute nearest neighbor graphs: each data point corresponds to a vertex and edges between vertices indicate nearest neighbors. The edges are weighted by the Euclidean distances between the data points.
- Compute distance matrix: for vertices that do not have edges between them, compute the shortest-path between them and use the distances summed over path as a proxy to their true distance.
- Diagonalization (classic MDS): convert the distance matrix to Gram matrix (ie, inner product matrix) and diagonalize Gram matrix and compute the embedding
Step 3 needs to convert distance matrix to Gram matrix. Use following formula (presented as a lemma in the lecture):

Let d_{ij} be the distance between vertex i and j . Construct a matrix S whose element $S_{ij} = d_{ij}^2$. Define a column vector $u = [1 \ 1 \ 1 \ \dots \ 1 \ 1]'$, which has N ones and N is the number of data points. Then, the Gram matrix is defined as

$$G = - \left(I - \frac{1}{N} uu' \right) S \left(I - \frac{1}{N} uu' \right)$$

where I is the identity matrix. G_{ij} gives the inner product between i -th data point and j -th data points.

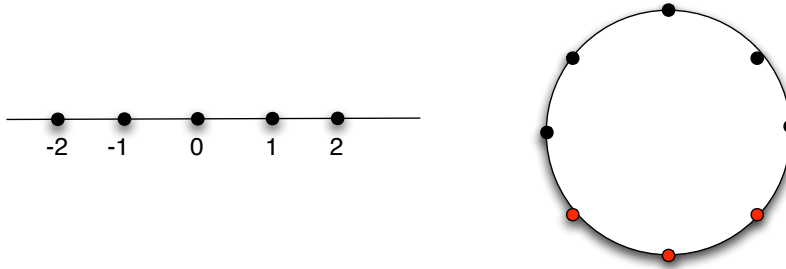


Figure 1: Linear and nonlinear manifolds

In the following, you are required to write a few lines of R codes to test the algorithm out on simple cases.

1.2 Linear

Consider a set of data points sampled in regular interval from a straight line. The left plot of Fig. 1 shows such 5 data points with their coordinates on the real axis. The interval is size 1. (You can use more than 5 data points, obviously.)

Specifically, in step 1 of the Isomap algorithm, choose the number of nearest neighbors to be 2. Namely, nearest neighbors of a data point are the ones immediately to the left and to the right.

Step 2 requires using dynamic programming on general graph. For a simple case like this, you can compute them explicitly.

Q1 Turn in your codes used for step 2 and step 3. Plot out the embeddings by both PCA and Isomap. Are they the same as original? How many significant eigenvalues (ie, dimensionality) returned by each method? Plot out these eigenvalues.

1.3 Nonlinear

Consider a set of data points *equally sampled* from a half-circle, as the black points in the right plot of Fig. 1. The 2D coordinates of these data points are $[\cos(\theta_i), \sin(\theta_i)]$ where θ_i is an angle between 0 and π . To get equally spaced sampling, the θ_i for all data points should be equally spaced in the interval, for instance, $0, \pi/4, \pi/2, 3\pi/4, \pi$ for 5 data points.

Q2 Apply PCA and Isomap to this type of data points (choose the number of nearest neighbors to be 2). Plot out the embeddings by the PCA and Isomap. How many significant eigenvalues (ie, dimensionality) returned by each method? Plot out these eigenvalues.

Q3 Which of the embeddings is the correct one, in your opinion? Why so? Particularly, the half circle can be parameterized by a one dimensional variable such as arc length (thus embedded as a one dimensional object). Does the Isomap algorithm find the arc length for the embedding?

Q4 Now make things slightly complicated and interesting. Consider in this case, data points equally sampled from a *full* circle (as the red and black points in Fig. 1). Run Isomap algorithm on it. Check your embeddings. How many significant eigenvalues now? What is the embedding? Is the embedding still a line, or the embedding a circle, or something else? Print out your embedding, turn in your codes, and plot of eigenvalues.

1.4 Project ideas

For those who are interested in implementing one or two algorithms in R, send me an email feisha@cs.berkeley.edu

Those algorithms are not very complicated to implement (at least in Matlab). Therefore, it would be good for you to implement if your primary working environment is not Matlab.