HW for 2018-07-04

(due: 2018-07-07)

For our last assignment, we will put together several ideas from the class in the context of coordinate embedding. Throughout this assignment, let $X \in [0,1]^2$ be a set of points chosen uniformly at random and let

$$f(x,y) = \begin{bmatrix} x + \sin(x) \\ y + \cos(y) \\ xy \end{bmatrix}.$$

That is, the data points live in a three dimensional space, but belong to a "hidden" two-dimensional manifold. Our goal will be to try to recover a coordinate system for this manifold.

1: From points to graphs Let $x^1, \ldots, x^m$ be random points in $[0,1]^2$, and let $y^i = f(x^i) \in \mathbb{R}^3$ be their images. Write a routine to construct the ϵ*-NN distance* matrix $W(\epsilon)$ between the points y^i; that is,

$$w_{ij} = \begin{cases} \|y_i - y_j\|, & \text{if } \|y_i - y_j\| \leq \epsilon \\ 0, & \text{otherwise.} \end{cases}$$

Write another routine to compute the two largest eigenvalues and corresponding eigenvectors of $L = D - W$ where D is the diagonal matrix with $d_{ii} = \sum_j w_{ij}$. The smallest eigenvalue is zero, with an associated eigenvector of all ones; why? Take the next two eigenvalues in increasing order of magnitude to be λ_2 and λ_3, and let

$$\hat{X}(\epsilon) = \begin{bmatrix} v_2/\sqrt{\lambda_2} & v_3/\sqrt{\lambda_3} \end{bmatrix}$$

This coordinate embedding is sometimes called a *Laplacian eigenmap*. Write a code to compute this.

2: Nearest neighbor Starting from the matrix W, we can use the Floyd-Warshall algorithm to compute the matrix M whose entries are distances[1] between nodes via paths in W (code is included in the repository). Let $M^{(2)}$

[1] We have already used D to refer to the diagonal matrix of weighted degrees in the previous problem, so D for distance is out. We use M for metric instead.

be the matrix of elementwise square distances. Let us scale and center this matrix, i.e. compute

$$B = -\frac{1}{2} J M^{(2)} J^T, J = I - \frac{ee^T}{n}$$

Now use the coordinate system

$$\hat{X} = \begin{bmatrix} v_1 \sqrt{\lambda_1} & v_2 \sqrt{\lambda_2} \end{bmatrix}$$

where v_1 and v_2 are the eigenvectors of B associated with the largest eigenvalue. This is the *Isomap* embedding. Write a code to compute this.

3: Kernelized comparisons In the previous two exercises, we have two coordinate systems for the underlying manifold: the original system where the x_i live, and a new system where the $\hat{x}_i$ live. For the x coordinate system, let the *spline energy* for f be

$$\sum_{j=1}^{3} |s_j|_{\mathcal{H}}^2$$

where each s_j is a thin plate spline (with linear tail) for the mapping from the x points in $\mathbb{R}^2$ to the $f(x)$ points in $\mathbb{R}^3$. Note that if f is linear with respect to x, this embedding will be zero – explain why. Compare the spline energy for the original coordinate system to the analogous energy for the Laplace eigenmap and isomap embeddings (as a function of ϵ). What coordinate system results in the smoothest representation for f?