Data Visualization
STAT 890 / 442, CM 462

Assignment 3
Fall 2006
Department of Statistics and Actuarial Science
University of Waterloo

Due: Monday November 20, at the start of class
Instructor: Ali Ghodsi
MS 6081G x37316, aghodsib@uwaterloo.ca

Policy on Lateness: Slightly late assignments (up to 24 hs after due date) are accepted with 10% penalty. No assignment are accepted after 24 hs after the due date.

1. In Table 1 we are given the road distances between towns. The aim is to construct a geographical map of these towns based on this information. We expect that Landmark Multidimensional Scaling (LMDS) method will produce a configuration which is close to the true map of these towns.

   Given n data points and m landmarks, recall that in LMDS we have matrices D and K:

   $$D = \begin{pmatrix} E & F \\ F^T & G \end{pmatrix}$$

   Where E is a m by m block and F is a m by $n-m$ block.

   $$K = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

   Where A is a m by m block and B is a m by $n-m$ block.

   a) Write a function which takes E as the input and returns A as the output. (Submit your function for this part)

   b) Write a function which takes F and E as the inputs and returns B as the output. (Submit your function for this part)

   c) Write a function which takes E , F and d as the inputs and computes the d-dimensional representation by LMDS. (Submit your function for this part)

   d) Use your program and produce the two-dimensional representation of the data in Table 1. (Submit a single plot for this part.)

	1	2	3
1			
2	244		
3	218	350	
4	284	77	369
5	197	167	347
6	312	444	94
7	215	221	150
8	469	583	251
9	166	242	116
10	212	53	298
11	253	325	57
12	270	168	284

Table 1: Road Distances Between Towns

2. LLE algorithm consists of three steps:

   (a) Identify the neighbours of each data point x_i. This can be done by finding the k nearest neighbours, or by choosing all points within some fixed radius, ϵ.

   (b) Compute the weights that best linearly reconstruct x_i from its neighbours.

   (c) Find the low-dimensional embedding vector y_i which is best reconstructed by the weights determined in the previous step.

   After finding the nearest neighbours in the first step, the second step must compute a local geometry for each locally linear patch. This geometry is characterized by linear coefficients that reconstruct each data point from its neighbours. The following cost function measures the reconstruction error.

$$\min_{w} \sum_{i=1}^{n} ||\mathbf{x}_i - \sum_{j=1}^{k} w_{ij} \mathbf{x}_{N_i(j)}||^2$$

   where the neighbours of x_i are represented by $\mathbf{x}_{N_i(j)}$. i.e. $N_i(j)$ is the index of the jth neighbour of the ith point.

   w_{ij} are computed to minimize this cost function subject to the constraint that the rows of the weight matrix sum to one, $\sum_{j=1}^{k} w_{ij} = 1$.

   The constrained weights that minimize these reconstruction errors obey several important symmetries: for any particular data point, they are invariant to rotations, rescalings, and translations of that data point and its neighbors. Use simple numerical examples to show that the constrained weights are

a) invariant to rotations

b) invariant to rescalings

c) invariant to translations

d) In the third step of the algorithm, each high dimensional input x_i is mapped to a low dimensional output y_i representing global internal coordinates on the manifold. This is done by choosing the d-dimensional coordinates of each output y_i to minimize the embedding cost function:

$$\min_Y \sum_{i=1}^{n} ||y_i - \sum_{j=1}^{k} w_{ij} y_j||^2$$

subject to the constraints:

$$\sum_{i=1}^{n} y_i = 0$$

$$\frac{1}{n} Y Y^T = I_{dXd}$$

Where $Y = [y_1 \dots y_n]$. This can be done by finding the bottom $d+1$ eigenvectors of the matrix, $M = (I - W)(I - W)^T$. The bottom eigenvector of this matrix, which we discard, is the unit vector with all equal components. Clearly and briefly explain why discarding this eigenvector enforces the constraint that the outputs have zero mean(i.e. $\sum_{i=1}^{n} y_i = 0$).

Only for GradStudents
Due: Friday December 1, at the start of class

3. (MDS) Let $\Lambda = diag(\lambda_1, \ldots, \lambda_p)$, where $\lambda_1 \geq \cdots \geq \lambda_p$ are real numbers, and let $\tilde{\Lambda} = diag(\tilde{\lambda_1}, \ldots, \tilde{\lambda_p})$, where $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_p \geq 0$ are non-negative numbers. Show that minimizing

$$Tr(\Lambda - G\tilde{\Lambda}G^T)^2$$

over orthogonal matrices G is equivalent to maximizing

$$Tr(\Lambda G\tilde{\Lambda}G^T) = \sum_{i,j=1}^{p} \lambda_i \tilde{\lambda}_j g_{ij}^2 = \sum_{i=1}^{p} \lambda_i h_i = \phi(h)$$

where

$$h_i = \sum_{j=1}^{p} \tilde{\lambda}_j g_{ij}^2 \geq 0 \quad and \quad \sum_{i=1}^{p} h_i = \sum_{j=1}^{p} \tilde{\lambda}_j$$

Show that $\phi(h)$ is maximized over such vectors h when $h_i = \tilde{\lambda}_i$ for $i = 1, \ldots, p$; that is, when $G = I$.