

## Homework 6. Manifold Learning

*Instructor: Yuan Yao**Due: 1 weeks later*

The problem below marked by \* is optional with bonus credits.

1. *Order the faces:* The following dataset contains 33 faces of the same person ( $Y \in \mathbb{R}^{112 \times 92 \times 33}$ ) in different angles,

<https://yao-lab.github.io/data/face.mat>

You may create a data matrix  $X \in \mathbb{R}^{n \times p}$  where  $n = 33, p = 112 \times 92 = 10304$  (e.g. `X=reshape(Y,[10304,33])'`; in matlab).

- (a) Explore the MDS-embedding of the 33 faces on top two eigenvectors: order the faces according to the top 1st eigenvector and visualize your results with figures.
  - (b) Explore the ISOMAP-embedding of the 33 faces on the  $k = 5$  nearest neighbor graph and compare it against the MDS results. Note: you may try Tenenbaum's Matlab code <https://yao-lab.github.io/data/isomapII.m>
  - (c) Explore the LLE-embedding of the 33 faces on the  $k = 5$  nearest neighbor graph and compare it against ISOMAP. Note: you may try the following Matlab code <https://yao-lab.github.io/data/lle.m>
2. *Manifold Learning:* The following codes by Todd Wittman contain major manifold learning algorithms talked on class.

<http://math.stanford.edu/~yuany/course/data/mani.m>

Precisely, eight algorithms are implemented in the codes: MDS, PCA, ISOMAP, LLE, Hessian Eigenmap, Laplacian Eigenmap, Diffusion Map, and LTSA. The following nine examples are given to compare these methods,

- (a) Swiss roll;
- (b) Swiss hole;
- (c) Corner Planes;
- (d) Punctured Sphere;
- (e) Twin Peaks;
- (f) 3D Clusters;
- (g) Toroidal Helix;
- (h) Gaussian;
- (i) Occluded Disks.

Run the codes for each of the nine examples, and analyze the phenomena you observed.

\*Moreover if possible, play with t-SNE using scikit-learn manifold package:

<http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>

or any other implementations collected at

<http://lvdmaaten.github.io/tsne/>

3. *Nyström method*: In class, we have shown that every manifold learning algorithm can be regarded as Kernel PCA on graphs: (1) given  $N$  data points, define a neighborhood graph with  $N$  nodes for data points; (2) construct a positive semidefinite kernel  $K$ ; (3) pursue spectral decomposition of  $K$  to find the embedding (using top or bottom eigenvectors). However, this approach might suffer from the expensive computational cost in spectral decomposition of  $K$  if  $N$  is large and  $K$  is non-sparse, e.g. ISOMAP and MDS.

To overcome this hurdle, Nyström method leads us to a scalable approach to compute eigenvectors of low rank matrices. Suppose that an  $N$ -by- $N$  positive semidefinite matrix

$K \succeq 0$  admits the following block partition

$$K = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}. \quad (1)$$

where  $A$  is an  $n$ -by- $n$  block. Assume that  $A$  has the spectral decomposition  $A = U\Lambda U^T$ ,  $\Lambda = \text{diag}(\lambda_i)$  ( $\lambda_1 \geq \lambda_2 \geq \dots \lambda_k > \lambda_{k+1} = \dots = 0$ ) and  $U = [u_1, \dots, u_n]$  satisfies  $U^T U = I$ .

- (a) Assume that  $K = XX^T$  for some  $X = [X_1; X_2] \in \mathbb{R}^{N \times k}$  with the block  $X_1 \in \mathbb{R}^{n \times k}$ . Show that  $X_1$  and  $X_2$  can be decided by:

$$X_1 = U_k \Lambda_k^{1/2}, \quad (2)$$

$$X_2 = B^T U_k \Lambda_k^{-1/2}, \quad (3)$$

where  $U_k = [u_1, \dots, u_k]$  consists of those  $k$  columns of  $U$  corresponding to top  $k$  eigenvalues  $\lambda_i$  ( $i = 1, \dots, k$ ).

- (b) Show that for general  $K \succeq 0$ , one can construct an approximation from (2) and (3),

$$\hat{K} = \begin{bmatrix} A & B \\ B^T & \hat{C} \end{bmatrix}. \quad (4)$$

where  $A = X_1 X_1^T$ ,  $B = X_1 X_2^T$ , and  $\hat{C} = X_2 X_2^T = B^T A^\dagger B$ ,  $A^\dagger$  denoting the Moore-Penrose (pseudo-) inverse of  $A$ . Therefore  $\|\hat{K} - K\|_F = \|C - B^T A^\dagger B\|_F$ . Here the matrix  $C - B^T A^\dagger B =: K/A$  is called the (generalized) *Schur Complement* of  $A$  in  $K$ .

- (c) Explore Nyström method on the Swiss-Roll dataset ([http://yao-lab.github.io/data/swiss\\_roll\\_data.mat](http://yao-lab.github.io/data/swiss_roll_data.mat) contains 3D-data  $X$ ; <http://yao-lab.github.io/data/swissroll.m> is the matlab code) with ISOMAP. To construct the block  $A$ , you may choose either of the following:

$n$  random data points;

\* $n$  landmarks as minimax  $k$ -centers (<https://yao-lab.github.io/data/kcenter.m>);

Some references can be found at:

[dVT04] Vin de Silva and J. B. Tenenbaum, “Sparse multidimensional scaling using landmark points”, 2004, downloadable at <http://pages.pomona.edu/~vds04747/public/papers/landmarks.pdf>;

[P05] John C. Platt, “FastMap, MetricMap, and Landmark MDS are all Nyström Algorithms”, 2005, downloadable at <http://research.microsoft.com/en-us/um/people/jplatt/nystrom2.pdf>.

(d) \*Assume that  $A$  is invertible, show that

$$\det(K) = \det(A) \cdot \det(K/A),$$

(e) \*Assume that  $A$  is invertible, show that

$$\text{rank}(K) = \text{rank}(A) + \text{rank}(K/A).$$

(f) \*Can you extend the identities in (c) and (d) to the case of noninvertible  $A$ ? A good reference can be found at,

[Q81] Diane V. Quellet, “Schur Complements and Statistics”, *Linear Algebra and Its Applications*, 36:187-295, 1981. <http://www.sciencedirect.com/science/article/pii/0024379581902329>