Boston University
Department of Electrical and Computer Engineering

ENG EC 500 B1 (Ishwar) Introduction to Learning from Data

Matlab Exercise 5

Issued: Wed 2 Dec 2015 **Due:** 9am Wed 9 Dec 2015 PHO440 box + Blackboard
Required reading: Your notes from lectures.

- This homework assignment requires some programming background in MATLAB. Please refer to the following link for an introduction (or review) of MATLAB:
`http://www.math.ucsd.edu/~bdriver/21d-s99/matlab-primer.html`
- The maximum expected run time of code for any problem in this assignment is 10-15 minutes.
- You will be making two submissions: (1) A paper submission in the box outside PHO440. (2) An electronic submission of all your matlab code to blackboard (in a single zipped file appropriately named as described below).
- **Paper submission:** This must include all plots, figures, tables, numerical values, derivations, explanations (analysis of results and comments), and also printouts of all the matlab .m files that you either created anew or modified. Submit color printouts of figures and plots whenever appropriate. Color printers are available in PHO307 and PHO305. Be sure to annotate figures, plots, and tables appropriately: give them suitable *titles* to describe the content, label the *axes*, indicate *units* for each axis, and use a *legend* to indicate multiple curves in the plots. Please also explain each figure properly in your solution.
- **Blackboard submission:** All the matlab .m files (and only .m files) that you either create anew or modify must be appropriately named and placed into a **single** directory which should be zipped and uploaded into the course website. Your directory must be named as follows: `<yourBUemailID>_matlab5`. For example, if your BU email address is `mary567@bu.edu` you would submit a single directory named: `mary567_matlab5.zip` which contains all the matlab code (and only the code).
- For this assignment, only submit **one** .m file per problem with the following naming convention: `<yourBUemailID>_matlab5_<prb_num>.m`. For example if your id is `charles500` and the .m file is for problem 5.1, then you must submit a file named: `charles500_matlab5_1.m`. Note that the dot . in 5.1 is replaced with an underscore (this is important). *Running the .m file for a problem should generate the results for all parts of that problem, i.e., it should be self-contained.*

Problem 5.1 Kernel Principal Component Analysis (KPCA): In this problem, we will attempt to "un-roll" a 3-dimensional helix and swiss roll dataset using kernel PCA. Ideally, the helix unravels to become a circle, and the swiss roll a flat surface. We have provided the helix and swiss roll datasets in `helix.mat` and `swiss.mat`, respectively. The 3d points in these datasets, variable `X`, are generated from an underlying function, $f(t)$. For the purposes of visualizing the lower-dimensional "un-rolling", we have provided the underlying variable `t`, corresponding to each point.

The following is high-level pseudo-code for kernel PCA.

1. Compute the *centered* $n \times n$ Gram matrix: $\widetilde{\mathbf{K}}$.
2. Compute the q largest eigenvalues $\lambda_1 \geq \ldots \geq \lambda_q$ and the corresponding *orthonormal* eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_q$ of $\widetilde{\mathbf{K}}$.
3. Construct $\mathbf{V}_q = [\mathbf{v}_1, \ldots, \mathbf{v}_q]$ and $\Lambda_q^{1/2} = \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_q})$.
4. Encode training set as columns of: $\Lambda_q^{1/2}\mathbf{V}_q^\top$.

We will consider the following kernel matrices of the form $\mathbf{K}_{i,j} = K(x_i, x_j)$:

1. Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle + c$
2. Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\alpha \langle \mathbf{x}_i, \mathbf{x}_j \rangle + c)^r$
3. Radial basis function (rbf) kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\big)$

By default, these kernels cannot be assumed to be centered (the implicit feature data will not necessarily have zero mean). To solve this issue, we can center the gram matrix with a centering matrix $\mathbf{H}$ as follows:

$$\widetilde{\mathbf{K}} = \mathbf{HKH}$$

where, $\mathbf{H} = (\mathbf{I} - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T)$ and n is the number of points ($\mathbb{1}_n$ is a vector of 1s of length n). $\widetilde{\mathbf{K}}$ is then used in KPCA.

Isometric feature mapping (Isomap) can be considered as a special case kernel of KPCA with the following centered kernel:

$$\mathbf{K}_{\text{Isomap}} = -\frac{1}{2}\mathbf{H}\Delta\mathbf{H}$$

where Δ is the element-wise squared geodesic distance matrix computed in Isomap. The construction of Δ is as follows:

- Create a k-nearest neighbor undirected connected graph,[1] with Euclidean distance between points as the edge weights. For this exercise we assume that k is selected such that **all** the points lie in one connected component.
- Compute the approximate geodesic distance between any pair of points i, j, on the graph as the length of the shortest path $d_{i,j}$ between them on the graph using Dijkstra's algorithm (we have provided this functionality for you in `dijk.m` and `dijkstra.m` from Mark Steyver's group at Stanford – please refer to the provided files for proper usage and syntax)
- For all i, j, compute $\Delta_{i,j} = (d_{i,j})^2$.

[1] Points i and j are connected by an edge if i is among the k nearest neighbors of j *or* j is among the k nearest neighbors of i.

Perform KPCA with the Isomap kernel: $\mathbf{K}_{\text{Isomap}}$.

(a) On two separate figures, plot the 3-d points helix and swiss roll data. Color the points, with the underlying variable `t`. For the 3-d viewpoint use: `view([12 20])`.
Useful MATLAB functions: `scatter3, grid on`.

(b) **Implement** kernel PCA with the following kernels: linear, polynomial ($c = 0, d = 3$), and radial basis function ($\sigma = 4$). Embed the 3-d data points in the helix and swiss roll datasets into 2-d space. (i) Use `scatter` to plot the lower dimensional points (colored with variable t) for each kernel. You should have 6 plots in total, 3 for each dataset. Label your plots accordingly. In addition, for ease in grading, please call `rng('default')` before any non-deterministic function (i.e., `eigs`) such that the same plots are always generated.

(c) What similarities or differences do you observe between the kernels? Which ones (if any) succeed in unraveling a dataset?

(d) **Implement** kernel PCA using $\mathbf{K}_{\text{Isomap}}$ as a kernel (as we have described above). Set $k = 7$ in your nearest neighbor graph. Embed the 3-d data points in the helix and swiss roll datasets into 2-d. (i) Plot the edges of the k-nearest neighbor graph. For the 3-d viewpoint use: `view([12 20])`. Repeat this for each dataset (2 plots for this part). (ii) Again, use `scatter` to plot the lower dimensional points (colored with variable t) with the Isomap kernel. You should have another 2 plots for this part.

(e) Does Isomap succeed in unraveling a dataset? How does this compare to the results earlier in (b), with the other kernels? What are some trade-offs between the methods?