Fachgebiet Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Exercise Sheet 2: Unsupervised Learning

Deadline: see website

In this exercise sheet you will have to write both code and a report (as PDF). Both have to be handed in using PASSR (see link on website). Please run the test scripts (see website) before submitting your solution.

Exercises

Part 1: Implementation

Exercise 1 (2 pts)

Write a function `PCA` with signature

```
[Z, U, D] = PCA(X, m)
```

which receives a $d \times n$ matrix X and the number of components m to be used, and which returns the principal components, as well as the projected data points in a $m \times n$ matrix Z.

U and D should contain the principal components: U is a $d \times d$ matrix, which contains the principal directions, and D is a $1 \times d$ vector, which contains the principal values, sorted in descending order (i.e. $D_1 \geq D_2 \ldots$).

Exercise 2 (4 pts)

Write a function `Isomap` with signature

```
Y = Isomap(X, m, n_rule, param)
```

which receives a $d \times n$ matrix X containing the data points, and which calculates an m-dimensional embedding $Y \in R^{m \times n}$ using the Isomap algorithm. The parameter `n_rule` determines the method (`'knn'` or `'eps-ball'`) which is used to build the graph. `param` is the corresponding parameter (k or ϵ, respectively). The function should be robust against malicious parameters, use the Matlab-function `error` for error reporting. Especially you should check, whether the resulting graph is connected.

Exercise 3 (4 pts)

Write a function `LLE` with signature

```
Y = LLE(X, m, n_rule, param)
```

which implements the LLE algorithm. The parameters have the same meaning as in the function `Isomap`.

Part 2: Application

Exercise 4 (3 pts)

Apply PCA to the `usps` data set (see website). Write a Matlab script (or a function without any arguments, to be able to define local functions, which Matlab does not allow in scripts), which does the follwing:

1. Load the `usps` data set
2. For each digit:
   (a) Extract all the `X` data for this digit.

(b) Calculate the PCA of this data.

(c) Plot the principal values (as a bar plot, see `bar`) and the first 5 principal directions as images (using `imagesc`).

3. Now add Gaussian noise to the images (see `randn`). Select an appropriate variance on your own, such that the resulting images are very noisy.

4. For each digit:

   (a) Extract all the `X` data for this digit.

   (b) Calculate the PCA of this data, where `m` should be tuned by hand, such that the reconstructed (denoised) images are fairly good. The reconstruction y of a data point x by the k largest eigenvectors $v_1, \ldots, v_k$ of the covariance matrix is given by

$$y = \sum_{i=1}^{k} v_i x^\top v_i. \tag{1}$$

   The reconstruction error of x is therefore $\|x - y\|$.

   (c) For 5 examples of your choice, plot the noisy image and its denoised reconstruction (using `imagesc`).

Remark: use `subplot` to arrange the images in a single figure.

Exercise 5 (3 pts)

Apply LLE and Isomap on the data sets which you can find on the website. Write a script which does the follwing:

1. For all data sets:

   (a) Load the data set.

   (b) Apply Isomap on the data set, where the parameters should be chosen appropriately (depending on the data set).

   (c) Plot the resulting embedding in a 2-dimensional coordinate system, for example using `scatter`. Show the name of the data set and the used method and parameter values in the title of the plot (using `title`).

   (d) Repeat everything for LLE.

Exercise 6 (4 pts)

In this exercise the influence of noise on LLE and Isomap should be studied, using the example of the `flatroll` data set. Write a script which does the follwing:

1. Load the data set.

2. Add Gaussian noise with variance 0.2 and 1.8 to the data set (this results in 2 noisy data sets).

3. Apply LLE and Isomap on both data sets, where the neighborhood graph should be constructed using k-nn. Try to find both a good value for k and a value which is obviously too large.

4. Plot for each combination of method (LLE and Isomap), noise (0.2 and 1.8) and k (good and too large one) the neighboring graph (for example using `scatter` to plot nodes and `line` to plot edges) and the resulting embedding (i.e. 8 plots per method in total).