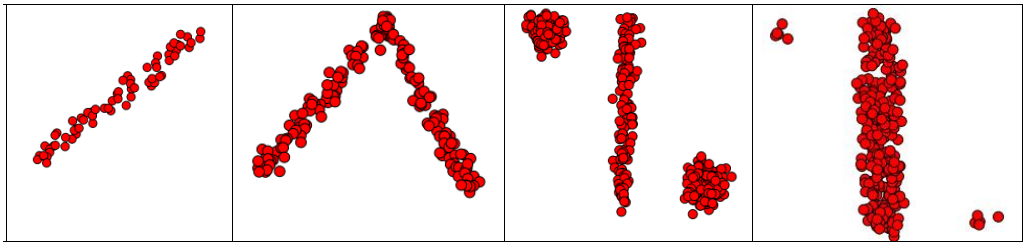
Assignment 8

Machine Learning, Summer term 2013, Ulrike von Luxburg

Solutions due by June 10

Exercise 1 (Direction of principal components, 1 point) Guess and plot the direction of eigenvectors on each image. Draw roughly the data projected on these eigenvectors.



Exercise 2 (Expressing the variance, 2 points) A computer game company runs a survey among visitors of their website. Around 1000 people participate in this survey and they provide their 1- age, 2- time spent playing with computer, 3- time spent in facebook and 4- time spent doing sport. Then they run a PCA on the data.

- What does it mean if a single eigenvector covers 90% of the data variance?
- How would you interpret the results if the eigenvector $v_1 = [0, 1, 1, 1]^T$ covers 85% of the data variance.

Exercise 3 (Generating samples from a Gaussian distribution, 5 points) Assume you are given the mean μ and the covariance matrix Σ of a d-dimensional normal density $\mathcal{N}(\mu, \Sigma)$ and you want to sample n points from this density. The following matlab code will do this for you:

```
S1 = chol(Sigma);
X = repmat(mu,n,1) + randn(n,d)*S1;
```

The command `S1 = chol(A)` generates an upper triangular matrix `S1` which satisfies `A=S1'*S1`. This decomposition is called the Cholesky decomposition. An alternative method is to decompose `A` to eigenvectors and eigenvalues by `[V,D] = eig(A)` and then form `S2` by `S2=V*sqrt(D)`. However, the Cholesky decomposition is numerically more stable and computationally faster than eigen decomposition method.

- Show that in eigen decomposition, $A = S2 \cdot S2'$.
- Generate $n = 2000$ points in 3 dimensional space from a Gaussian distribution with mean `mu=[0,0,0]` and Covariance `Sigma=[2 0 0;0 1 0;0 0 4]`. Plot it with `plot3`.
- What are the eigenvalues and eigenvectors of the covariance matrix `Sigma`?
- Assume you know eigenvalues and eigenvectors of your covariance matrix:

$$\Lambda = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}, V = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

  Generate $n = 400$ points in 2 dimensional space from a Gaussian distribution with mean zero and covariance matrix corresponding to these eigenvalues and eigenvectors ($\Sigma = V \Lambda V'$). Plot the points and guess the approximate direction of principal components in the figure.

Web page: `http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/teaching/2013-ss-vorlesung-ml/`
Login: "machine", Password: "learning"

- Add the eigenvectors in V to your plot. Compare your guessed directions with these eigenvectors.

Exercise 4 (PCA, 4 points) Implement PCA in matlab. Do it in a three line matlab code: Subtract the mean of your data, calculate the covariance matrix C, and find its eigenvalues and eigenvectors using the matlab command `[V,D] = eig(C)`.
To test your code, generate 500 samples from a Gaussian distribution with mean $\mu = [1, 1]$ and covariance $\Sigma = [2, -1; -1, 2]$. For generating the points you can either use your code from Exercise 3, or use the matlab command `normrnd`. Apply your PCA code on this data and compare the result with the eigenvectors of the covariance matrix Σ.

Exercise 5 (PCA on USPS data, 2 points) Apply the PCA method on images of digits 5 from USPS dataset. The dataset is available in the course website from Assignment 1. Plot the image of the first and the second principal components as 16x16 grayscale images. You can either use your PCA implementation from Exercise 4, or use the matlab command `princomp`.

Exercise 6 (Isomap on USPS data, 5 points) In this exercise you will implement the Isomap method to embed digits 1,2,3,4 from USPS dataset into $\mathbb{R}^2$. The code for building kNN graph and the Isomap algorithm itself is provided in the course web page.

- Load the data from `usps_train.mat`. Select 300 example from each of digits $\{1, 2, 3, 4\}$ and put it in variable `X`. Put the corresponding labels in `Y`. Do not forget to convert the data to double.
- Set the connectivity parameter in the kNN graph to $k = 10$ and use the following code to plot the embedding in 2 dimensional space using Isomap. Read the manual of the command `scatter` to understand how does it work.

```
A = buildKnnGraph(X,k);
D = graphallshortestpaths(A,'Directed', false);
xy = Isomap(D,2);

figure;
scatter(xy(:,1),xy(:,2),10,Y,'filled');
```

- Play with the parameter k. Describe the effect of the parameter on the embedding.
- Plot the embedding of the data into 2 dimensional space spanned by the first two principal components of PCA. You can either use your PCA implementation from Exercise 4, or use the matlab command `princomp` to perform PCA.