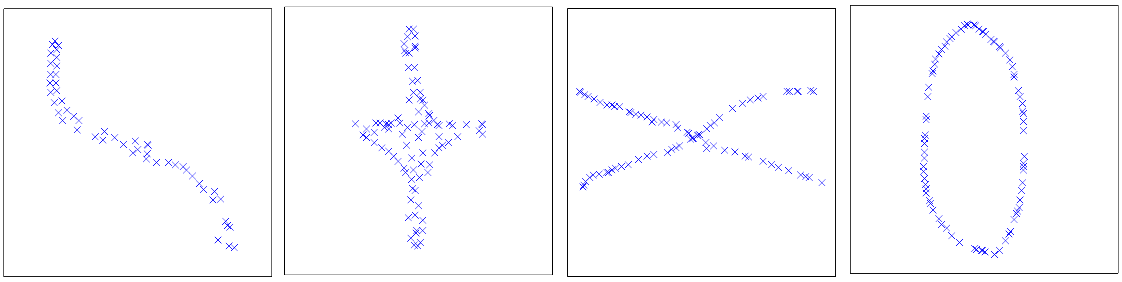
Assignment 7

Machine Learning, Summer term 2014, Ulrike von Luxburg

To be discussed in exercise groups on June 2-4

Exercise 1 (Direction of principal components, 1 point) Below are a number of 2D-data sets. Plot the two principal components.



Exercise 2 (Interpreting principal components, 2 points) A carsharing service runs a survey among 1000 students, who provide information concerning their 1- income, 2- distance they cover by car per month, 3- distance they cover by bike per month, 4- distance they cover by public transport per month, 5- distance they cover by foot per month. Then they run a PCA on the data. Provide answers to the following questions:

- What would it mean if a single eigenvector covered 95% of the total data variance?
- How would you interpret the result if the eigenvector $v_1 = [0, 0, 1, -1, 0]$ covers 90% of the total data variance?
- Why might it be necessary to rescale the data before running PCA in order to obtain a sensible result?

Exercise 3 (Generating samples from a Gaussian distribution, 0.5+0.5+0.5+1+0.5 points) 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. Assuming that Σ is positive definite, 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(Sigma)` generates an upper triangular matrix `S1` which satisfies `Sigma=S1'*S1`. This decomposition is called the Cholesky decomposition. An alternative method, which also works when Σ is only positive semi-definite, is to decompose Σ to eigenvectors and eigenvalues by `[V,D] = eig(Sigma)` and then form `S2` by `S2=V*sqrt(D)`. However, the Cholesky decomposition is numerically more stable and computationally faster than eigen decomposition method.

(a) Show that in eigen decomposition, $\Sigma = S2 \cdot S2'$.

(b) 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`.

(c) What are the eigenvalues and eigenvectors of the covariance matrix `Sigma`?

Web page: `http://www.informatik.uni-hamburg.de/ML/contents/people/luxburg/teaching/2014-ss-vorlesung-ml/`
Login: "machine", Password: "learning"

(d) 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.

(e) Add the eigenvectors in V to your plot. Compare your guessed directions with these eigenvectors.

Exercise 4 (PCA, 2+1 points)

(a) 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)`.

(b) To test your code (if you could not solve part (a), you can use the MATLAB command `pca`) 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, 1+3 points)

(a) Apply the PCA method on images of digits 5 from USPS dataset (use the training data of the complete dataset — available on the course webpage from Assignment 4). Plot the first and the second principal components as 16x16 grayscale images. You can either use your PCA implementation from Exercise 4 or the MATLAB command `pca`.

(b) Choose three images of digits 5 from USPS dataset at random and project them onto 1- the first principal component, 2- the first and the second principal component in $\mathbb{R}^{256}$ (i.e. as a result you should obtain vectors in the original space — this is View 1 in the notation of the lecture notes). Create a 3×3-subplot (use `help subplot` in case you do not know how this works) showing the original images in the first row, the results from 1 in the second row, and the results from 2 in the third row (using `imagesc`).

Exercise 6 (Isomap on USPS data, 1+1+1 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 on the course web page.

In preparation for the following, load the data from `usps_train_complete.mat` (available on the course webpage from Assignment 4). Select 300 examples from each of digits $\{1, 2, 3, 4\}$ and put them in variable `X`. Put the corresponding labels in `Y`.

(a) 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 it works.

```
A = buildKnnGraph(X,k);
D = graphallshortestpaths(A,'Directed', false);
xy = Isomap(D,2);
figure;
scatter(xy(:,1),xy(:,2),10,Y,'filled');
```

(b) Play with the parameter k. Describe the effect of the parameter on the embedding.

(c) Project the data onto the first two principal components of PCA in $\mathbb{R}^2$ (i.e. as a result you should obtain vectors in $\mathbb{R}^2$ — this is View 2 in the notation of the lecture notes). Plot the embedding, again using the command `scatter`. You can either use your PCA implementation from Exercise 4 or the MATLAB command `pca` to perform PCA.