Machine Learning
Exercise 8

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

June 16, 2015

1 PCA optimality principles

Proof what is given on slide 04:36.

a) That is, for data $D = \{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^d$ and *orthonormal* $V \in \mathbb{R}^{d\times p}$, first prove

$$\widehat{\mu}, \widehat{z}_{1:n} = \underset{\mu, z_{1:n}}{\text{argmin}} \sum_{i=1}^n \|x_i - V z_i - \mu\|^2 \quad \Rightarrow \quad \widehat{\mu} = \langle x_i \rangle = \frac{1}{n}\sum_{i=1}^n x_i \ , \quad \widehat{z}_i = V^\top (x_i - \mu) \ . \tag{1}$$

b) Now, with $\tilde{x}_i = x_i - \widehat{\mu}$, prove

$$\widehat{V} = \underset{V\in\mathbb{R}^{d\times p}}{\text{argmin}} \sum_{i=1}^n \|\tilde{x}_i - VV^\top \tilde{x}_i\|^2 \quad \Rightarrow \quad V = (v_1\ v_2\ \cdots\ v_p) \tag{2}$$

where the latter are the p largest eigenvectors of $X^\top X$, and $X_{i\cdot} = \tilde{x}_i^\top$. Guide:

- The column vectors of V provide a "partial" orthonormal coordinate system. You may introduce a matrix W with remaining orthonormal column vecturs such that $WW^\top + VV^\top = \mathbf{I}$ and therefore $\tilde{x} = WW^\top \tilde{x} + VV^\top$. (Intuition: this decomposes any x in a part that lies within the sub-vector space spanned by V, and a part orthogonal to this sub-vector space.)
- Now rewrite the objective as $\sum_{j=1}^{d-p} w_j X^\top X w_j$, where we sum over the column vectors w_j of W, and included the summation over the data in $X^\top X$.
- Now prove that choosing W to contain the smallest eigenvectors of $X^\top X$ minimizes the objective. (Intuition: the objective is the squared distance of $\tilde{x}$ to the sub-vector space spanned by V.)

c) In the above, is the orthonormality of V an essential assumption?

d) Proove that you can compute the V also from the SVD of X (instead of $X^\top X$).

2 PCA and reconstruction on the Yale face database

On the webpage find and download the Yale face database http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/data/yalefaces.tgz. (Optinally use `yalefaces_cropBackground.tgz`.) The file contains gif images of 165 faces.

a) Write a routine to load all images into a big data matrix $X \in \mathbb{R}^{165\times 77760}$, where each row contains a gray image.

In Octave, images can easily be read using `I=imread("subject01.gif");` and `imagesc(I);`. You can loop over files using `files=dir(".");` and `files(:).name`. Python tips: `import matplotlib.pyplot as plt import scipy as sp plt.imshow(plt.imread(file_name)) u, s, vt = sp.sparse.linalg.svds(X, k=neigenvalues)`

b) Compute the mean face $\mu = 1/n \sum_i x_i$ and center the whole data, $X \leftarrow X - \mathbf{1}_n \mu^\top$.

c) Compute the singular value decomposition $X = UDV^\top$ for the data matrix.[1] In Octave/Matlab, use the command `[U, S, V] = svd(X, "econ")`, where the `econ` ensures we don't run out of memory.

d) Map the whole data set to $Z = XV_p$, where $V_p \in \mathbb{R}^{77760\times p}$ contains only the first p columns of V. Assume $p = 60$. The Z represents each face as a p-dimensional vector, instead of a 77760-dimensional image.

[1] This is alternative to what was discussed in the lecture: In the lecture we computed the SVD of $X^\top X = (UDV^\top)^\top (UDV^\top) = VD^2V^\top$, as U is orthonormal and $U^\top U = \mathbf{I}$. Decomposing the covariance matric $X^\top X$ is a bit more intuitive, decomposing X directly is more efficient and amounts to the same V.

e) Reconstruct all images by computing $\tilde{X} = \mathbf{1}_n \mu^\top + \boldsymbol{Z} V_p^\top$. Display the reconstructed images (by reshaping each row of $\tilde{X}$ to a 320×243-image) – do they look ok? Report the reconstruction error $\|X - \tilde{X}\|^2 = \sum_{i=1}^n \|x_i - \tilde{x}_i\|^2$. Repeat for various PCA-dimensions $p = 1, 2, ...$