Exercises
Introduction to Machine Learning
FS 2018

Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: https://las.inf.ethz.ch/teaching/introml-s18
Email questions to:
Aytunc Sahin, aytunc.sahin@inf.ethz.ch

Series 5, May 4th, 2018 (Clustering and PCA)

Problem 1 (K-means convergence):

In the K-means clustering algorithm, you are given a set of n points $x_i \in \mathbb{R}^d$, $i \in \{1, \dots, n\}$ and you want to find the centers of k clusters $\mu = (\mu_1, \dots, \mu_k)$ by minimizing the average distance from the points to the closest cluster center. Formally, you want to minimize the following loss function

$$L(\mu) = \sum_{i=1}^{n} \min_{j \in \{1,\dots,k\}} \|x_i - \mu_j\|_2^2.$$

To approximate the solution, we introduce new assignment variables $z_i \in \arg\min_{j \in \{1,\dots,k\}} \|x_i - \mu_j\|_2^2$ for each data point x_i. The K-means algorithm iterates between updating the variables z_i (*assignment step*) and updating the centers $\mu_j = \frac{1}{|\{i:z_i=j\}|} \sum_{i:z_i=j} x_i$ (*refitting step*). The algorithm stops when no change occurs during the *assignment step*.

Show that K-means is guaranteed to converge (to a local optimum). *Hint:* You need to prove that the loss function is guaranteed to decrease monotonically in each iteration until convergence. Prove this separately for the *assignment step* and the *refitting step* .

Solution 1:

To prove convergence of the K-means algorithm, we show that the loss function is guaranteed to decrease monotonically in each iteration until convergence for the *assignment step* and for the *refitting step*. Since the loss function is non-negative, the algorithm will eventually converge when the loss function reaches its (local) minimum.

Let $z = (z_1, \dots, z_n)$ denote the cluster assignments for the n points.

(i) *Assignment step*
We can write down the original loss function $L(\mu)$ as follows:

$$L(\mu, z) = \sum_{i=1}^{n} \|x_i - \mu_{z_i}\|_2^2$$

Let us consider a data point x_i, and let z_i be the assignment from the previous iteration and z_i^* be the new assignment obtained as:

$$z_i^* \in \arg\min_{j \in \{1,\dots,k\}} \|x_i - \mu_j\|_2^2$$

Let z^* denote the new cluster assignments for all the n points. The change in loss function after this assignment step is then given by:

$$L(\mu, z^*) - L(\mu, z) = \sum_{i=1}^{n} \left(\|x_i - \mu_{z_i^*}\|_2^2 - \|x_i - \mu_{z_i}\|_2^2 \right) \leq 0$$

The inequality holds by the rule z_i^* is determined, i.e. to assign x_i to the nearest cluster.

(ii) *Refitting step*
We can write down the original loss function $L(\mu)$ as follows:

$$L(\mu, z) = \sum_{j=1}^{k} \Big(\sum_{i:z_i=j} \|x_i - \mu_j\|_2^2 \Big)$$

Let us consider the j^{th} cluster, and let μ_j be the cluster center from the previous iteration and μ_j^* be the new cluster center obtained as:

$$\mu_j^* = \frac{1}{|\{i : z_i = j\}|} \sum_{i:z_i=j} x_i$$

Let μ^* denote the new cluster centers for all the k clusters. The change in loss function after this refitting step is then given by:

$$L(\mu^*, z) - L(\mu, z) = \sum_{j=1}^{k} \Big(\big(\sum_{i:z_i=j} \|x_i - \mu_j^*\|_2^2 \big) - \big(\sum_{i:z_i=j} \|x_i - \mu_j\|_2^2 \big) \Big) \leq 0$$

The inequality holds because the update rule of μ_j^* essentially minimizes this quantity.

Problem 2 (K-medians clustering):

In this exercise, you are asked to derive a new clustering algorithm that would use a different loss function given by

$$L(\mu) = \sum_{i=1}^{n} \min_{j \in \{1,\ldots,k\}} \|x_i - \mu_j\|_1.$$

(a) Find the update steps for both z_i and for μ_j in this case.

(b) What can you say about the convergence of your algorithm?

(c) In which situation would you prefer to use K-medians clustering instead of K-means clustering?

Solution 2:

(a) As in the K-means algorithm, let's again introduce hidden variables $z_i = \arg\min_{j \in 1,\ldots,k} ||x_i - \mu_j||_1$ for each data point x_i. Then the initial problem

$$\mu = \arg\min_{\mu} \sum_{i=1}^{n} \min_{j \in 1,\ldots,k} ||x_i - \mu_j||_1$$

can be rewritten in a different form (because we know where exactly the minimum is achieved):

$$\mu = \arg\min_{\mu} \sum_{i=1}^{n} ||x_i - \mu_{z_i}||_1$$

In order to find the solution with respect to μ_j with fixed z_i, let's leave only the data points that correspond to the j^{th} component:

$$\mu_j = \arg\min_{\mu_j} \sum_{i:z_i=j} ||x_i - \mu_j||_1$$

$$\mu_j = \arg\min_{\mu_j} \sum_{i:z_i=j} \sum_{q=1}^{d} |x_{i,q} - \mu_{j,q}|$$

This can again be separated component-wise:

$$\mu_{j,q} = \arg\min_{\mu_{j,q}} \sum_{i:z_i=j} |x_{i,q} - \mu_{j,q}|$$

Again, as in the K-means algorithm, we proceed by finding the derivative of the functional and setting it to zero. In order to get rid of the L_1 norm, we also separate the functional into the sum over those $x_{i,q}$ that are smaller than $\mu_{j,q}$ and those that are larger:

$$\sum_{i:z_i=j, x_{i,q} \le \mu_{j,q}} |x_{i,q} - \mu_{j,q}| + \sum_{i:z_i=j, x_{i,q} > \mu_{j,q}} |x_{i,q} - \mu_{j,q}| = \sum_{i:z_i=j, x_{i,q} \le \mu_{j,q}} (\mu_{j,q} - x_{i,q}) + \sum_{i:z_i=j, x_{i,q} > \mu_{j,q}} (x_{i,q} - \mu_{j,q})$$

The derivative of every bracket in the sum is either $+1$ or -1, and the number of $+1$'s is exactly $|\{i : z_i = j, x_{i,q} \le \mu_{j,q}\}|$. Therefore, we need to set

$$|\{i : z_i = j, x_{i,q} \le \mu_{j,q}\}| - |\{i : z_i = j, x_{i,q} > \mu_{j,q}\}| = 0$$

This means that $\mu_{j,q}$ is nothing but the *median* of all the numbers $x_{i,q}$, $i : z_i = j$.

The resulting algorithm then iterates between two steps:

- $z_i = \arg\min_{j \in 1,\dots,k} ||x_i - \mu_j||_1$
- $\mu_{j,q} = \text{median}(x_{i,q}, i : z_i = j), \forall j = 1, \dots, k; \forall q = 1, \dots, d.$

(b) You can prove the same convergence properties for K-medians as for K-means.

(c) In comparison with K-means, K-medians clustering is particularly robust to outliers. Thus, if we expect out input data to have many outliers, it is preferable to use K-medians clustering.

Problem 3 (PCA):

Suppose we have a dataset with 4 points:

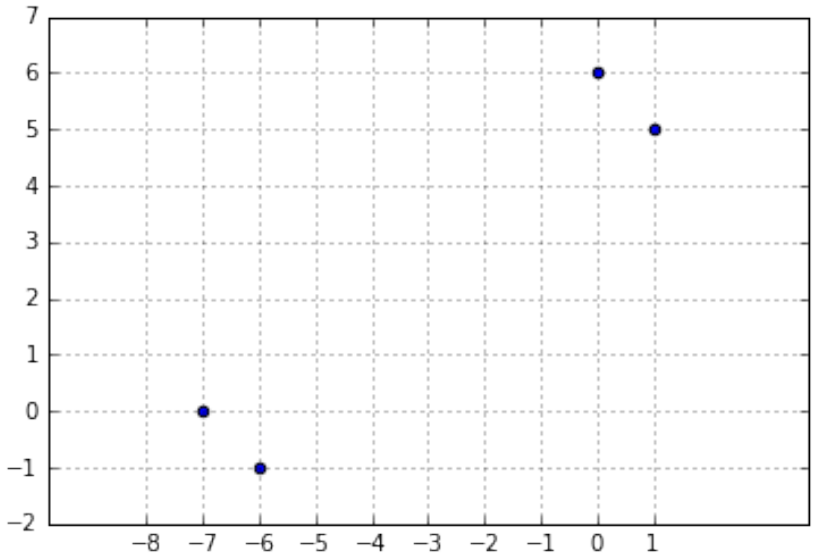
$$\mathcal{D} = \{(1, 5), (0, 6), (-7, 0), (-6, -1)\}$$

(a) Plot the dataset and try to guess two principal components ($k = 2$).

(b) Compute the empirical covariance matrix, its eigenvalues and eigenvectors. Do the eigenvectors correspond to your guess of principal components? Please do not forget the assumptions of PCA. (The dataset should be centered and we want unit eigenvectors.)

Solution 3:

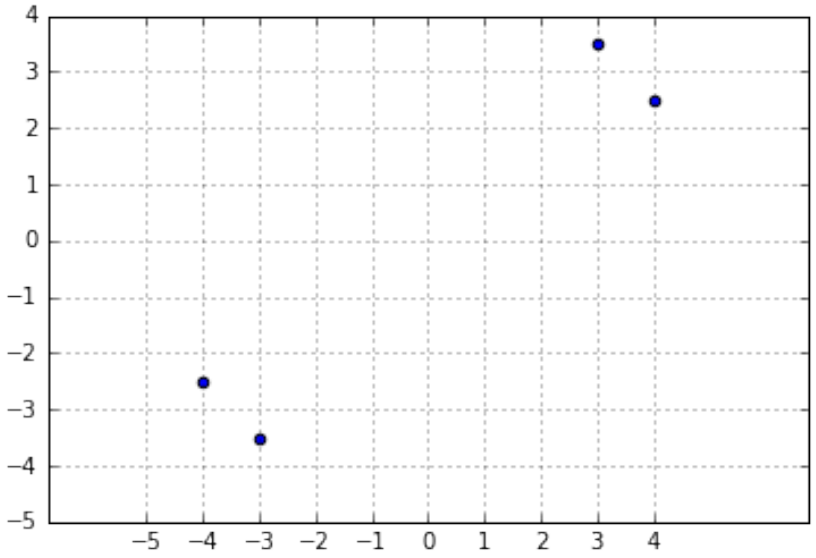
(a) Plot of the original dataset:



(b) We first need to center the data by subtracting from it its mean $(-3,\, 2.5)^T$, obtaining

$$\begin{aligned}
\boldsymbol{x_1} &= (4,\, 2.5)^T \\
\boldsymbol{x_2} &= (3,\, 3.5)^T \\
\boldsymbol{x_3} &= (-4,\, -2.5)^T \\
\boldsymbol{x_4} &= (-3,\, -3.5)^T.
\end{aligned}$$

Plot of the centered dataset:

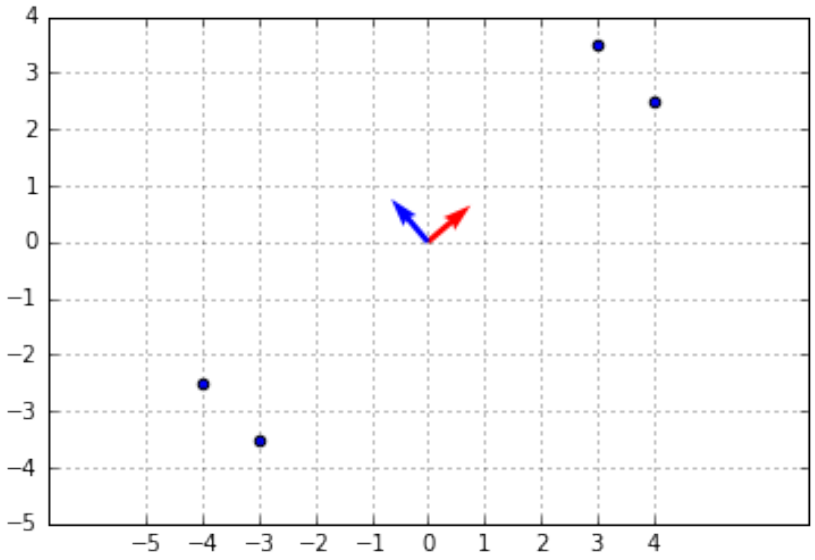


For the empirical covariance matrix, we obtain

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i}\boldsymbol{x_i}^T = \frac{1}{4} \cdot \begin{pmatrix} 50 & 41 \\ 41 & 37 \end{pmatrix} = \begin{pmatrix} 12.5 & 10.25 \\ 10.25 & 9.25 \end{pmatrix}.$$

The unit-length eigenvectors of Σ are $v_1 = (0.76045416, 0.64939162)^T$ and $v_2 = (-0.64939162, 0.76045416)^T$ with eigenvalues $w_1 = 21.25301161$ and $w_2 = 0.49698839$, respectively.

Plot of the centered dataset with principal components 1 (red) and 2 (blue):



For a nice visualization of PCA, also see `http://setosa.io/ev/principal-component-analysis`.