National Economics University, Vietnam

Faculty of Mathematics Economics

Data Science in Economics and Business

Machine Learning 2

Homework Week 1: Principal Component Analysis

Student: Nguyễn Anh Tú - ID: 11207333

Problem 1

Explain the math behind Principal Component Analysis algorithm.

Solution.

We consider an independent and identically distributed (i.i.d.) dataset $X = x_1, x_2, \ldots, x_n$ with mean 0 and $x_n \in \mathbb{R}^D$. In general, if a dataset does not have a non-zero mean, we can standardize it before other steps.

PCA is the method in which we will find a new basis, a.k.a a projection matrix, where the original dataset can be projected onto it to reduce the data dimension but important information still retained.

We assume there exists a low-dimensional compressed representation of a data point x_n given by:

$$z_n = B^T x_n$$

where we compressed x_n of D dimensions to z_n of M dimensions ($M < D$), and the projection matrix is:

$$B = [b_1, b_2, \ldots, b_M] \in \mathbb{R}^{D \times M}$$

Note that z also has zero mean: $\mathbf{E}_z[z] = \mathbf{E}_x[B^T x] = B^T \mathbf{E}_x[x] = 0$

We need to find a matrix B that retains as much information as possible when compressing data by projecting it onto the subspace spanned by the columns $b_1, b_2, \ldots, b_M$ of B. Retaining most information after data compression is equivalent to capturing the largest amount of variance in the low-dimensional code.

We maximize the variance of the low-dimensional code using a sequential approach. First, we consider the case when X is projected onto a single vector $b \in \mathbb{R}^D$ (basically, this is when the projection matrix B is just a vector and we want to keep only one most important feature). Our aim is to maximize the variance of the projected data, i.e:

$$\text{maximize } V = \frac{1}{N}\sum_{n=1}^{N} z_n^2$$

where $z_n = b^T x_n$. Substituting this into the expression of V:

$$\begin{aligned} V &= \frac{1}{N}\sum_{n=1}^{N}(b^T x_n)^2 = \frac{1}{N}\sum_{n=1}^{N} b^T x_n x_n^T b \\ &= b^T(\frac{1}{N}\sum_{n=1}^{N} x_n x_n^T)b = b^T S b \end{aligned}$$

where S is the covariance matrix of the original dataset.

Here, we observe that increasing the magnitude of b will increase V, thus making this optimization problem impossible to solve. Therefore, we restrict all solutions to $||b||^2 = 1$ which results in a constrained optimization problem:

$$\begin{aligned} &\text{find } \underset{b}{\operatorname{argmax}}\ b^T S b \\ &\text{subject to } ||b||^2 = 1 \end{aligned}$$

The Lagrangian function for the problem is:

$$\mathcal{L}(b, \lambda) = b^T S b + \lambda(1 - b^T b)$$

Note that b is a single vector then λ is just a number (not a vector) because we only have one condition. Take the partial derivative of $\mathcal{L}$ with respect to b and λ to 0:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial b} &= 2b^T S + 2\lambda b^T = 0 \iff S \cdot b = \lambda b \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= 1 - b^T b \iff b^T b = 1 \end{aligned}$$

At this point, we can see that b and λ are a pair of eigenvector and eigenvalue of S. Then we can rewrite the variance V as:

$$V = b^T S b = b^T \lambda b = \lambda$$

The variance of the data projected onto a one-dimensional subspace equals the eigenvalue that is associated with the basis vector b that spans this subspace.

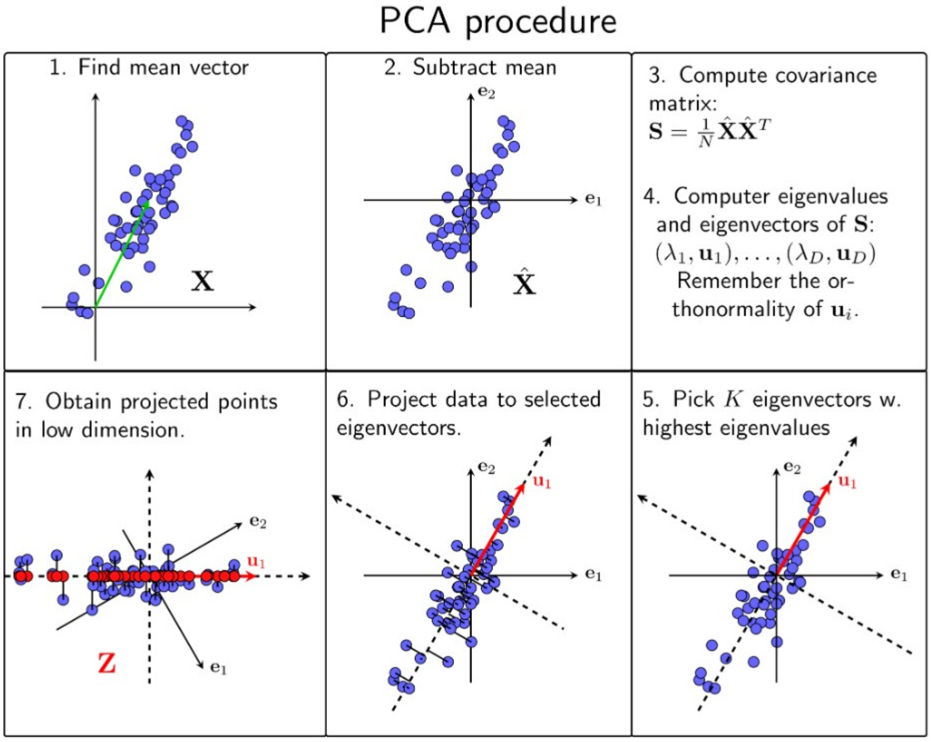
Therefore, to maximize the variance of the low-dimensional code, we choose the basis vector associated with the largest eigenvalue principal component of the data covariance matrix. This eigenvector is called the first principal component.

The finding of vector b above can be viewed as the finding of a direction or an axis that data varies the most. Suppose we want to find another direction besides the one we have found, we can construct the similar vector projection problem to find the largest variance possible, or actually the largest eigenvalue possible. Now what we derive will be the second largest eigenvalue. Therefore, if we want to find M axes to project data, we can choose M eigenvectors $b_1, b_2, \dots, b_M$ of the data variance matrix that associates with M largest eigenvalues, each eigenvector corresponding to a new axis of projection. The projection matrix B is formed by those eigenvectors, i.e $B = [b_1, b_2, \dots, b_M]$.

Finally, the projection of dataset $X \in \mathbb{R}^{D \times N}$ onto $B \in \mathbb{R}^{D \times M}$ given by $\widetilde{X} = B^T X$ will result in new representation of data, i. e $\widetilde{X} \in \mathbb{R}^{M \times N}$ (M features and N samples).

Note In case me in the future forget how this algorithm works, remember features of $\widetilde{X}$ are not derived by removing some less important features of X (and keeping the remaining) but by synthesizing information from the whole dataset X and compressing them into M **new** features.

Summary (image source):


PCA procedure
1. Find mean vector
$\mathbf{X}$
2. Subtract mean
$\mathbf{e}_2$
$\mathbf{e}_1$
$\hat{\mathbf{X}}$
3. Compute covariance matrix: $\mathbf{S} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T$
4. Computer eigenvalues and eigenvectors of $\mathbf{S}$: $(\lambda_1, \mathbf{u}_1), \dots, (\lambda_D, \mathbf{u}_D)$ Remember the orthonormality of $\mathbf{u}_i$.
7. Obtain projected points in low dimension.
$\mathbf{e}_2$
$\mathbf{u}_1$
$\mathbf{Z}$
$\mathbf{e}_1$
6. Project data to selected eigenvectors.
$\mathbf{e}_2$
$\mathbf{u}_1$
$\mathbf{e}_1$
5. Pick K eigenvectors w. highest eigenvalues
$\mathbf{e}_2$
$\mathbf{u}_1$
$\mathbf{e}_1$