

# Machine Learning, Tutorial 11

## Universität Bern

Abdelhak Lemkhenter (abdelhak.lemkhenter@inf.unibe.ch)

### PCA

PCA is a dimensionality reduction algorithm where each data point is projected to the first  $k$  eigenvectors of their covariance matrix. It is important that the data is first normalised, as follows:

- Let  $\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$
- Replace each  $x^{(i)}$  with  $x^{(i)} - \mu$
- Let  $\sigma_j^2 = \frac{1}{m} \sum_i (x_j^{(i)})^2$
- Replace each  $x_j^{(i)}$  with  $x_j^{(i)} / \sigma_j$

If you have prior knowledge that the input data has zero mean, you can skip the first two steps. If you have prior knowledge that the dimension of the data coordinates have the same scale, you can skip the third and fourth step.

1. There was a beer drinking competition at the Oktoberfest. There were two rounds, one on Saturday and one on Sunday. The participants needed to drink as many mugs of beer as possible in 30 minutes. The data below shows their results. The first row shows the amount they drank on Saturday, and the second show their result on Sunday.

	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
$mugs(Saturday)$	3	2	5	4
$mugs(Sunday)$	1	2	3	4

- Find the first principal component.  
**Solution.** Because the data coordinates have the same meaning (number of mugs), we do not need to perform the normalization ( $3^{rd}$  and  $4^{th}$ ) steps. First we extract the mean,

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)} = \frac{1}{4} \left( \begin{pmatrix} 1 \\ 3 \end{pmatrix} + \begin{pmatrix} 2 \\ 2 \end{pmatrix} + \begin{pmatrix} 5 \\ 3 \end{pmatrix} + \begin{pmatrix} 4 \\ 4 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 2.5 \end{pmatrix}, \quad (1)$$

$$X_c = X - \mu \cdot \mathbf{1}^T = \begin{pmatrix} -0.5 & -1.5 & 1.5 & 0.5 \\ -1.5 & -0.5 & 0.5 & 1.5 \end{pmatrix}, \quad (2)$$

where  $X_c$  denotes the centred data. Then we need to compute the covariance matrix, which can be done with the following formula,

$$\Sigma = \frac{1}{m} X_c \cdot X_c^T = \begin{pmatrix} 5/4 & 3/4 \\ 3/4 & 5/4 \end{pmatrix}. \quad (3)$$

To compute the eigenvalues, we need to set the characteristic polynomial to 0.

$$\det(\Sigma - \lambda I) = p(\lambda) = (5/4 - \lambda)(5/4 - \lambda) - 9/16 = 0, \quad (4)$$

which leads to the eigenvalues  $\lambda_1 = 2$  and  $\lambda_2 = 0.5$ . The first principal component is simply the eigenvector  $\mathbf{v}_1$  corresponding to the largest eigenvalue. This can be computed from the following equation,

$$(\Sigma - \lambda_1 I) \mathbf{v}_1 = \begin{pmatrix} -3/4 & 3/4 \\ 3/4 & -3/4 \end{pmatrix} \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} = 0, \quad (5)$$

which yields  $v_{11} = v_{21}$ . Using the convention of  $\|\mathbf{v}_1\| = 1$ , we get  $\mathbf{v}_1 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^T$ .

- Find the second principal component.

**Solution:** The second principal component is the eigenvector corresponding to the second eigenvalue. It can be computed in the same way as above,  $\mathbf{v}_2 = (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})^T$ .

- Model the data as the sum of one principal component and some noise. Remove the noise from the data.

**Solution:** Removing the noise of the data means to project it to its principal subspace. It consists of three steps: extract the mean from the data, project it to the first few (in our case one) principal components, and add back the mean. We already computed  $X_c$ , now we need to project it to the first principal component,

$$X_p = \mathbf{v}_1^T X_c = \left(-\frac{2}{\sqrt{2}}, -\frac{2}{\sqrt{2}}, \frac{2}{\sqrt{2}}, \frac{2}{\sqrt{2}}\right). \quad (6)$$

$X_p$  can be thought of as the compressed data. We can get the denoised data  $X_d$  by the following formula,

$$X_d = \mathbf{v}_1 \mathbf{v}_1^T X_c + \mu \cdot \mathbf{1}^T = \begin{pmatrix} 2.5 & 2.5 & 4.5 & 4.5 \\ 1.5 & 1.5 & 3.5 & 3.5 \end{pmatrix}. \quad (7)$$

**NOTE** In equation (2) and (7) we used the vector  $\mathbf{1}^T$ . This is simply a notation for the vector  $\mathbf{1}^T = (1, 1, \dots, 1)$ , where the length of the vector is the number of data points  $m$ .

2. There is a nice theorem in numerical linear algebra, that says the following. Suppose  $M$  is an  $m \times n$  matrix whose entries are real numbers. Then there exists a factorisation of the form

$$M = USV^T, \quad (8)$$

where  $U$  and  $V$  are orthogonal matrices, and  $S$  is diagonal, and  $s_i \geq 0$  for every diagonal element of  $S$ . The  $s_i$  diagonal elements are called singular values, and the factorisation itself is called singular value decomposition (SVD). This decomposition is unique (in most cases). How would you compute the principal components using SVD?

**Solution:** The principal components are the eigenvectors of the covariance matrix of the data. The covariance matrix is expressed in the following form,

$$\Sigma = \frac{1}{m} X_c X_c^T. \quad (9)$$

Factorising  $\frac{1}{\sqrt{m}} X_c$  with SVD, we get  $\frac{1}{\sqrt{m}} X_c = USV^T$ . We can write

$$\Sigma = \frac{1}{m} X_c X_c^T = USV^T V S U^T = US^2 U^T. \quad (10)$$

It is easy to see, that the columns of  $U$  (denoted by  $u_i$ ) are eigenvectors of  $\Sigma$  and the eigenvalues are the singular values squared  $\lambda_i = s_i^2$ .

$$\Sigma u_i = US^2 U^T u_i = US^2 e_i = U(s_i^2 e_i) = s_i^2 U e_i = s_i^2 u_i, \quad (11)$$

where  $e_i = (0, 0, \dots, 1, \dots, 0)^T$ , the  $i^{th}$  element of  $e_i$  is 1, all the others are 0. There is an interesting thing to notice here. The factorization of  $\Sigma = US^2 U^T$  is a valid SVD factorization too. You can compute the SVD of  $\Sigma = U_\Sigma S_\Sigma V_\Sigma^T$ . Because the SVD is unique (apart from multiplying  $u_i$  with  $\pm 1$ ),  $U_\Sigma = V_\Sigma = U$  and  $S_\Sigma = S^2$ . Computing the eigenvectors this way usually works faster in practice.

## ICA

3. We consider the cocktail party problem given by equation 12 where  $x$  and  $s$  in  $\mathbb{R}^n$  are respectively the mixed and source components.  $A \in \mathbb{R}^{n \times n}$  is the mixing matrix and the goal is to derive  $W = A^{-1}$  using only  $x$ .

$$x = As \quad (12)$$

- In the ICA algorithm, we make the hypothesis that  $p(s) = \prod_i p(s_i)$ . What condition on the  $s$  is this equivalent to?
- Derive the joint distribution of joint distribution of  $x$ .
- We assume the cumulative density function of each  $s_i$  is given by the sigmoid function  $g$ . What properties are always satisfied by a cdf? What is the probability density function of  $s_i$  under this assumption?
- The log-likelihood for a single training example is given by equation 13. Derive the gradient update step for  $W$ . **Hint:**  $\nabla_W |W| = |W|(W^T)^{-1}$ .

$$l(W) = \sum_{j=1}^n \log(g'(w_j^T x)) + \log|W| \quad (13)$$

### Solution:

- The hypothesis is that the source components are independent.
- Using the change of variable formula, we obtain that  $p(x) = \prod_{i=1}^n p_s(w_i^T x) |W|$ .
- $F$  is a cdf if and only if:
  - $F$  is positive;
  - $F$  is right-continuous;
  - $F$  is monotonically non-decreasing;
  - $\lim_{t \rightarrow +\infty} F(t) = 1$  and  $\lim_{t \rightarrow -\infty} F(t) = 0$ .

The probability density function is given by the derivative of the sigmoid function:  $g'(t) = g(t)(1 - g(t))$ .

- The derivation of the gradient is given by:

$$\begin{aligned}
\nabla_W l(W) &= \nabla_W \left( \sum_{j=1}^n \log(g'(w_j^T x)) \right) + \nabla_W \log|W| \\
\nabla_W \log|W| &= |W|(W^T)^{-1} \times \frac{1}{|W|} = (W^T)^{-1} \\
\nabla_{w_i} \log(g'(w_j^T x)) &= \nabla_{w_i} \log(g(w_j^T x)) + \nabla_{w_i} \log(1 - g(w_j^T x)) \\
&= \frac{g(w_j^T x)(1 - g(w_j^T x))}{g(w_j^T x)} x^T - \frac{g(w_j^T x)(1 - g(w_j^T x))}{(1 - g(w_j^T x))} x^T \\
&= 1 - 2g(w_j^T x) x^T \\
\nabla_W l(W) &= \left( \begin{bmatrix} 1 - 2g(w_1^T x) \\ 1 - 2g(w_2^T x) \\ \vdots \\ 1 - 2g(w_n^T x) \end{bmatrix} x^T + (W^T)^{-1} \right)
\end{aligned}$$