Principal Component Analysis

1 How it works

1.1 Just a little more theory to better understand

Let $X \in \mathcal{M}_{n,p}(\mathbb{R})$ be a matrix of our dataset, where n is the number of samples, each composed of p parameters. We want to find a matrix

$$R \in \mathcal{M}_{n,k}(\mathbb{R}), \quad k < p.$$

The first step is to compute $X^\top X$. Indeed, if we take a random direction $w \in \mathbb{R}^p$ such that $\|w\| = 1$, then the data projected on this axis is given by Xw. Because we are looking for the direction that keeps the most information, which means maximizing the variance of the projection, we are basically looking for

$$\begin{aligned} \mathrm{Var}(Xw) &= \mathbb{E}[(Xw)^2] - \mathbb{E}[Xw]^2 \\ &= \frac{1}{n}\|Xw\|^2 \\ &\propto w^\top X^\top X w. \end{aligned}$$

If necessary, we define

$$\tilde{X}_{(i)} = X_{(i)} - \mathrm{mean}(X_{(i)}),$$

so we can assume that $\mathbb{E}[X] = 0$. This is in fact the very first thing to do before computing $X^\top X$.

In order to find our first direction, we need to solve

$$\arg\max_{\|w\|=1} \left\{ w^\top X^\top X w \right\} = \arg\max \left\{ \frac{w^\top X^\top X w}{w^\top w} \right\}.$$

We recognize the

Thus, we found our first direction $w_1 = v_{\max}$. Our intuition tells us that we now have to find the second vector by using the exact same idea, but on a new matrix $\tilde{X}$ where all the information given by w_1 has been removed:

$$\tilde{X}_k = X - \sum_{s=1}^{k-1} X w_s w_s^\top.$$

We end up with the same Rayleigh quotient as before but with $\tilde{X}_k$. It turns out that the k-th vector w_k we are looking for is the eigenvector corresponding to the k-th largest eigenvalue. If $\{\lambda_1, \ldots, \lambda_p\}$ are sorted in decreasing order, then the k vectors we are looking for are the $\{v_1, \ldots, v_k\}$ corresponding to them.

$$R = (Xw_i)_i$$

1.2 The practical implementation

Without entering into details, those eigenvectors can be computed using the QR algorithm. In this implementation, the Gram–Schmidt process is used to compute the QR decomposition. Then, the inverse iteration algorithm is applied to find the corresponding eigenvectors. The inverse iteration algorithm requires matrix inversions, which are performed using Gaussian elimination, even though faster methods exist.