# Principal Component Analysis with Numpy

March 5, 2018

## 1 Principal Component Analysis with Numpy

Principal Component Analysis or PCA is an important method heavily applied in data science. PCA is used either in data visualization, data compression, multivariate outlier detection noise reduction and others.

Alternatives for the computation of PCA with help of Numpy.

Let our data matrix $X$ be of $n \times p$ size, where $n$ is the number of observations and $p$ is the number of features observed.

The covariance of our data is a measure of the extent to which correponding features move in the same direction from our data. We can compute the covariance between two data matrices $X$,$Y$ as:

$$Cov(X, Y) = E[(X - E[X])^\top (Y - E[Y])]$$

The covariance of our data $X$ is

$$\Sigma = Cov(X, X) = E[(X - E[X])^\top (X - E[X])]$$

If our data is mean centered:

$$\Sigma = Cov(X, X) = E[X^\top X] \approx \frac{X^\top X}{n - 1}$$

### 1.1 Solution with eigendecomposition

There are several ways to compute the PCA solution. Maybe the easier is with eigendecomposition of the covariance matrix. You can compute the $P$ matrix of eigenvector which diagonalizes the empirical covariance matrix $\Sigma$

$$P^{-1} \Sigma P = S$$

Where, where $S$ is the diagonal matrix of eigenvalues of $\Sigma$ and $P$ is a matrix ($p \times p$) containing the eigenvectors or principal components. The variance explained by each eigenvalue $P_i$ is:

$$Var_i = \frac{S_{i,i}^2}{\sum_i S_{i,i}^2}$$

By this way, we have decomposed our data with help of $P_k$ orthogonal vectors, so that:

$$X = TP^\top$$

If we only select some of the eigenvectors, $P^*$, we are approximating our data as:

$$X = T^* P^{*\top} + E$$

where T can be computed through projecting our data $X$ onto $P$ or $P^*$:

$$T = XP^\top$$

### 1.2 Work to do

Create your own function to compute PCA from a data matrix $X$. You can make use of the *np.linalg.eig()* function. Generate some multivariate data through *numpy.multivariate_normal*.

### 1.3 Bonus

You are involved in an experiment with measurements of mRNA abundance of 3 genes involved in neurogenesis (TNFa,IL6,TGFb). The research consortium extracted the expression of each gene from 15447 tissue samples. You can find this dataset in the file *neurogenesis.csv*. The main goal is to observe differences on the genes covariability among a large population and check whether they are correlated with some genetic features.

However, the design was flawed as the protocol did not order to store the age of the subject corresponding to each sampled tissue. As age will be the main factor altering the correlation between TNFa,IL6,TGFb, the data will be mainly useless.

When you receive the data, you end up calling the project coordinator telling him that you might have a way to remove the variability explained by age, without knowing each sample's age. Your idea is that, if age is producing the main variability in your data, it would correlate wtih the main component of variance.

### 1.4 Activity

Please consider the following actions on your assignment:

1. Import your data $X$ into Python (use *numpy.genfromtxt()* )
2. Using your implementation, compute a PCA of your input data $P$, plot a 2D scoreplot and check that matches figure 1.
3. Extract the main *principal component* from your model that explains maximum variance (first eigenvector, or *first loading vector*) $\vec{p}_1$ and the associated scores vector $\vec{t}_1$.
4. Create a new dataset $\bar{X}$, free of the of the variance explained by $\vec{t}_1, \vec{p}_1$.
5. Compute a PCA of $\bar{X}$, plot the new 2D scoreplot.
6. Discuss why you know you solved the problem correcty.