ECE544NA: PCA, python + numpy + TensorFlow tutorial

Raymond Yeh

University of Illinois at Urbana Champaign

yeh17@illinois.edu

September 13, 2016

PCA: Least Squares Formulation

ILLINOIS

1. Given a dataset of $\{\vec{x}_1, ...\vec{x}_N\}$, where $\vec{x}_i \in \mathbb{R}^d$. We hope to reduce the dimension of the data to $m < d$, using linear transformations.
2. Let $\{\vec{u}_1, \vec{u}_2 ... \vec{u}_d\}$ be a set of orthonormal vectors, then without loss of generality, we can write, each $\vec{x}_i$ as

$$\vec{x}_i = \sum_{j=1}^{d} z_{ji} \vec{u}_j \tag{1}$$

3. Recall that the definition of orthonormal vectors,

$$\vec{u}_j^\intercal \vec{u}_l = \delta_{jl} \tag{2}$$

where

$$\delta_{jl} = \begin{cases} 0, j \neq l \\ 1, j = l \end{cases} \tag{3}$$

1. Given a $\vec{x}_i$, we can solve for $z_{ji} = \vec{u_j}^\intercal \vec{x}_i$
2. We wish to reduce the dimension of the data to $m < d$, meaning we only use m coefficients z_j, and the remaining $d - m$ coefficients will be replaced with constants b_j, then the reduced dimension vector $\hat{\vec{x}}_i$ can be written as:

$$\hat{\vec{x}}_i = \sum_{j=1}^{M} z_{ji} \vec{u}_j + \sum_{j=m+1}^{d} b_j \vec{u_j} \tag{4}$$

3. Next, we will find the best approximation in the least squares sense. Meaning we minimize the following,

$$L_m = \frac{1}{2} \sum_{i=1}^{N} ||\vec{x}_i - \hat{\vec{x}}_i||^2 \tag{5}$$

PCA: Least Squares Formulation

1. Expand using Equation (4) and orthonormality Equation (2)

$$L_m = \frac{1}{2}\sum_{i=1}^{N}||\vec{x}_i - \hat{\vec{x}}_i||^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=m+1}^{d}(z_{ji} - b_j)^2 \tag{6}$$

2. Set the derivative of L_m with respect to b_i to zero, then

$$b_j = \frac{1}{N}\sum_{i=1}^{N} z_j i = \vec{u_j}^\intercal(\frac{1}{N}\sum_{i=1}^{N}\vec{x_i}) = \vec{u_j}^\intercal\bar{\vec{x}} \tag{7}$$

3. Substitute b_j back in to Equation (5)

$$L_m = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=m+1}^{d}(\vec{u_j}^\intercal(\vec{x_i} - \bar{\vec{x}}))^2 \tag{8}$$

PCA: Least Squares Formulation

ILLINOIS

1. Some rewriting:

$$L_m = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=m+1}^{d}(\vec{u_j}^{\intercal}(\vec{x_i} - \bar{\vec{x}}))^2 \tag{9}$$

$$= \frac{1}{2}\sum_{j=m+1}^{d}\sum_{i=1}^{N}(\vec{u_j}^{\intercal}(\vec{x_i} - \bar{\vec{x}}))(\vec{u_j}^{\intercal}(\vec{x_i} - \bar{\vec{x}}))^{\intercal} \tag{10}$$

$$= \frac{1}{2}\sum_{j=m+1}^{d}\vec{u_j}^{\intercal}(\sum_{i=1}^{N}((\vec{x_i} - \bar{\vec{x}}))((\vec{x_i} - \bar{\vec{x}}))^{\intercal})\vec{u_j} \tag{11}$$

$$= \frac{1}{2}\sum_{j=m+1}^{d}\vec{u_j}^{\intercal}\Sigma\vec{u_j} \tag{12}$$

1. Lastly, we will need to find the set of $\vec{u}_j$ that minimize L_m. Minimum occurs when the basis satisfy,

$$\Sigma \vec{u}_j = \lambda_j \vec{u}_j \tag{13}$$

,where $\Sigma = \sum_{i=1}^{N} ((\vec{x}_i - \bar{\vec{x}}))((\vec{x}_i - \bar{\vec{x}}))^\intercal)$ is the covariance matrix of the data set $\{\vec{x}_i\}$.

2. Using the Equation (13) and L_m in Equation (12). The minimum is at

$$L_m = \sum_{j=m+1}^{d} \vec{u}_j^\intercal \Sigma \vec{u}_j = \sum_{j=m+1}^{d} \vec{u}_j^\intercal \lambda_j \vec{u}_j = \sum_{j=m+1}^{d} \lambda_j \tag{14}$$

3. Therefore, to minimize the L_m, we should remove dimension where $\vec{u}_j$ has the smallest λ_j.

1. Let $X = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_N]$
2. Compute $\bar{\vec{x}} = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$
3. Compute $\Sigma = (X - \bar{\vec{x}})(X - \bar{\vec{x}})^\intercal$
4. Compute Eigenvectors and Eigenvalues of Σ, keep the m dimensions with the largest Eigenvalues.
5. Lastly, to project examples on to the lower dimensional space, $\vec{z} = U^\intercal X$, where $U = [\vec{u_1}, \vec{u_2}, ..., \vec{u_m}]$, and $\vec{u}_j$ are the Eigenvectors.

1. PCA involves computing $XX^{\intercal}$, X has the dimension of $d \times N$. When $d >> N$, then compute Eigenvalues and Eigenvectors on $X^{\intercal}X$ instead to save computation.
2. Claim: If $\vec{v}$ is Eigenvector of $X^{\intercal}X$, then $\vec{u} = X\vec{v}$ is the Eigenvector of $XX^{\intercal}$.
3. Proof:

$$X^{\intercal}X\vec{v} = \lambda\vec{v} \tag{15}$$

$$X(X^{\intercal}X\vec{v}) = X(\lambda\vec{v}) \tag{16}$$

$$(XX^{\intercal})(X\vec{v})) = \lambda(X\vec{v}) \tag{17}$$

$$(XX^{\intercal})(\vec{u})) = \lambda(\vec{u}) \tag{18}$$

Ipython Notebook will be posted on the website.