Dimension Reduction

Sunmook Choi felixchoi@korea.ac.kr

August 24, 2023

1 Principal Component Analysis

1.1 Prerequisites

Definition 1 (Orthogonal Complement)**.** *Let V be a vector space and U be its subspace. The orthogonal complement of U is the set*

$$U^{\perp} = \{\boldsymbol{v} \in V : \langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0,\ \forall \boldsymbol{u} \in U\}. \tag{1.1}$$

Then, the set $U^{\perp}$ is a subspace of V.

Definition 2 (Direct Sum)**.** *Let U_1, U_2 be subspaces of V. For each $\boldsymbol{v} \in V$, if there exist $\boldsymbol{u}_1 \in U_1$ and $\boldsymbol{u}_2 \in U_2$ uniquely such that $\boldsymbol{v} = \boldsymbol{u}_1 + \boldsymbol{u}_2$, then V is the direct sum of U_1 and U_2, and we write $V = U_1 \oplus U_2$.*

Theorem 3. *If U is a subspace of $\mathbb{R}^n$, then the following hold.*

- $\dim(U) + \dim(U^{\perp}) = n$
- $\mathbb{R}^n = U \oplus U^{\perp}$ *(orthogonal decomposition of $\mathbb{R}^n$)*
- $(U^{\perp})^{\perp} = U$

Definition 4 (Projection)**.** *Let V be a vector space. If $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_k\}$ is an orthonormal basis for the subspace U of V, then the orthogonal projection of $\boldsymbol{v} \in V$ onto U is the vector*

$$proj_U \boldsymbol{v} = \langle \boldsymbol{v}, \boldsymbol{u}_1 \rangle \boldsymbol{u}_1 + \cdots + \langle \boldsymbol{v}, \boldsymbol{u}_k \rangle \boldsymbol{u}_k. \tag{1.2}$$

Definition 5 (Orthogonal Decomposition)**.** *Let V be a vector space with $dim(V) = n$ and let U be its subspace. Let $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_k\}$ and $\{\boldsymbol{u}_{k+1}, \dots, \boldsymbol{u}_n\}$ be orthonormal bases for U and*

$U^\perp$, respectively. Then, the orthogonal decomposition of $\boldsymbol{v} \in V$ is

$$\boldsymbol{v} = \sum_{i=1}^{k} a_i \, \boldsymbol{u}_i + \sum_{i=k+1}^{n} b_i \, \boldsymbol{u}_i \ \in U \oplus U^\perp \tag{1.3}$$

where $a_i = \langle \boldsymbol{v}, \boldsymbol{u}_i \rangle$ for $i = 1, \ldots, k$ and $b_i = \langle \boldsymbol{v}, \boldsymbol{u}_i \rangle$ for $i = k+1, \ldots, n$. Here, $\sum_{i=1}^{k} a_i \, \boldsymbol{u}_i$ and $\sum_{i=k+1}^{n} b_i \, \boldsymbol{u}_i$ are the orthogonal projections onto U and $U^\perp$, respectively.

Definition 6 (Variance and Covariance)**.** *For a random variable X, the variance of X is defined to be*

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2. \tag{1.4}$$

For two random variables S and Y, the covariance of X and Y is defined to be

$$Cov(X, Y) = \mathbb{E}[(X - \mathbb{E}(X))(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \tag{1.5}$$

For a random vector $X = (X_1, \ldots, X_n)^\top$, the covariance matrix Σ is defined whose (i, j)-entry is defined to be

$$\Sigma_{ij} = \Sigma_{ji} = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]. \tag{1.6}$$

Definition 7 (Sample Covariance Matrix)**.** *Let $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$ be a dataset, that is, let $\boldsymbol{x}_i \in \mathbb{R}^d$ be a sample data for $i = 1, \ldots, n$. Let $\bar{\boldsymbol{x}} = (\bar{x}_1, \ldots, \bar{x}_d)$ be a sample mean vector of X such that $\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} X_{ij}$. Then the sample covariance matrix $\mathcal{S}$ is defined to be*

$$\mathcal{S} = \frac{1}{n-1} (X - \bar{\boldsymbol{x}})^\top (X - \bar{\boldsymbol{x}}) \in \mathbb{R}^{d \times d}. \tag{1.7}$$

If the population mean of X_j is known as μ_j, then the sample covariance is defined to be

$$\mathcal{S} = \frac{1}{n} (X - \boldsymbol{\mu})^\top (X - \boldsymbol{\mu}) \in \mathbb{R}^{d \times d} \tag{1.8}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$.

Remark 8. The denominator $n - 1$ in Eq. 7 is due to Bessel's correction, which makes the estimator unbiased. A sample covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ is positive semi-definite, that is, $\boldsymbol{z}^\top \Sigma \boldsymbol{z} \geq 0$ for any $\boldsymbol{z} \in \mathbb{R}^d$.

1.2 Principal Components

For any sample covariance matrix $\mathcal{S} \in \mathbb{R}^{d \times d}$, it is positive semi-definite and real symmetric. Positive semi-definiteness implies that its eigenvalues are nonnegative, and real symmetry implies that it is orthogonally diagonalizable.

Let $\lambda_1, \ldots, \lambda_d$ be eigenvalues of $\mathcal{S}$ such that $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$, and let $\boldsymbol{v}_i$'s be orthonormal eigenvectors corresponding to λ_i's for $i = 1, \ldots, d$. Here, we call the eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ the top k principal components of X where $k \leq d$.

1.2.1 Problem Setup

Given a dataset $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$ and a positive integer $k \leq d$, find the best linear projection $\tilde{X}$ of X onto a lower dimensional subspace U of $\mathbb{R}^d$, $\dim(U) = k$. Here, the 'best' linear projection $\tilde{X} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n]^\top$ is the linear projection which minimizes

$$\frac{1}{n} \sum_{i=1}^{n} \| \tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i \|^2. \tag{1.9}$$

The compressed/encoded data can be expressed in two ways: $\boldsymbol{z_i} \in U$ and $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^d$ which lie in different dimensional spaces. In this sense, $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^d$ is called a reconstructed data. Note that PCA can be also considered finding $\tilde{X}$ which has the smallest reconstruction loss. We additionally assume that the population mean of each dimension of data is zero. Then, the sample covariance $\mathcal{S}$ of X will be $\mathcal{S} = \frac{1}{n} X^\top X = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i^\top \boldsymbol{x}_i$.

1.2.2 Projection Perspective

Claim 1. *The vector $\tilde{\boldsymbol{x}}_i$ is the projection of $\boldsymbol{x}_i$ onto the subspace spanned by top k principal components of X.*

Proof. Let $\mathcal{B} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ be a set of orthonormal vectors and U be the spanned subspace. The projection $\tilde{\boldsymbol{x}}_i$ of $\boldsymbol{x}_i$ onto U is

$$\tilde{\boldsymbol{x}}_i = \sum_{j=1}^{k} (\boldsymbol{x}_i \boldsymbol{v}_j^\top) \boldsymbol{v}_j, \tag{1.10}$$

and we want to find

$$\mathcal{B}^* = \arg\min_{\mathcal{B}} \frac{1}{n} \sum_{i=1}^{n} \|\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i\|^2. \tag{1.11}$$

By simple computation, we have the following.

$$\begin{aligned} \|\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i\|^2 &= \|\tilde{\boldsymbol{x}}_i\|^2 - 2\tilde{\boldsymbol{x}}_i \boldsymbol{x}_i^\top + \|\boldsymbol{x}_i\|^2 && (1.12) \\ &= \sum_{j=1}^{k} (\boldsymbol{x}_i \boldsymbol{v}_j^\top)^2 - 2 \sum_{j=1}^{k} (\boldsymbol{x}_i \boldsymbol{v}_j^\top) \boldsymbol{v}_j \boldsymbol{x}_i^\top + \| \boldsymbol{x}_i\|^2 && (\because \mathcal{B} \text{ is orthonormal}) \\ &= \| \boldsymbol{x}_i\|^2 - \sum_{j=1}^{k} (\boldsymbol{x}_i \boldsymbol{v}_j^\top)^2 && (1.13) \end{aligned}$$

Therefore, it is equivalent to find

$$\mathcal{B}^* = \arg\max_{\mathcal{B}} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} (\boldsymbol{x}_i \boldsymbol{v}_j^\top)^2. \tag{1.14}$$

Since $\boldsymbol{x}_i\boldsymbol{v}_j^\top$ is a scalar, we know that $\boldsymbol{x}_i\boldsymbol{v}_j^\top = \boldsymbol{v}_j\boldsymbol{x}_i^\top$. Then we have

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}(\boldsymbol{x}_i\boldsymbol{v}_j^\top)^2 = \frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{n}(\boldsymbol{v}_j\boldsymbol{x}_i^\top)(\boldsymbol{x}_i\boldsymbol{v}_j^\top) = \sum_{j=1}^{k}\boldsymbol{v}_j\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^\top\boldsymbol{x}_i\right)\boldsymbol{v}_j^\top = \sum_{j=1}^{k}\boldsymbol{v}_j\mathcal{S}\boldsymbol{v}_j^\top. \tag{1.15}$$

Now, we are into an optimization problem of finding

$$\mathcal{B}^* = \arg\max_{\mathcal{B}}\sum_{j=1}^{k}\boldsymbol{v}_j\mathcal{S}\boldsymbol{v}_j^\top \tag{1.16}$$

where $\mathcal{B}$ is orthonormal. In order to find a solution, we solve the following Lagrangian:

$$\mathcal{L}(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k,\lambda_1,\ldots,\lambda_k) = \sum_{j=1}^{k}\Big[\boldsymbol{v}_j\mathcal{S}\boldsymbol{v}_j^\top - \lambda_j(\boldsymbol{v}_j\boldsymbol{v}_j^\top - 1)\Big] \tag{1.17}$$

Recall that if $y = \boldsymbol{x}A\boldsymbol{x}^\top$ and $z = \boldsymbol{x}\boldsymbol{x}^\top$, then $\frac{dy}{d\boldsymbol{x}} = 2\boldsymbol{x}A$ and $\frac{dz}{d\boldsymbol{x}} = 2\boldsymbol{x}$ for $\boldsymbol{x}\in\mathbb{R}^d$ and $A\in\mathbb{R}^{d\times d}$.

$$\frac{\partial\mathcal{L}}{\partial\lambda_j} = \boldsymbol{v}_j\boldsymbol{v}_j^\top - 1 = 0 \quad\Rightarrow\quad \boldsymbol{v}_j\boldsymbol{v}_j^\top = \|\boldsymbol{v}_j\| = 1 \tag{1.18}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{v}_j} = 2\boldsymbol{v}_j\mathcal{S} - 2\lambda_j\boldsymbol{v}_j = 0 \quad\Rightarrow\quad \boldsymbol{v}_j\,\mathcal{S} = \lambda_j\,\boldsymbol{v}_j \quad\text{and}\quad \boldsymbol{v}_j\,\mathcal{S}\,\boldsymbol{v}_j^\top = \lambda_j \tag{1.19}$$

Thus, $\boldsymbol{v}_j$ is the eigenvector of $\mathcal{S}$ corresponding to the j th largest eigenvalue, that is, it is the top j th principal component of X. □

We can also prove the claim by using mathematical induction.

- Prove that it holds for $k = 1$.
- Assuming that it holds for $k-1$, prove that it also holds for k. That is, if the top $k-1$ principal components $\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{k-1}\}$ maximizes $\sum_{j=1}^{k-1}\boldsymbol{v}_j\mathcal{S}\boldsymbol{v}_j^\top$, then $\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{k-1},\boldsymbol{v}_k\}$ also maximizes $\sum_{j=1}^{k}\boldsymbol{v}_j\mathcal{S}\boldsymbol{v}_j^\top$.
- Notice that the choice of k does not affect the resulting set of vectors $\boldsymbol{v}_j$.
- The low-dimension k only decides how many vectors of $\boldsymbol{v}_j's$ we use to make a linear projection.

We have concluded that PCA minimizes the reconstruction loss $\frac{1}{n}\sum_{i=1}^{n}\|\,\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i\,\|^2$. Let's take a look at what the minimized loss will be. Let $\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_d\}$ be the all principal components of X, that is, $\boldsymbol{x}_i = \sum_{j=1}^{d}(\boldsymbol{x}_i\boldsymbol{v}_j^\top)\,\boldsymbol{v}_j$. Then, for the projection $\tilde{\boldsymbol{x}}_i$ onto U, we have $\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i = \sum_{j=k+1}^{d}(\boldsymbol{x}_i\boldsymbol{v}_j^\top)\boldsymbol{v}_j \in U^\perp$ for all $i = 1,\ldots,n$. Thus the reconstruction loss is

$$\frac{1}{n}\sum_{i=1}^{n}\|\,\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i\,\|^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=k+1}^{d}(\boldsymbol{x}_i\boldsymbol{v}_j^\top)^2 = \sum_{j=k+1}^{d}\boldsymbol{v}_j\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^\top\boldsymbol{x}_i\right)\boldsymbol{v}_j^\top = \sum_{j=k+1}^{d}\lambda_j. \tag{1.20}$$

That is, the loss is the sum of the eigenvalues corresponding to the remaining principal components of X.

1.2.3 Maximum Variance Perspective

"Retaining most information after data compression is equivalent to capturing the largest amount of variance in the low-dimensional code." (Harold Hotelling, 1933)

Fix $k \leq d$. Let $\tilde{X} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n]^\top$ be the "best" linear projection of X in the projection perspective, that is, $\tilde{\boldsymbol{x}}_i = \sum_{j=1}^{k} (\boldsymbol{x}_i \boldsymbol{v}_j^\top) \boldsymbol{v}_j$ where $\boldsymbol{v}_j$ is the top j th principal component of X.

Claim 2. *PCA finds the subspace that maximizes the variance of the projected data, that is, $\tilde{X}$ can be obtained by a basis $\mathcal{B} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ that maximizes the variance V_j of $\{\boldsymbol{x}_1 \boldsymbol{v}_j^\top, \ldots, \boldsymbol{x}_n \boldsymbol{v}_j^\top\}$ with $V_1 \geq V_2 \geq \cdots \geq V_k$.*

Proof. Assuming that $\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{v}_j^\top] = 0$, we have $Var_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{v}_j^\top] = \mathbb{E}_{\boldsymbol{x}}[(\boldsymbol{x}\boldsymbol{v}_j^\top)^2] - \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{v}_j^\top]^2 = \mathbb{E}_{\boldsymbol{x}}[(\boldsymbol{x}\boldsymbol{v}_j^\top)^2]$.

$$\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i \boldsymbol{v}_j^\top)^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{v}_j \boldsymbol{x}_i^\top)(\boldsymbol{x}_j \boldsymbol{v}_j^\top) = \frac{1}{n}\boldsymbol{v}_j \left(\sum_{i=1}^{n} \boldsymbol{x}_i^\top \boldsymbol{x}_i\right) \boldsymbol{v}_j^\top = \boldsymbol{v}_j \mathcal{S} \boldsymbol{v}_j^\top \tag{1.21}$$

The sample variance of X is equal to the objective in the projection perspective, which completes the proof. □

The maximum variance that PCA can capture/explain is $\sum_{j=1}^{k} \lambda_j$, and the lost variance by data compression via PCA is $\sum_{j=k+1}^{d} \lambda_j$.

2 Autoencoder

Autoencoder (AE) is an encoder-decoder network to find a latent space of given dataset. It tries to find latent vectors by reducing the dimension of data and by restoring the original data from them. The natural bottleneck network forces the encoder network to compress information into latent vectors. Additionally, the encoder network should find features of the original data so that the decoder network can reconstruct the original ones from the latent vectors. AE has three important attributes: dimensionality reduction, data-specific, and lossy.

(i) **Dimensionality Reduction**: The encoder compresses an input into a low dimensional latent vector and the decoder reconstructs the input from this latent vector as an output.

(ii) **Data Specific**: To find well-trained encoder-decoder network which compresses data meaningfully, data should have strong correlations between input features, and/or data should be similar to what they have been trained on.

(iii) **Lossy**: Since the dimension of latent vector is smaller than the original input, the reconstructed input is bound to have lower quality than the original input.

Optimizing the network can be considered as an unsupervised learning since it does not need any label for input. The training algorithm of AE is described as below.

Algorithm 1 Training Autoencoder

Input: Data batches $\mathcal{D} = \{\mathcal{B}_j\}_{j=1}^{B}$, $\mathcal{B}_j = \{x_i^{(j)}\}_i$, encoder Ψ, and decoder Φ

for each batch $j = 1, \ldots, B$ **do**
  Initialize $L = 0$ ▷ L for accumulation of loss
  for each data $x_i^{(j)} \in \mathcal{B}_j$ **do**
    Compute $\tilde{x}_i^{(j)} = \Phi(\Psi(x_i^{(j)}))$ and the loss $\mathcal{L}(x_i^{(j)}, \tilde{x}_i^{(j)})$
    $L = L + \frac{1}{B}\mathcal{L}(x_i^{(j)}, \tilde{x}_i^{(j)})$
  end for
  Take a gradient descent step on $\nabla_{\boldsymbol{\theta}} L$ ▷ $\boldsymbol{\theta}$: the set of parameters in Ψ and Φ
end for

Dimensionality reduction methods can be used for data visualization. In PCA, it attempts to find a lower dimensional hyperplane which maximizes the variance of original data. On the other hand, since AE is defined as nonlinear neural networks, it is capable of learning a nonlinear manifold describing the data in a lower dimensionality.

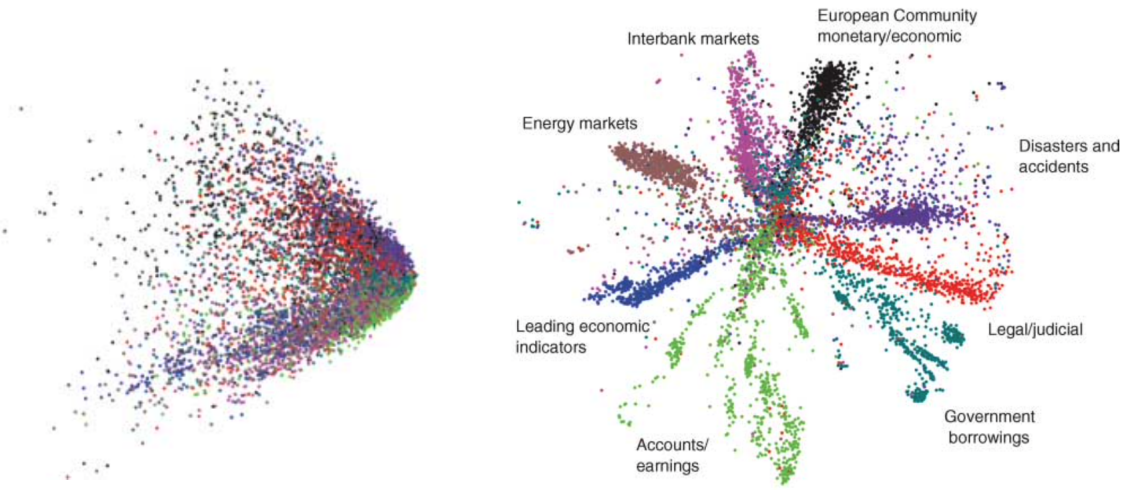

Figure 2.1: The left shows the two-dimensional codes produced by LSA (a well-known document retrieval method based on PCA), while the right shows the codes produced by an autoencoder.

Figure 2 shows that autoencoder clearly learns a nonlinear manifold, which can provide clear visualization.

2.1 Denoising Autoencoder

References

[1] G.E. Hinton and R. R. Salakhutdinov, *Reducing the Dimensionality of Data with Neural Networks*, Science (2006).