ECE 586 Application: Principal Component Analysis (PCA)

Henry D. Pfister
Duke University

October 12, 2022

1 A Few Questions

1.1 What affine subspace best approximates a given set of points in $\mathbb{R}^n$?

For a given set of N data points $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N \in \mathbb{R}^n$, what p-dimensional affine subspace $W \subset \mathbb{R}^n$ best approximates these points in the sense that the error

$$\frac{1}{N}\sum_{i=1}^{N} \|\underline{x}_i - P_W(\underline{x}_i)\|^2$$

is minimized. We refer to this as the *empirical PCA problem*. Recall that an *affine subspace* $W = \text{span}\{\underline{w}_1, \ldots, \underline{w}_p\} + \underline{w}_0$ is defined by linearly independent vectors $\underline{w}_0, \ldots, \underline{w}_p$ where adding $\underline{w}_0$ has the effect of translating the subspace $\text{span}\{\underline{w}_1, \ldots, \underline{w}_p\}$. For such an affine subspace, we define $B \in \mathbb{R}^{n\times p}$ by $B = [\underline{w}_1, \ldots, \underline{w}_p]$ and the projection onto W is defined by $P_W(\underline{x}) = B(B^TB)^{-1}B^T(\underline{x} - \underline{w}_0) + \underline{w}_0$. Also, choosing $\underline{w}_0 = \frac{1}{N}\sum_{i=1}^{N} \underline{x}_i$ is optimal because this corresponds to first removing the mean and then solving the problem for a set of zero-mean vectors.

Let $A = [\underline{x}_1 - \underline{w}_0, \ldots, \underline{x}_N - \underline{w}_0]$ be the mean-corrected data matrix whose columns are the mean-corrected data points. Then, the solution to this problem can be computed using the SVD $A = U\Sigma V^T$ of A. In particular, we can choose $B = U_p \triangleq [\underline{u}_1, \ldots, \underline{u}_p]$ to be the first p columns of U so that $P_W(\underline{x}) = U_pU_p^T(\underline{x} - \underline{w}_0) + \underline{w}_0$. When used for dimension reduction, the idea is to store $\underline{y}_i = U_p^T\underline{x}_i \in \mathbb{R}^p$ instead of $\underline{x}_i \in \mathbb{R}^n$ and recreate the approximation $\hat{\underline{x}}_i = U_p\underline{y}_i + \underline{w_0}$ whenever it is needed.

1.2 What is the best p-variable approximation of n random variables?

Let $\underline{X} = (X_1, X_2, \ldots, X_n)^T$ be a vector of n real random variables and let $U_p = [\underline{u}_1, \ldots, \underline{u}_p]$ be a matrix with orthogonal columns whose i-th column is $\underline{u}_i \in \mathbb{R}^n$. Consider the linear transformation to $\underline{Y} = (Y_1, Y_2, \ldots, Y_p)^T$ defined by

$$Y_i = [U_p^T\underline{X}]_i = \sum_{j=1}^{n} u_{j,i}X_j.$$

For the random vector $\underline{Y}$, one can show that the covariance is given by

$$\text{Cov}(Y_i, Y_k) = \sum_{j=1}^{n}\sum_{l=1}^{n} u_{j,i}\text{Cov}(X_j, X_l)u_{l,k} = \underline{u}_i^TK\underline{u}_k,$$

where $\text{Cov}(X, Y) \triangleq \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$ and $[K]_{j,l} = \text{Cov}(X_j, X_l)$. For $p < n$, the key question is "How can we choose $\underline{u}_1, \ldots, \underline{u}_p$ to maximize the variance in $X_1, X_2, \ldots, X_n$ that is explained by $Y_1, \ldots, Y_p$?". We refer to this as the *probabilistic PCA problem*.

Mathematically, this question can be formalized by the sequential computation, for $i = 1, \ldots, p$, of the vectors

$$\begin{aligned}\underline{u}_i &= \arg\max_{\underline{u}\in\mathbb{R}^n:\|\underline{u}\|=1} \text{Var}(Y_i) \quad \text{subject to } \text{Cov}(Y_i, Y_k) = 0\ \forall k \in \{1, \ldots, i-1\}\\ &= \arg\max_{\underline{u}\in\mathbb{R}^n:\|\underline{u}\|=1} \underline{u}^TK\underline{u} \quad \text{subject to } \underline{u}^TK\underline{u}_k = 0\ \forall k \in \{1, \ldots, i-1\}.\end{aligned}$$

Since K is symmetric and symmetric matrices have a complete set of orthogonal eigenvectors, the second line is solved by choosing $\underline{u}_i$ to be the i-th normalized eigenvector of K (i.e., associated with the i-th largest eigenvalue). Thus, the implied transformation (i.e., dimension reduction) is given by $\underline{Y} = U_p^T \underline{X}$.

1.3 How are these two problems related?

While these two problems may seem quite different (e.g., one is deterministic and the other is probabilistic), they are actually quite similar. To see this, we observe that each vector $\underline{x}_i$ in the first problem can be seen as a sample of the random vector $\underline{X} = (X_1, \ldots, X_n)^T$ in the second problem. Then, focusing on the second problem, we can assume the distribution of the random vector $\underline{X}$ is the empirical distribution defined by the data set in the first problem. Using this interpretation, the empirical covariance matrix of the data is given by

$$\hat{K} = \frac{1}{N} \sum_{i=1}^{N} \underline{x}_i \underline{x}_i^T = \frac{1}{N} A A^T,$$

where A is the mean-corrected data matrix from the first problem. Using the SVD $A = U \Sigma V^T$, it follows that

$$\hat{K} = \frac{1}{N} U \Sigma V^T V \Sigma^T U^T = U \left(\frac{1}{N} \Sigma^2 \right) U^T.$$

From this, we see that the columns of $U = [\underline{u}_1, \ldots, \underline{u}_n]$ define a set n orthogonal eigenvectors for $\hat{K}$ where $\hat{K} \underline{u}_i = \left(\frac{1}{N} \sigma_i^2 \right) \underline{u}_i$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. Thus, if we solve the second problem by finding the p eigenvectors of $\hat{K}$ with largest eigenvalues, then the desired dimension reduction is given by $\underline{Y} = U_p^T \underline{X}$ where $U_p \triangleq [\underline{u}_1, \ldots, \underline{u}_p]$. But, based on this, the U_p matrix from the second problem is identical to the U_p matrix computed in the first problem! Finally, by treating $\underline{x}_i$ as a realization of $\underline{X}$ and applying the dimension reduction, one gets a sample $\underline{y}_i = U_p^T \underline{x}_i$ of the random vector $\underline{Y}$.

1.4 How does this compare with random projections?

Instead of carefully designing the subspace used to approximate data vectors, one can simply project the data vectors onto a randomly chosen subspace. This method is called *random projections* and it proceeds by choosing a random matrix $P \in \mathbb{R}^{n \times p}$ and compressing $\underline{x}_i$ to $\underline{y}_i = P^T \underline{x}_i$. Classification can be done by using only the $\underline{y}_i$'s. For schemes that need $\underline{x}_i$, we can use the projection $\hat{\underline{x}}_i = P(P^T P)^{-1} \underline{y}_i + \underline{w_0}$ to approximate it. This computation is often simplified because $P^T P \approx I$ (e.g., if entries of P are independent random variables with mean 0 and variance $\frac{1}{n}$). Two popular examples are where each entry is a standard Gaussian scaled by $\frac{1}{\sqrt{n}}$ and where each entry takes the values $\frac{1}{\sqrt{n}}, -\frac{1}{\sqrt{n}}$ with equal probability. The idea of random projections is approach is typically motivated by the Johnson-Lindenstrauss Lemma which states that compression with a random matrix introduces negligible errors in the pairwise Euclidean distances between all data points as long as p grows faster than $\ln N$.

1.5 Connection to Low-Rank Approximation

The SVD and PCA also have a close connection to low-rank approximation of matrices in the Frobenius norm. For a matrix $A \in \mathbb{C}^{m \times n}$, the Frobenius norm is defined by

$$\|A\|_F \triangleq \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{i,j}|^2}$$

and it is induced by the Hilbert-Schmidt inner product $\langle A | B \rangle \triangleq \mathrm{Tr}(B^H A)$ on $\mathbb{C}^{m \times n}$.

Definition 1. Let $A \in \mathbb{C}^{m \times n}$ have SVD $A = U \Sigma V^H$ where $\underline{u}_1, \ldots, \underline{u}_m$ and $\underline{v}_1, \ldots, \underline{v}_n$ are the columns of U and V. Then, the k-truncated SVD expansion of A is given by

$$\mathcal{T}_k(A) \triangleq \sum_{i=1}^{k} \sigma_i \, \underline{u}_i \, \underline{v}_i^H.$$

Since the ordering of equal singular values is not specified, we note that $\mathcal{T}_k(A)$ is uniquely defined if and only if $\sigma_{k+1} < \sigma_k$.

Theorem 1. *In terms of the Frobenius norm, the best rank-k approximation of $A \in \mathbb{C}^{m\times n}$ is characterized by the k-truncated SVD expansion. In particular, the minimum error is given by*

$$\min_{B\in\mathbb{C}^{m\times n}:\, rank(B)=k} \|A - B\|_F = \|A - \mathcal{T}_k(A)\|_F$$

and it is achieved by $\mathcal{T}_k(A)$ even when it is not unique.

2 Exercises

The following exercises highlight PCA as an application of linear algebra for dimension reduction.

Exercise 1. (15 pt) Use the SVD to numerically compute the p-dimensional PCA of the data matrix

$$D = \begin{bmatrix} 4 & 1 & -2 \\ -3 & 4 & -1 \\ -4 & 0 & 1 \\ 2 & -2 & 3 \end{bmatrix}.$$

For each $p \in \{0, 1, 2\}$, print the resulting post-PCA data matrix with the mean added back in and rounded to 2 decimals digits. Hint: Make sure to compute the mean along the correct dimension (e.g., the data vectors have dimension 4).

Exercise 2. (25 pt) Use the steps outlined in Section 1.1 to solve the empirical PCA problem for the MNIST digit 2. Plot, as small MNIST images, the first 30 reconstruction vectors (i.e., the first 30 columns of U in the SVD) with and without mean removal. What do you observe about these images? How many of these look a lot like the digit 2? Hint: The first few should look like a 2 but not all of them.

Exercise 3. (30 pt) Do the following (separately) for each of the 10 digits in the given MNIST dataset:

1. Use the given samples to solve the empirical PCA problem *with* mean removal as outlined in Section 1.1.
2. Use the given samples to solve the empirical PCA problem *without* mean removal (i.e., set $\underline{w}_0 = \underline{0}$) using the steps in Section 1.1.
3. Plot the first 12 image samples of this digit in three rows:
   (a) the original image
   (b) the image after projection onto $p = 3$ dimensions (with mean removal)
   (c) the image after projection onto $p = 4$ dimensions (without mean removal)
4. Plot samples from the 9 other classes in three rows. If you are currently working on the empirical PCA problem for class j, please plot:
   (a) the original images (for classes $i \neq j$)
   (b) the images after projection onto $p = 3$ dimensions (classes $i \neq j$ projected onto $p = 3$ dimensions from class j, with mean removal)
      i. note: when adding back in the mean after projection, add back in the mean from class i.
   (c) the images after projection onto $p = 4$ dimensions (classes $i \neq j$ projected onto $p = 4$ dimensions from class j, without mean removal)
5. For the case without mean removal, plot the magnitude of the first 12 singular values in decreasing order using a logarithmic scale for the y-axis.

Exercise 4. (20 pt) Repeat Exercise 6 of the Least Squares project after using PCA to reduce the dimension of the whole data set. Do this 4 times using $p = 20, 40, 70, 100$ dimensions. How do the error rates compare to the results from the Least Squares project. Hint: When using `nunmpy.linalg.svd`, make sure to use the correct matrix for projection (e.g., the columns of `u` if called with A or the rows of `vh` if called with A^T).

Exercise 5. (20 pt) Repeat Exercise 6 of the Least Squares project instead using random projections (e.g., with a Gaussian matrix) to reduce the dimension of the whole data set. Do this 4 times using $p = 20, 40, 70, 100$ dimensions. How do the error rates compare to the Exercise 4 and the results from the Least Squares project.