Lecture 13

MFDS Team

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Introduction

- In the previous lecture, we discussed dimensionality reduction using PCA
- In this lecture, we will see the use of linear algebra in practical implmentation of PCA.
- We will also study the challenges encountered when PCA is used in problems of larger dimensions.
- Finally, we elaborate the key steps of PCA in practice.

Revision of PCA problem

- We derived the matrix $\mathbf{B}$ used in generation of lower-dimensional representation $\boldsymbol{z}$ and the compressed data $\tilde{\boldsymbol{x}}$.
- The data covariance matrix $\mathbf{S}$ was used to derive $\mathbf{B}$
- Recall that linear relationship connecting the original data $\boldsymbol{x}$, its low-dimensional code $\boldsymbol{z}$ and the compressed data $\tilde{\boldsymbol{x}}$: $\boldsymbol{z} = \boldsymbol{B}^T\boldsymbol{x}$, and $\tilde{\boldsymbol{x}} = \boldsymbol{B}\boldsymbol{z}$.

Eigenvector Computation and Low Rank Approximation

- In the previous sections, we obtained the basis of the principal subspace as the eigenvectors that are associated with the largest eigenvalues of the data covariance matrix.

$$S = \frac{1}{N} \sum_{i=1}^{N} x_n x_n^T \tag{1}$$

- Equivalently we get

$$S = \frac{1}{N} X X^T \tag{2}$$

$$X = [x_1, ..., x_N] \in \mathbb{R}^{D \times N} \tag{3}$$

- Note that X is a $D \times N$ matrix,

Eigenvector Computation and Low Rank Approximation

- To get the eigenvalues and the corresponding eigenvectors of S, we can follow two approaches
- We can perform an eigendecomposition and compute the eigenvalues and eigenvectors of S directly.
- We can also use a singular value decomposition. Since S is symmetric and factorizes into XX^T, the eigenvalues of S are the squared singular values of X.
- Assume the SVD of X as $X = U\Sigma V^T$. Then

$$S = \frac{1}{N}XX^T = \frac{1}{N}U\Sigma\Sigma^T U^T$$

Eigenvector Computation and Low Rank Approximation

- The columns of U are the eigenvectors of S.
- The eigenvalues λ_d of S are related to the singular values of X via

$$\lambda_d = \frac{\sigma_d^2}{N} \tag{4}$$

- This relationship between the eigenvalues of S and the singular values of X provides the connection between the maximum variance view and the singular value decomposition.

PCA Using Low-Rank Matrix Approximations

- To maximize the variance of the projected data PCA chooses the columns of U to be the eigenvectors that are associated with the M largest eigenvalues of the data covariance matrix S
- The Eckart-Young theoremoffers a direct way to estimate the low-dimensional representation.
- Consider the best rank-M approximation of X defined as $\tilde{X}_M$

$$\tilde{X}_M = \text{argmin}_{rank(A)<=M} \|X - A\|_2 \tag{5}$$

PCA Using Low-Rank Matrix Approximations

- The Eckart-Young theorem states that the best rank M approximation $\tilde{X}_M$ is given by truncating the SVD at the top-M singular value.

$$\tilde{X}_M = U_M \Sigma_M V_M^T$$

- U_M is an orthogonal matrix
- V_M is an orthogonal matrix
- Σ_M has M largest singular values of X as diagonal entries

Practical Aspects of PCA

- Finding eigenvalues and eigenvectors is also important in other fundamental machine learning methods that require matrix decompositions
- In theory we can solve for the eigenvalues as roots of the characteristic polynomial.
- However for matrices larger than 4 by 4 this is not possible because we would need to find the roots of polynomial of degree 5 or higher.

Practical Aspects of PCA

- However the Abel-Ruffini theorem states that there exists no algebraic solution to this problem for polynomials of degree 5 or more.
- Therefore, in practice, solve for eigenvalues or singular values using iterative methods, which are implemented in all modern packages for linear algebra
- In many applications we only require a few eigenvectors.

Practical Aspects of PCA

- It would be wasteful to compute the full decomposition,and then discard all eigenvectors with eigenvalues that are beyond the first few.
- It turns out that if we are interested in only the first few eigenvectors (with the largest eigenvalues), then iterative processes, which directly optimize these eigenvectors, are computationally more efficient than a full eigendecomposition

Practical Aspects of PCA

- In the extreme case of only needing the first eigenvector, a simple method called the power iteration is very efficient.
- Power iteration chooses a random vector x_0 that is not in the null space of S and follows the iteration for $k = 0, 1, \ldots$

$$x_{k+1} = \frac{Sx_k}{\|Sx_k\|} \tag{6}$$

- This sequence of vectors converges to the eigenvector associated with the largest eigenvalue of S.

PCA in High Dimension

1. In order to do PCA, we need to compute the data covariance matrix.
2. In D dimensions, the data covariance matrix is a $D \times D$ matrix.
3. Computing the eigenvalues and eigenvectors of this matrix is computationally expensive as it scales cubically in D.
4. Therefore, PCA, as we discussed earlier, will be infeasible in very high dimensions.
5. In the following, we provide a solution to this problem for the case that we have substantially fewer data points than dimensions, i.e., $N << D$

PCA in High Dimension

- Assume we have a centered dataset $x_1, ..., x_N, x_n \in \mathbb{R}^D$. Then the data covariance matrix is given as $S = \frac{1}{N}XX^T$
- $$Sb_m = \frac{1}{N}XX^T b_m = \lambda_m b_m \tag{7}$$ $$\frac{1}{N}X^T XX^T b_m = \lambda_m X^T b_m \tag{8}$$ $$\frac{1}{N}X^T X c_m = \lambda_m c_m \tag{9}$$
- Here $c_m = X^T b_m$.
- The nonzeroeigenvalues of XX^T is same as the nonzero eigenvalues of $X^T X$.

PCA in High Dimension

- Now that we have the eigenvectors of $\frac{1}{N}X^TX$,
- We need to derive the eigenvectors of XX^T, which we still need for PCA

$$\frac{1}{N}XX^TXc_m = \lambda_m Xc_m \tag{10}$$

- Here, we recover the data covariance matrix again.
- This now also means that we recover Xc_m as an eigenvector of S.

Key Steps of PCA in Practice

- In the following, we will go through the individual steps of PCA using a running example.
- We are given a two dimensional dataset and we want to use PCA to project it onto a one-dimensional subspace. The key steps are given below
  - Mean subtraction
  - Standardization
  - Eigendecomposition of the covariance matrix
  - Projection

Key Steps of PCA in Practice

Mean subtraction

1. We start by centering the data by computing the mean of the dataset and subtracting it from every single data point.
2. This ensures that the dataset has mean 0.
3. Mean subtraction is not strictly necessary but reduces the risk of numerical problems

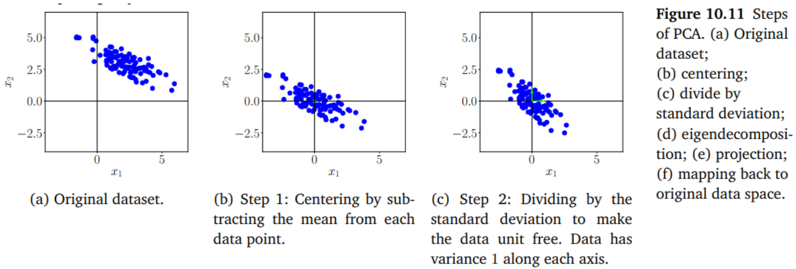
Key Steps of PCA in Practice

Standardization

1. Divide the data points by the standard deviation σ_d of the dataset for every dimension d.
2. Now the data is unit free, and it has variance 1 along each axis, which is indicated by the standardization
3. This step completes the standardization of the data.

Key Steps of PCA in Practice

Figure: PCA



(a) Original dataset.

(b) Step 1: Centering by subtracting the mean from each data point.

(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

Figure 10.11 Steps of PCA. (a) Original dataset; (b) centering; (c) divide by standard deviation; (d) eigendecomposition; (e) projection; (f) mapping back to original data space.

Key Steps of PCA in Practice

Eigendecomposition of the covariance matrix

1. Compute the data covariance matrix and its eigenvalues and corresponding eigenvectors.
2. Since the covariance matrix is symmetric, the spectral theorem states that we can find an orthonormal basis of eigenvectors.
3. The eigenvectors are scaled by the magnitude of the corresponding eigenvalue.

Key Steps of PCA in Practice

Projection

1. We can project any data point $x_* \in \mathbb{R}^d$ onto the principal subspace:
2. To get this right, we need to standardize x_* using the mean and standard deviation of the training data in the d th dimension

$$x_*^{(d)} = \frac{x_*^{(d)} - \mu_d}{\sigma_d}, \quad d = 1, ..., D \tag{11}$$

3. Here $x_*^{(d)}$ is the d th component of x_*.

Key Steps of PCA in Practice

1. We obtain the projection as

$$\tilde{x} = BB^T x_* \tag{12}$$

2. The coordinates are

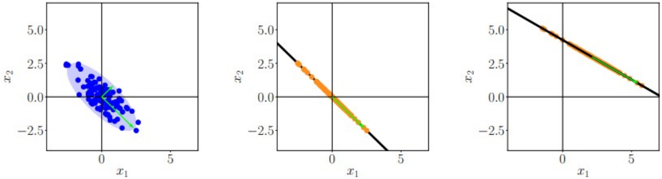
$$z_* = B^T x_* \tag{13}$$

   with respect to the basis of the principal subspace.

3. Here, B is the matrix that contains the eigenvectors that are associated with the largest eigenvalues of the data covariance matrix as columns.

Key Steps of PCA in Practice

Figure: PCA



(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).

(e) Step 4: Project data onto the principal subspace.

(f) Undo the standardization and move projected data back into the original data space from (a).

Summary of PCA

1. We derived PCA from two perspectives: (a) maximizing the variance in the projected space; (b) minimizing the average reconstruction error.
2. We took high-dimensional data $x \in \mathbb{R}^D$and used a matrix B to find a lower-dimensional representation $z \in \mathbb{R}^M$
3. The columns of B are the eigenvectors of the data covariance matrix S that are associated with the largest eigenvalues.
4. Once we have a low-dimensional representation z, we can get a high-dimensional version of it as Bz.