PRS L6

Principal Component Analysis – PCA

Principal Component Analysis – PCA

- Objective: dimensionality reduction, compression, visualization of higher dimensional data (>3D)
- Given a set of data points lying in a high dimensional space (ND), our goal is to reduce the dimensionality (KD) of the data points while preserving as much information as possible. (e.g. 7D points (7 features for each point) will be reduced to 2D (2 features))
- The data will be projected in a smaller K dimensional subspace

Principal Component Analysis – PCA

- **PCA Projection**: ND space -> KD space where K < N
- **PCA Reconstruction**: KD space -> ND space where K < N

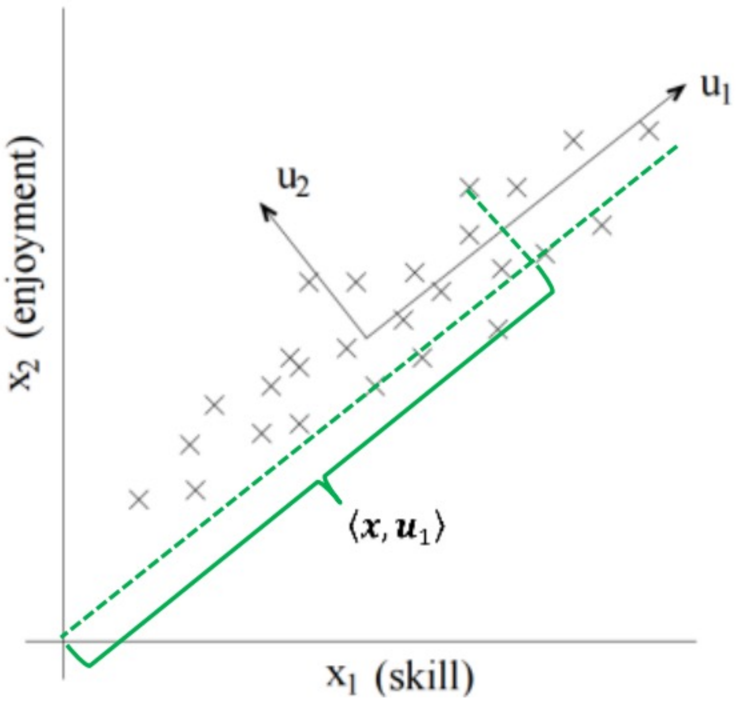
Idea: Compute the top K “major axis of variation” (the directions on which the data approximately lies) – such that when the data is projected to a smaller space, the variance of the projected data is maximized

Principal Component Analysis – PCA

PCA Projection example 2D -> 1D

- We have a set of 2D points (skill, enjoyment)
- Consider two vectors *u1* and *u2*
- If we project the 2D points onto the vector *u2* we obtain scalar values with a small spread (variance)
- If we project the 2D points onto *u1* the spread is much larger

- If we had to choose a single vector we would prefer to project onto *u1* since the variance is maximized and we don’t lose as much information



Principal Component Analysis – PCA

- How to find these vectors (top K "major axis of variation") that are used by the PCA projection?
  - the variance of the projected data should be maximized

- Use the covariance matrix
- Find the axes along which the covariance matrix is maximal with eigen decomposition
  - Use SVD (Singular Value Decomposition) to find the eigenvalues and eigenvectors of the convariance matrix
  - Keep top K eigenvalues that have the largest values
  - The eigenvector corresponding to the largest eigenvalue is the first principal axis, the second eigenvector is the second principal axis and so on
- Math details:
  - https://leimao.github.io/article/Principal-Component-Analysis/
  - http://cs229.stanford.edu/notes2020spring/cs229-notes10.pdf

Practical Work

1. **Read the data** from pca2d.txt and initialize a Mat
    - X has size [1000 x 7] 1000 samples of points with 7 features)

2. **Data normalization** Calculate the mean vector (mean of each column) and subtract it from the data points. e.g. mean vector for X is an array of 7 values.

Practical Work

3. Compute the **covariance matrix**

   `X.t()*X/(n-1);` (e.g C [7x7])

4. Perform the **eigenvalue decomposition** on the covariance matrix C

```
from numpy import linalg as LA
Lambda, Q = LA.eig(C) # Lambda – eigenvalues, Q - eigenvectors
# (e.g. Lambda [7x1], Q [7x7])
```

5. Print the eigenvalues.
   - For pca2d: First eigenvalue is 8090.21
   - For pca3d: First eigenvalue is 5462.33

Practical Work

6. Calculate the PCA coefficients (projection) and compute the reconstruction of X from the projection using k = 1 principal components.

PCA projection using k = 1 components

- Take the first k columns of Q (the top k eigenvectors). You can use *Rect* from opencv to extract a submatrix) => Q_k
- The PCA projection of X: $X_{coeff} = X * Q_k$ (X_{coeff} of size [1000 x 1])
- The PCA reconstruction (approximation): $X_k = X * Q_k * Q_k^T$ (X_k has the same size as X e.g [1000 x 7])

Practical Work

7. Evaluate the mean absolute difference between the original points (X) and their approximation using k principal components (X_k). When computing the mean use all the values from X and Xk. K=1.

    For pca2d: Mean absolute difference using only one dimension: 22.43

    For pca3d: Mean absolute difference using only one dimension: 14.50

8. Set k = 2. For the input data from pca2d.txt select the first two columns from Xcoeff as x and y coordinates and plot the data as black 2D points on a white background (check with Fig. 2).

Practical Work

9. Set k = 3. For input data from pca3d.txt select the first three columns from Xcoeff and plot the data as a grayscale image. Use the first two components as x and y coordinates and the third as intensity values.

10. Automatically select the required k which retains a given percent of the original variance. For example, find k for which the k^{th} approximate retains 99% of the original variance. The percentage of variance retained is given by

$$\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{d} \lambda_i$$