COMP 2211 Exploring Artificial Intelligence

Supplementary Notes: Principal Component Analysis (PCA)

Prof. Song Guo, Dr. Desmond Tsoi & Dr. Huiru Xiao

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology, Hong Kong SAR, China

Principal Component Analysis

- Principal component analysis (PCA) is a dimension-reduction method that can be used to reduce a large set of variables (normally correlated) into a smaller set of (uncorrelated) variables, called principal components, which still contain most of the information.
- Basically PCA is nothing else but a projection of some higher dimensional data into a lower dimension.
- Analogy: When we watch TV, we see a 2D-projection of 3D objects.
- But what we want is not only a projection but a projection which retains as much information as possible.
- A good way is to find the longest axis of the object and after that turn the object around this axis so that we again find the second longest axis.
- Mathematically this can be done by calculating the so called eigenvectors of the covariance matrix, after that we just use the n eigenvectors with the biggest eigenvalues to project the object into n-dimensional space.

Steps

1. Take the whole dataset consisting of d+1 dimensions and ignore the labels such that our new dataset becomes d dimensional.
2. Compute the means for every dimension of the whole dataset.
3. Compute the covariance matrix of the whole dataset.
4. Compute eigenvectors and the corresponding eigenvalues.
5. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix **W**.
6. Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace.

Step 1: Take the whole dataset consisting of d+1 dimensions and ignore the labels such that our new dataset becomes d dimensional

- Let our data be the scores of 5 students:

Student	COMP 2011	COMP 2012	COMP 2211	Label
1	90	60	90	?
2	90	90	30	?
3	60	60	60	?
4	60	60	90	?
5	30	30	30	?

  which forms a 2D matrix as follows

$$M = \begin{bmatrix} 90 & 60 & 90 \\ 90 & 90 & 30 \\ 60 & 60 & 60 \\ 60 & 60 & 90 \\ 30 & 30 & 30 \end{bmatrix}$$

Step 2: Compute the means for every dimension of the whole dataset

- 1st dimension:

$$Mean = (90 + 90 + 60 + 60 + 30)/5 = 66$$

- 2nd dimension:

$$Mean = (60 + 90 + 60 + 60 + 30)/5 = 60$$

- 3rd dimension:

$$Mean = (90 + 30 + 60 + 90 + 30)/5 = 60$$

So, the mean is

$$\overline{M} = \begin{bmatrix} 66 & 60 & 60 \end{bmatrix}$$

Step 3: Compute the covariance matrix of the whole dataset

- The covariance of two variables X and Y using the following formula

$$cov(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{x})(Y_i - \overline{y}) \qquad M = \begin{bmatrix} 90 & 60 & 90 \\ 90 & 90 & 30 \\ 60 & 60 & 60 \\ 60 & 60 & 90 \\ 30 & 30 & 30 \end{bmatrix}$$

- Using the above formula, we can find the covariance of M. Also, the result would be a square matrix of 3×3 dimensions.

Calculation of Covariance Matrix

- $cov(COMP2011, COMP2011) =$
  $\frac{1}{4}((90-66)(90-66)+(90-66)(90-66)+(60-66)(60-66)+(60-66)(60-66)+(30-66)(30-66)) = 630$
- $cov(COMP2011, COMP2012) =$
  $\frac{1}{4}((90-66)(60-60)+(90-66)(90-60)+(60-66)(60-60)+(60-66)(60-60)+(30-66)(30-60)) = 450$
- $cov(COMP2011, COMP2211) =$
  $\frac{1}{4}((90-66)(90-60)+(90-66)(30-60)+(60-66)(60-60)+(60-66)(90-60)+(30-66)(30-60)) = 225$
- $cov(COMP2012, COMP2012) =$
  $\frac{1}{4}((60-60)(60-60)+(90-60)(90-60)+(60-60)(60-60)+(60-60)(60-60)+(30-60)(30-60)) = 450$
- $cov(COMP2012, COMP2211) =$
  $\frac{1}{4}((60-60)(90-60)+(90-60)(30-60)+(60-60)(60-60)+(60-60)(60-90)+(30-60)(30-60)) = 0$
- $cov(COMP2211, COMP2211) =$
  $\frac{1}{4}((90-60)(90-60)+(30-60)(30-60)+(60-60)(60-60)+(90-60)(90-60)+(30-60)(30-60)) = 900$

As the covariance matrix is symmetric along the diagonal, we do not need to compute $cov(COMP2012, COMP2011)$, $cov(COMP2211, COMP2011)$, and $cov(COMP2211, COMP2012)$.

So the covariance matrix is

$$C = \begin{bmatrix} 630 & 450 & 225 \\ 450 & 450 & 0 \\ 225 & 0 & 900 \end{bmatrix}$$

Observations based on the covariance matrix

- The covariance between COMP2011 and COMP2012 is positive 450, and the covariance between COMP2011 and COMP 2211 is positive 225. This means the scores tend to co-vary in a positive way, i.e. as the scores on COMP2011 goes up, scores on COMP2012 and COMP 2211 also tend to go up, and vice versa.
- The covariance between COMP2012 and COMP 2211 is 0. This means there tends to be no predictable relationship between the movement of COMP2012 and COMP2211 scores.

Compute eigenvectors and the corresponding eigenvalues

- Let **A** be a square matrix, **v** be a vector and λ is a scalar that satisfies $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. λ is called eigenvalue associated with eigenvector **v** of **A**.
- The eigenvalues of **A** are roots of the characteristic equation

$$det(A - \lambda I) = 0$$

- Calculating $det(A - \lambda I)$ first where I is an identity matrix:

$$det\left(\begin{bmatrix} 630 & 450 & 225 \\ 450 & 450 & 0 \\ 225 & 0 & 900 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

$$= det\left(\begin{bmatrix} 630-\lambda & 450 & 225 \\ 450 & 450-\lambda & 0 \\ 225 & 0 & 900-\lambda \end{bmatrix}\right)$$

- Now equating the above to zero:

$$-\lambda^3 + 1980\lambda^2 - 1002375\lambda + 50118750 = 0‘$$

Compute eigenvectors and the corresponding eigenvalues

- Solving the equation for the value of λ, we get the following values.

$$\begin{aligned}\lambda_1 &= 56.025\\ \lambda_2 &= 786.388\\ \lambda_3 &= 1137.587\end{aligned}$$

- After solving for eigenvectors, we would get the following solution for the corresponding eigenvalues.

$$\begin{aligned}\mathbf{v_1} &= (0.6487899, -0.74104991, -0.17296443)\\ \mathbf{v_2} &= (-0.3859988, -0.51636642, 0.7644414)\\ \mathbf{v_3} &= (-0.65580225, -0.4291978, -0.62105769)\end{aligned}$$

Step 4: Sort the eigevectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d×k dimensional matrix **W**

- After sorting the eigenvalues in decreasing order, we have

$$\begin{bmatrix} 1137.587 \\ 786.388 \\ 56.025 \end{bmatrix}$$

- For our example, where we are reducing a 3-dimensional data to a 2-dimensional data, we are combining the two eigenvectors with the highest eigenvalues to construct our d×k dimensional eigenvector matrix **W**.
- So, eigenvectors corresponding to two maximum eigenvalues are:

$$\mathbf{W} = \begin{bmatrix} -0.65580225 & -0.3859988 \\ -0.4291978 & -0.51636642 \\ -0.62105769 & 0.7644414 \end{bmatrix}$$

Use this d×k eigenvector matrix to transform the samples onto the new subspace

- We use the 3× 2 matrix $\mathbf{W}$ to transform our samples via the equation:

$$\mathbf{y} = \mathbf{x}\mathbf{W}$$

- So, we have

$$\mathbf{y} = \begin{bmatrix} 90 & 60 & 90 \\ 90 & 90 & 30 \\ 60 & 60 & 60 \\ 60 & 60 & 90 \\ 30 & 30 & 30 \end{bmatrix} \begin{bmatrix} -0.65580225 & -0.3859988 \\ -0.4291978 & -0.51636642 \\ -0.62105769 & 0.7644414 \end{bmatrix}$$

$$= \begin{bmatrix} -140.67 & 3.08 \\ -116.28 & -58.28 \\ -102.36 & -8.28 \\ -121.00 & 14.66 \\ -51.18 & -4.14 \end{bmatrix}$$

Implement PCA using NumPy

```python
import numpy as np # Import NumPy

x = np.array( [[90, 60, 90],    # Original data
[90, 90, 30],
[60, 60, 60],
[60, 60, 90],
[30, 30, 30]] )

covariance_matrix = np.cov(x , rowvar = False, bias=False) # Find covariance matrix
print("Covariance Matrix:\n", covariance_matrix)

# Compute eigenvalues, eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)

# Sort the eigenvalues and eigenvectors
sorted_index = np.argsort(eigenvalues)[::-1]
sorted_eigenvalue = eigenvalues[sorted_index]
sorted_eigenvectors = eigenvectors[:,sorted_index]
print("Eigenvalues:\n",sorted_eigenvalue)
print("Eigenvectors:\n",sorted_eigenvectors)

# Form transformation matrix
W = np.array([sorted_eigenvectors[:,0], sorted_eigenvectors[:,1]]).T
print("W:\n", W)

# Compute the result, i.e., the data in reduced dimensions
y = x.dot(W)
print("y:\n", y)
```

That's all!

Any questions?