Machine Learning - Unsupervised Learning

Dimensionality Reduction

Principal Component Analysis (PCA)

Dataset:

The Dataset is taken from UCI Machine Learning Repository. The Dataset has 10 real attributes and one categorical attribute, which is a class label. Since here we are using a Principal Component Analysis (PCA) which is Unsupervised Learning model which doesn't need class labels attribute.

Description:

MAGIC gamma telescope data 2004
Number of instances = 19020
Number of attributes = 11(including the class)
Missing attribute values = None
Class distribution = g, h (gamma - 12332, hadron- 6688)
The dataset has 10 real (continuous) and one categorical attribute (class label)

Implementation:

Z- Score Normalization:

This is also called as a Standard Score Normalization.

Here each value in the dataset is replaced by its ***z-score.***

$$x_i' = \frac{x_i - \hat{\mu}}{\hat{\sigma}}$$

where hmu is the sample mean and $\hat{\sigma}^2$ is the sample variance of X. After transformation, the new attribute has mean $\hat{\mu}' = 0$, and standard deviation $\hat{\sigma}' = 1$.

The *sample standard deviation* is given as the positive square root of the sample variance

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2}$$

The *standard score*, also called the *z-score* , of a sample value x_i is the number of standard deviations away the value is from the mean

$$z_i = \frac{x_i - \hat{\mu}}{\hat{\sigma}}$$

Put differently, the z-score of x_i measures the deviation of x_i from the mean value $\hat{\mu}$, in units of $\hat{\sigma}$.

MANUALLY CALCULATED Z-SCORE

	0	1	2	...	7	8	9
0	-0.577226	-0.336804	-0.381130	...	-0.405842	0.476816	-1.497866
1	-0.510969	-0.570027	-0.648595	...	-0.490094	-0.815418	0.153125
2	2.568278	6.205858	2.615783	...	-2.183030	1.889224	0.842635
3	-0.694768	-0.687259	-1.029478	...	-0.355359	-0.658804	-1.031463
4	0.516622	0.476384	0.711157	...	1.036620	-0.881039	2.176427
5	-0.038386	-0.056186	0.176651	...	0.459251	-0.920690	0.592534
6	-0.118105	-0.262977	0.440519	...	0.496333	-0.875523	0.338138
7	-0.624601	-0.459046	-0.577497	...	-0.152636	-1.027996	0.579634
8	1.014607	1.326507	2.812149	...	2.061502	-0.873147	0.728062
9	-0.153156	-0.380564	-0.521422	...	-0.332787	-0.757413	-1.225308
10	0.224873	0.421324	1.075112	...	1.129370	-0.679284	1.729911
11	-0.811872	-0.311844	-0.817876	...	-0.825012	-0.620034	-0.414427
12	-0.179825	0.028837	0.463583	...	-0.699390	-1.044461	-0.208257
13	-0.473715	-0.529581	-0.697263	...	-0.404766	0.547191	-2.039834
14	0.170142	0.595519	0.781620	...	1.372350	-0.741438	2.391950
15	-0.561236	0.015106	-0.421546	...	0.620235	0.781165	0.271752

Sample Covariance Matrix:

Sample Covariance Matrix The *sample covariance matrix* is given as

$$\widehat{\boldsymbol{\Sigma}} = E\big[(\mathbf{X} - \hat{\boldsymbol{\mu}})(\mathbf{X} - \hat{\boldsymbol{\mu}})^T\big] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1d} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{d1} & \hat{\sigma}_{d2} & \cdots & \hat{\sigma}_d^2 \end{pmatrix} \tag{2.29}$$

Instead of computing the sample covariance matrix element-by-element, we can obtain it via matrix operations. Let $\mathbf{Z}$ represent the centered data matrix, given as the matrix of centered attribute vectors $Z_i = X_i - \mathbf{1} \cdot \hat{\mu}_i$, where $\mathbf{1} \in \mathbb{R}^n$

$$\mathbf{Z} = \mathbf{D} - \mathbf{1} \cdot \hat{\boldsymbol{\mu}}^T = \begin{pmatrix} | & | & & | \\ Z_1 & Z_2 & \cdots & Z_d \\ | & | & & | \end{pmatrix}$$

Alternatively, the centered data matrix can also be written in terms of the centered points $\mathbf{z}_i = \mathbf{x}_i - \hat{\boldsymbol{\mu}}$

$$\mathbf{Z} = \mathbf{D} - \mathbf{1} \cdot \hat{\boldsymbol{\mu}}^T = \begin{pmatrix} \mathbf{x}_1^T - \hat{\boldsymbol{\mu}}^T \\ \mathbf{x}_2^T - \hat{\boldsymbol{\mu}}^T \\ \vdots \\ \mathbf{x}_n^T - \hat{\boldsymbol{\mu}}^T \end{pmatrix} = \begin{pmatrix} — & \mathbf{z}_1^T & — \\ — & \mathbf{z}_2^T & — \\ & \vdots & \\ — & \mathbf{z}_n^T & — \end{pmatrix}$$

In matrix notation, the sample covariance matrix can be written as

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}(\mathbf{Z}^T\mathbf{Z}) = \frac{1}{n}\begin{pmatrix} Z_1^T Z_1 & Z_1^T Z_2 & \cdots & Z_1^T Z_d \\ Z_2^T Z_1 & Z_2^T Z_2 & \cdots & Z_2^T Z_d \\ \vdots & \vdots & \ddots & \vdots \\ Z_d^T Z_1 & Z_d^T Z_2 & \cdots & Z_d^T Z_d \end{pmatrix} \tag{2.30}$$

The sample covariance matrix is thus given as the pair-wise *inner or dot products* of the centered attribute vectors, normalized by the sample size.

In terms of the centered points $\mathbf{z}_i$, the sample covariance matrix can also be written as a sum of rank-one matrices obtained as the *outer-product* of each centered point

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i \cdot \mathbf{z}_i^T \tag{2.31}$$

```
COVARIANCE MATRIX USING MANUAL CALCULATION
[[ 1.          0.77051204  0.70245376 -0.63099908 -0.59814513 -0.36855649
  -0.11974713  0.01338923 -0.00877749  0.41846607]
 [ 0.77051204  1.          0.71751686 -0.60977867 -0.58114073 -0.26696092
  -0.17623422  0.03974361  0.06606138  0.33681611]
 [ 0.70245376  0.71751686  1.         -0.8508498  -0.80883501 -0.15986295
   0.09515719  0.01545474 -0.18667516  0.43704076]
 [-0.63099908 -0.60977867 -0.8508498   1.          0.97641185  0.11227223
  -0.12189934 -0.01129407  0.2352719  -0.3283323 ]
 [-0.59814513 -0.58114073 -0.80883501  0.97641185  1.          0.10015922
  -0.11876891 -0.0109656   0.22979854 -0.30462468]
 [-0.36855649 -0.26696092 -0.15986295  0.11227223  0.10015922  1.
   0.27404482  0.00255284 -0.05568862 -0.20672963]
 [-0.11974713 -0.17623422  0.09515719 -0.12189934 -0.11876891  0.27404482
   1.         -0.01719669 -0.18627459  0.03702517]
 [ 0.01338923  0.03974361  0.01545474 -0.01129407 -0.0109656   0.00255284
  -0.01719669  1.          0.00465913  0.01142661]
 [-0.00877749  0.06606138 -0.18667516  0.2352719   0.22979854 -0.05568862
  -0.18627459  0.00465913  1.         -0.22055558]
 [ 0.41846607  0.33681611  0.43704076 -0.3283323  -0.30462468 -0.20672963
   0.03702517  0.01142661 -0.22055558  1.        ]]
(19020, 10)
```

Dominant Eigen Value and Eigen Vector of the Covariance Matrix $\sum$ via the Power-iteration Method:

$$\mathbf{x_0} = \begin{pmatrix} x_{0,1} \\ x_{0,2} \\ \vdots \\ x_{0,d} \end{pmatrix} \tag{1}$$

Let, $\mathbf{x_0}$ be a starting vector in R^d, where d is a vector of appropriate dimensions. In each iteration i, we compute the new vector: $\mathbf{x_i} = \mathbf{\Sigma x_{i-1}}$. We then find the element of $\mathbf{x_i}$ that has the maximum absolute value, say at index m_i. For the next round, to avoid numerical issues with large values, we re-scale $\mathbf{x_i}$ by dividing all elements by x_{i,m_i}, so that the largest value is always 1 before we begin the next iteration. To test convergence, you may compute the norm of the difference between the scaled vectors from the current iteration and the previous one, and you can stop if this norm falls below some threshold, say 0.000001. That is, to stop, check if $\|\mathbf{x_i} - \mathbf{x_{i-1}}\| < 0.000001$ is true. For the final eigen-vector, make sure to normalize it, so that it has unit length. Also, the ratio $x_{i,m_i}/x_{i-1,m_{i-1}}$ gives you the largest eigenvalue. Verify your answer using the numpy `linalg.eig` function.

Covariance Matrix → Σ

Eigen Value → λ

Eigen vector → V

$$X_0 = \begin{pmatrix} x_{0,1} \\ x_{0,2} \\ \vdots \\ x_{0,d} \end{pmatrix} \quad \text{——— (1).}$$

X_0 is a starting vector in R^d

d - no. of attributes in a Σ matrix.

Power-iteration method:-

(i - iteration).

$X_i = \Sigma X_{i-1}$ $\quad (X_{new} = \Sigma X_{old})$

$max\text{-}val = max(x_i)$ [maximum absolute value].

$x_i' = \frac{x_i}{max\text{-}val}$. [re-scale vector]

largest value in x_i' is 1.

Convergence:

If ($\|x_i' - x_{i-1}\| < 0.000001$) → threshold value.

stop's execution.

$x_{i-1} = x_i'$

Eigen vector = $\frac{x_i'}{\|x_i\|}$ (unit normalization)

Eigen value = max_val

Note:- Length of Eigen vector = 1. (unit vector).

```
 THE DOMINANT EIGEN VALUE FOR COVARIANCE MATRIX USING MANUAL CALCULATION
4.223990734177301

 THE DOMINANT EIGEN VECTOR FOR COVARIANCE MATRIX USING MANUAL CALCULATION
[[ 0.40827675]
 [ 0.39505702]
 [ 0.44892194]
 [-0.44161248]
 [-0.42923781]
 [-0.14468793]
 [ 0.00519843]
 [ 0.01235159]
 [-0.09717502]
 [ 0.2566813 ]]
```

First Two Dominant Eigen Vectors of ∑

First sort the Eigen values (λ) of the Covariance Matrix (∑) in a descending order and take first two values from it (Since we are looking for two dominant eigenvalues).

Sort the Eigen vectors (V) corresponding to the two dominant Eigen values (λ) and take two eigen vectors.

```
 First Two Dominant Eigenvectors of Covariance Matrix
[[-0.40827678  0.22569874]
 [-0.39505705  0.24528158]
 [-0.44892193 -0.09331051]
 [ 0.44161246  0.1883758 ]
 [ 0.42923779  0.19595182]
 [ 0.14468798 -0.48690288]
 [-0.00519836 -0.59959056]
 [-0.01235159  0.03147723]
 [ 0.09717497  0.45670351]
 [-0.2566813  -0.01246824]]
(10, 2)
```

Projection of datapoints on the subspace spanned by these two eigen vectors

To compute the datapoints, multiply are Z-score matrix x 2-dominant eigen vectors

)

$$\mathbf{D = ZxV2}$$

```
 projection of data points spanned by 2-dominant eigenvector
[[ 1.01654976 -0.36925645]
 [ 1.67996111 -0.54922547]
 [-5.96624169  1.8313162 ]
 ...
 [-2.76395431 -0.49118566]
 [-4.64718554  2.31055878]
 [-3.35429914  4.76432253]]
```

Variance of the datapoints

The variance of the datapoints spanned by two dominant eigenvectors is equal to the sum of the two dominant eigenvalues.

Variance $\sigma = \sum_{k=1}^{n} \lambda(k)$ k – number of dominant eigenvalues

```
THE VARIANCE OF DATA POINTS ON PROJECTED SUBSPACE :  5.799178801077083
```

Covariance Matrix ($\sum$) in its Eigen-decomposition form ($V\lambda V^T$)

V- Eigen vector matrix

λ- Eigen value matrix (non-diagonal elements are zero, and diagonal elements are eigen values)

$$\sum = V\lambda V^T$$

$V\lambda V^T$ should equal to Covariance Matrix ($\sum$)

```
 COVARIANCE MATRIX IN EIGEN-DECOMPOSITION FORM UVUT
[[ 1.          0.77051204  0.70245376 -0.63099908 -0.59814513 -0.36855649
  -0.11974713  0.01338923 -0.00877749  0.41846607]
 [ 0.77051204  1.          0.71751686 -0.60977867 -0.58114073 -0.26696092
  -0.17623422  0.03974361  0.06606138  0.33681611]
 [ 0.70245376  0.71751686  1.         -0.8508498  -0.80883501 -0.15986295
   0.09515719  0.01545474 -0.18667516  0.43704076]
 [-0.63099908 -0.60977867 -0.8508498   1.          0.97641185  0.11227223
  -0.12189934 -0.01129407  0.2352719  -0.3283323 ]
 [-0.59814513 -0.58114073 -0.80883501  0.97641185  1.          0.10015922
  -0.11876891 -0.0109656   0.22979854 -0.30462468]
 [-0.36855649 -0.26696092 -0.15986295  0.11227223  0.10015922  1.
   0.27404482  0.00255284 -0.05568862 -0.20672963]
 [-0.11974713 -0.17623422  0.09515719 -0.12189934 -0.11876891  0.27404482
   1.         -0.01719669 -0.18627459  0.03702517]
 [ 0.01338923  0.03974361  0.01545474 -0.01129407 -0.0109656   0.00255284
  -0.01719669  1.          0.00465913  0.01142661]
 [-0.00877749  0.06606138 -0.18667516  0.2352719   0.22979854 -0.05568862
  -0.18627459  0.00465913  1.         -0.22055558]
 [ 0.41846607  0.33681611  0.43704076 -0.3283323  -0.30462468 -0.20672963
   0.03702517  0.01142661 -0.22055558  1.        ]]
```

Implementation of Principal Component Analysis by using Subroutine:

Algorithm 7.1: Principal Component Analysis

PCA $(\mathbf{D}, \alpha)$:

1 $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ // compute mean

2 $\mathbf{Z} = \mathbf{D} - \mathbf{1} \cdot \boldsymbol{\mu}^T$ // center the data

3 $\boldsymbol{\Sigma} = \frac{1}{n}\left(\mathbf{Z}^T\mathbf{Z}\right)$ // compute covariance matrix

4 $(\lambda_1, \lambda_2, \cdots, \lambda_d) = \text{eigenvalues}(\boldsymbol{\Sigma})$ // compute eigenvalues

5 $\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_d \end{pmatrix} = \text{eigenvectors}(\boldsymbol{\Sigma})$ // compute eigenvectors

6 $f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, for all $r = 1, 2, \cdots, d$ // fraction of total variance

7 Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality

8 $\mathbf{U}_r = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \end{pmatrix}$ // reduced basis

9 $\mathbf{A} = \{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{U}_r^T\mathbf{x}_i, \text{for } i = 1, \cdots, n\}$ // reduced dimensionality data

Algorithm 7.1 gives the pseudo-code for the principal component analysis algorithm. Given the input data $\mathbf{D} \in \mathbb{R}^{n \times d}$, it first centers it by subtracting the mean from each point. Next, it computes the eigenvectors and eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$. Given the desired variance threshold α, it selects the smallest set of dimensions r that capture at least α fraction of the total variance. Finally, it computes the coordinates of each point in the new r-dimensional principal component subspace, to yield the new data matrix $\mathbf{A} \in \mathbb{R}^{n \times r}$.

Since, we have already calculated the mean and centered data in the z-normalization step. Here, we directly compute the covariance matrix.

Choosing the Dimensionality

Often we may not know how many dimensions, r, to use for a good approximation. One criteria for choosing r is to compute the fraction of the total variance captured by the first r principal components, computed as

$$f(r) = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_r}{\lambda_1 + \lambda_2 + \cdots + \lambda_d} = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{\sum_{i=1}^{r} \lambda_i}{var(\mathbf{D})} \tag{7.25}$$

Given a certain desired variance threshold, say α, starting from the first principal component, we keep on adding additional components, and stop at the smallest value r, for which $f(r) \geq \alpha$. In other words, we select the fewest number of dimensions such that the subspace spanned by those r dimensions captures at least α fraction of the total variance. In practice, α is usually set to 0.9 or higher, so that the reduced dataset captures at least 90% of the total variance.

Total Variance

Note that the total variance of $\mathbf{D}$ is invariant to a change in basis vectors. Therefore, we have the following identity

$$var(\mathbf{D}) = \sum_{i=1}^{d} \sigma_i^2 = \sum_{i=1}^{d} \lambda_i$$

1. **Here we will find the Principal Vectors that preserves the 95% variance of the data. (*fig:* 1(a))**
2. **Will compute the datapoints using Principal Vectors as the new basis vectors. (*fig:* 1(b))**

```
The number of eigevectors that preserves the variance of 95 % is 7
eigen vector  [[-4.08276775e-01  2.25698741e-01 -5.00639811e-04  2.53364917e-02
  -1.19474438e-01 -1.49895160e-02 -4.37462245e-01  7.53225050e-01
  -9.04660487e-02  1.79955894e-02]
 [-3.95057049e-01  2.45281583e-01 -1.52717635e-01  9.46212428e-02
  -5.78974762e-02 -1.68904043e-01 -4.95341859e-01 -5.39478658e-01
   4.27867208e-01 -1.17981303e-02]
 [-4.48921929e-01 -9.33105148e-02 -7.06014805e-02  5.74952569e-02
  -1.59226791e-02 -2.76416862e-02 -1.07193101e-03 -3.21598696e-01
  -8.15749467e-01  1.07611883e-01]
 [ 4.41612458e-01  1.88375795e-01  1.28919623e-01 -1.08617795e-01
  -1.80669622e-01 -1.04311902e-01 -3.30736734e-01 -7.19134376e-02
  -1.48746906e-01  7.48911506e-01]
 [ 4.29237789e-01  1.95951819e-01  1.39197755e-01 -1.16313517e-01
  -2.05565490e-01 -1.15773238e-01 -3.89073611e-01 -1.05167458e-01
  -3.15173270e-01 -6.53467431e-01]
 [ 1.44687983e-01 -4.86902882e-01 -3.77273475e-01  2.13916540e-01
   2.27372884e-02 -7.21816382e-01 -1.13592072e-01  1.39311777e-01
  -6.97106831e-03 -1.01361748e-04]
 [-5.19836359e-03 -5.99590556e-01 -5.83262999e-02  5.35141492e-02
  -5.98664166e-01  4.59923140e-01 -2.33844650e-01 -3.57336827e-02
   9.11194260e-02  2.71596438e-03]
 [-1.23515912e-02  3.14772268e-02 -6.02869888e-01 -7.94685553e-01
  -8.11764733e-03  5.67924976e-02  1.43239128e-02  1.72572649e-02
  -9.22610587e-03 -8.71296824e-06]
 [ 9.71749700e-02  4.56703506e-01 -4.69918810e-01  3.86270303e-01
  -5.25793320e-01  3.00595105e-03  3.64958044e-01  2.65508688e-02
  -3.93710146e-02 -5.78335826e-03]
 [-2.56681301e-01 -1.24682373e-02  4.53523095e-01 -3.63893657e-01
  -5.21326430e-01 -4.58710859e-01  3.18607890e-01 -1.64290736e-02
   1.08866787e-01 -5.26146252e-03]]
```

fig: **1(a)**

```
COORDINATES OF THE FIRST TEN DATA POINTS
[[ 1.01654976 -0.36925645 -0.80995074  1.12162721  0.52498314  0.46510731
   0.03388353]
 [ 1.67996111 -0.54922547  0.98161468 -0.22118145 -0.13761196 -0.48799867
  -0.71280668]
 [-5.96624169  1.8313162  -1.49720548  3.72895739 -0.48762911 -3.41897421
  -1.84986395]
 [ 2.55706027 -0.03162913  0.44138683  0.03143789  0.67608057 -0.25175859
  -0.9603356 ]
 [-1.65585251 -0.55290202  0.55551247 -1.77161148 -0.87096642 -0.80381284
   0.06940903]
 [-0.80452283 -1.57572136 -0.16764975 -0.53153486  0.10124447 -0.45857728
   0.1824198 ]
 [-0.74806576 -1.19404497 -0.0084875  -0.64048741  0.2273275   0.13446004
   0.27858938]
 [ 0.64600539 -1.02760041  0.75535631 -0.47465107  0.19880914 -0.52707914
   0.11000122]
 [-3.63809698 -2.49323102 -2.15451949 -1.07040856 -0.3848824  -0.90627546
  -0.55716894]
 [ 0.56596174 -1.13679926 -0.18709999  0.53788967  0.79383922  0.61664008
  -0.46480103]]
```

fig: **1(b)**

Covariance of the Projected data points:

Here we will compute the covariance of the projected data points.

Sum of Eigenvalues corresponding to the Principal Vectors on which data is projected:

We will take the corresponding eigenvalues of principal vectors and sum it.

```
 The Covariance Matrix of the Projected data points

[[ 4.22399090e+00  1.82305392e-16  1.50924955e-16 -4.78551653e-16
   2.04346412e-16  6.87380985e-17 -9.77650021e-16]
 [ 1.82305392e-16  1.57518790e+00 -5.74660077e-16  9.82973488e-17
  -6.48902593e-16 -1.49804226e-16  2.94751955e-16]
 [ 1.50924955e-16 -5.74660077e-16  1.01180532e+00 -1.75805158e-15
   4.59499246e-17  4.66970778e-17 -2.58515023e-16]
 [-4.78551653e-16  9.82973488e-17 -1.75805158e-15  9.94012697e-01
  -3.38787299e-16  2.40956921e-16  6.41617849e-17]
 [ 2.04346412e-16 -6.48902593e-16  4.59499246e-17 -3.38787299e-16
   7.42019394e-01  3.47986624e-16  2.88961517e-16]
 [ 6.87380985e-17 -1.49804226e-16  4.66970778e-17  2.40956921e-16
   3.47986624e-16  6.50086467e-01 -3.59100528e-17]
 [-9.77650021e-16  2.94751955e-16 -2.58515023e-16  6.41617849e-17
   2.88961517e-16 -3.59100528e-17  4.07538598e-01]]
```

fig: **Covariance matrix of the projected data points**

Verification:

The sum of the eigenvalues corresponding to principal vectors = Trace of the Covariance of Projected Data Matrix (sum of the diagonal elements of the covariance matrix)

```
COVARIANCE OF THE PROJECTED DATA POINTS :  9.604641277555682
SUM OF THE EIGENVALUES CORRESPONDING TO THE PRINCIPAL VECTORS ON WHICH DATA IS PROJECTED :  9.604641277555682
```

References:

https://archive.ics.uci.edu/ml/machine-learning-databases/magic/magic04.data

https://repo.palkeo.com/algo/information-retrieval/Data%20mining%20and%20analysis.pdf

https://en.wikipedia.org/wiki/Principal_component_analysis