Clustering and Dimensionality Reduction in Better Life Index 2022

Francesco Natali - Matricola 1945581

Introduction

This study aims to explore the Better Life Index 2022 dataset, which includes data from 41 countries across 24 variables. The objective is to employ various clustering techniques in combination with dimensionality reduction methods to categorize countries based on their quality of life indicators.
To ensure a proper analysis, missing values in the dataset, represented as "NaN", must be addressed. To achieve this, we replace each missing value with the mean of its three nearest neighbors, ensuring a more precise estimation. The following command is used for this process:

```
Xi = knnimpute (X',3);    where X is pur Data Matrix
```

Sandardize Data

The dataset is subsequently standardized to make possible the comparison among the variables and of our results. I recall that a standardized variable has mean=0 and variance=1.:

```
Xs = zscore(Xi,1);
```

Now I calculate even the correlation Matrix Rx

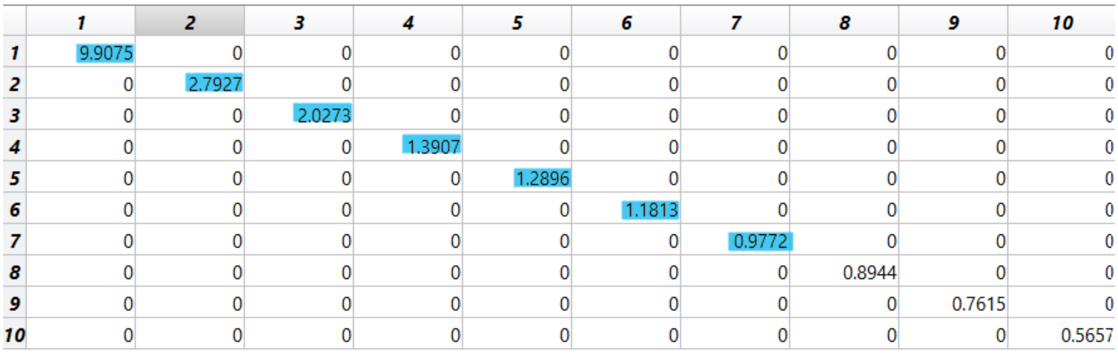
```
    Rx = (1/41) * Xs' * Xs;
```

1 Compute the PCA to determine the number of PRINCIPAL COMPONENTS

To determine the number of principal components, we apply PCA and use Kaiser's rule, selecting components with eigenvalues greater than 1.

```
[A,L] = eigs(Rx,24);
RQ = trace(L(1:7,1:7))/24;
Y = Xs * A(:,1:7);
```

where A is the matrix of eigenvectors; L the matrix of the eigenvalues. As we can see from the result, we've 7 principal components (The seventh taken for approximation, because very close to 1).



	1	2	3	4	5	6	7	8	9	10
1	9.9075	0	0	0	0	0	0	0	0	0
2	0	2.7927	0	0	0	0	0	0	0	0
3	0	0	2.0273	0	0	0	0	0	0	0
4	0	0	0	1.3907	0	0	0	0	0	0
5	0	0	0	0	1.2896	0	0	0	0	0
6	0	0	0	0	0	1.1813	0	0	0	0
7	0	0	0	0	0	0	0.9772	0	0	0
8	0	0	0	0	0	0	0	0.8944	0	0
9	0	0	0	0	0	0	0	0	0.7615	0
10	0	0	0	0	0	0	0	0	0	0.5657

```
RQ = trace(L(1:7,1:7))/24;
```

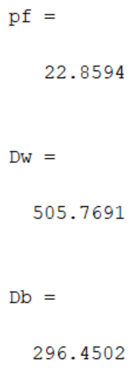
Seven components explain 81.53% of the total variance, justifying their use in subsequent clustering analyses. Now we compute the Component Score Matrix, that we will use for the k-means clustering

```
Y = Xs * A(:,1:7);
```

2 Compute K-MEANS on the number of component identified in STEP 1 & Identify best K with pF

K-means is applied to the principal components. The best number of clusters is selected using the pseudo-F statistic. We know that for an optimal data's partition we need to find the one with maximum ratio between deviance between and deviance within, that is exactly the pseudo-F index. Let's start computing this index considering K=2 clusters:

```
[cv2,Xm2] = kmeans(Y,2,'rep',100);
I2 = eye(2);
Ukm2 = I2(cv2,:);
[pf,Dw,Db] = psF(Y,Ukm2);
```



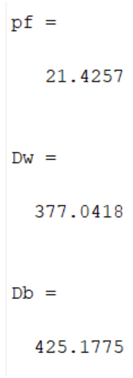
```
pf =

   22.8594


Dw =

  505.7691


Db =

  296.4502
```

The number of clusters is set to 2, and the analysis is repeated 100 times to ensure stability. In the computation, the variable Ukm2 represents the classification labels assigned to each observation, while corresponds to the centroids of the clusters. The matrix is a binary, row-stochastic matrix associated with , ensuring a proper representation of the cluster assignments.

As a final result, the computed pseudo-F statistic is 22.8594.

Now I repeat the same steps for K=3 and K=4 clusters to compare the values of the pseudo-F index

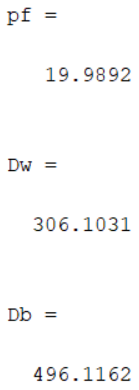
```
[cv3,Xm3]=kmeans(Y,3,'rep',100);
I3=eye(3);
Ukm3=I3(cv3,:);
[pf,Dw,Db] = psF(Y,Ukm3);
```



```
pf =

   21.4257


Dw =

  377.0418


Db =

  425.1775
```

```
[cv4,Xm4]=kmeans(Y,4,'rep',100);
```

```
I4=eye(4);
Ukm4=I4(cv4,:);
[pf,Dw,Db] = psF(Y,Ukm4);
```



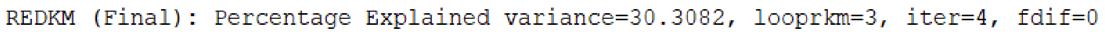
```
pf =

   19.9892


Dw =

  306.1031


Db =

  496.1162
```

For K=3 we obtain pf=21.4257 while for K=4 the result is pf=19.9892. So the highest values of the Pseudo-F index is obtained with K=2 clusters.

3 Compute Reduced K-Means (RKM)

By now we are cnsidering the number of cluster is equal to 2, and the individuated Principal Components are 7. RKM integrates PCA into clustering to reduce the dimensionality while maintaining cluster structure, that is useful to optimize the reconstruction of the data by means of a simultaneous clustering of units and variables:

```
[Urkm,Arkm,Yrkm,frkm,inrkm] = REDKM(Xs,2,7,100);
```



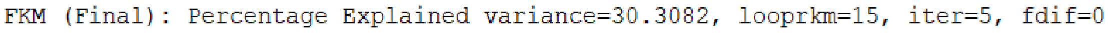
```
REDKM (Final): Percentage Explained variance=30.3082, looprkm=3, iter=4, fdif=0
```

The explained variance is 73.92%, lower than PCA that was is equal to 81.53% but it's obviously higher due to the fact that it has been compuetd on 41 obs.

4 Compute Factorial K-Means (FKM)

So, even for the computation of the Factorial K-Means, we will utilize the values obtained in the previous points. So with K=2 cluster and 7 Principal Components. FKM optimizes clustering by projecting all data points into a reduced space, so we want to reconstruct the data in the reduced space by means of the reduced centroids matrix:

```
[Ufkm,Afkm,Yfkm,ffkm,infkm] = FKM(Xs,2,7,100);
```



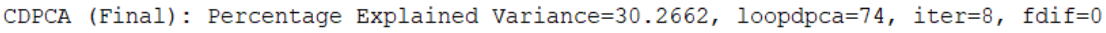
```
FKM (Final): Percentage Explained variance=30.3082, looprkm=15, iter=5, fdif=0
```

The explained variance is 30.31%.

5 Compute Clustering and Disjoint PCA (CDPCA)

CDPCA simultaneously clusters observations and assigns variables to different components, where each class of variables is summarised by PC of maximal variance. The command used is similar to the last two that we've seen:

```
[Vcdpca,Ucdpca,Acdpca,Ycdpca,fcdpca,incdpca] = CDPCA(Xs,2,7,100);
```



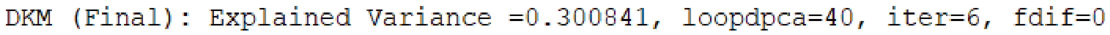
```
CDPCA (Final): Percentage Explained Variance=30.2662, loopdpca=74, iter=8, fdif=0
```

The explained variance is 30.29%, that is almost equal to the explained variance obtained with the FKM. These values compared with the PCA's one are definetly higher but, in absolotue terms, they are still quite small.

6 Compute Double K-Means (DKM)

This method extends CDPCA by enforcing equal weighting among variables in each component:

```
[Vdkm,Udkm,Ymdkm,fdkm,indkm] = DKM(Xs,2,7);
```



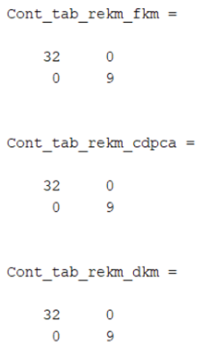
```
DKM (Final): Explained Variance =0.300841, loopdpca=40, iter=6, fdif=0
```

The explained variance is 30.08%, confirming similarities with FKM and CDPCA.

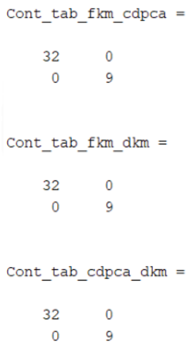
7 Compare the results by using the confusion matrix (CONTINGENCY TABLE) between 4, 5, 6 and 7

A contingency table is constructed to compare clustering results:

```
Cont_tab_rekm_fkm = Urkm' * Ufkm;
Cont_tab_rekm_cdpca = Urkm' * Ucdpca;
Cont_tab_rekm_dkm = Urkm' * Udkm;
Cont_tab_fkm_cdpca = Ufkm' * Ucdpca;
Cont_tab_fkm_dkm = Ufkm' * Udkm;
Cont_tab_cdpca_dkm = Ucdpca' * Udkm;
```



```
Cont_tab_rekm_fkm =

    32     0
     0     9


Cont_tab_rekm_cdpca =

    32     0
     0     9


Cont_tab_rekm_dkm =

    32     0
     0     9
```



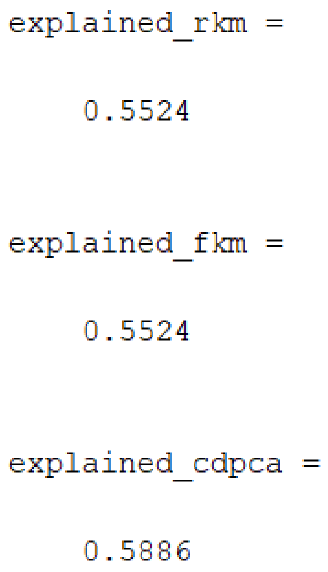
```
Cont_tab_fkm_cdpca =

    32     0
     0     9


Cont_tab_fkm_dkm =

    32     0
     0     9


Cont_tab_cdpca_dkm =

    32     0
     0     9
```

The output obtained by each one of these 6 commands is a 2x2 matrix with diagonal elements equal to 32 and 9; and off-diagonal elements equal to 0. 32 and 9 are the number of variables in the two clusters.

8 Compute the explained variance of the different methods from 4 to 6

The final explained variance values are computed as:

```
explained_rkm = sum(var(Yrkm))/24;
explained_fkm = sum(var(Yfkm))/24;
explained_cdpca = sum(var(Ycdpca))/24;
```



```
explained_rkm =

    0.5524


explained_fkm =

    0.5524


explained_cdpca =

    0.5886
```

The highest variance is explained by CDPCA (57.64%), followed by FKM (56.67%) and RKM (53.11%).

Conclusion

This study demonstrated how PCA-based clustering methods improve interpretability. Among them, CDPCA provides the highest explained variance. Future studies could explore Bayesian clustering approaches to improve classification robustness.

Matlab code

```
run("BLI2022.m")

% 1. Standardize Data;

Xs=zscore(X,1); %standardized data
Rx=(1/41)*Xs'*Xs; %correlation matrix
```

```
%2. Compute PCA to determine the number of Principal Components;

[A,L]=eigs(Rx,24); %A matrix of eigenvectors
%L matrix of eigenvalues
RQ=trace(L(1:7,1:7))/24; %Rating Quotient
Y=Xs*A(:,1:7); %Component Score matrix

%3 & 4. Compute K-means on the number of component identified in the step
% and Identify the best K with Pf

[cv2,Xm2]=kmeans(Y,2,'rep',100); %k-means application
I2=eye(2);
Ukm2=I2(cv2,:);
[pf,Dw,Db] = psF(Y,Ukm2) %pseudo-F index

[cv3,Xm3]=kmeans(Y,3,'rep',100);
I3=eye(3);
Ukm3=I3(cv3,:);
[pf,Dw,Db] = psF(Y,Ukm3)

[cv4,Xm4]=kmeans(Y,4,'rep',100);
I4=eye(4);
Ukm4=I4(cv4,:);
[pf,Dw,Db] = psF(Y,Ukm4)

%5. Compute Reduced K-means
[Urkm,Arkm, Yrkm,frkm,inrkm]=REDKM(Xs,2,7,100); %Reduced K-Means

%6. Compute Factorial K-means
[Ufkm,Afkm, Yfkm,ffkm,infkm]=FKM(Xs,2,7,100); %Factorial K-Means

%7. Compute Clustering and Disjoint PCA
[Vcdpca,Ucdpca,Acdpca, Ycdpca,fcdpca,incdpca]=CDPCA(Xs,2,7,100); %CDPCA

%8. Compute the Double K-means
[Vdkm,Udkm,Ymdkm, fdkm,indkm]=DKM(Xs,2,7, 100); %Double K-Means

%9. Compare the results by using the confusion matrix (contingency table) between
Cont_tab_rekm_fkm=Urkm'*Ufkm; %contingency tables
```

```
Cont_tab_rekm_cdpca=Urkm'*Ucdpca;
Cont_tab_rekm_dkm=Urkm'*Udkm;
Cont_tab_fkm_cdpca=Ufkm'*Ucdpca;
Cont_tab_fkm_dkm=Ufkm'*Udkm;
Cont_tab_cdpca_dkm=Ucdpca'*Udkm;

%10. Compute the explained variance of the different methods from 4 to 6
explained_rkm=sum(var(Yrkm))/24 %explained variances
explained_fkm=sum(var(Yfkm))/24
explained_cdpca=sum(var(Ycdpca))/24
```