Grenoble INP - Ense3

AI and Autonomous systems

Lab 3 - M2-MARS

Principal Component Analysis (PCA)

Author :
Souhaiel BEN SALEM

3 novembre 2021

Lab Objectives: The objective of this lab is to illustrate principal component analysis (PCA) on the olympic and iris datasets, compare its performance to LDA for dimensionality reduction and explore how it can improve the performance of a (SVM) classifier .

Introduction:

Principal components analysis (PCA) is one of a family of techniques for taking high-dimensional data, and using the dependencies between the variables to represent it in a more tractable, lower-dimensional form, without losing too much information. PCA is one of the simplest and most robust ways of doing such dimensionality reduction. It is also one of the oldest, and has been rediscovered many times in many fields, so it is also known as the Karhunen-Loève transformation, the Hotelling transformation, the method of empirical orthogonal functions, and singular value decomposition.

1 Notebook 1: PCA olympic decathlon data

1. **Q: Based on the summary statistics above, can you find the units for the score of the running event, and for the jumping or throwing ones?**
   The units for the score of the running events (100m, 400m,1500m and 110 hurdles) is **seconds (s)** and the unit for the jumping(long jump and high jump)/throwing (short put, discus and javeline) is the **meter (m)**.

1.1 Part I. PCA on raw data:

1. **Q: Does it seems possible to reduce the dimension of this dataset using PCA? Then how many components do you think are sufficient to explain these olympic data (justify)?**
   Yes it is clearly possible to the dimension of this dataset using PCA. In fact, we can clearly see that most of the dataset variance can be described by the **two first principal components PC1 and PC2**.

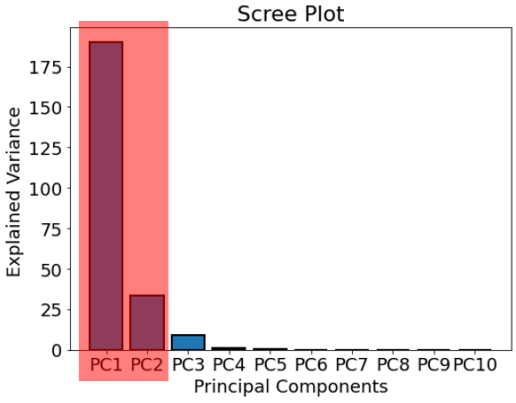

Figure 1: the weight of each PC in explainig the variance of the dataset

	sdev	varprop	cumprop
	Standard deviation	Proportion of Variance	Cumulative Proportion
PC1	13.780719	0.811567	0.811567
PC2	5.804303	0.143973	0.955540
PC3	2.960659	0.037459	0.992999
PC4	0.940615	0.003781	0.996780
PC5	0.698113	0.002083	0.998863
PC6	0.346530	0.000513	0.999376
PC7	0.247437	0.000262	0.999637
PC8	0.231472	0.000229	0.999866
PC9	0.156289	0.000104	0.999971
PC10	0.082793	0.000029	1.000000

Figure 2: PCA1 and PCA2 explain more than 95% of the total variance

2. **What is the essential loadings for the first principal component PC1? Thus what is the meaning of PC1? Same question for PC2.**
   The loading for PC1 are as follow:

   - 1 for the **1500m running event**
   - Around 0.05 for **Javelin** , around 0.2 for **discus throw** and around 0 for the other variable.

   the first principal component PCA1 is positively associated with longer times on the 1500. So, slower runners will have higher value on this component.

   The loading for PC2 are as follow:

   - 1 for the **Javelin** event
   - around 0 for the other variables (events)

   the second principal component PCA2 is positively associated with longer distances of the javelin throw event. So, participants with high performance (longer throwing distance) will have higher value on this component.

3. **How can we interpret the score of the athlete 9? Same question for athlete 23 and 16.**

   - **Athlete 9**: has a relatively high positive score on PC1 which means he is **a slow runner** and almost a 0 score on PC2 which mean he is **an average javeline thrower**.
   - **Athlete 23**: has a relatively high negative score on PC2 which means **his performance in the Javelin throwing event was poor** and almost very low negative score on PC1 which means he is **slightly better-than-average runner**.
   - **Athlete 16**: has a high positive score on both PC1 and PC2 which means he is **a slow runner** and **a good javelin thrower**.

4. **Why was this result expected just by looking at the descriptive statistics table of the variables displayed at the beginning of the notebook?**
   The results were expected because, as indicated by the statistics table in the beginning of the notebook, the 1500m running event and javelin event have the highest standard deviation which means that the variance is highest for these two events. In other words, most of the athletes performed similarly in the other events and most of the difference in performance occured during the 1500m and javelin events, so logically these two features (events) would be sufficient to draw meaningful information about the performance of each athlete (sample).

5. **Can we deduce that only these two events, namely the 1500-metre run and javelin throw, are really significant to measure the performance of decathlon athletes?**
   These two events are not enough enough to conclude on the performance of an athlete. For instance, we only took on account two factors (features) when in reality there are alot of events that athletes who performed poorly on the first two events could have performed well on.
   In addition, the variables used were not scaled (one measures the speed in seconds and the other one measures the strength of the throw by comparing distances measured in meter). Moreover, PCA tries to get the features with maximum variance and the variance is high for high magnitude features (1500m event vs 400m event for example) which makes it biased towards these events.

1.2 Part II. Standardizing: scale matters

1. **Check that PCA is just an eigendecomposition of the sample covariance matrix**
   We can compute the sample covariance matrix of **Xs** which is $\mathbf{Xs}^\top\mathbf{Xs}$ and then use the np.linalg.eigh() to get its eigendecomposition.Afterwards we can extract the two eigenvectors corresponding to the two largest eigenvalues. We can verify that these two eigenvectors are indeed the loadings for the two principal components PC1 and PC2.

```
from sklearn.preprocessing import StandardScaler

# Center and reduce the variables
scaler = StandardScaler()
Xs = scaler.fit_transform(olympic)

# Make PCA on standardized variables
pca_s = PCA() # estimate only 2 PCs
Xs_pc = pca_s.fit_transform(Xs) # project the original data into the PCA space
Xs_transp=Xs.T
print(Xs.shape)
print(Xs_transp.shape)
cov_mat=Xs_transp @ Xs
print(cov_mat.shape)
w,v=np.linalg.eigh(cov_mat)
print(w.shape)
print(v.shape)
v_trans=v.T
PC1_loadings=v_trans[-1:]
PC2_loadings=v_trans[-2:-1,:]
print('PC1_loading are:',PC1_loadings.T)
print('PC2_loading are:',PC2_loadings.T)
```

Figure 3: Computing the eigendecomposition of Xs and extracting the first two eigenvectors

```
PC1_loading are: [[-0.41588233]
 [ 0.39405149]
 [ 0.26910572]
 [ 0.21228177]
 [-0.35584739]
 [-0.43348158]
 [ 0.17579228]
 [ 0.38408214]
 [ 0.17994361]
 [-0.17014262]]
PC2_loading are: [[ 0.14880813]
 [-0.1520815 ]
 [ 0.48353737]
 [ 0.0278985 ]
 [ 0.35215981]
 [ 0.0695682 ]
 [ 0.50333471]
 [ 0.14958202]
 [ 0.371957  ]
 [ 0.42096528]]
```

Figure 4: the eigenvectors corresponding to the loadings of PC1 and PC2

2. **After standarization of the olympic data, is the 1500 event still the more important to explain the variance?**
   After standarization, we can see that the 1500m event is not the more important in explaining the variance along PC1 as all the other events have larger loading (weights). It does not also have the most significant weight that explains the variance along PC2 as the ”poid” and ”disc” events have larger loadings.
   This is also shown by the biplot as the 1500 event vector does not have the biggest magnitude out of all the features like before.

3. **In this standardized case what are the overall meanings of the first two principal components? Are the loadings also consistent with these conclusions?**
   The first principle component PC1 characterizes the performance in the **speed** events (the speed events have higher loadings compared to strength events so they characterize the variance along PC1) whereas The second principle component PC2 characterizes the performance in the **strength** events.

4. **How can you interpret that the arrows for the jump events are in the opposite direction than the sprint ones?**
   The fact that the arrows for the jump events are in the opposite direction than the sprint ones means that these events are **negatively correlated** which in itself means that good sprinters are not

necessarily good at jumping events. This is explained by the fact that an athlete needs explosive power more than speed to be successful at the jumping event,

5. **How can you interpret the score of athlete 1, 16, 32 respectively (weakness/strength)?**
   - **Athlete 1** is a good runner as he scored negatively along the PC1 which is mostly correlated with sprinting events and he had an average performance in the strength events as he scored nearly 0 on the PC2 axis.
   - **Athlete 16** has a very good performance in strength events and a slightly lower than average performance when it comes to speed events.
   - **Athlete 32** has poor performance in both categories (strength events and speed events) as he scored highly positively on PC1 and highly negatively on PC2.
6. **In your opinion, is it better (i.e. more useful) to perform PCA on standardized or non-standardized data for this example?**
   Performing PCA on standardized variables is more useful because before applying PCA we need to make sure that variables are not measured in different scales (e.g: kilograms, kilometers, centimeters, ...); otherwise, the PCA outputs obtained will be severely affected (biased towards the feature with larger variance).
   Scaling and standardizing data ensures that we get the best components axis to project our data on and derive the most useful and relevant insights.
7. **Is it still possible to explain most of the variance with a much reduced number of components?**
   The cumulative explained variance plot shows that in order to explain most of the variance (at least 90%) we need 6 components. Two components are not enough anymore.

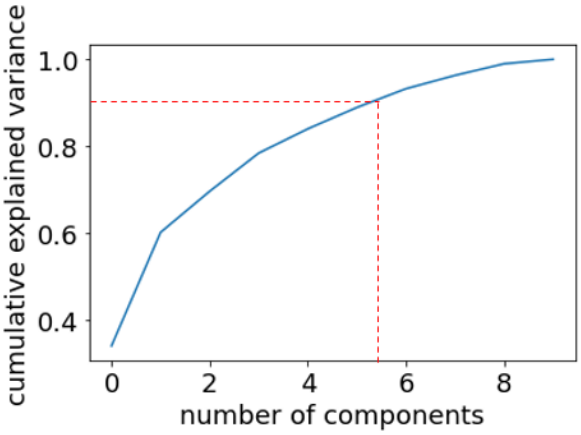

Figure 5: cumulative explained variance

8. **Why this is rather a good news for the sporting interest of decathlon?**
   The fact that variance needs more than half of the features **(events)** to be explained means that most of the sporting events are competitive and gives the audience a much better experience watching the Dechatlon Olympic event.

2 Notebook 2: PCA vs LDA

1. **Does the PCA requires the class label to transform the dataset (justify)?**
   PCA is an unsupervised technique , thus it does not require class labels to transform the dataset. Instead, PCA geometrically project a data set onto fewer dimensions, where the new variables are the principal components. This is done in such a way that the principal components are orthogonal and have the largest possible variances. In fact, we don't apply PCA to labeled data because sometimes labels might not be correlated with the variance of the features and that makes this technique susceptible to throwing strong classification signals away.

2. **How many PCA principal component seems sufficient to mostly explain the variance of this dataset?**
   From the cumulative explained variance plot, we can see that more than 90% of the data variance can be explained by **the two first principal components PC1 and PC2**.

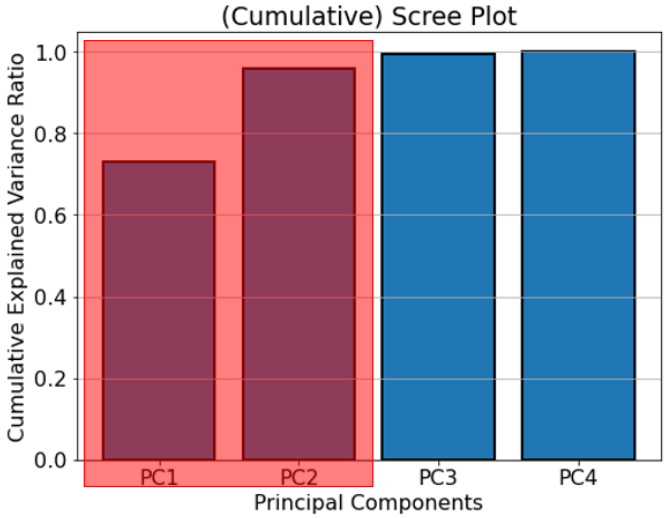

Figure 6: PC1 and PC2 explain most of the variance

3. **How many PCA principal component seems sufficient to get a quite accurate separation between the classes?**
   The first two principal components seem enough to get a quite accurate class separation but if we want to be more accurate (to completely eliminate the overlapping between classes) we should take the first three components PC1, PC2 and PC3 as they explain almost the totality of the variance.

4. **What is the 'explained variance ratio' criterion for PCA, i.e. variance of what? Same question for LDA**.
   for PCA , the explained variance ratio indicates how informative a principal component PCx in explaining the total variance of the dataset. Mathematically speaking, this is the ratio of the eigenvalue of the covariance matrix associated with a certain principal component (i.e eigenvector and the sum of all eigenvalues of the covariance matrix (i.e all the variance in the data).
   In LDA, we are observing the **discriminant components** and each contribute to the **In-between-classes** variance. So, when we talk about explained variance ration for LDA, we are trying to find out how much a certain discriminant component contributes to the overall in-between-class variance.
5. **Explain why the explained variance ratio is 1 for the LDA with two components**.
   For the Iris dataset, we have 3 classes (i.e: Iris Setosa, Iris Versicolour and Iris Virginica). Since for LDA classifier, there is a dimensionality reduction by linear projection onto a $K-1$ dimensional space, where K is the total number of target classes which 3 in our case , **it is expected that two discriminant components would totally explain the variance**.
6. **Does LDA allows us to get an accurate separation between the classes when using only the first LDA vector?**
   The first LDA vector would allow for accurate between-class separation but if we need to get better insights into the distribution of points within each class we need to also consider the second discriminant component.
7. **Do you think that PCA can be useful to transform this dataset for visualization or dimension reduction?**
   PCA will cluster the data by projecting it on the principal components basis. However PCA does not consider labels so it can be useful to eliminate some redundancy in the data but not for classification task.

3 Notebook 2: SVM face Recognition

1. **What is the interest to perform dimension reduction (PCA) before applying the classifier?**
   Applying the PCA algorithm help reduce the number of informative dimensions we need to understand our dataset.In our case, the eigenfaces dataset has 1850 features and 1217. These numbers would make it difficult for the svm classifier to learn all the features from that number of samples.
   After reducing the dimensions of our data, the SVM algorithm would perform better in the new basis the data is projected in.
2. **Plot the mcr errors curves as a function of the number of retained pca features. What happen if we decrease/increase this number?**
   When augmenting the number of components, the misclassification rate (msc) would drop.
   For this purpose, I wrote a function that returns the **accurary** and **MSE** for each number of components, starting from 1 and going up to a number n set by the user.

```
def pca_SVC_mse_accuracy(n):
    """
    Creates a PCA model with n components,
    reduces the dimensionality of the validation data set,
    then computes two error metrics:
    * Percent correct predictions using SVC
    * MSE (X_hat minus original X)
    """


    # split into a training and testing set
    X_training, X_testing, y_training, y_testing = train_test_split(
    X, y, test_size=0.25, random_state=42)
    X_std, _, _ = get_normed_mean_cov(X_training)
    X_std_validation, _, _ = get_normed_mean_cov(X_testing)

    #########
    # building PCA/SVC
    # Train on training data

    # Make PCA
    # Make PCA
    pca = PCA(n_components=n, whiten=True)
    pca.fit(X_std)
    X_red = pca.transform(X_std)

    # build SVC
    linclass = SVC()
    linclass.fit(X_red,y_training)
    ########

    # Transform inputs and feed them to linear classifier
    y_validation = y_testing
    X_red_validation = pca.transform(X_std_validation)
    yhat_validation = linclass.predict(X_red_validation)

    # Evaluate confusion matrix for predictions
    cm = confusion_matrix(y_validation,yhat_validation)
    total = cm.sum(axis=None)
    correct = cm.diagonal().sum()
    accuracy = 1.0*correct/total


    # MSE associated:
    X_orig = X_std
    X_hat = pca.inverse_transform( pca.transform(X_orig) )
    mse = ((X_hat - X_orig)**2).mean(axis=None)

    return mse, accuracy


mses = []
accuracies = []
N = 50
for i in range(1,N):
    m, a = pca_mse_accuracy(i)

    print ("%d-component PCA: MSE = %0.4g, Accuracy = %0.2f"%(i,m,a*100.0))

    mses.append((i,m))
    accuracies.append((i,a))

mses = np.array(mses)
accuracies = np.array(accuracies)
```

Figure 7: MSE and Accuracy computing

The results for $n = 33$ are as follow:

```
1-component PCA: MSE = 0.7764, Accuracy = 43.61
2-component PCA: MSE = 0.6116, Accuracy = 43.28
3-component PCA: MSE = 0.5366, Accuracy = 47.87
4-component PCA: MSE = 0.4829, Accuracy = 49.51
5-component PCA: MSE = 0.4487, Accuracy = 49.51
6-component PCA: MSE = 0.4195, Accuracy = 51.80
7-component PCA: MSE = 0.3979, Accuracy = 53.11
8-component PCA: MSE = 0.3797, Accuracy = 55.74
9-component PCA: MSE = 0.3624, Accuracy = 54.43
10-component PCA: MSE = 0.3463, Accuracy = 57.38
11-component PCA: MSE = 0.3318, Accuracy = 60.98
12-component PCA: MSE = 0.318, Accuracy = 63.28
13-component PCA: MSE = 0.3055, Accuracy = 62.62
14-component PCA: MSE = 0.2951, Accuracy = 64.92
15-component PCA: MSE = 0.2853, Accuracy = 66.89
16-component PCA: MSE = 0.276, Accuracy = 67.87
17-component PCA: MSE = 0.2669, Accuracy = 69.51
18-component PCA: MSE = 0.2589, Accuracy = 69.51
19-component PCA: MSE = 0.2518, Accuracy = 73.44
20-component PCA: MSE = 0.2453, Accuracy = 72.79
21-component PCA: MSE = 0.2389, Accuracy = 75.08
22-component PCA: MSE = 0.2328, Accuracy = 78.36
23-component PCA: MSE = 0.2271, Accuracy = 79.34
24-component PCA: MSE = 0.2215, Accuracy = 79.67
25-component PCA: MSE = 0.2164, Accuracy = 79.67
26-component PCA: MSE = 0.2115, Accuracy = 80.00
27-component PCA: MSE = 0.2068, Accuracy = 80.98
28-component PCA: MSE = 0.2024, Accuracy = 80.98
29-component PCA: MSE = 0.1984, Accuracy = 80.00
30-component PCA: MSE = 0.1944, Accuracy = 80.98
31-component PCA: MSE = 0.1907, Accuracy = 82.95
32-component PCA: MSE = 0.187, Accuracy = 81.97
```

Figure 8: MSE and Accuracy computed as a function of the number of PCs

Afterwards, we can plot the accuracy as a function of the number of components:

```
# Now plot the accuracy
plt.plot(accuracies[:,0],accuracies[:,1],'bo-')
plt.title("Percent Correct: Accuracy of Predictions",size=14)
plt.xlabel("Number of Principal Components")
plt.ylabel("Percent Correct")
plt.show()
```

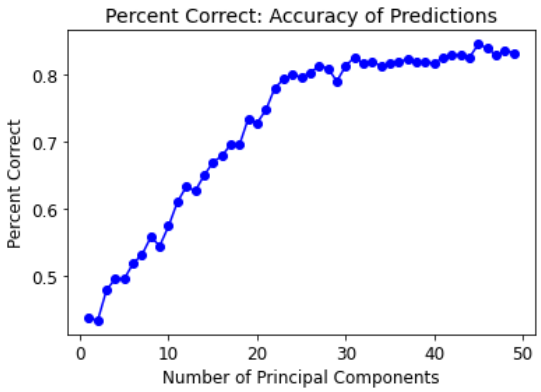

Figure 9: MSE and Accuracy computed as a function of the number of PCs

Conclusion:

Principal component analysis (PCA) has been called one of the most valuable results from applied linear algebra. It provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it. During this Lab we examined the PCA technique for dimensionality reduction and saw the importance of data standardization, its advantages and disadvantages , how it compares to LDA and how it can be used as a preprocessing technique to assist an SVM classifier.