After choosing K-components in PCA how do we find out which components(names of the columns) have algorithm selected?

Asked 10 months ago Active 10 months ago Viewed 623 times

1

I am new to Data Science and I need some help to understand PCA.I know that each of columns constitute one axis,but when PCA is done and components are reduced to some k value,How to know which all columns got selected?

python-3.x k-means pca sklearn-pandas

asked May 26 '19 at 20:25

Ravi Biradar

41 ● 6

How is this python-3.x? There is no python code here, nor is there any reference to python libraries. – Finomnis May 26 '19 at 20:55

Show us the scikit-learn code you are having trouble with. – J_H May 26 '19 at 22:10

I am not facing code issue,I am not able to understand after PCA which all the columns it is taking.Ex: let's say we have 4 columns a,b,c and d. We found after PCA just the 2 components are able to explain 95% of the data.So which are the columns referring to these components? – Ravi Biradar May 27 '19 at 7:12

2 Answers

Active Oldest Votes

3

In PCA you compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components.
Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on.

Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data. s there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set.

According to my experience, if the percentage of cumulative sum of Eigen values can over 80% or 90%, the transformed vectors will be enough to represent the old vectors.

To explain clearly lets use @Nicholas M's code.

```
pca = PCA(n_components=1)
pca.fit(X)
```

You must increase the n_components to get %90 variance.

Input:

```
pca.explained_variance_ratio_
```

Output:

```
array([0.99244289])
```

On this example just 1 component is enough.

I hope its all clear to understand.

Resources:
https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60
https://towardsdatascience.com/a-step-by-step-explanation-of-principal-component-analysis-b836fb9c97e2

edited May 28 '19 at 8:58

answered May 27 '19 at 19:18

Alperen Tahta
188 ● 9

How are the components and columns related?How this results in dimensionality reduction? – Ravi Biradar May 28 '19 at 8:39

In PCA we not just reduce dimensions also we change the dimesions. As ı mention, 10-dimensional data gives you 10 principal components and this components are our new columns but with one difference, the new columns carry the maximum possible information in the first column. – Alperen Tahta May 28 '19 at 8:44

0

You have to look at Eigenvectors of the PCA. Each Eigenvalues are the "force" of each "*new axis*" and the eigenvector provide the linear combination of your original features.

With scikit-learn, you should look at the attribute **components**_

```
import numpy as np
from sklearn.decomposition import PCA
X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
pca = PCA(n_components=2)
pca.fit(X)
print(pca.components_) # << eigenvector matrix
```

answered May 27 '19 at 18:26

Nicolas M.
1,037 ● 1 ● 6 ● 19

How are the components and columns related?How this results in dimensionality reduction? –

variance covered is equal to the sum of kept Eigenvalues/ sum of all eigenvalues. If you have 10 features, **components**_ will be a 10x10 matrix. If you need only the 4 first components to have 80% of the variance, you keep only the first 4 columns to have a 10x4 matrix. – Nicolas M. May 28 '19 at 8:48

How do we know for sure that first 4 components represent first 4 columns only? – Ravi Biradar May 28 '19 at 8:58

Actually that 4 component not represents just first four column,but represent the all data because your new columns(Principle Components) carry the most of the data on first columns. If you worry about the reason of it you can look at towardsdatascience.com/... – Alperen Tahta May 28 '19 at 11:19