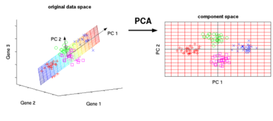
Dimensional Reduction Illustration

Untuk menceritakan bagaimana wajah pacar kita waktu SMA, tidak perlu disebutkan hidungnya mancung, kulitnya halus, rambutnya indah tergerai dan sebagainya. Tapi cukup katakan 'Pacar saya waktu SMA orangnya cantik'. Kata 'cantik' sudah mampu menggambarkan uraian sebelumnya.

Misalnya kita memiliki 20 variabel, kita reduksi menjadi 4 variabel, yang mana 4 variabel tersebut mampu merepresentasikan 20 variabel.

Unsupervised Learning (Dimensional Reduction)



- Original data space menggunakan 3 variabel yaitu Gene1, Gene2, Gene3. Hasil clustering 4 kelompok.
- Kemudian direduksi menjadi 2 variabel yaitu PC1 dan PC2. Dan hasil clustringnya tetap 4 dan lebih sederhana serta lebih cepat.

Principal Component Analysis (PCA)

1. Dimensionality Reduction
2. Increasing interpretability but at the same time minimizing information loss

Main Purpose:

1. Data Compression - run the algorithm quickly: it's not intended to improve the performance in accuracy
2. Data Visualization - understand data better

Basic Idea

PCA reduces the dimensions of a data set by projecting the data onto a lower dimensional subspace.

Variabel
X1, X2, X3,.....,Xp

→ **PCA**
PC1, PC2, PC3,.....,PCp

Jumlah PC yang dipilih<p

Variabel X1, X2, X3, ..., XP akan direduksi menggunakan PCA yang mana akan membentuk Variabel Baru, PC1, PC2, PC3,...,PCP

Nilai PC yang diambil kurang dari p. Misalnya P=10 maka diambil kurang dari 10.

> PC Merupakan kombinasi linear dari variabel yang ada
>
> Informasi yang terkandung pada PC merupakan gabungan dari semua variabel dengan bobot tertentu

How to Search PC value?

PC1=B11*X1+B12*X2+B13*X3+.....+B1p*Xp
PC2=B21*X1+B22*X2+B23*X3+.....+B2p*Xp
.
.
PCp=Bp1*X1+Bp2*X2+Bp3*X3+.....+Bpp*Xp

- B11, B12 dan seterusnya merupakan tertentu.

> Kandungan informasi PC1 > PC2 > PC3 > PC4 > ..... > PCp.
>
> Antar PC saling independen atau tidak saling berkorelasi satu sama lain

> Besarnya kandungan informasi pada PC diukur oleh nilai eigenvalue.
>
> Nilai PC yang digunakan adalah yang memiliki eigenvalue > 1.

	Eigenvalues of the Correlation Matrix			
	Eigenvalue	Difference	Proportion	Cumulative
1	4.11495951	2.87623708	0.5879	0.5879
2	1.23872183	0.51290521	0.1770	0.7648
3	0.72581663	0.40938658	0.1037	0.8685
4	0.31643205	0.05845759	0.0452	0.9137
5	0.25797446	0.03593499	0.0369	0.9506
6	0.22203947	0.09798342	0.0317	0.9823
7	0.12405606		0.0177	1.0000

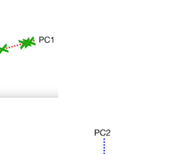
> Dari tabel dapat diketahui bahwa PC yang memiliki eigenvalue > 1 adalah PC1 dan PC2. Dengan demikian, yang mana terdapat 7 variabel akan direduksi menjadi 2 variabel.
>
> Informasi yang terkandung dalam PC1 adalah 58,79% (lihat proportion). Informasi yang terkandung dalam PC2 adalah 17,70% dan seterusnya
>
> Berdasarkan eigenvalue, kita hanya menggunakan PC1 & PC2 maka total informasi dari hasil reduksi adalah 76,48% maka sisanya merupakan informasi yang hilang.

Step by Step: The Data

- Contoh kita memiliki 2 variabel yaitu, Gene 1 and Gene 2, yang didalamnya terdapat 6 data.
- Calculate the mean by X. Berdasarkan 6 data, rata2 di sumbu x, y dan gen sama seperti k-means. Untuk menegetahui centroid (titik pusat).

Alert: there will be so many mathematical terminology when we try to talk about PCA. I'll try to make this quick to elaborate it in a nutshell

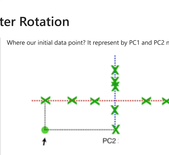
Step by Step: Data Projection



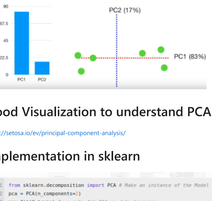
- Set the mean in position (0,0) by shift the data.

Step by Step: Calculate Sum of Square (Eigenvalue)

- Try to fit a line by finding largest sum of square between data point and (0,0). that is PC1



Step by Step: Find the PC2

> Find the PC2 that is perpendicular to PC1
>
> 
>

Step by Step: Calculate Eigenvalue

> Calculate Sum of Square (Eigenvalue) for PC2
>
> $d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances = SS(distances)
>
> SS(distances for PC2) = Eigenvalue for PC2
>
> Lastly, the **Eigenvalue for PC2** is the sum of squares of the distances between the projected points and the origin.
>
>
>

Mencari garis linear yang melewati sumbu(0, 0) yang mana memiliki nilai d terbesar.

- Mencari garis yang memiliki nilai sum of squared distances terbesar.
- sum of squared merupakan eigenvalue dari PC1

Step by Step: Result

- Mencari sum of squared untuk PC2 yang mana yang tegak lurus dengan garis merah PC1 dan melewati poin (0,0)

>
>
>
> Rotate it so PC1 need to be in horizontal position

After Rotation

> Where our initial data point? It represent by PC1 and PC2 now (the plot is different to get a better representation)
>
>
>

Step by Step: Calculate the Variation of PC1 and PC2

The bar plot in the left side shows the percentage of proportion(eigenvalue) between PC1 and PC2. To calculate it by using

proportion_PC1=(eigenvalue_in_PC1) / (eigenvalue_in_PC1+eigenvalue_in_PC2)

proportion_PC2=(eigenvalue_in_PC2) / (eigenvalue_in_PC1+eigenvalue_in_PC2)

Good Visualization to understand PCA

https://setosa.io/ev/principal-component-analysis/

Implementation in sklearn

```
from sklearn.decomposition import PCA # Make an instance of the Model
pca = PCA(n_components=2)
pca.fit(X_train) # we only fit PCA on data training
data = pca.transform(X)
```

executed in 9ms, finished 18:11:17 2020-04-06

PCA Applications

1. Handles clustering with lots of variables
2. Handles Multicolinearity in Linear Regression and Logistic Regression