SCIENCE ACADEMY
Excellence in Teaching & Learning

Principal Component Analysis (PCA)
(Unsupervised learning)

Hi guys, welcome to the Principal Component Analysis Lecture!

In this lecture, we will talk about “Principle Component Analysis (PCA)”. As an unsupervised learning statistical procedure, PCA involves only a set of features *(X_1, X_2, X_3, …, X_p).*

Good to know!

PCA was invented in 1901 by Karl Pearson. Pearson was the student of Sir. Galton who coined the term regression in 1875. The original work by Pearson can be found here.

✅ *Optional Readings and References:*
sklearn’s Official Documentation PCA
Introduction to Statistical Learning - Chapter 10
Machine Learning - A Probabilistic Perspective Chapter 12 is also a good read!

Dr. Junaid S. Qazi PhD

General message: *Key concepts along with significant commentary / text is provided in the slides, so that they serve as a reference for the respective theory lecture. However, the suggested readings are recommended to explore more on the topic under discussion!*

Principal Component Analysis (PCA)

Lets create a simple scenario to learn the idea behind PCA:

- Suppose, we have a dataset with *"p"* features *(X_1, X_2, X_3, …, X_p)* and *"n"* observations.
- If we want 2D scatterplots (with n observations or measurements in each plot), for *p = 10*, there will be 45 scatterplots "*p(p-1)/2*". What if you get more features?
- Another very important question is, do you need all those plots? Most likely, many of them may not be informative (I should say none of them will be informative, since each plot contain just a very small fraction of the total information present in the dataset with 10 features).
- Clearly, a better method is required to visualize such dataset of *"n"* observations when *"p"* is large.
- In particular, we would like to find a low-dimensional representation of the data that captures as much of the information as possible. *For instance, if we can obtain a two-dimensional representation of the data that captures most of the information, then we can plot the observations in this low-dimensional space.*

Principal Component Analysis (PCA)

Lets create a simple scenario to learn the idea behind PCA:

- Suppose, we have a dataset with *“p”* features $(X_1, X_2, X_3, \ldots, X_p)$ and *“n”* observations.
- If we want 2D scatterplots (with n observations or measurements in each plot), for $p = 10$, there will be 45 scatterplots “$p(p-1)/2$”. What if you get more features?
- Another very important question is, do you need all those plots? Most likely, many of them may not be informative (I should say none of them will be informative, since each plot contain just a very small fraction of the total information present in the dataset with 10 features).
- Clearly, a better method is required to visualize such dataset of *“n”* observations when *“p”* is large.
- In particular, we would like to find a low-dimensional representation of the data that captures as much of the information as possible. *For instance, if we can obtain a two-dimensional representation of the data that captures most of the information, then we can plot the observations in this low-dimensional space.*

Principal Component Analysis (PCA)

Lets create a simple scenario to learn the idea behind PCA:

- Suppose, we have a dataset with *"p"* features $(X_1, X_2, X_3, \ldots, X_p)$ and *"n"* observations.
- If we want 2D scatterplots (with n observations or measurements in each plot), for $p = 10$, there will be 45 scatterplots "$p(p-1)/2$". What if you get more features?
- Another very important question is, do you need all those plots? Most likely, many of them may not be informative (I should say none of them will be informative, since each plot contain just a very small fraction of the total information present in the dataset with 10 features).
- Clearly, a better method is required to visualize such dataset of *"n"* observations when *"p"* is large.
- In particular, we would like to find a low-dimensional representation of the data that captures as much of the information as possible. *For instance, if we can obtain a two-dimensional representation of the data that captures most of the information, then we can plot the observations in this low-dimensional space.*

Principal Component Analysis (PCA)

Lets create a simple scenario to learn the idea behind PCA:

- Suppose, we have a dataset with *"p"* features *(X_1, X_2, X_3, …, X_p)* and *"n"* observations.
- If we want 2D scatterplots (with n observations or measurements in each plot), for *p = 10*, there will be 45 scatterplots *"p(p-1)/2"*. What if you get more features?
- Another very important question is, do you need all those plots? Most likely, many of them may not be informative (I should say none of them will be informative, since each plot contain just a very small fraction of the total information present in the dataset with 10 features).
- Clearly, a better method is required to visualize such dataset of *"n"* observations when *"p"* is large.
- In particular, we would like to find a low-dimensional representation of the data that captures as much of the information as possible. *For instance, if we can obtain a two-dimensional representation of the data that captures most of the information, then we can plot the observations in this low-dimensional space.*

Principal Component Analysis (PCA)

Lets create a simple scenario to learn the idea behind PCA:

- Suppose, we have a dataset with *"p"* features *(X_1, X_2, X_3, …, X_p)* and *"n"* observations.
- If we want 2D scatterplots (with n observations or measurements in each plot), for *$p = 10$*, there will be 45 scatterplots *"$p(p-1)/2$"*. What if you get more features?
- Another very important question is, do you need all those plots? Most likely, many of them may not be informative (I should say none of them will be informative, since each plot contain just a very small fraction of the total information present in the dataset with 10 features).
- Clearly, a better method is required to visualize such dataset of *"n"* observations when *"p"* is large.
- In particular, we would like to find a low-dimensional representation of the data that captures as much of the information as possible. *For instance, if we can obtain a two-dimensional representation of the data that captures most of the information, then we can plot the observations in this low-dimensional space.*

Principal Component Analysis (PCA)

PCA provides a tool to do just this:

- It finds a low-dimensional representation of a data set that contains as much as possible of the variation.
- The idea is that each of the *“n”* observations lives in *“p-dimensional”* space, but not all of these dimensions are equally interesting.
- PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension.

Principal Component Analysis (PCA)

PCA provides a tool to do just this:

- It finds a low-dimensional representation of a data set that contains as much as possible of the variation.
- The idea is that each of the *“n”* observations lives in *“p-dimensional”* space, but not all of these dimensions are equally interesting.
- PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension.

Principal Component Analysis (PCA)

PCA provides a tool to do just this:

- It finds a low-dimensional representation of a data set that contains as much as possible of the variation.
- The idea is that each of the *“n”* observations lives in *“p-dimensional”* space, but not all of these dimensions are equally interesting.
- PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension.

Principal Component Analysis (PCA)

PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called Principal Components.

Principal Component Analysis (PCA)

PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called Principal Components.

If you recall, regression determines a line of best fit to a dataset, whereas PCA determines several orthogonal lines of best fit to the dataset.

These orthogonal (means at right angle) lines are perpendicular to each other in *p-dimensional* space *(p-dimensional space is the variable sample space, if the data set have 5 variables or features the sample space will be 5-dimensional)*.

Principal Component Analysis (PCA)

PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called Principal Components.

If you recall, regression determines a line of best fit to a dataset, whereas PCA determines several orthogonal lines of best fit to the dataset.
These orthogonal (means at right angle) lines are perpendicular to each other in *p-dimensional* space *(p-dimensional space is the variable sample space, if the data set have 5 variables or features the sample space will be 5-dimensional)*.

Let's explore with a simple dataset having two features or dimension *X* and *Y*.

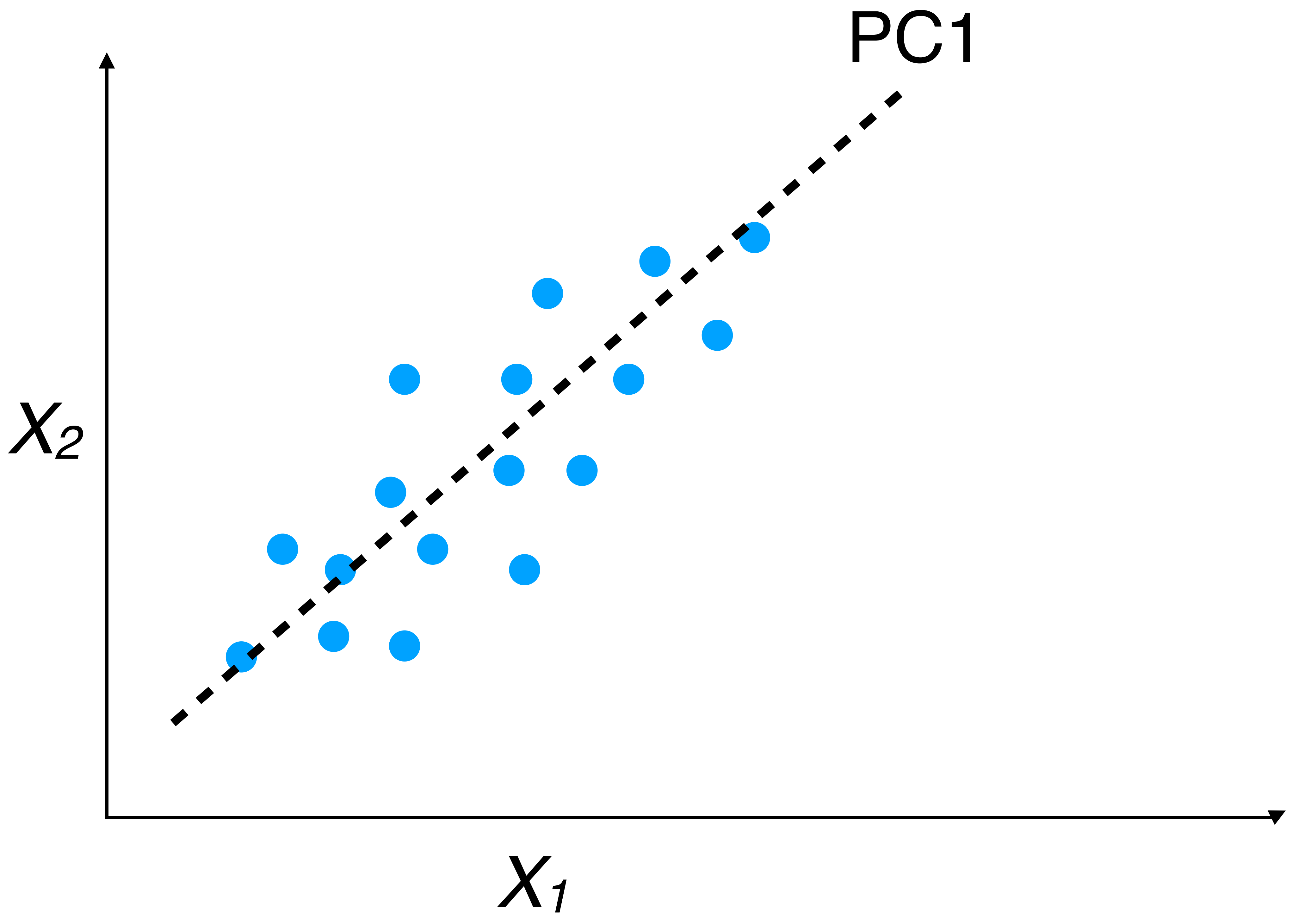
Principal Component Analysis (PCA)

Let's consider:

We have a dataset (shown in the plot) with 2 features or dimensions *X_1 and X_2.*

Black dotted is the regression line of the best fit to the data. This is the First Principal Component (PC1) capturing the maximum variance in the data.

PC1 determines the direction of highest variability in the data. Larger the variability captured in first component, larger the information captured by component. No other component can have variability higher than first principal component. The first principal component results in a line which is closest to the data i.e. it minimizes the sum of squared distance between a data point and the line.



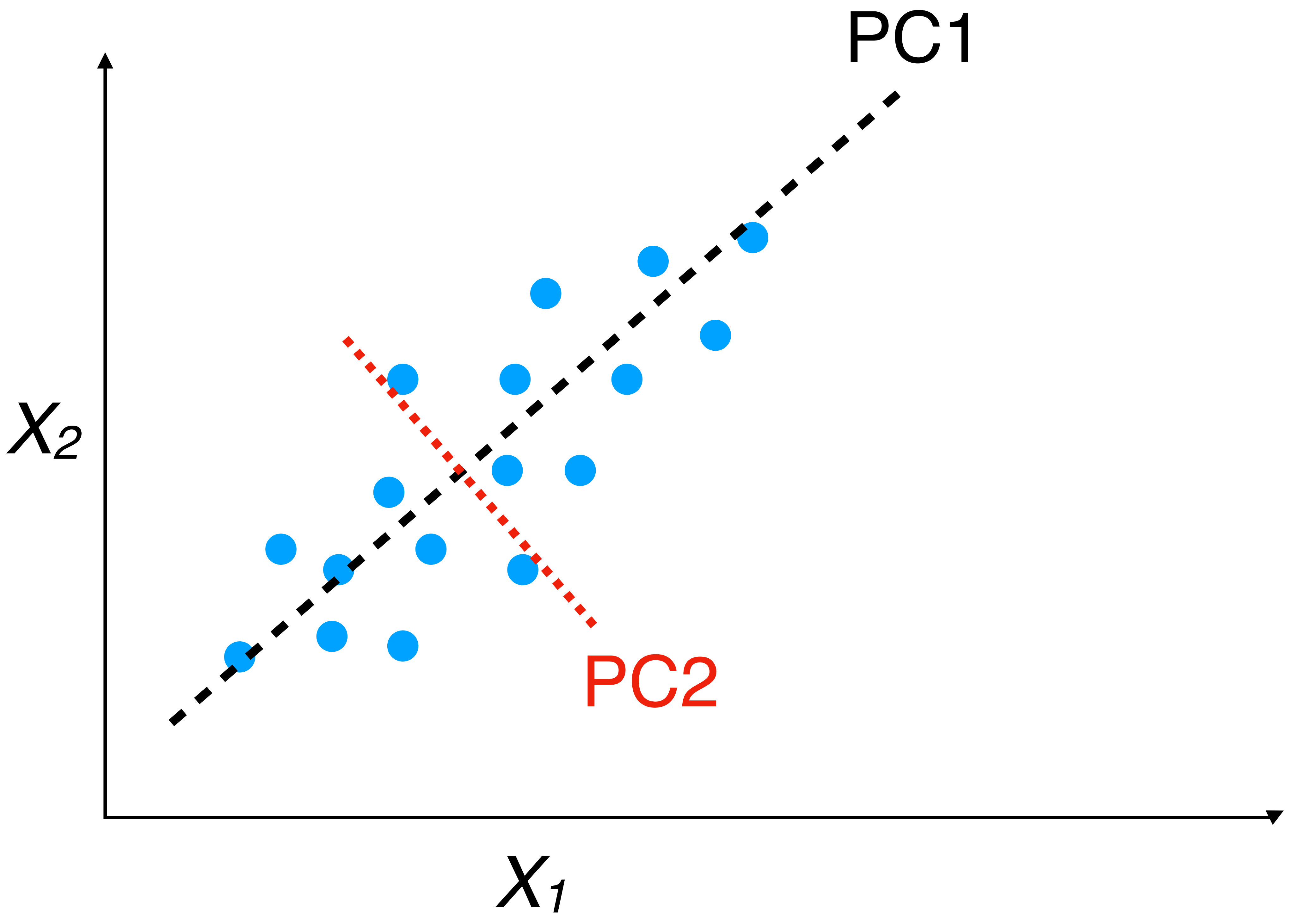
Principal Component Analysis (PCA)

Let's consider:

We have a dataset (shown in the plot) with 2 features or dimensions *X_1 and X_2.*

Black dotted is the regression line of the best fit to the data. This is the First Principal Component (PC1) capturing the maximum variance in the data.

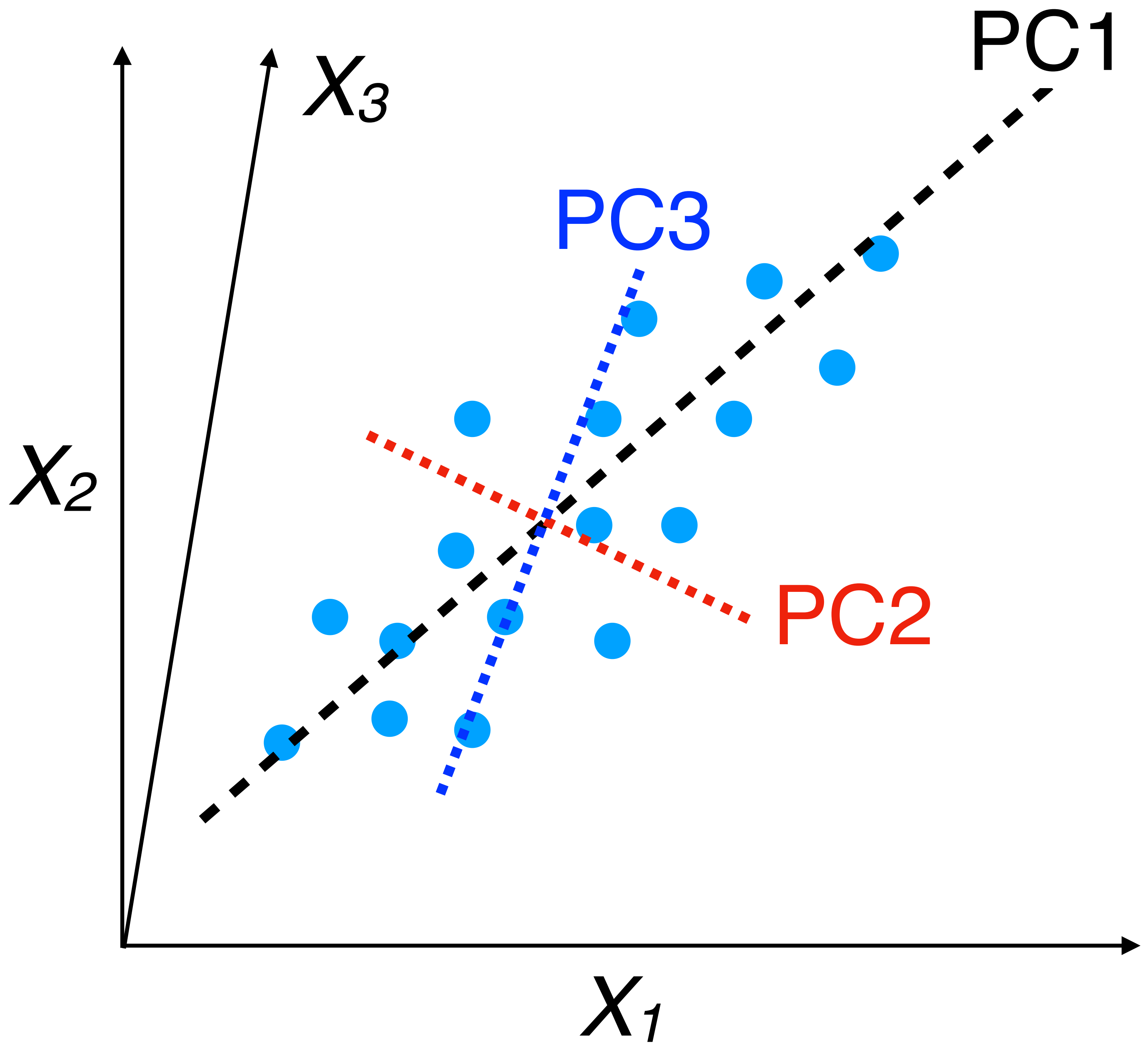
PC1 determines the direction of highest variability in the data. Larger the variability captured in first component, larger the information captured by component. No other component can have variability higher than first principal component. The first principal component results in a line which is closest to the data i.e. it minimizes the sum of squared distance between a data point and the line.



Let's add another line PC2, orthogonal to PC1. PC2 will capture the remaining variance in the dataset and is uncorrelated with PC1. (If the tow components are uncorrelated, their direction is orthogonal to each other in the sample space, as show in the figure.

Principal Component Analysis (PCA)

Let's add another dimension, $X_{3,}$ as shown in the plot below



All subsequent principal component follows a similar concept i.e. they capture the remaining variation without being correlated with the previous component.

Principal Component Analysis (PCA)

Good to remember:

- Using PCA on a dataset with large number of features, *p*, we can compress the amount of explained features to just a few components. However, the challenging part of PCA is to interpret the components.
- The principal components are supplied with normalized version of original features. This is because, the original features may have different scales. Imagine a data set with features measuring units as gallons, kilometre, light years etc. It is definite that the scale of variances in these variables will be large.
- Performing PCA on un-normalized features will lead to insanely large loadings for features with high variance. In turn, this will lead to dependence of a principal component on the feature with high variance, which is undesirable.

Principal Component Analysis (PCA)

Good to remember:

- Using PCA on a dataset with large number of features, *p*, we can compress the amount of explained features to just a few components. However, the challenging part of PCA is to interpret the components.
- The principal components are supplied with normalized version of original features. This is because, the original features may have different scales. Imagine a data set with features measuring units as gallons, kilometre, light years etc. It is definite that the scale of variances in these variables will be large.
- Performing PCA on un-normalized features will lead to insanely large loadings for features with high variance. In turn, this will lead to dependence of a principal component on the feature with high variance, which is undesirable.

Principal Component Analysis (PCA)

Good to remember:

- Using PCA on a dataset with large number of features, *p*, we can compress the amount of explained features to just a few components. However, the challenging part of PCA is to interpret the components.
- The principal components are supplied with normalized version of original features. This is because, the original features may have different scales. Imagine a data set with features measuring units as gallons, kilometre, light years etc. It is definite that the scale of variances in these variables will be large.
- Performing PCA on un-normalized features will lead to insanely large loadings for features with high variance. In turn, this will lead to dependence of a principal component on the feature with high variance, which is undesirable.

Principal Component Analysis (PCA)

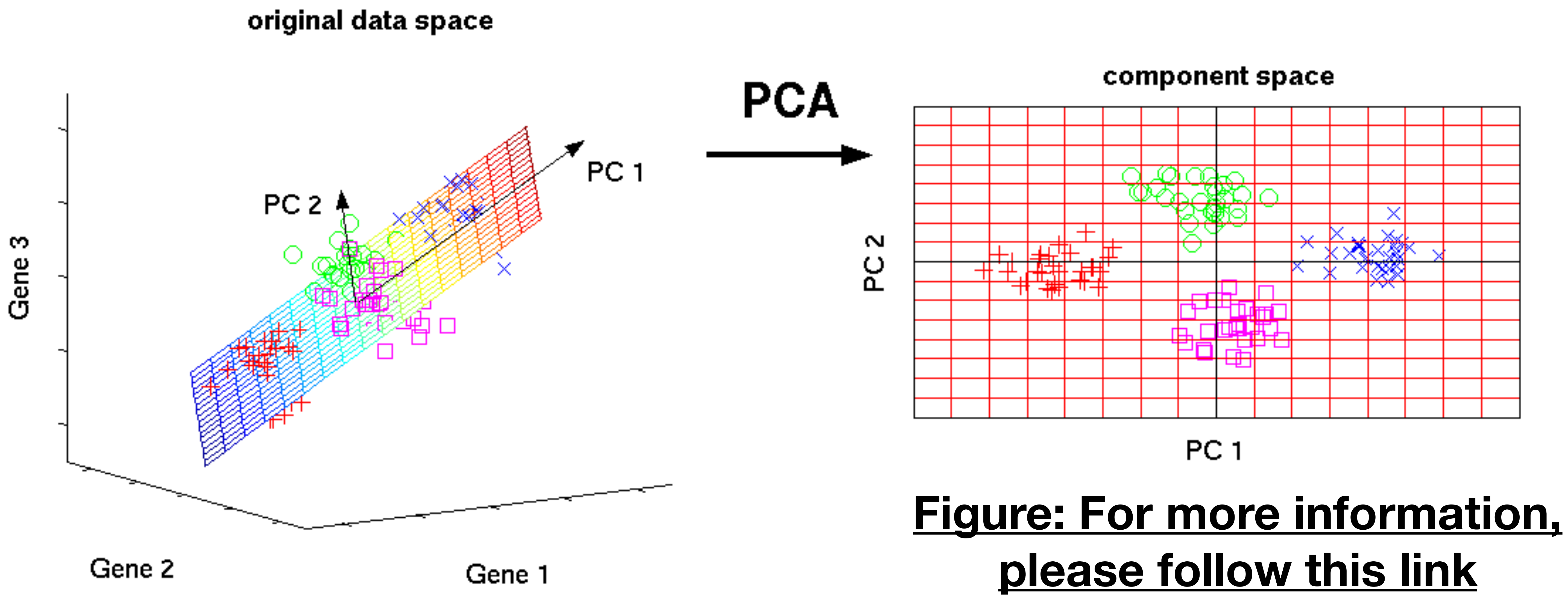


Figure: For more information, please follow this link

3-D gene expression data which are mainly located within a 2-D subspace. PCA is used to visualize these data by reducing the dimensionality of the data. The three original variables (genes) are reduced to a lower number of two new variables termed principal components (PCs). ***Left:*** *Using PCA, we can identify the two-dimensional plane that optimally describes the highest variance of the data. This two-dimensional subspace can then be rotated and presented as a two-dimensional component space* ***(right)****.*

Let’s move on to the jupyter notebook to learn how to perform PCA with scikit-learn in Python.

Good to remember!

One of the many confusing issues in statistics is the confusion between Principal Component Analysis (PCA) and Factor Analysis (FA). These two are technically different approaches and should not be combined. I found these links very useful and simple to do the comparisons.

http://psych.wisc.edu/henriques/pca.html

http://www2.sas.com/proceedings/sugi30/203-30.pdf