CS 559: Principal Component Analysis & Time Series

Lecture 9

Spring 2020

Announcement

- Homework 1,2 and Exam 1 are graded.
- Project is opened.
- Homework 3 due on 4/14th
- Video Presentation on clusters and PCA
  - Ravi Patel – clustering (either k-mean or hierarchal clustering)
  - Xianquing Zou – PCA
  - Due on 4/16th Thursday

Outline

- Highlights from Last Lecture– Unsupervised Learning I
- Principal Component Analysis (PCA)
- Time Series

Highlights from Last Lecture

- Unsupervised Learning
  - How it is different from supervised learning?
    - Data Structure – unlabeled data
    - Different goal – no prediction
  - What are we trying learn in unsupervised learning?
    - Learn more about data structure, meanings of data itself
- K-mean Clustering
  - Easy to implement but has uncertainties of making decisions on cluster numbers.
  - Elbow method
- Hierarchical Clustering
  - Options of making clusters (e.g., distance, linkage)
  - Similar to decision trees

PCA - Multicollinearity

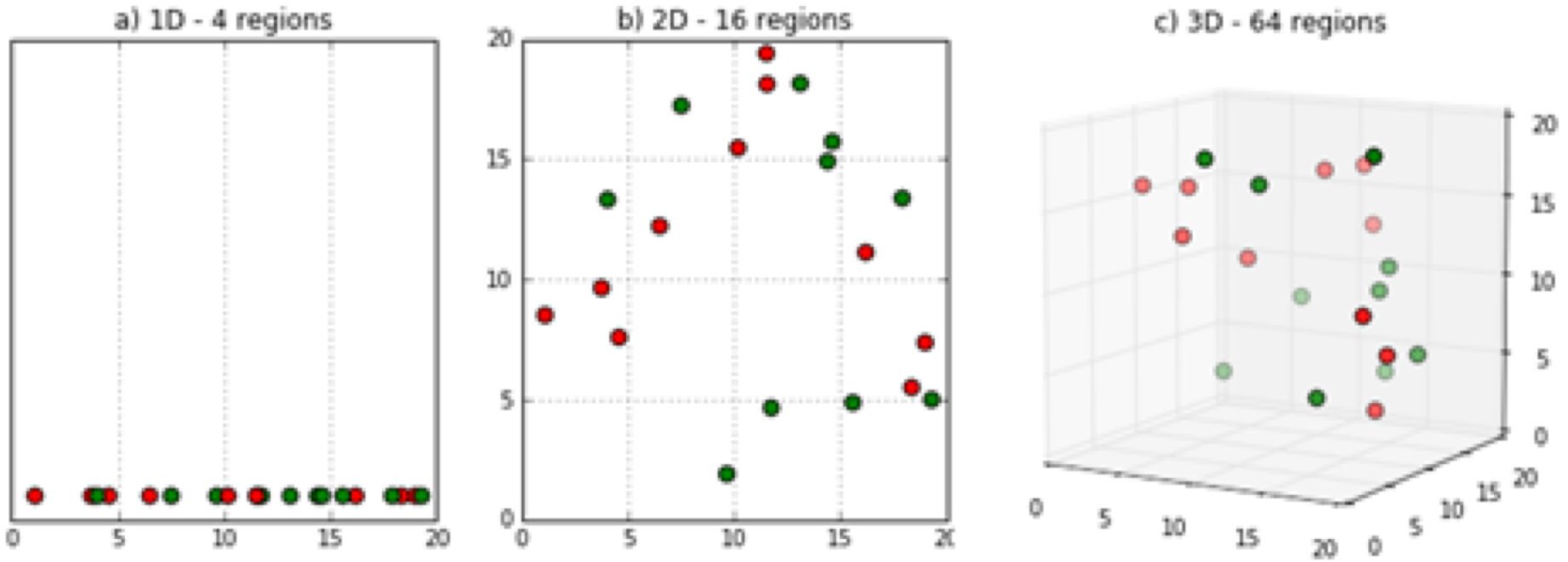
Multicollinearity is a phenomenon in which two or more predictor variables in a multiple regression model are *highly correlated*, meaning that one can be predicted from the others through linear formulae with *a substantial degree of accuracy*.

Issues:

- The regression coefficients of highly correlated variables might be *inaccurate* (high model variance).
- The estimate of one variable's impact on the dependent variable Y while controlling for the others tends to be *less precise*.
- The nearly collinear variables contain similar information about the dependent variable, which may lead to *overfitting*.
- The standard errors of the affected coefficients tend to be *large*.

PCA - Multicollinearity

- Given a number of observations, additional dimensions spread the points out further and further from one another.
- Sparsity becomes exponentially worse as the dimensionality of the data increases.
- The model **SVM** takes advantage of **the curse of dimensionality**.



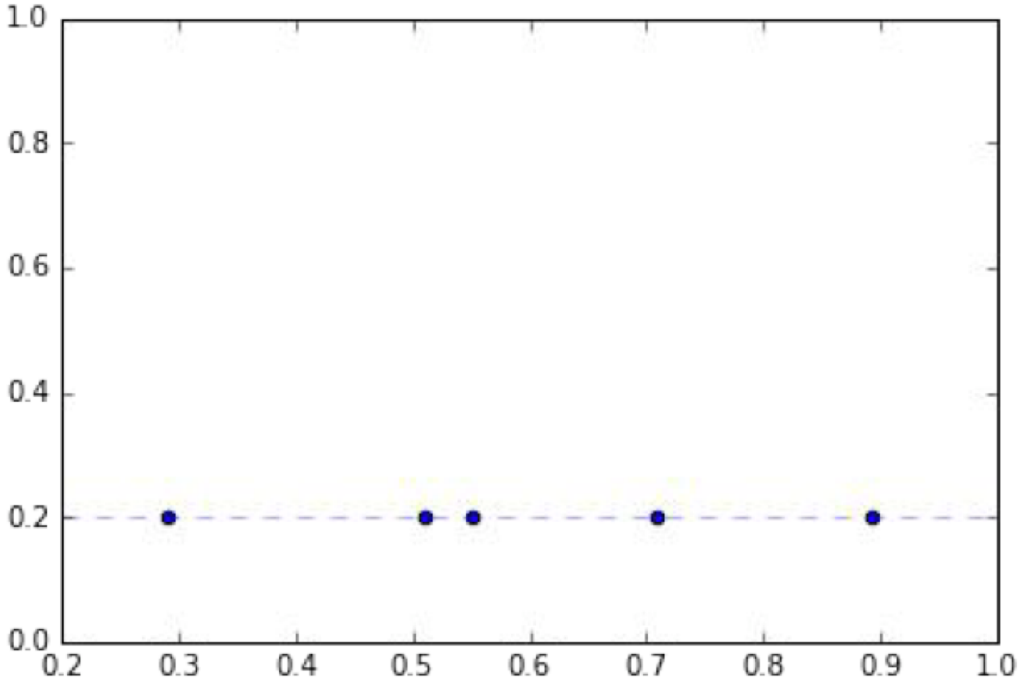
PCA

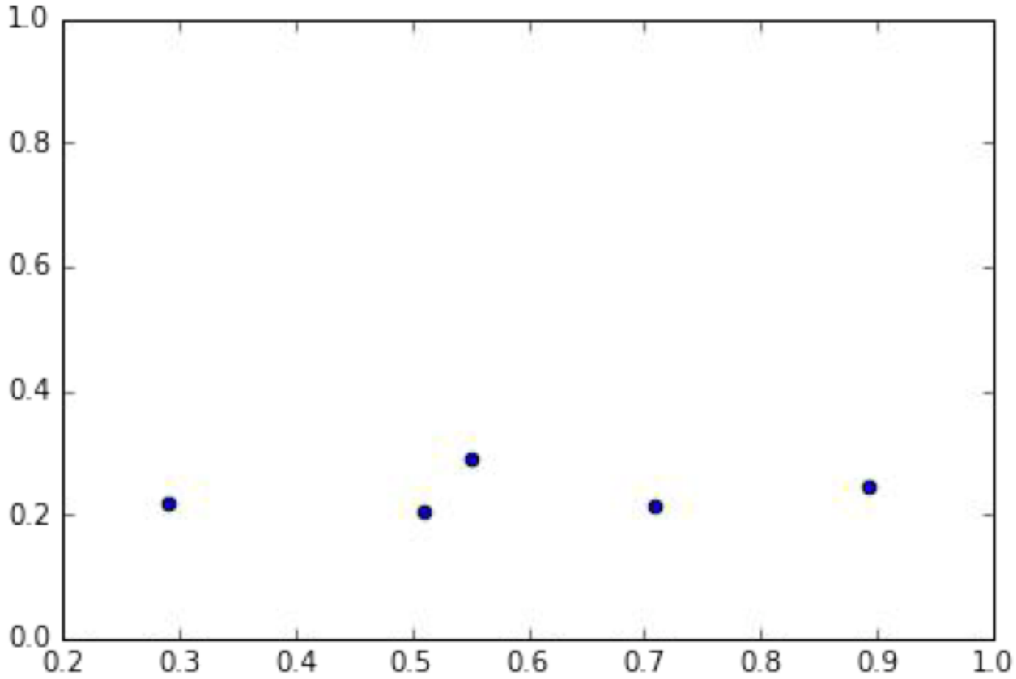
Principal component analysis (PCA) is a tool that finds a sequence of linear combinations of the variables to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components**.

Ideal input variables:

- Linearly uncorrelated
- Low-dimensional in the feature space

PCA – Geometric Motivation

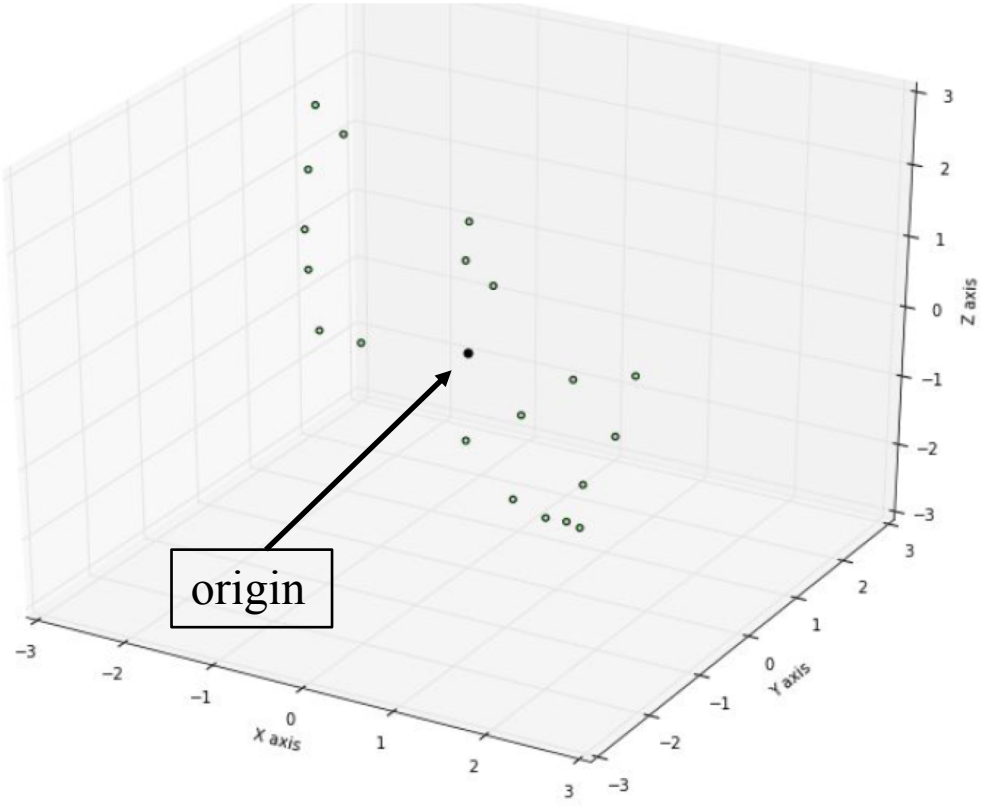


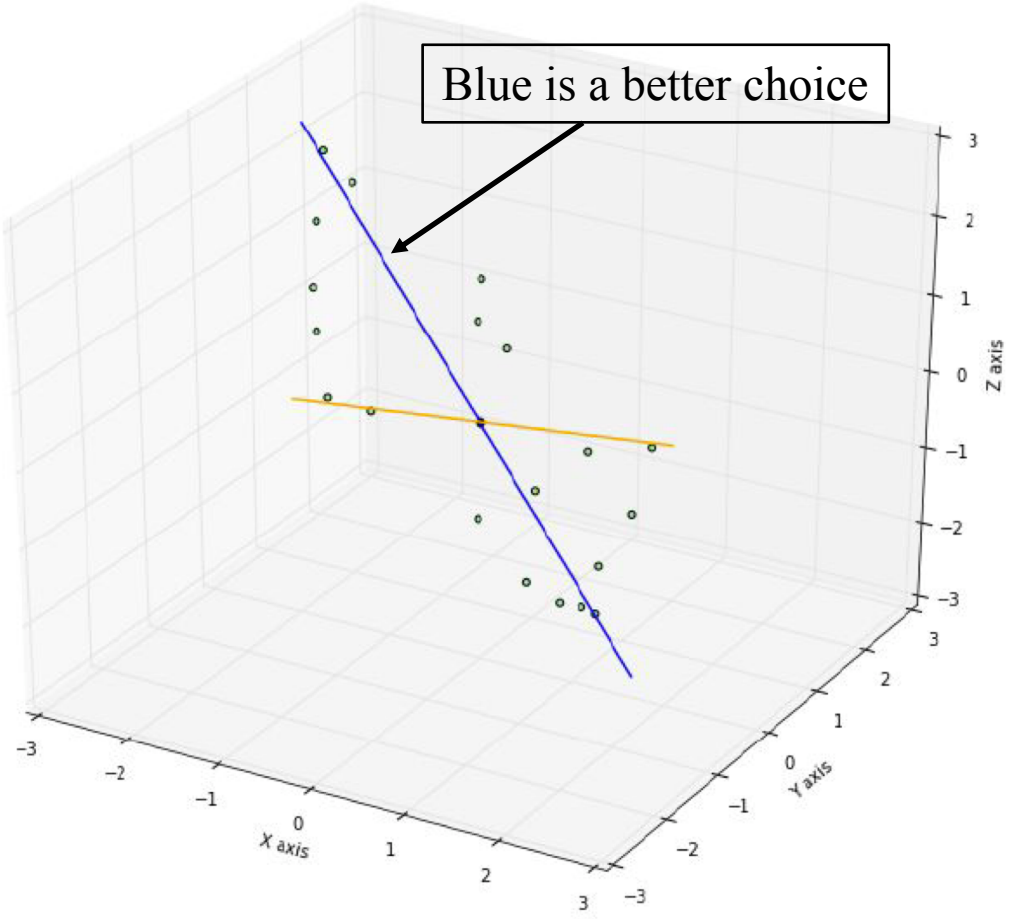


Left: We always do not need all the features. The *y* component of all the points are the same, it provides **NO** additional information.
Right: *y* values are restricted in a much smaller region than *x* values. This suggests that *x* component might provide more information.

PCA – Geometric Motivation

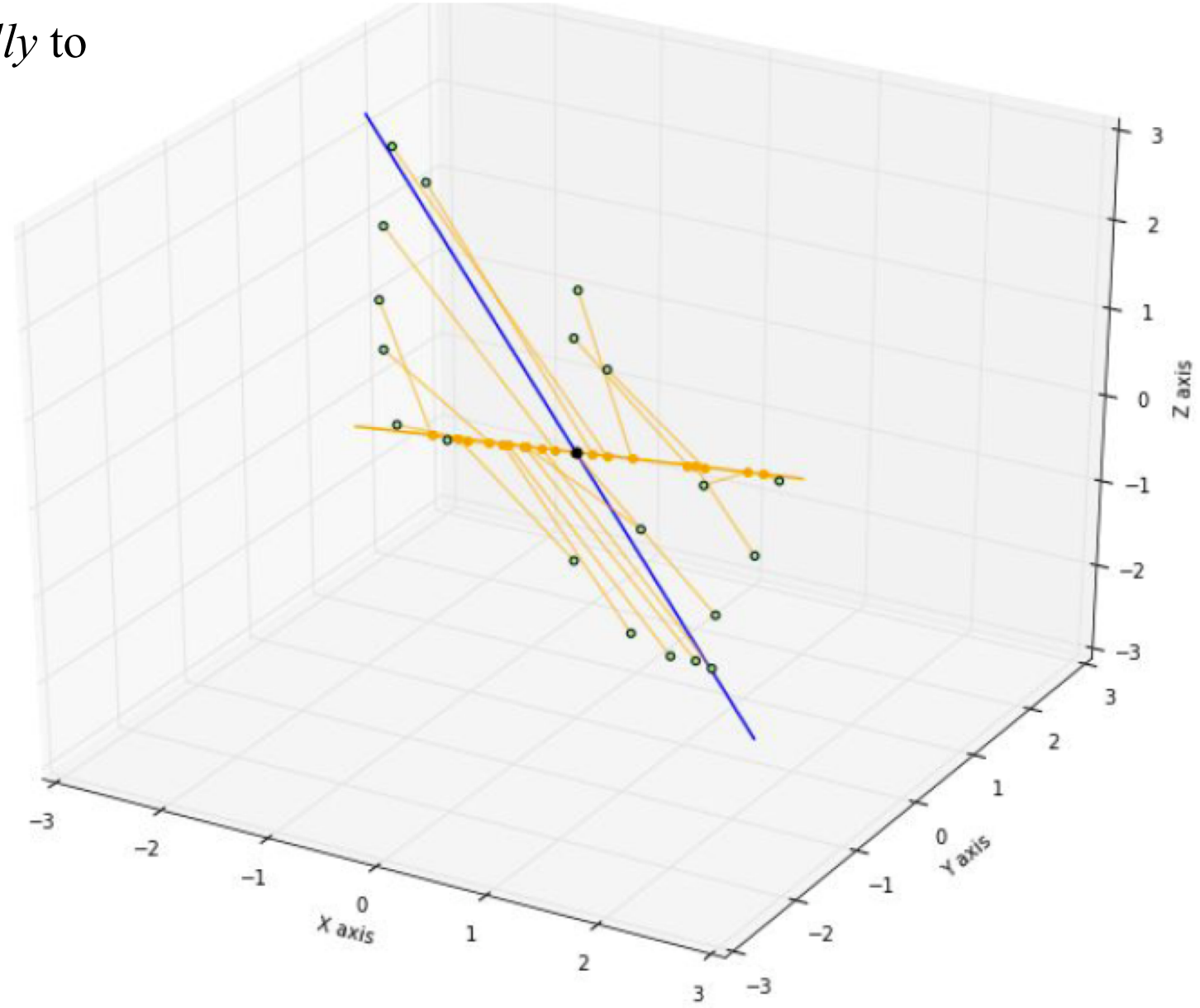
- Left: Consider a set of 20 points in a three dimensional space. In such a scenario, we have: 20 observations and 3 features.
- Right: We compare the **importance** of each direction. Note that the chosen directions in the example are *not parallel* to any coordinate axis.





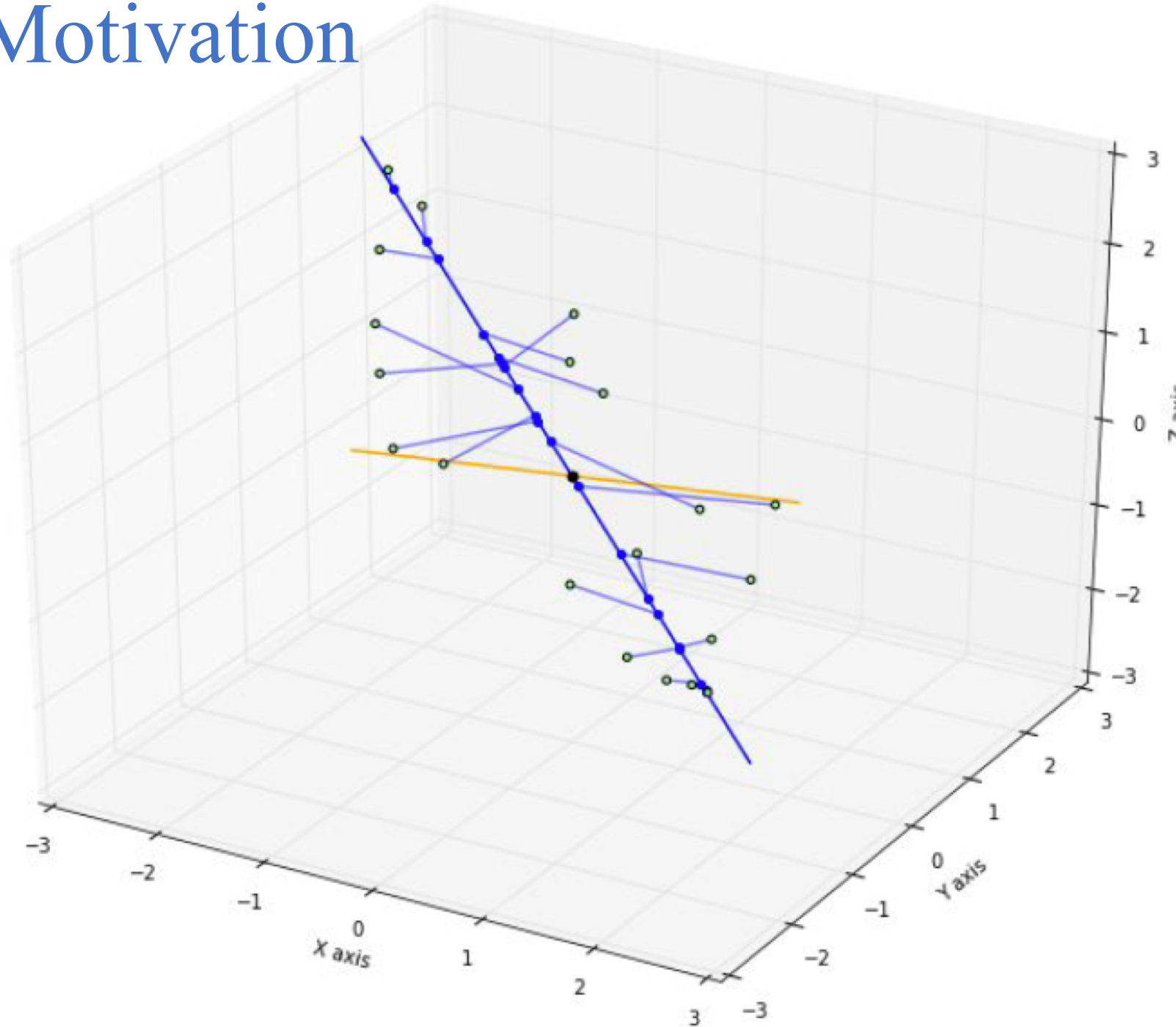
PCA – Geometric Motivation

- We project each observation *orthogonally* to the "orange" direction.



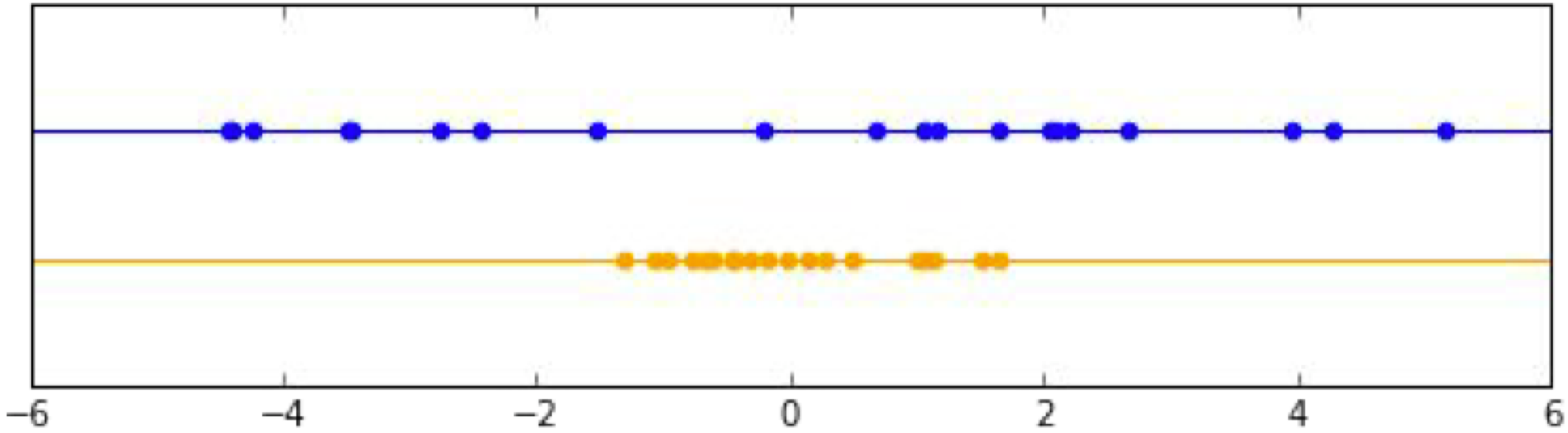
PCA – Geometric Motivation

- We project each observation *orthogonally* to the “blue” direction.



PCA – Geometric Motivation

- The projection of the observations into the “blue” direction is more widely spread than the one into the “orange” direction.



PCA – Geometric Motivation

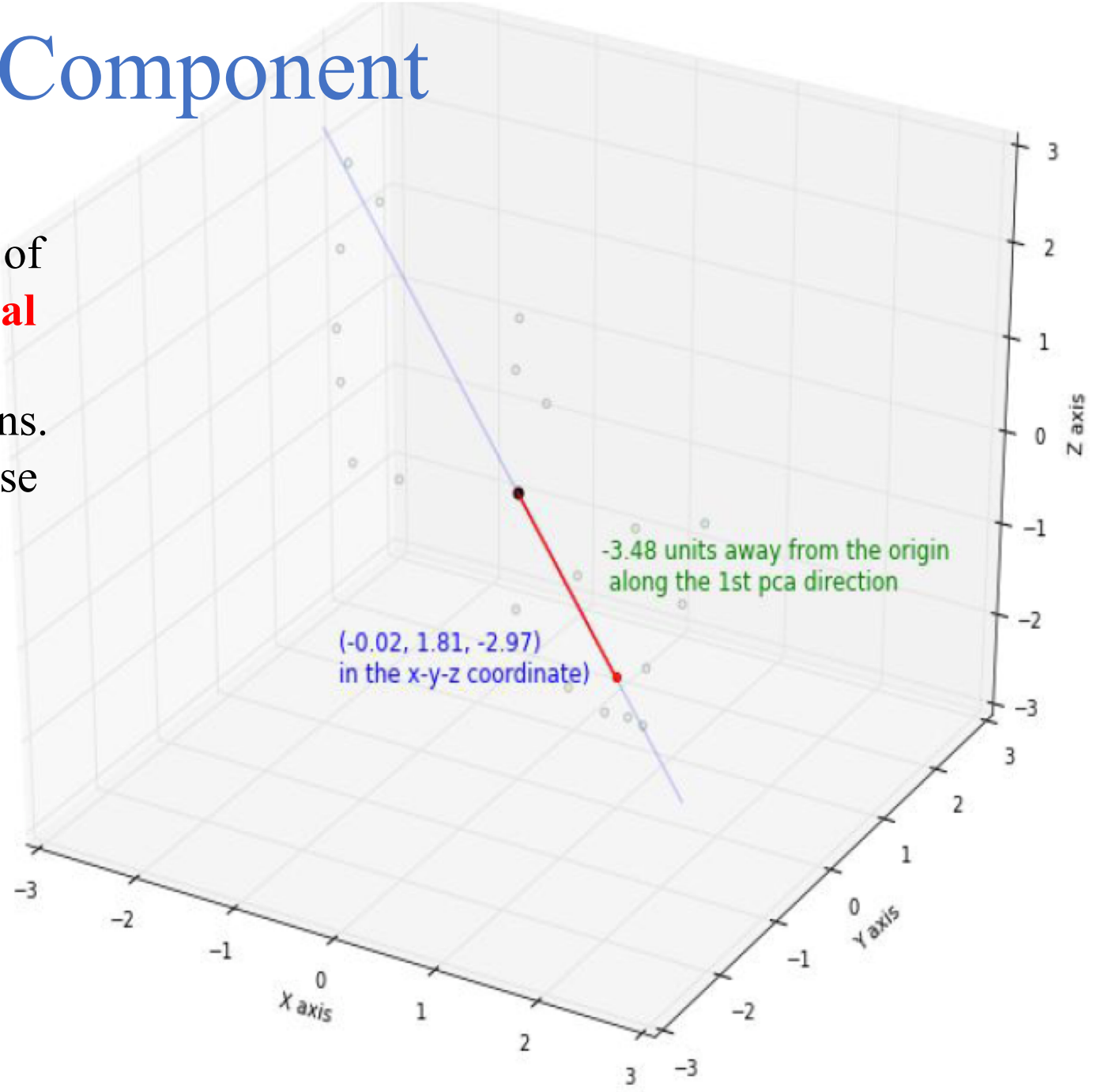
- The "blue" direction above is actually the first loading vector, which means:
  - it is the direction into which the projection of the observations is more widely spread than the projection into any other direction.
  - being a direction (vector), it has as many entries as the number of the features.

- The statements above characterize the principal direction. To find the principal direction we need to apply the technique of linear algebra.

- With the first loading vector (heuristically the most important one), we want to keep, for all the observations, only the information recorded in this direction.
  - This is done by orthogonal linear projection.
  - There are in general N (the number of samples) components for **principal component**.
  - There are in general p (the number of features) components for a **principal direction** (the loading vector).
  - The principal components live in the space of samples, while the principal directions live in the space of features.

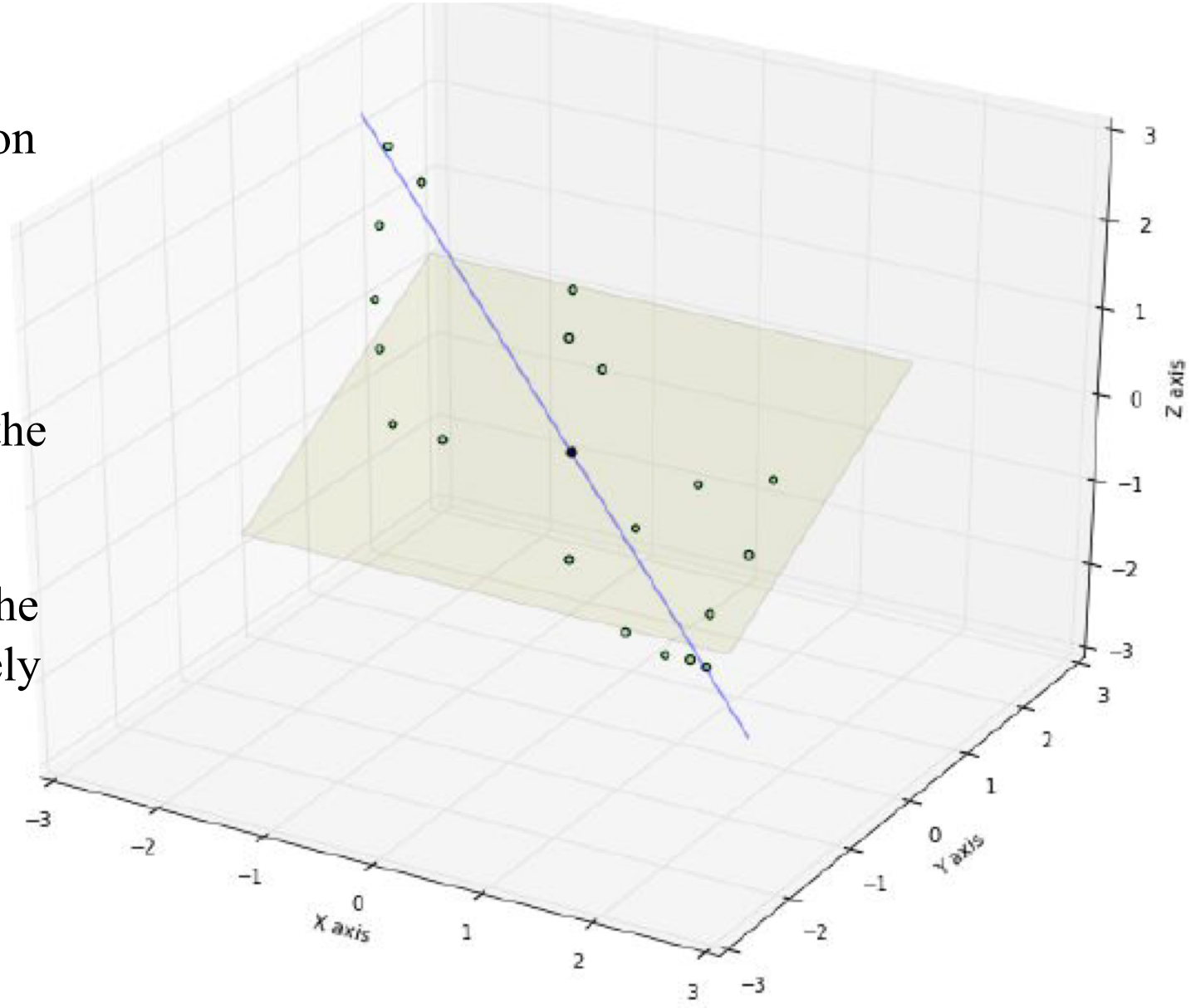
PCA – First Principal Component

- The first loading vector gives us a vector of length 20. This vector is the **first principal component**.
- We need 20 records for all the observations.
- We do not use the *xyz*-coordinates – we use **one coordinate**, the first principal component.
- The *red projection* can describe a certain length away from the origin along the principal direction and is a vector in the original *xyz*-coordinates.



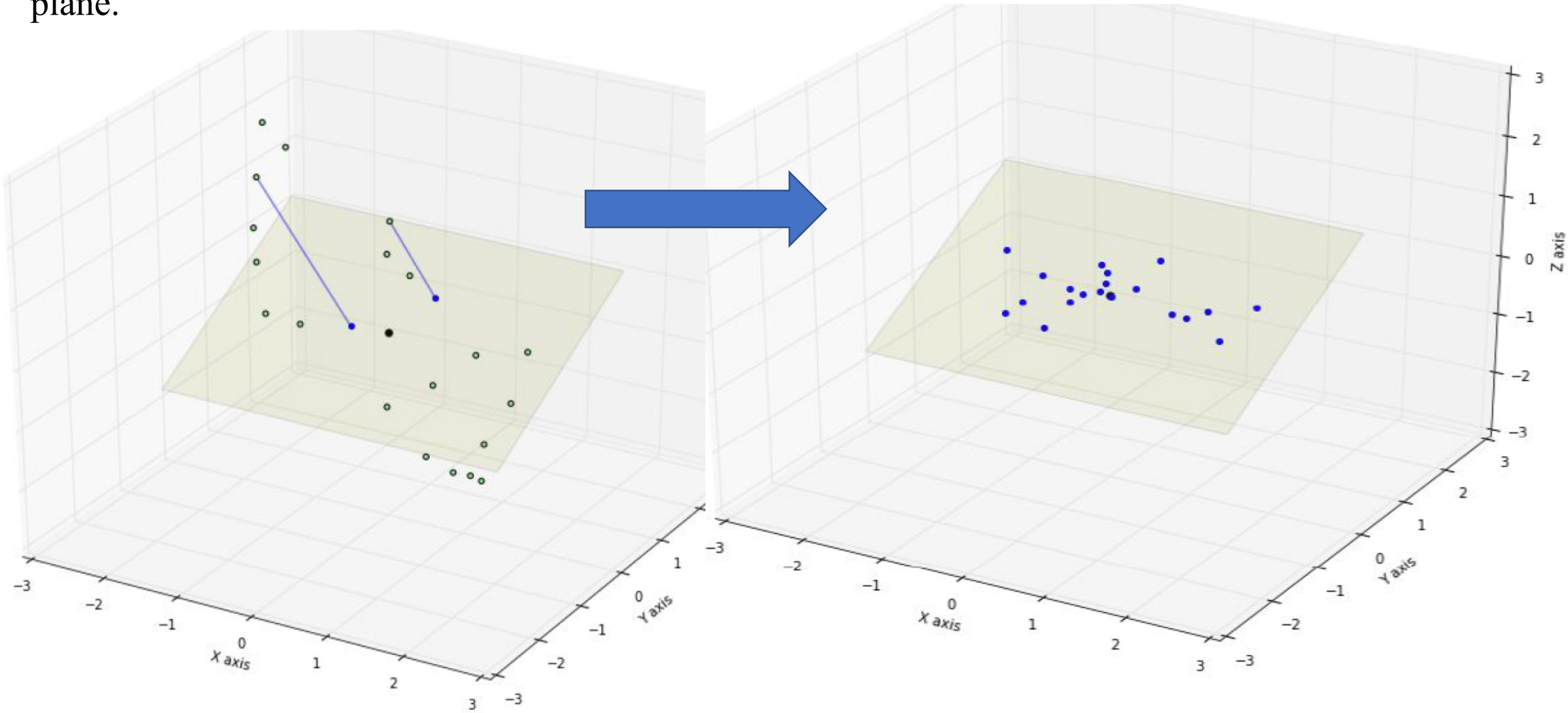
PCA – Second Principal Component

- The information stored is about the variation of the points across the whole sample set.
- Not all directions are born equal.
- The first principal component provides the most information but most likely not all.
- We remove the data information stored in the first principal component.
- Then find the new direction (orthogonal to the original principal direction) on which the projection of the observations is most widely spread.
- We consider the 2-D plane that is perpendicular to the first loading vector.



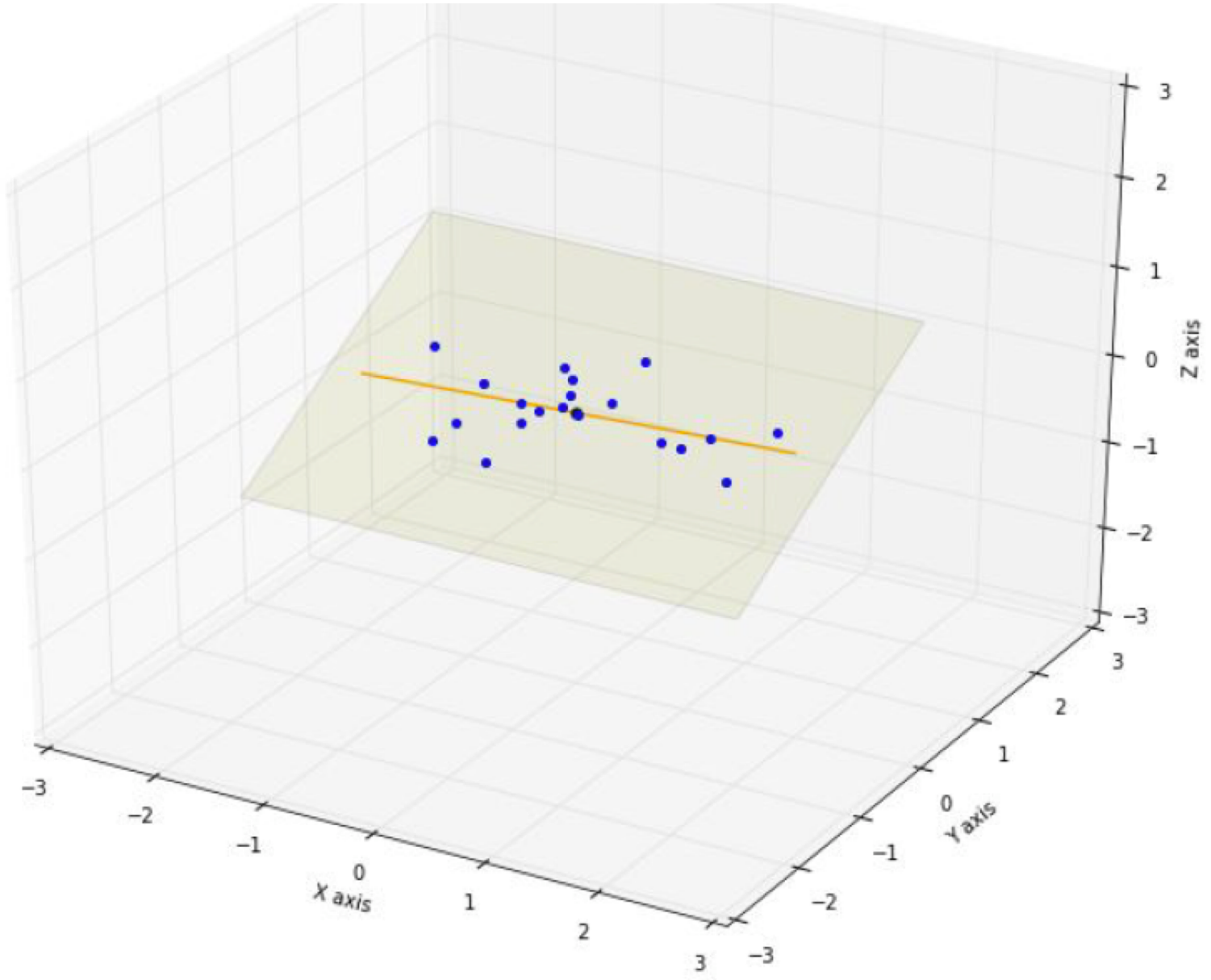
PCA – Second Principal Component

- We can remove the effect of the first principal component by first projecting the observations to this plane.



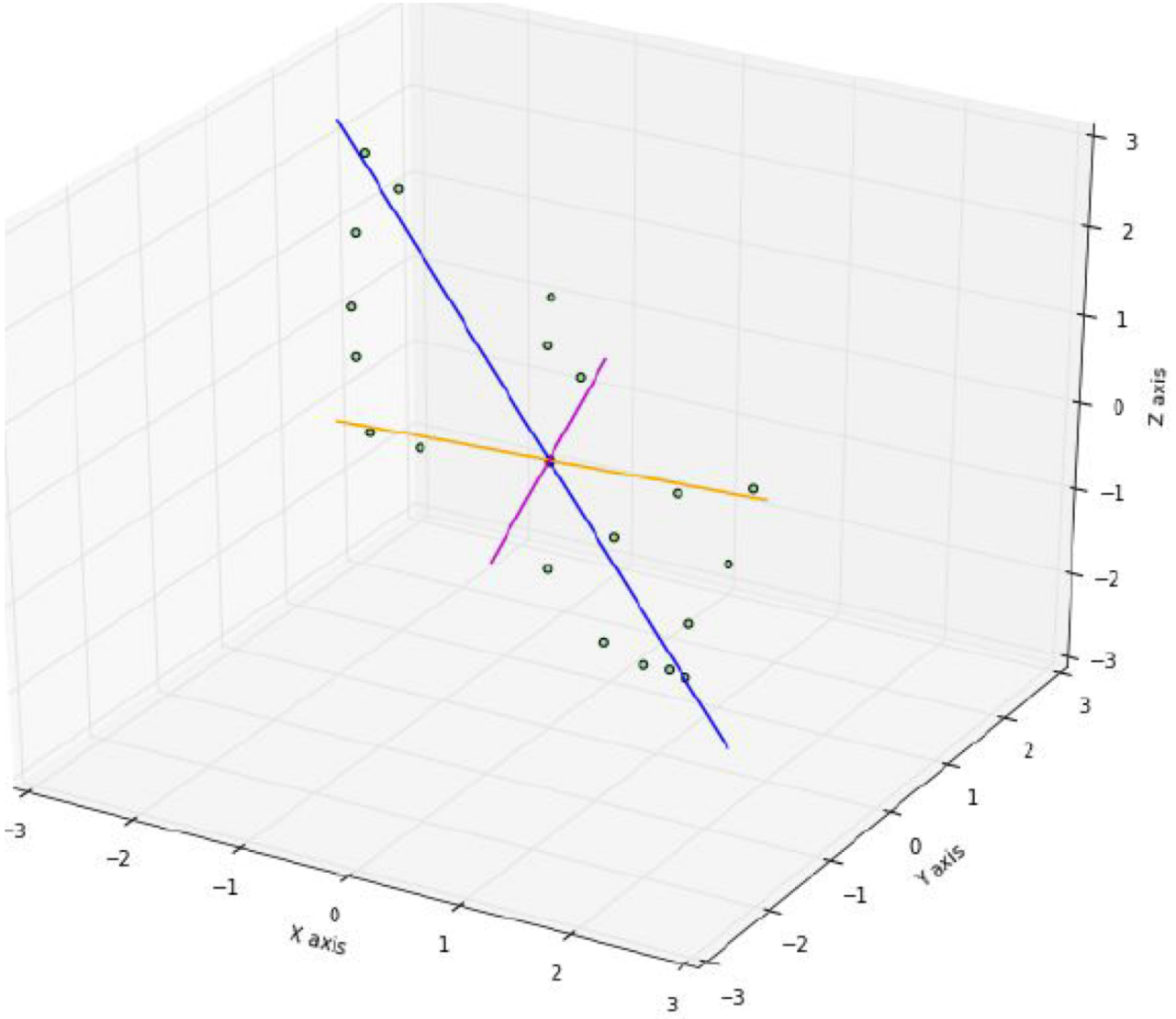
PCA – Second Principal Component

- We find the direction on which the projection is most widely spread.
- Since all the provided observations are now in **the plane**, the direction we find would be automatically in the plane and is automatically perpendicular to the first loading vector.
- This is the **second loading vector (second principal direction)**. The projected values of the observations to this direction is **the second principal component**.



PCA – Second Principal Component

- This process can be continued to retain more and more information from the raw data. However, we remove one dimension each time when we remove the information recorded in a principal direction.
- We cannot have more number of the principal components than the number of the original features we have.
- This induction process always terminates within a finite step.



PCA – The Mathematical Formulation

- The first (very important) step is to centralize the raw data. Assume that our data X is an n by p matrix. The average of each feature column is 0.
- We then project the data into any possible direction. A direction is represented by a unit vector $\hat{u}$ and the projection is $X\hat{u}^T$.
- We need to find the direction on which the the projection of the data is most widely spread.

$$\phi = \max Var(X\hat{u}^T) \ s.t. \left|\left|\hat{u}\right|\right| = 1$$

- The solution to the optimization problem above is the first loading vector (first principal direction), denoted by ϕ_1. The projection of our data X on the first loading vector is

$$Z_1 = X\phi_1^T$$

which is called the first principal component.

PCA – The Mathematical Formulation

- Once the first k-1 principal component have been found, the next one (if there is one) can be found inductively.
- We first remove the information stored in the first k-1 components from X (X_k denotes the resulting matrix).

$$X_k = X - \sum_{i=1}^{k-1} X\phi_i^T\phi_i$$

- With this matrix we solve the optimization problem again:

$$\phi_k = \max Var(X_k\hat{u}^T)\ \ s.t.\ \left|\left|\hat{u}\right|\right| = 1$$

- Again the solution ϕ_k is the k_{th} loading vector and the projection on this direction is called the k_{th} principal component.

$$Z_k = X_k\phi_k^T$$

PCA – The Properties

- There are most $\min(n, p)$ principal components (but we often assume p to be smaller among the two, so there are p of them).
- The variance of each principal component decreases: $Var(Z_1) \geq Var(Z_2) \geq \cdots \geq Var(Z_p)$
- The principal components $Z_1, Z_2, \ldots, Z_p$ are mutually uncorrelated.
- The principal loading vectors $\phi_1, \phi_2, \ldots, \phi_p$ are normalized and mutually perpendicular.
- The variance of the data along the principal directions (eigenvectors) are the corresponding positive eigenvalues.

PCA – Geometrical Meaning

- Geometrically we can imagine that the original data set sits inside $\mathbb{R}^f$ as a high dimensional scatterplot.
- The selection of the top p principal directions establishes a linear projection $\mathbb{R}^f \rightarrow \mathbb{R}^p$ into a lower dimensional space.
- There are many orthogonal linear projections from $\mathbb{R}^f$ to $\mathbb{R}^p$. But PCA is special that it collapses directions in which the variances are small and preserve those whose variances are larger.
- Those directions which get collapsed are interpreted as noise of the data.
- Even though the apparent dimension of the data is *f* dimensional, PCA hypothesizes that the true dimension of the data lies in a p dimensional linear space.
- In this sense, PCA is a de-noising process revealing the true nature of the data.
- Nonlinear projections into curved objects instead of linear spaces is called **manifold** learning as non-linear smooth objects are called manifolds in geometry.

Time Series

Time series - Introduction

- Our analyses have been limited to cross-sectional data – the variables we considered have theoretically been measured at a single point in time for each of the observations in our datasets.

- What if we have data on variables that are repeatedly measured over time? This kind of data is called longitudinal and involves following a phenomenon by measuring its evolution through time.

- Longitudinal data that has been recorded at regularly spaced time intervals for a given span of time comprises a time series, and will be the main subject of today's analysis.

Time series - Modeling

A time series has four components.

- Level – the mean value around which the series varies.
- Trend – the increasing or decreasing behavior of a variable with time.
- Seasonality – the cyclic behavior of time series.
- Noise – The error in the observations added due to environmental factors.

Parameter Calibration

- **Hit-and-try**: Start by visualizing the time-series and intuitively try some parameter values and change them over and over until you achieve a good enough fit. It requires a good understanding of the model you are trying.
- **Grid Search** – Try building a model for all possible combinations of parameters and select the one with minimum error.
- **Genetic Algorithm** – Works on the biological principle that a good solution will eventually evolve to the most “optimal” solution.

Time series - Modeling

Naïve Methods – These are sample estimation techniques, such as the predicted value is given the value equal to mean of preceding values of the time dependent variable, or previous actual value. These are used for comparison with sophisticated modelling techniques.

LSTM – Long Shor-Term Memory model (LSTM) is a recurrent neural network which is used for time series to account for long term dependencies. It can be trained with large amount of data to capture the trends in multi-variate time series.

Time series – Modeling: Auto Regression

- Each value in a time series is predicted from a linear combination of the previous p values:

$$AR(p): y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t$$

- y_t: A given value of the series
- C: The mean of series
- ϕ_i: The coefficients of lag y_{t-i}
- ϵ_t: The white noise component – error of prediction

- Without ϵ_t, the above recursive formula is deterministic!

Time series – Modeling: Moving Average

- For a stationary time series, a moving average model sees the value of a variable at time "t" as a linear function of residual errors from 'q' time steps preceding it.

- The residual error is calculated by comparing the value at the time 't' to moving average of the values preceding.

$$MA(q): y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

- y_t: A given value of the series
- c: The mean of series
- θ_i: The coefficients of error ϵ_{t-i}
- ϵ_t: The white noise component – error of prediction

- When ϵ_t all approach zero, the MA(q) model reduces to a **constant** time series.

Time series – Integrated Component

The integrated component I(d) refers to a time series that has been differenced *d* times; a differenced series represents the change between consecutive observations in the original series:

$$\Delta: y_t \rightarrow y_t - y_{t-1}$$

Note that as we difference multiple times, instead of differencing prior lags, we difference the previous difference.

$$\begin{aligned}\Delta^2(y_t) = \Delta\big(\Delta(y)\big) = \Delta(y)_t - \Delta(y)_{t-1} \\ = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ = y_t - 2y_{t-1} + y_{t-2}\end{aligned}$$

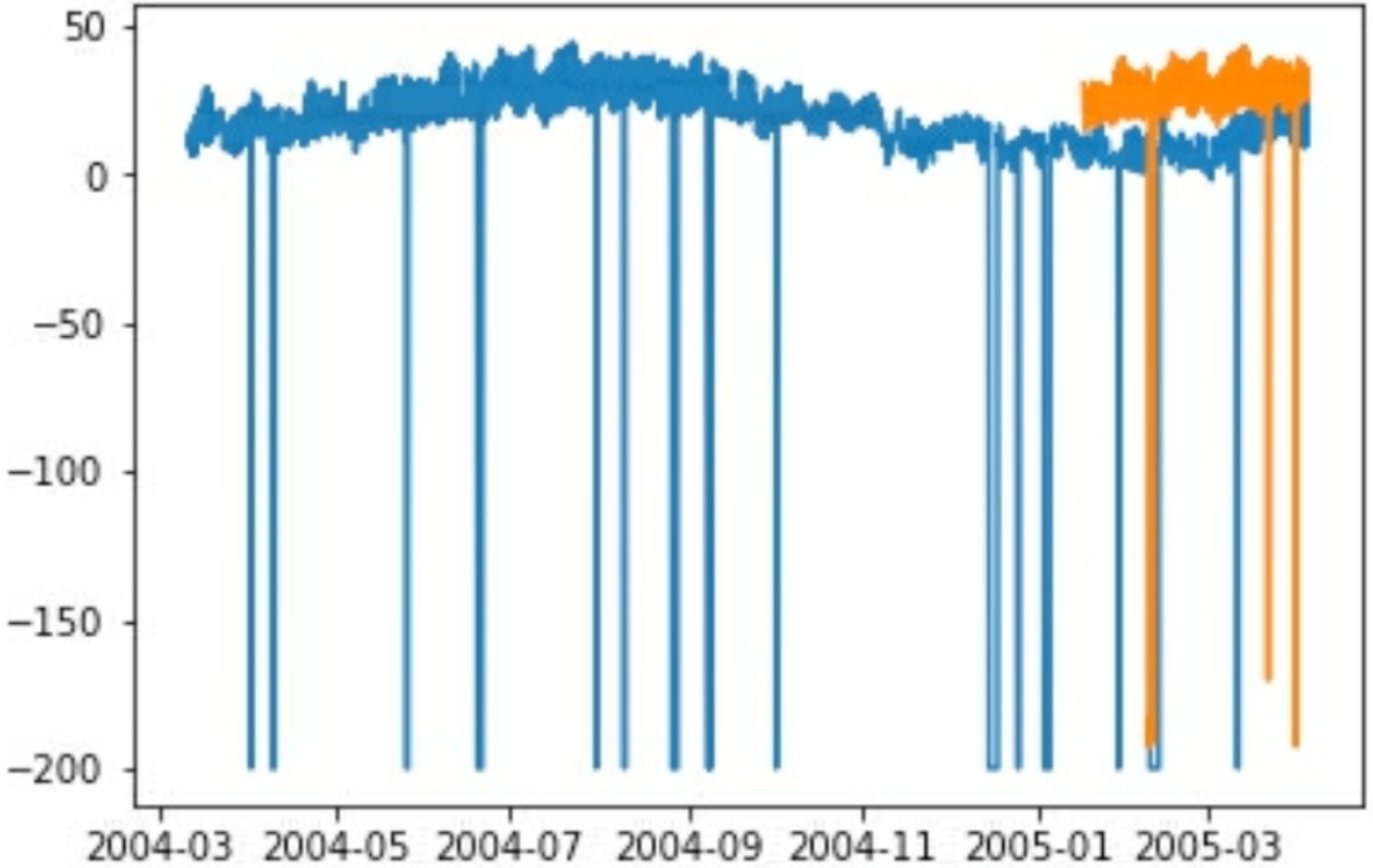
Time series – ARIMA

- An ARIMA(p,d,q) model is a linear time series model that combines the ideas of each of the previously discussed components:
  - The original time series has been differenced d times – I(d)
  - The resulting values are predicted from the previous p time series values - AR(p)
  - The resulting values are predicted from the previous q error terms – MA(q)
- An ARMA (p,q) model is the special case of ARIMA with d=0.
- Both ARMA and ARIMA are linear framework relating the current value of a time series to the past values and the past white noise shocks.
- The additional integrated component allows ARIMA to move beyond the range of stationary time series to model non-stationary time series.

Time series – ARIMA

Fitting Process:

1. Ensure that the time series is stationary.
    1. Use the residual values after detrending using linear regression.
    2. Use the residual values after seasonally decomposing.
    3. Possibly take differences d times to produce a stationary series.
2. Identify a reasonable subset of models.
    1. Determine possible values of p.
    2. Determine possible values of q.
3. Fit the models based on the parameter selections.
    1. Evaluate the model fit.
4. Make forecasts with the final selected model.


50
0
−50
−100
−150
−200
2004-03
2004-05
2004-07
2004-09
2004-11
2005-01
2005-03