



DAY 5

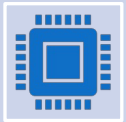
Instructor: Balu Mohandas Menon

Christian B. Wiberg
Philip Jess Teining

VISUALIZATION WITH MATPLOTLIB



Matplotlib is a multi-platform data visualization library built on NumPy arrays.



One of Matplotlib's most important features is its ability to play well with many operating systems and graphics backends

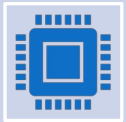


This cross-platform, everything-to-everyone approach has been one of the great strengths of Matplotlib.

VISUALIZATION WITH MATPLOTLIB



In recent years, however, ggplot and ggvis in the R language, along with web visualization toolkits based on D3js and HTML5 canvas, often make Matplotlib feel clunky and old-fashioned.

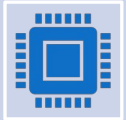


Still, we cannot ignore Matplotlib's strength as a well-tested, cross-platform graphics engine



Recent Matplotlib versions make it relatively easy to set new global plotting styles (see [Customizing Matplotlib: Configurations and Style Sheets](#)), and people have been developing new packages that build on its powerful internals to drive Matplotlib via cleaner, more modern APIs—for example, Seaborn (discussed in [Visualization With Seaborn](#)), ggpy, [HoloViews](#), [Altair](#), and even Pandas itself can be used as wrappers around Matplotlib's API

VISUALIZATION-WITH-SEABORN



Matplotlib's API is relatively low level. Doing sophisticated statistical visualization is possible, but often requires a lot of boilerplate code.



Matplotlib predated Pandas by more than a decade, and thus is not designed for use with Pandas DataFrames. In order to visualize data from a Pandas DataFrame, you must extract each Series and often concatenate them together into the right format. It would be nicer to have a plotting library that can intelligently use the DataFrame labels in a plot.



An answer to these problems is Seaborn. Seaborn provides an API on top of Matplotlib that offers sane choices for plot style and color defaults, defines simple high-level functions for common statistical plot types, and integrates with the functionality provided by Pandas DataFrames.

CURSE OF DIMENSIONALITY

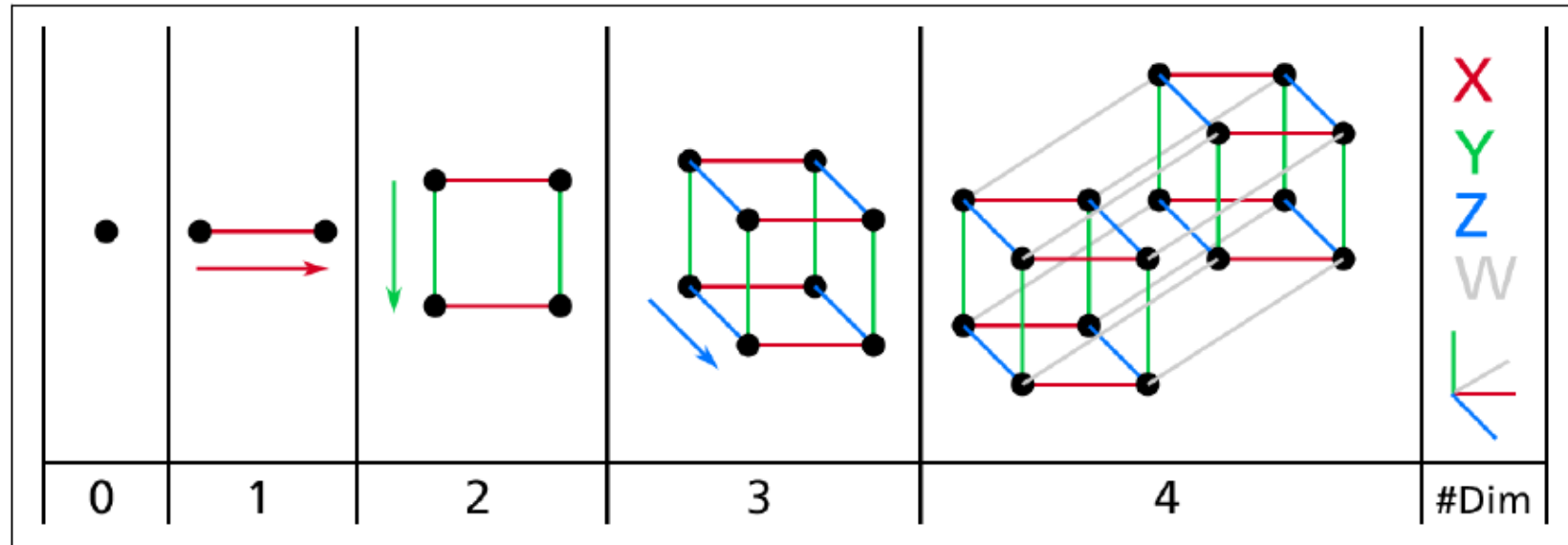
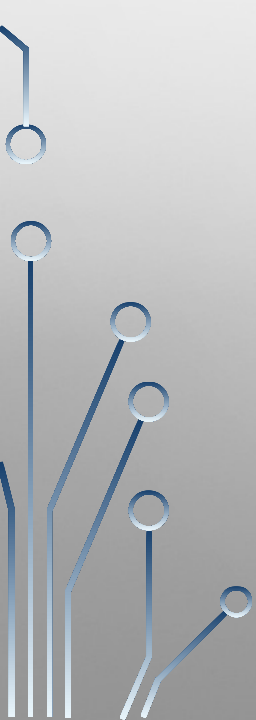
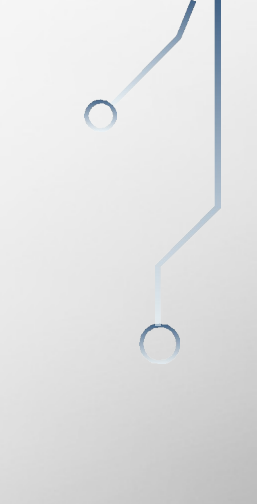
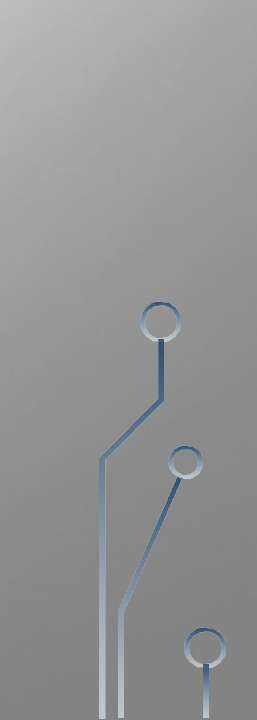


Figure 8-1. Point, segment, square, cube, and tesseract (0D to 4D hypercubes)²



CURSE OF DIMENSIONALITY

- Curse of Dimensionality refers to a set of problems that arise when working with high-dimensional data.
 - The dimension of a dataset corresponds to the number of attributes/features that exist in a dataset.
 - A dataset with a large number of attributes, generally of the order of a hundred or more, is referred to as high dimensional data
 - Some of the difficulties that come with high dimensional data manifest during analyzing or visualizing the data to identify patterns, and some manifest while training machine learning models.
 - ***The difficulties related to training machine learning models due to high dimensional data are referred to as the 'Curse of Dimensionality'.***
- 
- 
- 

CURSE OF DIMENSIONALITY : EXAMPLE 1

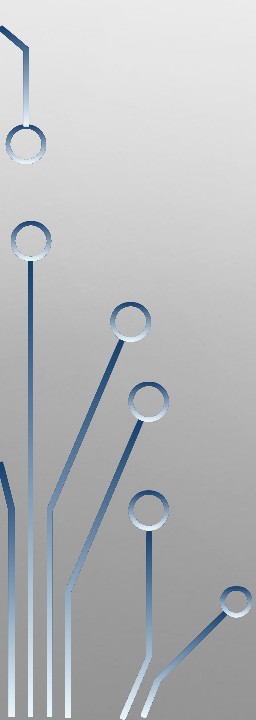
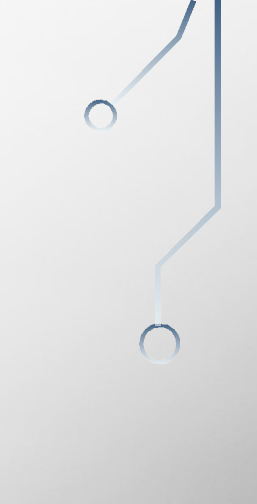
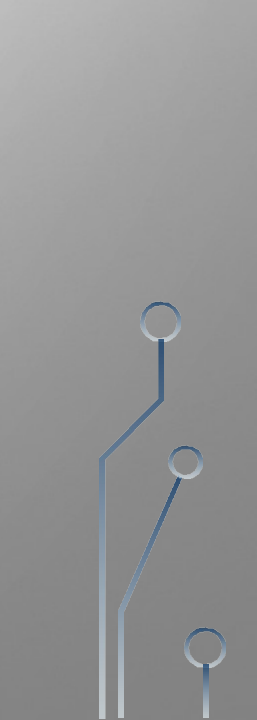
- Probably the kid will like to eat cookies, so let us assume that you have a whole truck with cookies having a different colour, a different shape, a different taste, a different price ...
- If the kid has to choose but only take into account one characteristic e.g. the taste, then it has four possibilities: sweet, salt, sour, bitter, so the kid only has to try four cookies to find what (s)he likes most.
- If the kid likes combinations of taste and colour, and there are 4 different colors, then he already has to choose among 4×4 different types;
- If he wants, in addition, to take into account the shape of the cookies and there are 5 different shapes then he will have to try $4 \times 4 \times 5 = 80$ cookies

CURSE OF DIMENSIONALITY : EXAMPLE 2

- It's easy to catch a caterpillar moving in a tube(1 dimension). It's harder to catch a dog if it were running around on the plane (two dimensions).
- It's much harder to hunt birds, which now have an extra dimension they can move in. If we pretend that ghosts are higher-dimensional beings, those are even more difficult to catch.



FEATURE SELECTION

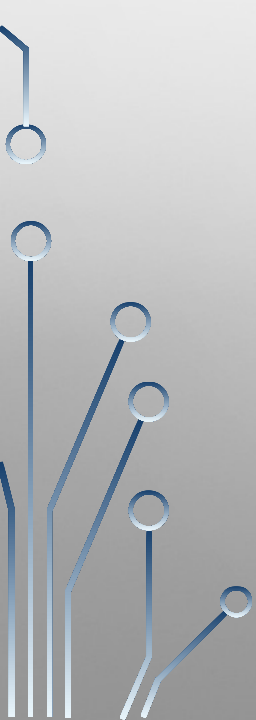
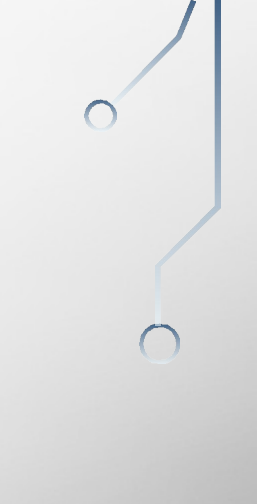
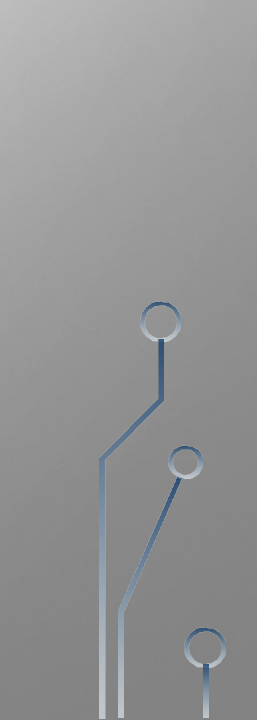
- To mitigate the problems associated with high dimensional data a suite of techniques generally referred to as 'Dimensionality reduction techniques are used. Dimensionality reduction techniques fall into one of the two categories- '**Feature selection**' or '**Feature extraction**'.
- 
- 
- 

FEATURE SELECTION TECHNIQUES

- In feature selection techniques, the attributes are tested for their worthiness and then selected or eliminated. Some of the commonly used Feature selection techniques are discussed below.
- **Feature Ranking:** Decision Tree models such as CART can rank the attributes based on their importance or contribution to the predictability of the model. In high dimensional data, some of the lower ranked variables could be eliminated to reduce the dimensions.
- **High Correlation filter:** In this technique, the pair wise correlation between attributes is determined. One of the attributes in the pairs that show very high correlation is eliminated and the other retained. The variability in the eliminated attribute is captured through the retained attribute.



INTRODUCING SCIKIT-LEARN

- There are several Python libraries which provide solid implementations of a range of machine learning algorithms. One of the best known is [Scikit-Learn](#), a package that provides efficient versions of a large number of common algorithms.
 - Machine learning is about creating models from data: for that reason, we'll start by discussing how data can be represented in order to be understood by the computer. The best way to think about data within Scikit-Learn is in terms of tables of data
- 
- 
- 

A decorative graphic consisting of blue lines and circles, resembling a circuit board or a network diagram, is positioned on the left side of the slide. The lines are of varying thickness and connect to small circles at various points.

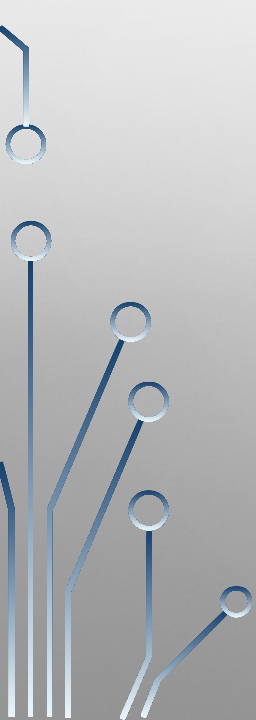
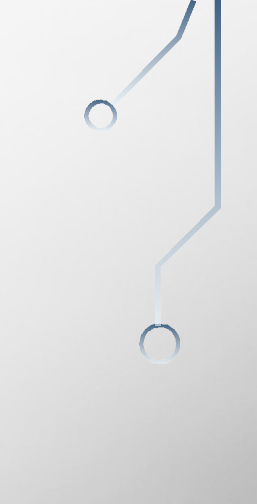
FEATURE EXTRACTION TECHNIQUES

PRINCIPAL COMPONENT ANALYSIS

- **Principal Component Analysis** (PCA) is an unsupervised statistical technique algorithm. PCA is a “**dimensionality reduction**” method.
- It reduces the number of variables that are correlated to each other into fewer independent variables without losing the essence of these variables.
- It provides an overview of linear relationships between inputs and variables.



WHEN TO USE PCA

- Whenever we want to ensure that variables in data are independent to each other.
 - When we want to reduce the number of variables in a data set with many variables in it.
 - When we want to interpret data and variable selection out of it.
- 
- 
- 