Dimensionality Reduction

- What Is Dimensionality Reduction?
- Dimensionality Reduction Methods
  - Principal Component Analysis
  - Attribute Subset Selection
  - Nonlinear Dimensionality Reduction Methods

What Is Dimensionality Reduction?

- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- **Dimensionality reduction**
  - Reducing the number of random variables under consideration, via obtaining a set of principal variables
- **Advantages of dimensionality reduction**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

Dimensionality Reduction Methods

- Dimensionality reduction methodologies
  - **Feature selection**: Find a subset of the original variables (or features, attributes)
  - **Feature extraction**: Transform the data in the high-dimensional space to a space of fewer dimensions
- Some typical dimensionality reduction methods
  - Principal Component Analysis
  - Attribute Subset Selection
  - Nonlinear Dimensionality Reduction

Principal Component Analysis (PCA)

- **Purpose**: Dimensionality reduction
- **Goal**: Reduce the number of variables while keeping the most important information
- **Used in**:
  - Simplifying complex datasets
  - Revealing patterns
  - Preprocessing for machine learning tasks (regression, clustering, etc.)

Key Concepts in PCA

- **Dimensionality Reduction**: Fewer variables, less complexity
- **Principal Components**: New variables that capture the essence of the data
  - Linear combinations of original variables
  - Sorted by importance (variance)

Principal Component Analysis (PCA)

- PCA: A statistical procedure that 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***
- The original data are projected onto a much smaller space, resulting in dimensionality reduction

Steps in PCA

- **Step 1: Normalize Data:**
  - Scale variables to fall within the same range
  - Prevents domination by variables with larger ranges
- **Step 2: Compute Principal Components:**
  - Find k orthonormal vectors that best represent the data
  - These vectors are the principal components (new variables)
- **Step 3: Sort Principal Components:**
  - Sorted by how much variance they capture
  - First component captures the most variance, second captures the next most, etc.
- **Step 4: Reduce Data:**
  - Discard components that capture little variance
  - Results in fewer dimensions, simpler data

Visual Example

- **Original variables**:
  - X1 and X2
- **After PCA**: Y1 and Y2
  - **Y1** captures the most variance
  - **Y2** captures the second most

Result: Data is re-expressed in terms of Y1 and Y2

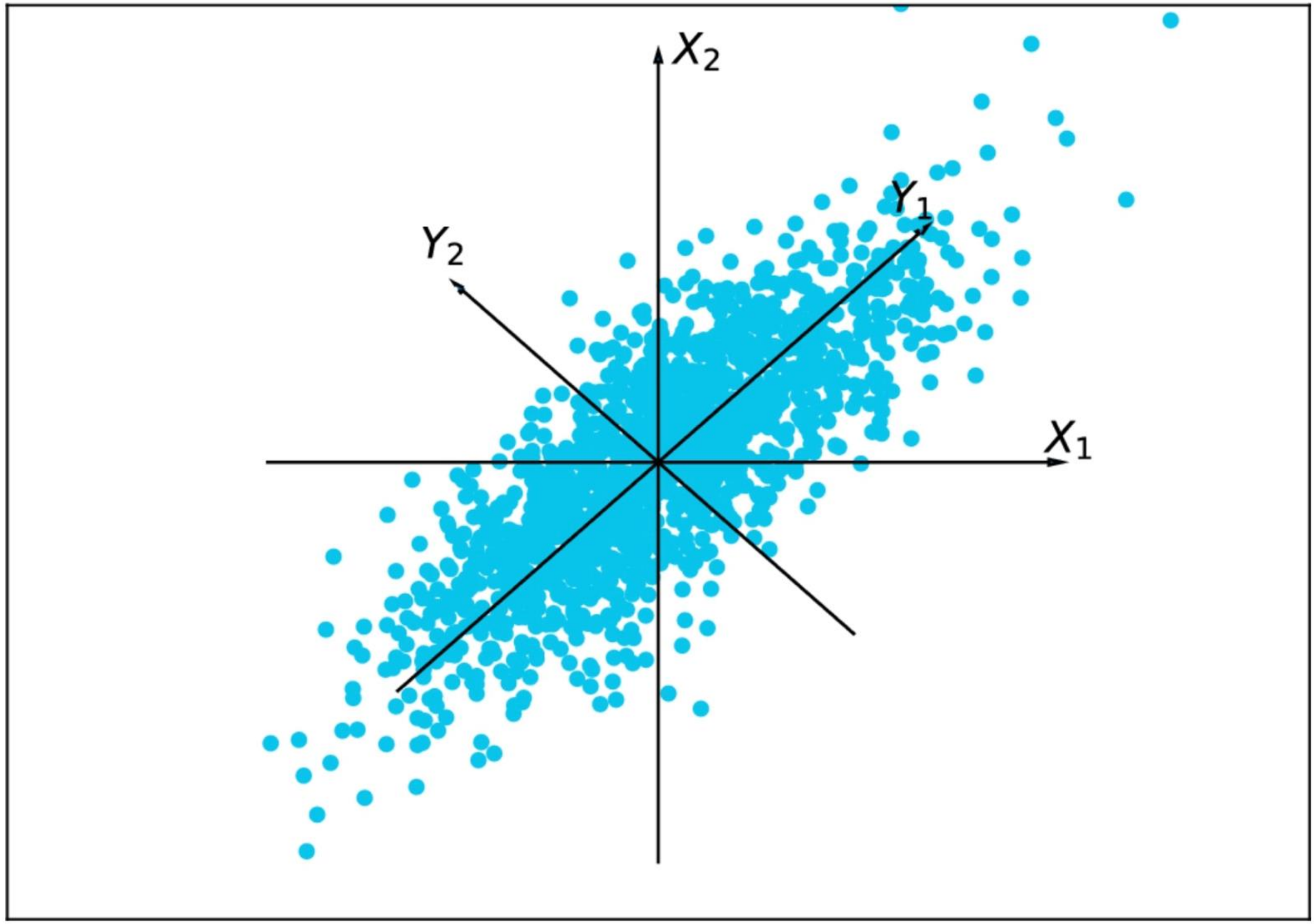


FIGURE 2.17 Principal components analysis. Y_1 and Y_2 are the first two principal components for the given data.

Next, checkout the Jypyter notebook

Principal components analysis

PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.

Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization.

See this YouTube video for a deeper understanding of how PCA works mathematically

1. Principal components analysis Intuition

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


import warnings
# Suppress FutureWarnings (optional)
warnings.simplefilter(action='ignore', category=FutureWarning)
```

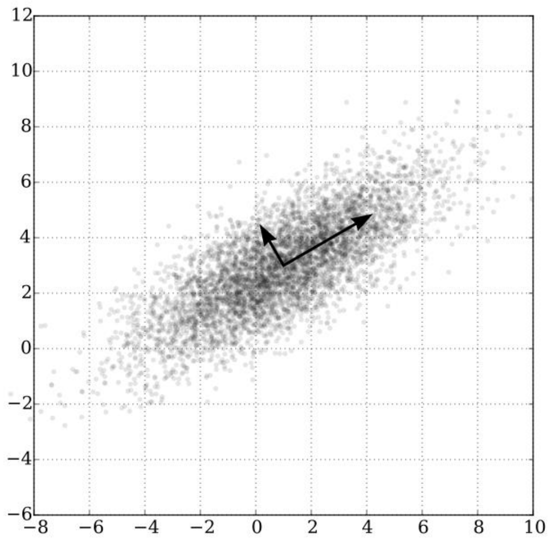
[2]:
```
df = pd.read_csv("height_weight.csv")
df.drop(columns=['Index'], inplace=True)
df.rename(columns={'Height(Inches)': 'height', 'Weight(Pounds)': 'weight'}, inplace=True)
df.weight = df.weight * 1.25 # artificially making everyone weigh 1.25 times because no one was overweight originally lol
df.head()
```

[2]:

	height	weight
0	65.78331	141.240625
1	71.51521	170.609125
2	69.39874	191.283625

Principal Component Analysis (Method)

- Given N data vectors from n-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute k orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the k principal component vectors
  - The principal components are sorted in order of decreasing "significance" or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, to reconstruct a good approximation of the original data)
- Works for numeric data only



Ack. Wikipedia: Principal Component Analysis