Unsupervised Learning

Madson L. D. Dias

Huawei / IFCE

March 26, 2021

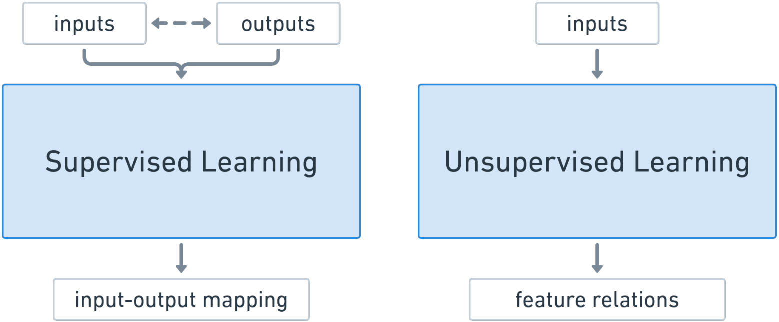
Agenda

Agenda

Agenda

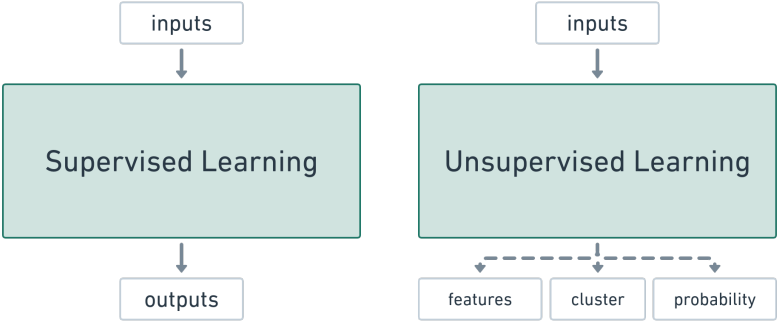
Unsupervised Learning

- A type of machine learning algorithm used to draw inferences from data sets consisting of **input data without labels**
- Task of inferring a function to describe **hidden structure** from unlabeled data.



Unsupervised Learning

- A type of machine learning algorithm used to draw inferences from data sets consisting of **input data without labels**
- Task of inferring a function to describe **hidden structure** from unlabeled data.



Agenda

1. Unsupervised Learning
   - Introduction
   - Applications
   - Supervised Learning vs. Unsupervised Learning
2. Clustering
   - Formal Definition
   - Types of clustering
   - Clustering using k-means
3. Dimensionality reduction
   - Principal Component Analysis

Applications of Unsupervised Learning

Clustering

- Process of **grouping data into different clusters** or groups.
- Same group $\rightarrow$ similar and different groups $\rightarrow$ dissimilar
- Used for analyzing and grouping data without labels

Dimensionality reduction

- Process of **reducing** the number of **variables**
- Simplify the data **without losing** too much information
- This method is also called feature extraction

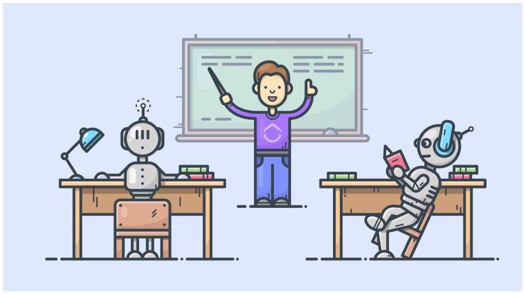
Applications of Unsupervised Learning

Anomaly detection

- Identification of **rare observations**, which brings suspicions
- The model is **trained** with a **lot of normal** instances
- Also called as Novelty Detection and Outlier Detection

Agenda

Supervised Learning vs. Unsupervised Learning



Parameter	Supervised Learning	Unsupervised Learning
Dataset	Labelled Dataset	Unlabelled Dataset
Learning	Guided learning	Guided by some metric
Complexity	Simpler method	Computationally complex
Accuracy	More Accurate	Less Accurate

Agenda

Agenda

Formal definition of Clustering

- Given a dataset $\mathcal{D} = \{\boldsymbol{x}_n \in \mathbb{R}^D\}_{n=1}^N$, where $\boldsymbol{x}$ is a vector with D dimensions and N is the number of elements of the data set[1]. The task of clustering is to find a function $g(\boldsymbol{x})$ in that $g : \mathbb{R}^D \to \mathbb{N}^K$, where $\mathcal{C} \in \mathbb{N}^K$ is a set of sets $\mathcal{C} = \{\mathcal{C}_k\}_{k=1}^{K<N}$ containing clusters $\mathcal{C}_k = \{\boldsymbol{x}_n : g(\boldsymbol{x}_n) = k\}$ such that

$$\mathcal{C}_k \neq \emptyset, \forall k \tag{1}$$

$$\bigcup_{k=1}^{K} \mathcal{C}_k = \mathcal{D} \tag{2}$$

$$\mathcal{C}_k \bigcap \mathcal{C}_j = \emptyset,\ \forall k, j \text{ where } k \neq j \tag{3}$$

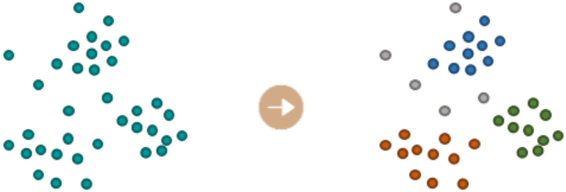
[1] The $\mathbb{R}^D$ space is called "data space" or "input space".

Agenda

Types of clustering algorithms

- Clustering itself can be categorized into two types
  - Hard Clustering
  - Soft Clustering
- We can also categorize clustering methods by their methodology and learning algorithms

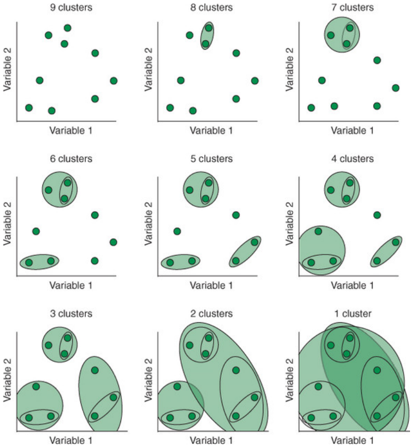
Density-based clustering



- In this method, the clusters are created based upon the **density of the data points** which are represented in the data space.
- Regions that become **dense** due to the huge number of data points residing in that region are considered as **clusters**
- Points in the **sparse** region are considered as noise or **outliers**

Hierarchical Clustering

- Divides the clusters based on the **distance metrics**
- **Agglomerative** or **Divisive**
- Create a **distance matrix** of all the existing clusters and perform the **linkage** between the clusters depending on the **criteria** of the linkage
  - Single Linkage
  - Complete Linkage
  - Average Linkage



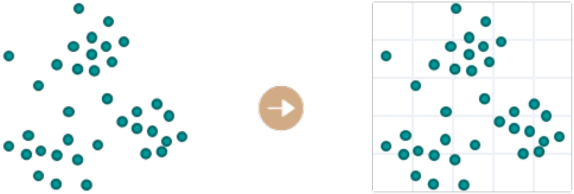
Partitioning Clustering

- The clusters are **partitioned** based upon the characteristics of the data points
- We need to specify the **number of clusters** to be created
- Iterative process to reassign the data points between clusters based upon the **distance**

Fuzzy clustering

- In fuzzy clustering, the assignment of the data points in any of the clusters is not decisive
- Here, one data point can belong to **more than one cluster**
- It provides the outcome as the **probability** of the data point belonging to each of the clusters.

Grid-based clustering



- The data set is represented into a **grid structure** which comprises of grids (also called cells).
- After partitioning the data sets into cells, it computes the **density** of the cells which helps in identifying the clusters.
- This makes it appropriate for dealing with **big data sets**

Agenda

k-means clustering

- One of the most popular choices for analysts to create clusters.
- We need to specify the number of clusters to be created for this clustering method.

Definition

- Given a dataset $\mathcal{D} = \{\boldsymbol{x}_n \in \mathcal{R}^D\}_{n=1}^N$, k-means clustering aims to partition the N observations into $K(\leq N)$ sets so as to minimize the *within-cluster* sum of squares (i.e. variance). Formally, the objective is to find:

$$\mathcal{C}^\star = \arg\min_{\mathcal{C}} \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{C}_k} \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2 \tag{4}$$

where $\boldsymbol{\mu}_k$ is the centroid of the cluster $\mathcal{C}_k$.

k-means clustering

Algorithm

1. Specity the number K of clusters to assingn
2. Randomly initialize K centroids
3. Repeat until the centroid positions do not change
    3.1. Assign each point to its closest centroid
    3.2. Compute the new centroid (mean) of each cluster

"Talk is cheap.
Show me the code."

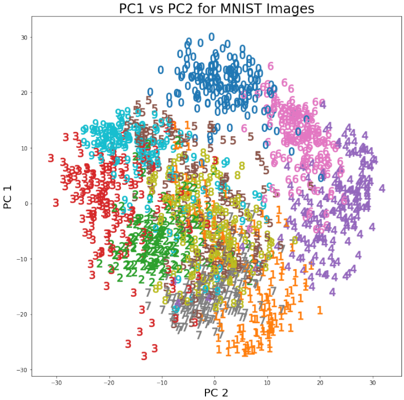
- Linus Torvalds

https://github.com/omadson/vds

Agenda

Dimensionality reduction

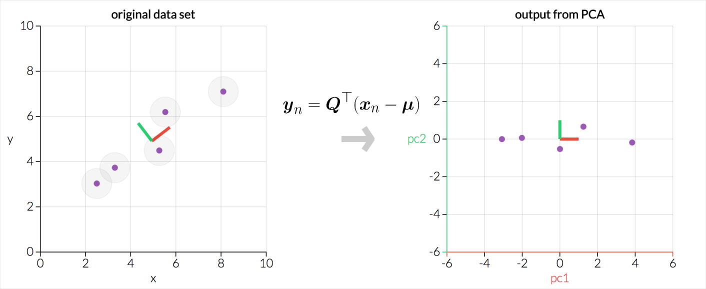
- Transformation of data from a high-dimensional space into a **low-dimensional space**
- Ideally close to its **intrinsic dimension**.
- Methods
  - Feature selection
  - Feature projection



Agenda

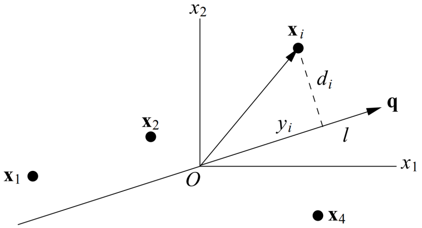
Principal Component Analysis

- PCA is often used to **reduce the dimensionality** of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.



Principal Component Analysis

- Consider a set of points $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$ in a D-dimensional space, such that their mean $\boldsymbol{\mu} = \mathbf{0}$, i.e., centroid is at the origin.



- The PCA want to finds a line l through the origin that maximizes the projections x'_n of the points $\boldsymbol{x}_n$ on l.
- Let $\boldsymbol{q}$ denote the unit vector along line l.

Principal Component Analysis

- The projection x'_n of $\boldsymbol{x}_n$ on l is $x'_n = \boldsymbol{x}_n^\top \boldsymbol{q}$.
- The mean squared projection is the variance V over all points

$$
\begin{aligned}
V &= \frac{1}{N} \sum_{n=1}^{N} {x'_n}^2 && (5) \\
&= \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{x}_n^\top \boldsymbol{q})^2 && (6) \\
&= \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{q}^\top \boldsymbol{x}_n)(\boldsymbol{x}_n^\top \boldsymbol{q}) && (7) \\
&= \boldsymbol{q}^\top \left[\frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top \right] \boldsymbol{q} && (8)
\end{aligned}
$$

Principal Component Analysis

- The middle factor is the covariance matrix $\mathbf{C}$ of the data points

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top \tag{9}$$

- We want to find a unit vector $\boldsymbol{q}$ that maximizes the variance V

$$\boldsymbol{q}^\star = \arg\max_{\boldsymbol{q}} \boldsymbol{q}^\top \mathbf{C} \boldsymbol{q}, \tag{10}$$

- Put $\|\boldsymbol{q}\| = 1$ constraint to avoid overflow
- The constraint optimization problem

$$\text{maximize}\, V = \boldsymbol{q}^\top \mathbf{C} \boldsymbol{q} \quad \text{subject to} \quad \|\boldsymbol{q}\| = 1. \tag{11}$$

Lagrange multiplier method

- Consider this problem

$$\text{maximize} \quad f(\boldsymbol{x}) \quad \text{subject to} \quad g(\boldsymbol{x}) = c. \tag{12}$$

- Lagrange multiplier method introduces a Lagrange multiplier λ to combine $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ as

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda(g(\boldsymbol{x}) - c) \tag{13}$$

- Then, we can solve using

$$\frac{\partial L(\boldsymbol{x}, \lambda)}{\partial \boldsymbol{x}} = 0, \quad \frac{\partial L(\boldsymbol{x}, \lambda)}{\partial \lambda} = 0 \tag{14}$$

Principal Component Analysis

- Use Lagrange multiplier method, combine V and the constraint
- Consider this problem

$$\text{maximize} \quad L = \boldsymbol{q}^\top \mathbf{C} \boldsymbol{q} - \lambda(\boldsymbol{q}^\top \boldsymbol{q} - 1) \tag{15}$$

- Now, we differentiate L with respect to $\boldsymbol{q}$ and λ:

$$\frac{\partial L}{\partial \boldsymbol{x}} = 2\boldsymbol{q}^\top \mathbf{C} - 2\lambda \boldsymbol{q}^\top = 0 \tag{16}$$

$$\frac{\partial L}{\partial \lambda} = \boldsymbol{q}^\top \boldsymbol{q} - 1 = 0 \tag{17}$$

- Eq. (16) gives

$$\boldsymbol{q}^\top \mathbf{C} = \lambda \boldsymbol{q}^\top \Longleftrightarrow \mathbf{C}\boldsymbol{q} = \lambda \boldsymbol{q} \tag{18}$$

- This is called an *eigenvector equation*

Principal Component Analysis

General PCA

- PCA transforms $\boldsymbol{x}_n$ into a new vector $\boldsymbol{y}_n$ through $\boldsymbol{Q}$ as follows:

$$\boldsymbol{x}_n' = \boldsymbol{Q}^\top(\boldsymbol{x}_n - \boldsymbol{\mu}) = \sum_{m=1}^{M} (\boldsymbol{x}_n - \boldsymbol{\mu})^\top \boldsymbol{q}_m \boldsymbol{q}_m \tag{19}$$

- Each component of $\boldsymbol{x}_n'$ is

$$x_{nm}' = (\boldsymbol{x}_n - \boldsymbol{\mu})^\top \boldsymbol{q}_m \tag{20}$$

- This is the projection of $\boldsymbol{x}_n - \boldsymbol{\mu}$ on $\boldsymbol{q}_m$.

“Talk is cheap.
Show me the code.”

- Linus Torvalds

https://github.com/omadson/vds

Bye-Bye!

Thank you!