DDA3020 Machine Learning: L14 Introduction to Unsupervised Learning

Jicong Fan
School of Data Science, CUHK-SZ

11/11/2025

Outline

1. Motivation
2. Definition
3. Main Approaches
   - Clustering
   - Dimensionality reduction
   - Density estimation
   - Autoencoder
   - Self-supervised Learning

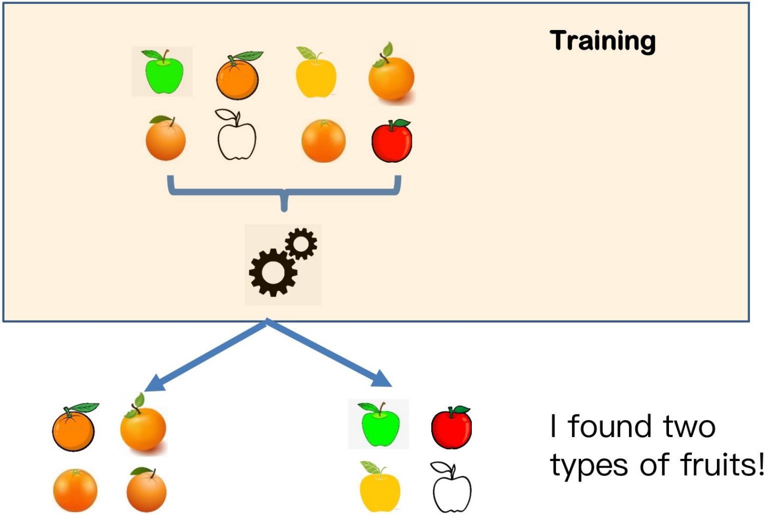
1. Motivation

2. Definition

3. Main Approaches
   - Clustering
   - Dimensionality reduction
   - Density estimation
   - Autoencoder
   - Self-supervised Learning

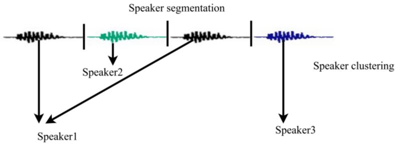
Unsupervised learning

A toy example of clustering.



Motivation of Unsupervised Learning

Motivation 1: Human has the ability of partitioning unlabeled data into some groups (*i.e.*, clustering), such that some useful structure of the data can be found.



Speaker diarization

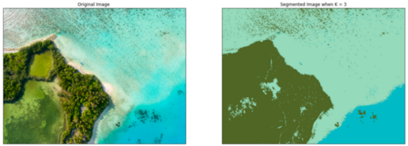
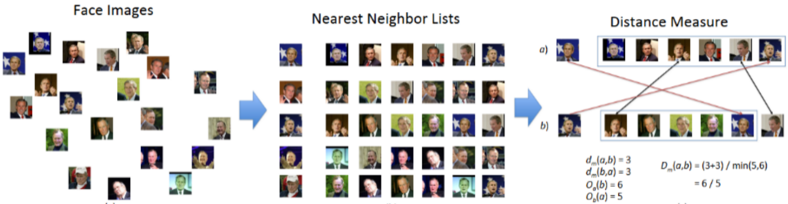


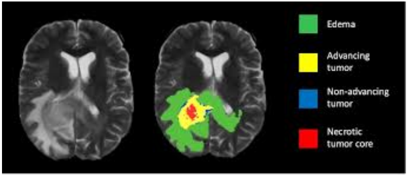
Image segmentation

Face clustering by identity



Motivation 2: In practice, it is difficult to obtain sufficient labels:

- **Labeling is expensive**: In some tasks, the labeling should be conducted by experts, such as medical imaging.
- **Labeling is time-consuming**: supervised learning of deep neural networks requires large-scale labeled databases. For example, ImageNet includes 1 million labeled images of 1,000 categories.



Thus, it is useful and necessary to conduct machine learning with unlabeled data, *i.e.*, unsupervised learning.

Definition of Unsupervised Learning

- In **unsupervised learning**, the dataset is a collection of **unlabeled examples** $\{\mathbf{x}_i\}_{i=1}^M$
- Again, $\mathbf{x}$ is a feature vector, and the goal of an **unsupervised learning algorithm** is to create a **model** that takes a feature vector x as input and either transforms it into another vector or into a value that can be used to solve a practical problem (anomaly detection, data compression, discovery of new species)

Supervised Learning vs. Unsupervised Learning

Supervised learning:

- Training set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$
- The goal is to fit the relationship from the input $\mathbf{x}$ to the label y, which represents the desired behavior for your model
- We can define some evaluation metrics based on the label, such as accuracy

Unsupervised learning:

- Training set $D = \{(\mathbf{x}_i)\}_{i=1}^N$
- The absence of labels means the absence of a solid reference point to judge the quality of your model
- The evaluation metrics should be defined based on your own task (*e.g.*, clustering, dimensionality reduction)

Unsupervised Learning

Main Approaches

- **Clustering**
  - Groups a set of objects in such a way that objects in the same group (called a **cluster**) are **more similar** (in some sense) to each other than to those in other groups (clusters).
- **Dimensionality Reduction by Principal Components Analysis**
  - Find a new low-dimensional space to re-present the data points in the original high-dimensional space, while the data structure is preserved as much as possible
- **Density Estimation**
  - Models the probability density function (pdf) of the unknown probability distribution from which the dataset has been drawn.
- **Autoencoder**
- **Self-supervised Learning**

There are several approaches to unsupervised learning. In later lectures, we will focus on clustering and dimensionality reduction.

Reference (Book1, Chapter 9):
https://www.dropbox.com/s/y9a7b0hzmuksqar/Chapter9.pdf?dl=0

Clustering

- Clustering is a problem of learning to assign labels to examples by leveraging an unlabeled dataset.
- The following example is k-means clustering, which will be introduced in detail in the next slides.

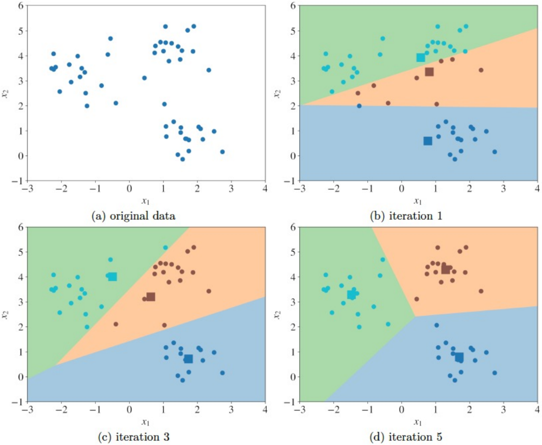
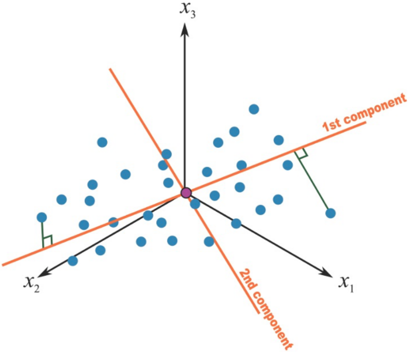


Figure 2: The progress of the k-means algorithm for $k = 3$.

1. Motivation
2. Definition
3. Main Approaches
   - Clustering
   - Dimensionality reduction
   - Density estimation
   - Autoencoder
   - Self-supervised Learning

Dimensionality reduction

- Principal component analysis (PCA): a technique that converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables, called **principal components**.
- In the following example, the task is finding two orthogonal coordinates (corresponding to the 1st and 2nd component) to represent the data points in the original 3-dimensional space, while reserving the data structure
- We will introduce the details of PCA in later lectures.



Density estimation

- In existing machine learning tasks, we always assume that the training set $D = \{(\mathbf{x}_i)\}_{i=1}^N$ is sampled from a distribution $P(\mathcal{X})$.
- However, in practice, we cannot explicitly write the underlying distribution.
- Thus, an important task is to estimate the distribution based on some observed data, called density estimation.

Kernel Density estimation

The most typical density estimation approach is kernel density estimation (KDE).

- Let $D = \{(x_i)\}_{i=1}^N$ be a one-dimensional dataset, whose examples were drawn from a distribution with an unknown probability density function (pdf) f. Our task is to model the shape of f based on D.
- We adopt the following kernel model:

$$\hat{f}_b(x) = \frac{1}{Nb} \sum_{i=1}^{N} k\left(\frac{x - x_i}{b}\right), \quad k(z) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-z^2}{2}\right),$$

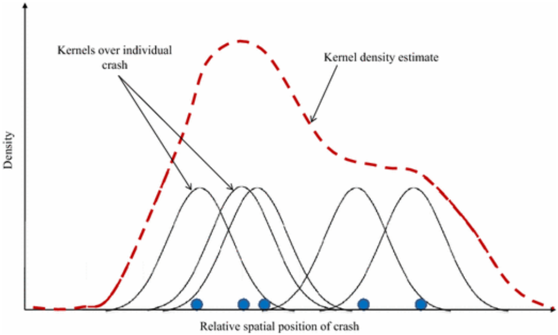
  where $b > 0$ is a hyper-parameter called kernel size, which can be tuned using K-fold cross-validation. Note that here we give the Gaussian kernel as an example, but you can choose other kernels in practice.

Kernel Density estimation

Example:

- Suppose D includes 5 one-dimensional points (see the blue points), we want to estimate the behind pdf based on these 5 points, using the KDE method.
- We adopt the following kernel model:

$$\hat{f}_b(x) = \frac{1}{Nb}\sum_{i=1}^{N} k\left(\frac{x - x_i}{b}\right), \quad k(z) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-z^2}{2}\right).$$



Kernel Density estimation

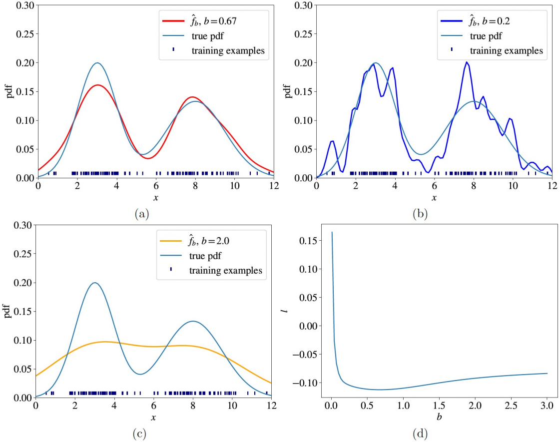


Figure 1: Kernel density estimation: (a) good fit; (b) overfitting; (c) underfitting; (d) the curve of grid search for the best value for b.

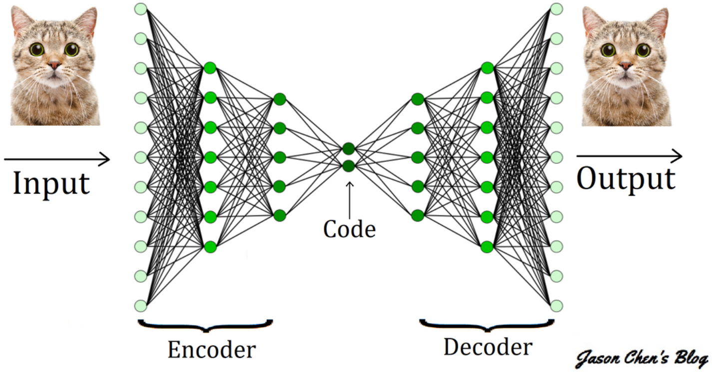
Further readings for KDE:

https://en.wikipedia.org/wiki/Kernel_density_estimation

Demo with code: https://scikit-learn.org/stable/modules/density.html

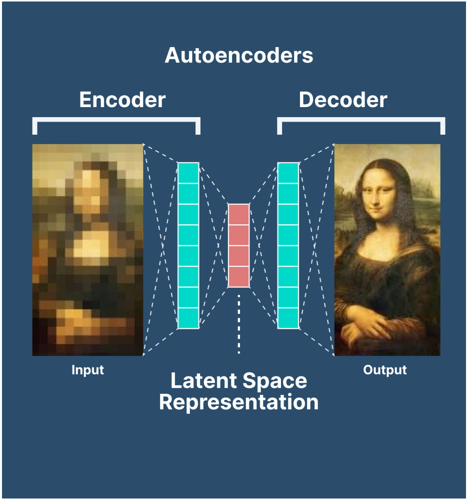
Autoencoder

- An autoencoder is a type of artificial neural network used to learn efficient coding of unlabeled data. Thus, it is unsupervised learning.
- In the following example, we have learned an autoencoder network that can reconstruct natural images. The activation vector of the middle layer can be seen as a low-dimensional representation of the input image.



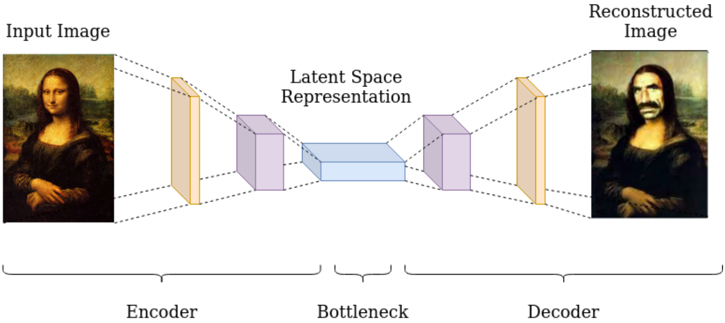
Autoencoder

- Autoencoder can be used to recover old/blurring images.



Autoencoder

- Autoencoder can be used to do style transfer.



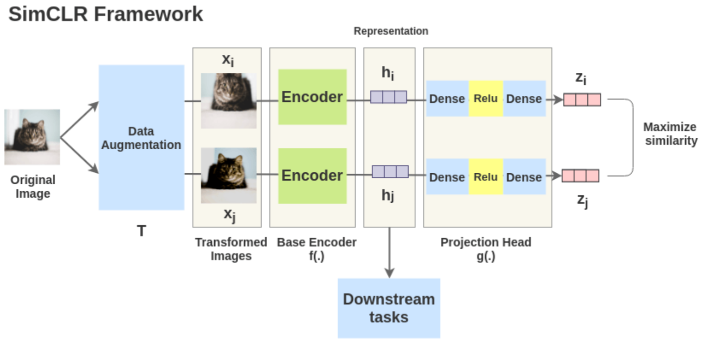
Autoencoder

Autoencoder can also be used to generate fake videos, called DeepFakes.
Let's see a video https://www.youtube.com/watch?v=AmUC4m6w1wo

Further readings for Autoencoder:
https://en.wikipedia.org/wiki/Autoencoder
Blog: https://theaisummer.com/deepfakes
Blog: https://zhuanlan.zhihu.com/p/58111908
Demo with code: https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial9/AE_CIFAR10.html

Self-supervised Learning

Contrastive self-supervised learning techniques are a promising class of methods that build representations by learning to encode what makes two things similar or different.



Further readings for self-supervised learning:
https://arxiv.org/pdf/2002.05709.pdf
https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-l
html
Code: https://github.com/sthalles/SimCLR

Self-supervised learning

Advantages of self-supervised learning (SSL):

- Pre-train large-scale deep neural networks: in the recent 3 years, there has been a trend of pre-training very large-scale deep neural networks, which can provide good feature representation for different downstream tasks. Self-supervised learning is a "perfect" approach for the large-scale pre-training task, since it requires very large-scale data, and the pre-trained model should be generalized to different downstream tasks.
- Sometimes, there are noisy labels or malicious labels in training set. The learned model by SSL will not be influenced by such labels. One interesting example is that SSL has been successfully applied to defend the backdoor attacks (an emerging topic about AI security). If interested, please read the second link below (*i.e.*, DBD).

Further readings for self-supervised learning:
`https://zhuanlan.zhihu.com/p/378360224`
Paper of DBD: `https://openreview.net/pdf?id=TySnJ-0RdKI`
Code of DBD: `https://github.com/SCLBD/DBD`