STOR566: Introduction to Deep Learning

Lecture 2: Overview of Machine Learning

Yao Li
UNC Chapel Hill

Aug 18, 2022

Materials are from *Deep Learning (UCLA)*

Outline

- Overview of machine learning
- Colab tutorial

Machine Learning: Overview

Human Learning

Observation **Learning** **Decision rule**

Machine Learning

Training Data

 → 0

 → 0

 → 1

Machine Learning

Decision rule

Machine Learning

Training Data | **Machine Learning** | **Decision rule**

 → 0

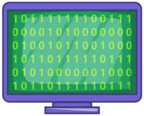
x_1 y_1

 → 0

x_2 y_2

 → 1

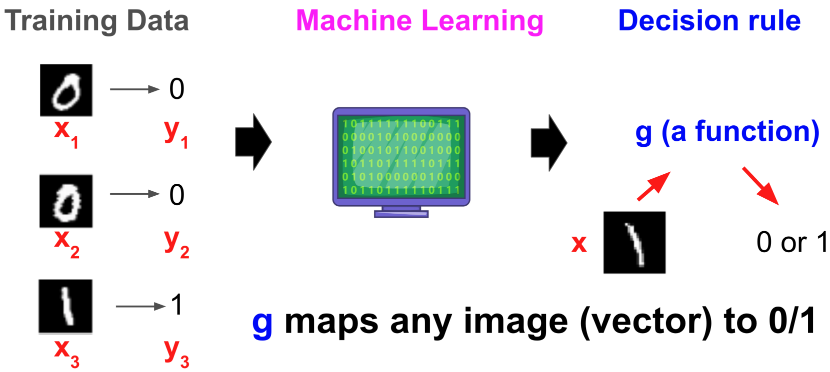
x_3 y_3



x_1: vector of pixel values [0, 24, 128, ...]

y_1: 0 or 1

Machine Learning



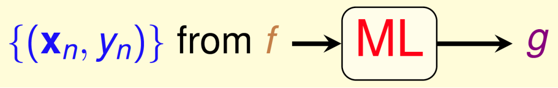
Formalize the Learning Problem

- Input: $\boldsymbol{x} \in \mathcal{X}$ (an image)
- Output: $y \in \mathcal{Y}$ (class)
- Target function to be learned:

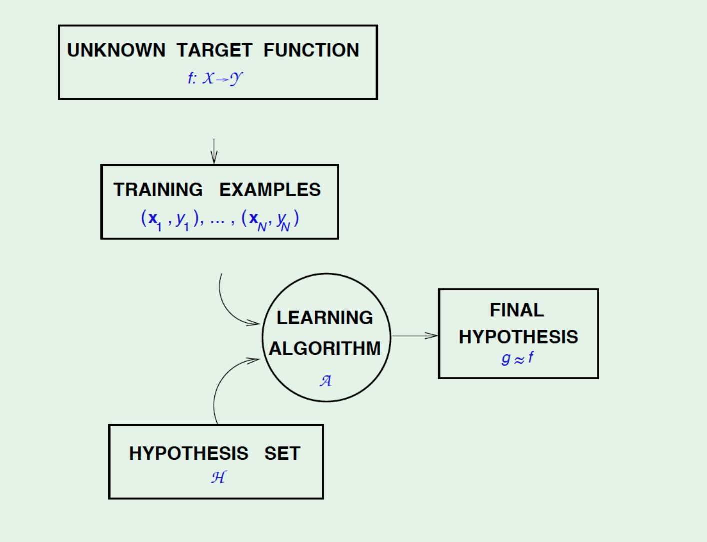
  $f : \mathcal{X} \to \mathcal{Y}$ (ideal image classification function)
- Data:

  $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_N, y_N)\}$
- Hypothesis (model)

  $g : \mathcal{X} \to \mathcal{Y}$ (**learned** formula to be used)



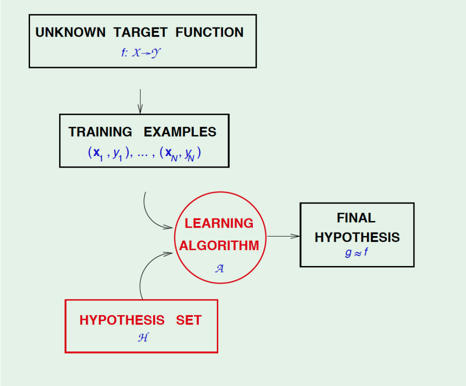
Basic Setup of Learning Problem



(Figure from "Learning from Data")

Learning Model

- A learning model has two components:
  - The hypothesis set $\mathcal{H}$:
    Set of candidate hypothesis (functions)
  - The learning algorithm:
    To pick a hypothesis (function) from the $\mathcal{H}$
    Usually **optimization algorithm** (choose the best function to minimize the training error)



Binary classification

- Data:
  - Features for each training example: $\{\boldsymbol{x}_n\}_{n=1}^N$, each $\boldsymbol{x}_n \in \mathbb{R}^d$
  - Labels for each training example: $y_n \in \{+1, -1\}$
- Goal: learn a function $f : \mathbb{R}^d \to \{+1, -1\}$
- Examples:
  - Credit approve/disapprove
  - Email spam/not-spam
  - patient sick/not sick
  - ...

Types of model (hypothesis)

- Linear hypothesis space:

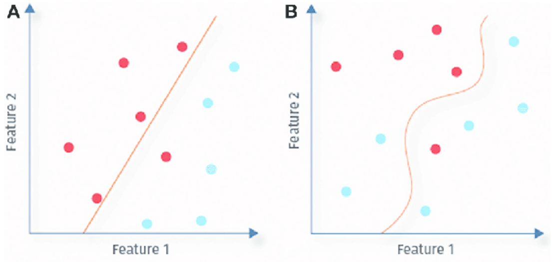
$$h(\boldsymbol{x}) = \text{sign}(\sum_{i=1}^{d} w_i x_i - \text{threshold})$$

- Feed forward (fully connected) network:

$$h(\boldsymbol{x}) = \text{sign}(W_L \cdots \sigma(W_2 \sigma(W_1 \boldsymbol{x} + b_1) + b_2) + b_L)$$

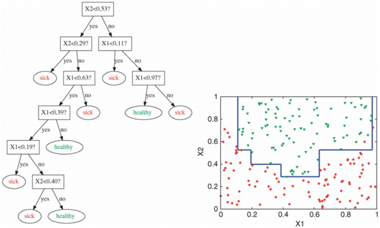
- Tree-based models
- ...

Types of model



Linear classification

Nonlinear classification



Tree-based classification

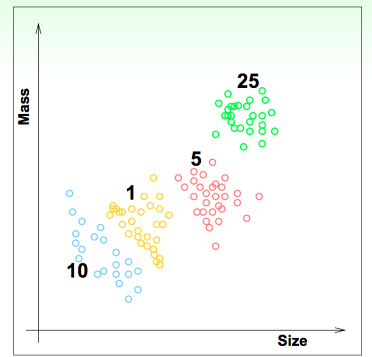
Other types of output space - Regression

- Regression: $y_n \in \mathbb{R}$ (output is a real number)
- Example:
  - Stock price prediction
  - Movie rating prediction
  - $\cdots$

Other types of output space - Multi-class prediction

Multi-class classification:

- $y_n \in \{1, \cdots, C\}$ (C-way classification)
- Example: Coin recognition
  - Classify coins by two features (size, mass) ($x_n \in \mathbb{R}^2$)
  - $y_n \in \mathcal{Y} = \{1c, 5c, 10c, 25c\}$
    ($\mathcal{Y} = \{1, 2, 3, 4\}$)
- Other examples: hand-written digits, $\cdots$

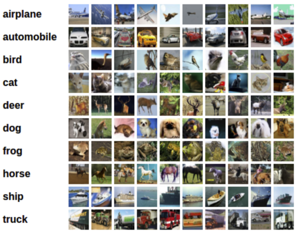


Other types of output space - Multi-class prediction

Multi-class classification:

- More examples: hand-written digit recognition, object classification, $\cdots$

MNIST



CIFAR

Other types of output space - Multi-label prediction

- Multi-class problem: Each sample only has one label
- Multi-label problem: Each sample can have multiple labels

Other types of output space - Multi-label prediction

- Multi-class problem: Each sample only has one label
- Multi-label problem: Each sample can have multiple labels
- Example:
  - Document categorization (news/sports/economy/$\cdots$)
  - Document/image tagging
  - $\cdots$

Other types of output space - Multi-label prediction

- Extreme classification (large output space problems):
  - Millions of billions of labels (but usually each sample only has few labels)
  - Recommendation systems: Predict a subset of preferred items for each user
  - Document retrieval or search: Predict a subset of related articles for a query
- Other examples:

1. A red stop sign sitting on the side of a road.
2. A stop sign on the corner of a street.
3. A red stop sign sitting on the side of a street.

Machine Learning Problems

Machine learning problems can usually be categorized into

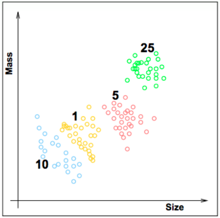
- Supervised learning
  (semi-supervised learning)
- Unsupervised learning
- Transfer learning

Unsupervised Learning (no y_n)

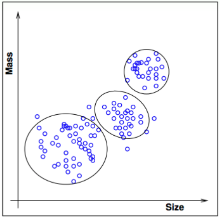
- Example: clustering

  Given examples $x_1, \ldots, x_N$, classify them into K classes

- Other unsupervised learning:
  - Outlier detection: $\{x_n\} \Rightarrow \text{unusual}(x)$
  - Dimensional reduction
  - $\ldots$

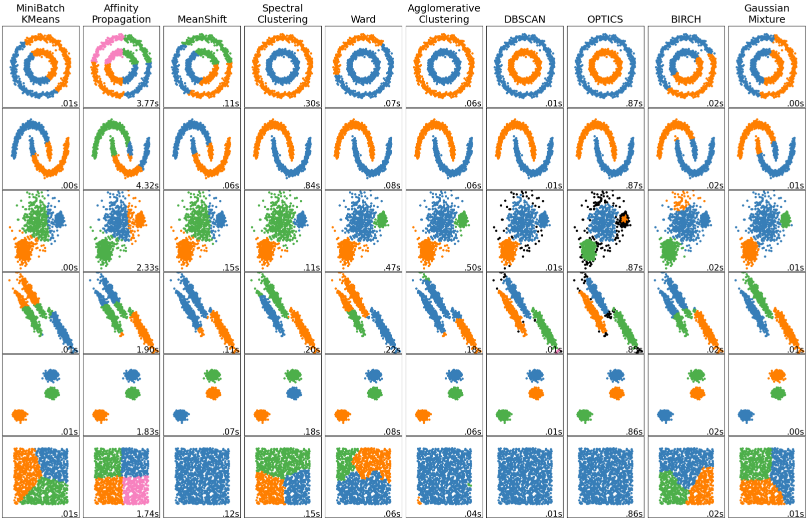


supervised multiclass classification



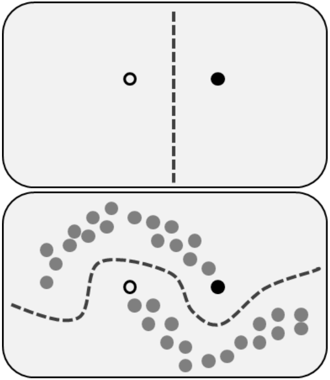
unsupervised multiclass classification
$\Longleftrightarrow$ 'clustering'

Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture
.01s
3.77s
.11s
.30s
.07s
.06s
.01s
.87s
.02s
.00s
.00s
4.32s
.06s
.84s
.08s
.06s
.01s
.87s
.02s
.01s
.00s
2.33s
.15s
.11s
.47s
.50s
.01s
.87s
.02s
.01s
.01s
1.90s
.11s
.20s
.22s
.18s
.01s
.85s
.02s
.01s
.01s
1.83s
.07s
.18s
.08s
.06s
.01s
.86s
.02s
.00s
.01s
1.74s
.12s
.15s
.06s
.04s
.01s
.86s
.02s
.01s

Semi-supervised learning

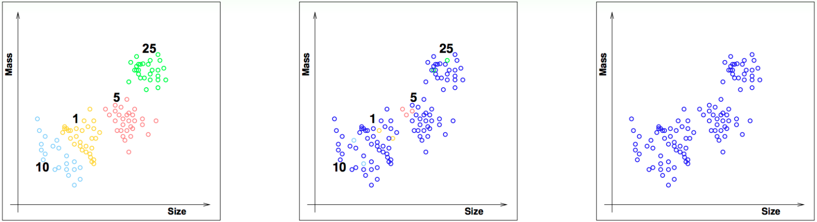
- Only some (few) $\boldsymbol{x}_n$ has y_n
- Labeled data is much more expensive than unlabeled data



(From Wikipedia)

Semi-supervised learning

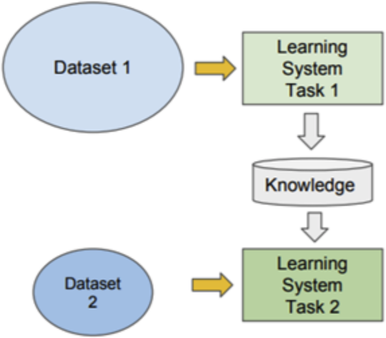
- Only some (few) $\boldsymbol{x}_n$ has y_n
- Labeled data is much more expensive than unlabeled data



supervised　semi-supervised　unsupervised (clustering)

Transfer learning

- Source dataset D_{source} and target dataset D_{target}
- How to leverage the information of D_{source} to improve the performance of target task?



Self-supervised learning

- The pretraining can be done with unlabeled data (easy to collect gigantic unlabeled data)
  - Example: We can get almost unlimited unlabeled text from Internet
- Define the training task based on unlabeled data
  - Example: predict a word in a sentence
- Transfer the model to end task

Original sentence:

In Autumn the leaves fall from the trees.

Masked sentence:

In Autumn the [ ] fall from the trees.

leaves
apples
raindrops
branches

Predicted words by the model

Masked language modeling

Contrastive learning

Conclusions

- Basic concept of learning:
  - Set up a hypothesis space (model class/potential functions)
  - Define an error measurement (define the quality of each function based on data)
  - Develop an algorithm to choose a good hypothesis based on the error measurement (optimization)
- Binary classification, multiclass, multilabel, etc.
- Different learning scenarios

Questions?