Lecture 1

What is ML?

(Tom Mitchell) A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Algorithms that find patterns in data

Methods that predict outcomes of a process by fitting a model

You can pretty much understand a "machine learning model" as a function that maps the inputs to outputs.

> The goal of machine learning is for the machines to figure out suitable functions themselves.

Think about complex tasks

- e.g. Want to balance a cup of water
  - **How many if - else** statements do we need?
    - Very hard because
      - need to understand image
      - need solution to balancing
- It is hard to program out rules. ML helps us with it.

Humans learn similarly, by looking at examples and learning from them

Examples

1. Email spam detection - given email content, is it spam or not?
2. Image recognition - given image of an animal, identify the species
3. Speech translation - given text in English translate to French

etc.

In this course, there are 2 main categories of ML problems

1) Supervised Learning

Given a dataset of (x, y)s, learn the mapping from x to y

Given an input $\vec{x} \in \mathbb{R}^d$, want to find parameters or configuration θ of the function / model f such that $f(\vec{x}, \theta)$ predicts output $y \in \mathbb{R}$

Specifically, given training data $\{(\vec{x}_i, \vec{y}_i)\}_{i=1}^N$, want to find θ s.t. $f(\vec{x}, \theta)$ is good (i.e. close to output y_i)

Tells us how far we are from the solution

For every input, there is a true outcome (why it's called supervised learning)

Two kinds of supervised learning: classification and regression

Classification

When y is discrete output, $y \in \{1, ..., C\}$ i.e. C classes

e.g.

email spam detection ($C = \{spam, not\ spam\}$

digit recognition - given image, predict the digit ($C = \{0, ..., 9\}$

$C = 2$ → binary classification

$C > 0$ → multiclass classification

Regression

When y is continuous output, $y \in \mathbb{R}$

e.g.

Given historical data, predict a stock's value

Given a video, predict a car's constant velocity

2) Unsupervised Learning

given an unlabelled data set(just x's), figure out an interesting structure (e.g. social network analysis for groups of friends)

Unlike supervised, we cannot say what is right or wrong cause we don't have a notion of what is correct

- training inputs $\{\vec{x}_i\}_{i=1}^N$
- Modeling, discovery
- **Dimensionality reduction**

  Given high dimensional input x, want lower dimensional encoding / representation

  e.g. Images: 1024 x 768 x 3 bytes to capture information, but has a lot of redundancy, so we can compress this.
- Feature selection (the process of reducing the number of input variables when developing a predictive model)

Clustering

Given data x, can you group them based on some similarity measure

e.g. Document clustering

Density Estimation

Give data x, can you find the probability distribution that generates the data x ?

e.g. What sentences are more likely to occur?

Central Problems in ML

1) Models

How do we choose a good model?

Density estimation - what prob. dist. should I choose?

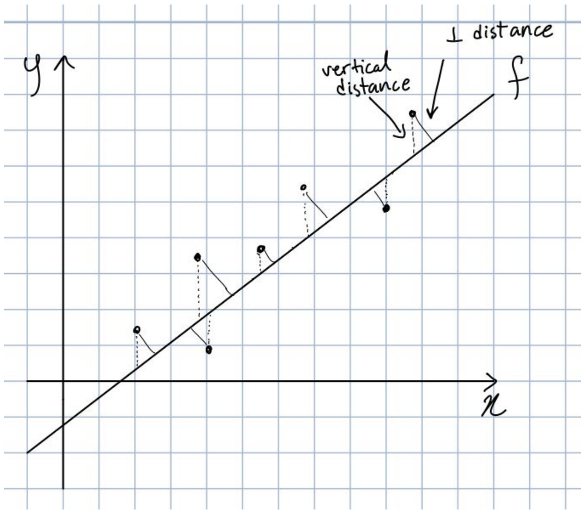
Regression - should I choose constant, linear, quadratic, etc.?

2) Loss Function

Provides a quantitative approach to measure how good a model is, $L \in [0, +\infty)$.

Usually when $L = 0$, the model is perfect

e.g. Regression



Depending on the task, you choose different loss functions.

e.g. Classification

$L =$ Count # of errors made

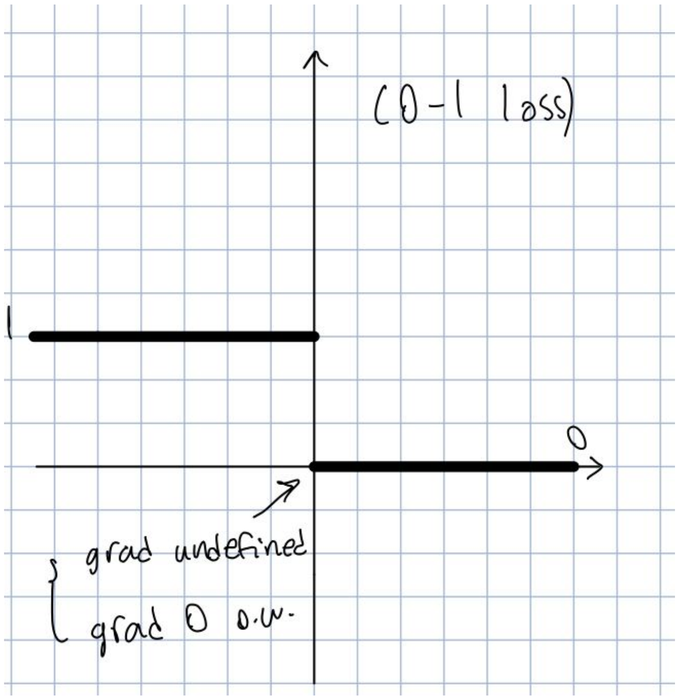
$= 1 - sign(y_i . f(x_i))$, where $y_i \in \{-1, 1\}$

count these

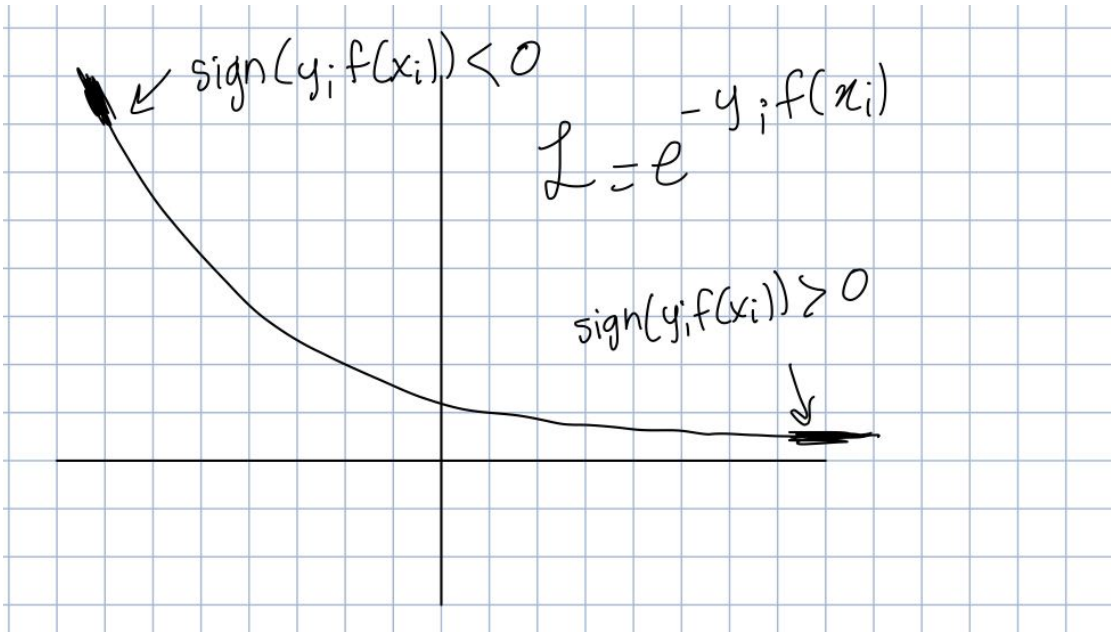
→ If y_i has same sign as $f(x_i)$ ⇒ right

→ otherwise ⇒ wrong

To tune θ we often look at gradients

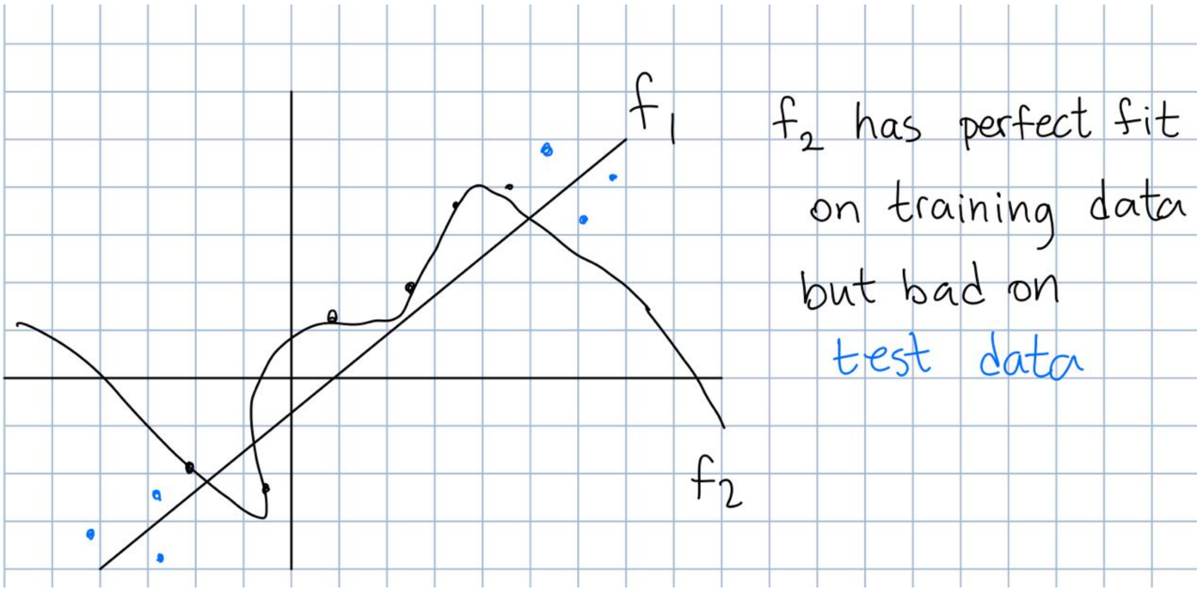


0 - 1 Loss is bad because it gives us "no" info.



3) Generalization

Ideally, model fitting well on training data is not sufficient, want to do well on test data as well.



Under-fitting - Can't fit well on training and testing data

Overfitting - Fit well on training but not testing

Want simple model that fits both data well enough (occam's razor)

Training vs. Testing data

- training data is collected before training the model (used for making the model)
- testing data is the new data (examples) you see after training model (used for checking the validity of the model)