Lecture 4

Introduction to Machine Learning

Ashish Dandekar

Lecture Overview

Formalism

Linear Regression

Statistical Learning

Introduction

Motivation

Something that is easy for a computer!

If $x = 2$,

$$
\begin{aligned}
f(x) &= 3x + 5, & y &= 11 \\
f(x) &= e^{\sin x}, & y &= 2.48 \\
f(x) &= x^2 + 0.2x, & y &= 4.4
\end{aligned}
$$

Given the functional form f and data x compute y.

Motivation

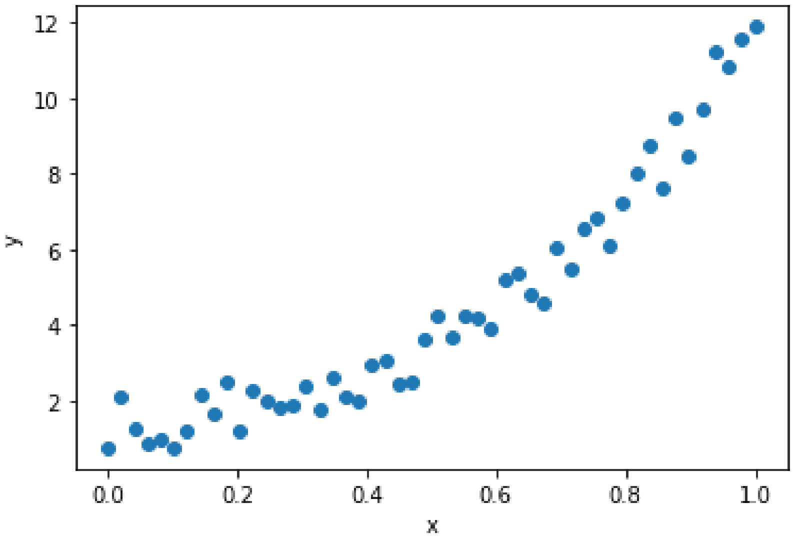
Something that is easy for a computer!

If $x = 2$,

$$
\begin{aligned}
f(x) &= 3x + 5, & y &= 11 \\
f(x) &= e^{\sin x}, & y &= 2.48 \\
f(x) &= x^2 + 0.2x, & y &= 4.4
\end{aligned}
$$

Given the functional form f and data x compute y.

Something that is *not* easy for a computer!

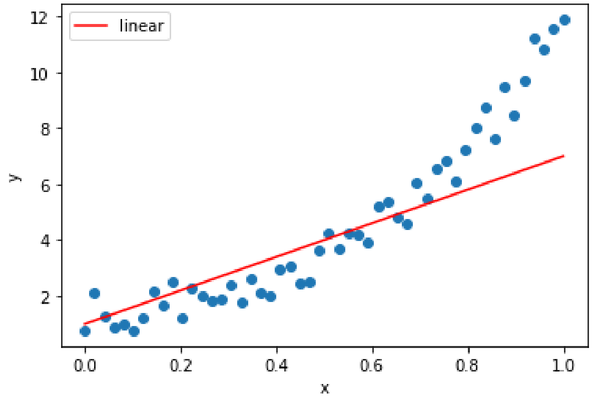


Given the data x and labels y, find the functional form f.

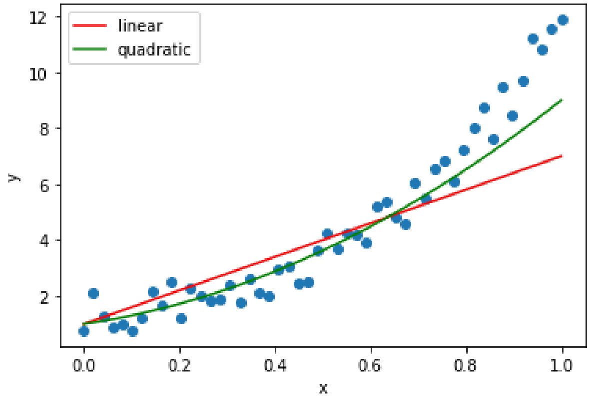
Motivation

Which function truly represents the data?

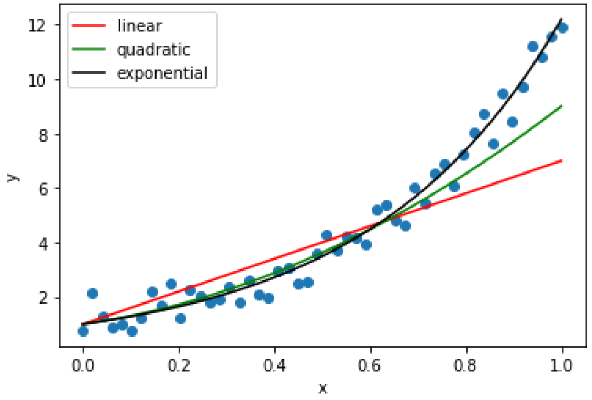
Linear function



Quadratic function



Exponential function



Definition

Machine Learning

Given a dataset, machine learning is the discovery of a function from *the set of possible functions* that *accurately* represents patterns in the dataset.

Hypothesis Set

Hypothesis set is set of all possible function that would map a dataset to the desired output.

- $H_{linear} = \{ax + b \mid a, b \in \mathbb{R}\}$
- $H_{quadratic} = \{ax^2 + bx + c \mid a, b, c \in \mathbb{R}\}$
- $H_{exponential} = \{e^{ax} \mid a \in \mathbb{R}\}$

Goodness of fit

A measure of evaluation to quantify how good a particular function fits the observed data.

For instance, root mean squared error (RMSE)

$$\sqrt{\frac{\sum_i (h(x_i) - y_i)^2}{n}}$$

Notation

Dataset

- A training dataset is denoted by D.
- Unless specified every dataset comprises of n datapoints.
- A **labeled** datapoint d_i is represented as a pair (x_i, y_i) where y_i is the label whereas x_i is a vector of the rest of the features of the datapoint.

Hypothesis Set

- Hypothesis set is denoted by H.

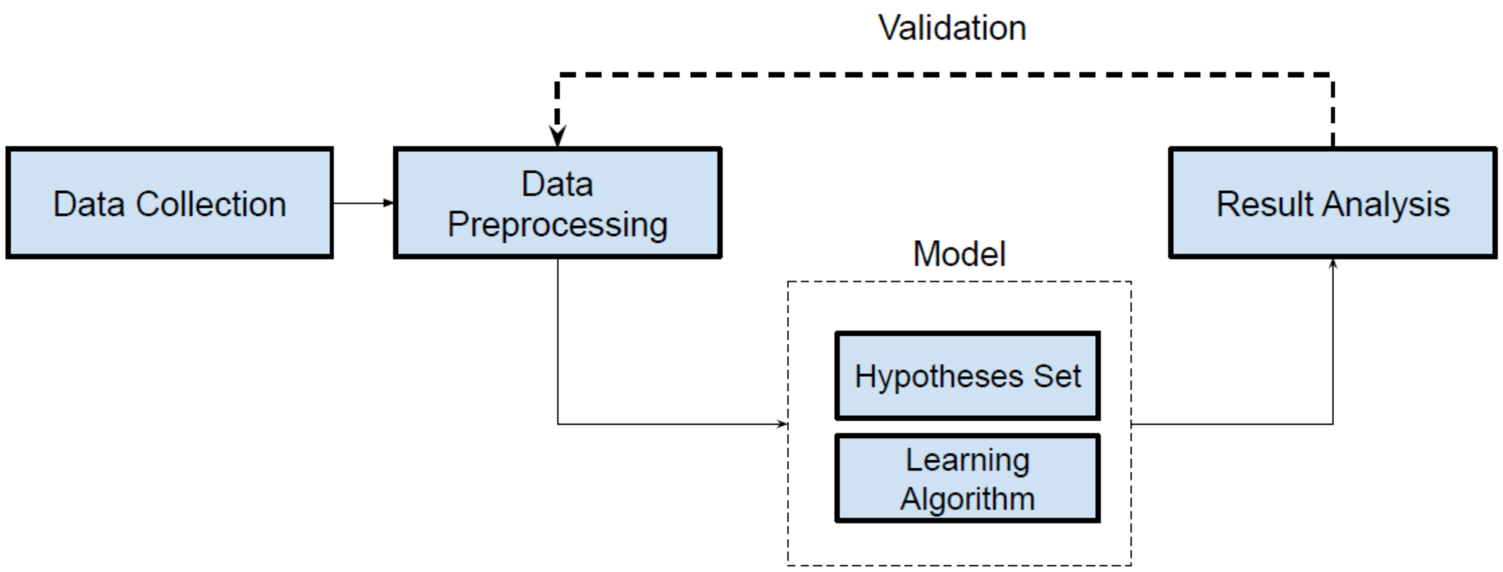
Goodness of fit

- It is generally known as a **loss function**.
- It is function of a hypothesis and the dataset. It quantifies the error under the specified hypothesis on the given dataset.
- It is denoted by $\ell_{\mathsf{D}}(h)$.

Notation

Given a dataset D, a hypothesis set H and the loss function ℓ machine learning can be defined as the following optimisation problem.

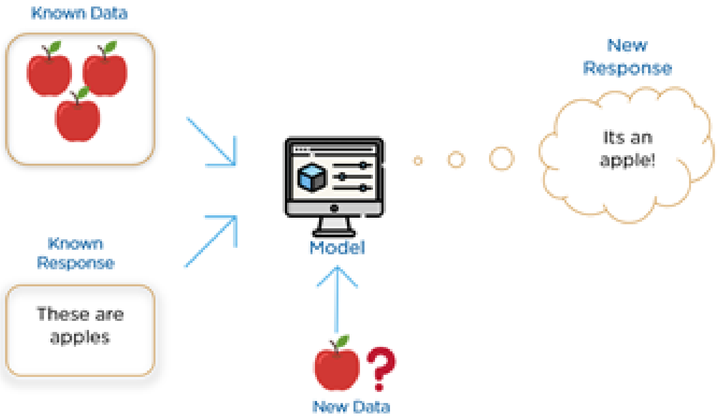
$$\hat{h} = \arg\min_{h \in \mathsf{H}} \ell_D(h)$$



Types of learning

Supervised Learning

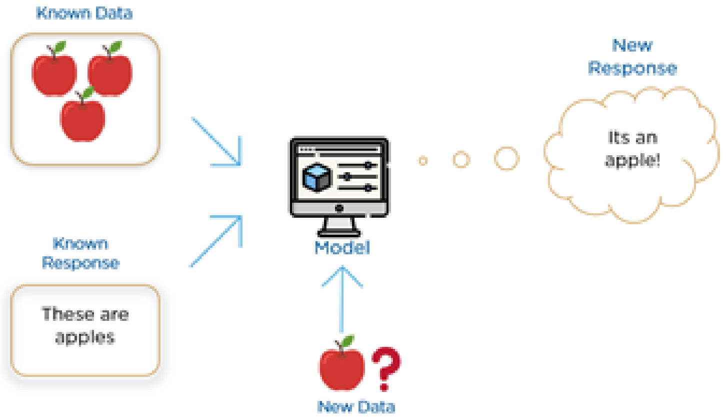
- **Inputs.** Labeled dataset.
- **Ouput.** A function that maps datapoints to the labels.



Types of learning

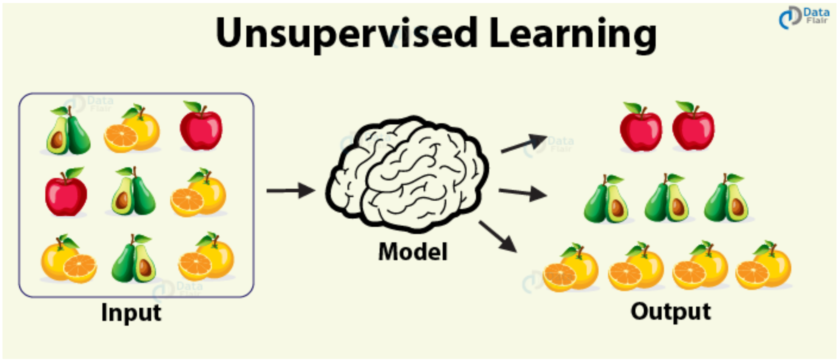
Supervised Learning

- **Inputs.** Labeled dataset.
- **Ouput.** A function that maps datapoints to the labels.



Unsupervised Learning

- **Inputs.** Labeled (or) unlabeled dataset.
- **Ouput.** A function that maps datapoints to clusters that capture *patterns* in the data.



Types of learning

Parametric Learning

- Hypothesis function takes a parametric form.
- Number of parameters are **not** proportional to the number of datapoints.

Examples: Linear regression, SVM, Logistic Regression, etc.

Types of learning

Parametric Learning

- Hypothesis function takes a parametric form.
- Number of parameters are not proportional to the number of datapoints.

Examples: Linear regression, SVM, Logistic Regression, etc.

Non-parametric Learning

- Hypothesis function doesn't necessarily have a parametric form.
- Number of parameters are proportional to the number of datapoints.

Examples: Kernel density estimation, k-NN clustering, etc.

Linear Regression

Equation of Line

❯ Setup
Derivation
Caveats

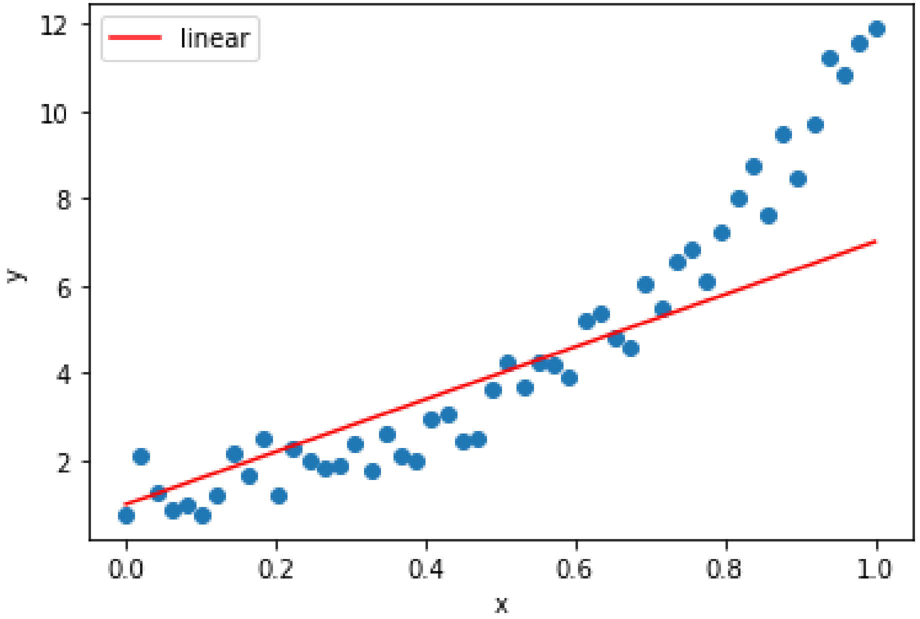
Equation of a line

$$y = a + bx$$

where a is the intercept and b is the slope.

In higher dimensions where $x \in \mathrm{R}^d$,

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_dx_d$$



Linear Hypothesis

Given a labeled dataset $D = \{(x_i, y_i)\}$ of n points where $x_i \in R^d$, $y_i \in R$.
Find a linear approximation $b \in R^{d+1}$ such that

$$\hat{y}_i = b_0 + \sum_j b_i x_{ij}$$

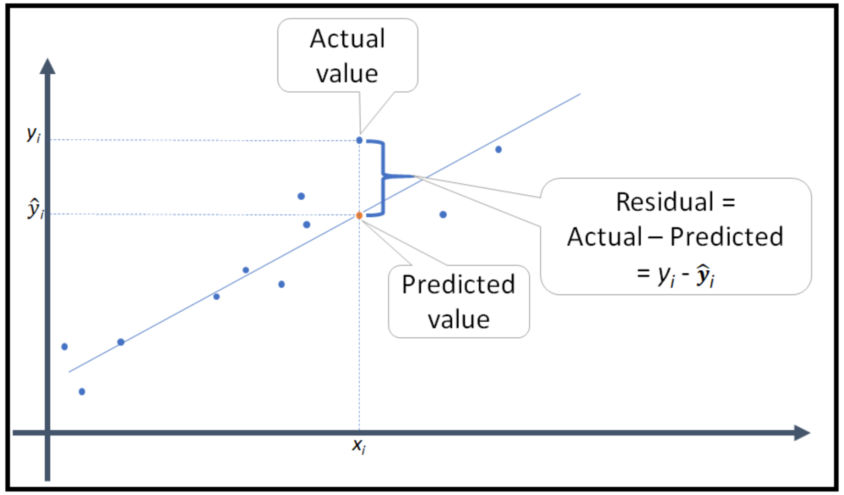
In matrix form,

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ . \\ . \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & . & . & x_{1m} \\ 1 & x_{21} & x_{22} & . & . & x_{2m} \\ . & & & & & \\ . & & & & & \\ 1 & x_{n1} & x_{n2} & . & . & x_{nm} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ . \\ . \\ b_m \end{bmatrix} = X\boldsymbol{b}$$

Goodness of fit

Mean squared error is used as the measure of goodness of fit.

$$
\begin{aligned}
\ell_D(b) &= \frac{1}{2n}\sum_i (y_i - \hat{y}_i)^2 \\
&= \frac{1}{2n}(Y - \hat{Y})^T(Y - \hat{Y}) \\
&= \frac{1}{2n}(Y^t - b^T X^T)(Y - Xb) \\
&= \frac{1}{2n}(Y^t Y - 2Y^T Xb + b^T X^T Xb)
\end{aligned}
$$



Ordinary Least Square

Setup
❯ Derivation
Caveats

- We need to find

$$\hat{b} = \arg\min_{b \in \mathrm{R}^{d+1}} \ell_D(b)$$

- Let's apply a recipe from calculus.

$$\begin{aligned}\frac{d\ell_D(b)}{db} &= \frac{1}{2n}\frac{d}{db}(Y^tY - 2Y^TXb + b^TX^TXb) \\ &\propto (X^TXb - X^TY)\end{aligned}$$

Equating the gradient to zero gives us:

$$\hat{b} = (X^tX)^{-1}(X^TY)$$

How to ensure invertibility?

Invertible Matrix

↓

Non-Singular Matrix

↓

Full-rank Matrix

↓

Linearly Independent Rows

How to ensure invertibility?

Invertible Matrix

↓

Non-Singular Matrix

↓

Full-rank Matrix

↓

Linearly Independent Rows

- The number of datapoints must be more than the number features.
- Duplicate datapoints must be removed.
- Features should not have perfect correlations.

Data preprocessing clearly plays a vital role!

Statistical Learning

Uncertainty in Data

Let R denotes the set of reasons for a set of observations O.

$$Pr[R = r \mid O = o] = \frac{Pr[O = o \mid R = r]Pr[R = r]}{\sum_{r'} Pr[O = o \mid R = r']Pr[R = r']}$$

Likelihood

$Pr[O = o \mid R = r]$ (easier to compute based on the hypothesis!)

Prior

$Pr[R = r]$ (assumed based on the background knowledge!)

Posterior

$Pr[R = r \mid O = o]$ (something we are interested in!)

Bayesian view of ML

We assume hypothesis as well as the data as a random variable.

$$Pr[H = h \mid D] = \frac{Pr[D \mid h]Pr[h]}{\sum_{h'} Pr[D \mid h']Pr[h']} \propto Pr[D \mid h]Pr[h]$$

Learning

$$\hat{\theta} = \arg\max_{\theta \in \Theta} Pr[\Theta = \theta \mid D]$$

Duality

Minimisation of loss translates to the maximisation of the data generation probability in the Bayesian framework.

Bayesian view of ML

Maximum Likelihood Estimation (ML)

$$\hat{\theta} = \underset{\theta \in \Theta}{\arg\max}\, Pr[D \mid \Theta = \theta]$$

Maximum Aposteriori Estimation (MAP)

$$\hat{\theta} = \underset{\theta \in \Theta}{\arg\max}\, Pr[D \mid \Theta = \theta] Pr[\Theta = \theta]$$

.

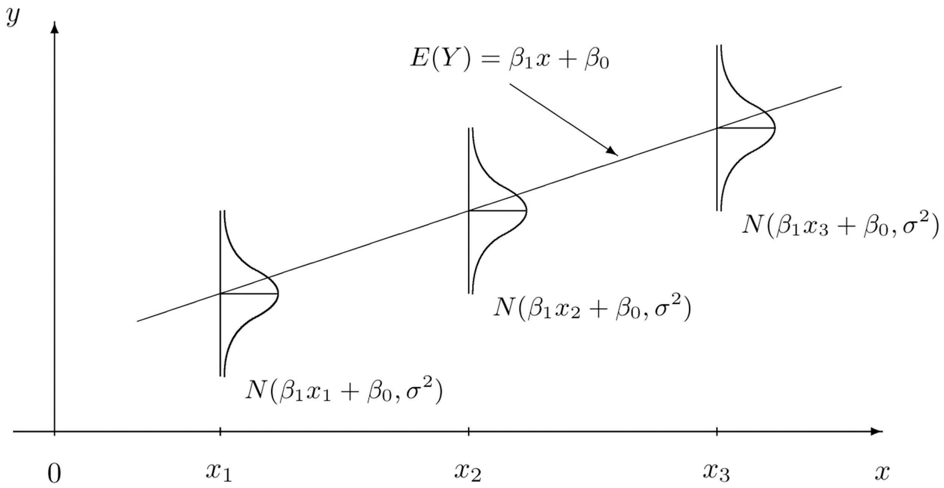
Prior probability

- Prior belief or probability represents what we believe to be the likelihood of an event occurring based on our knowledge, experience, or subjective judgment before observing any relevant data.
- Prior probability is typically assumed to follow a certain probability distribution based on the beliefs.
- In the absence of the prior belief, we assume it to be a uniform distribution. In such a case MAP estimation reduces to ML estimation.

Bayesian Regression

Data is noisy!

$$Y = Xb + \epsilon$$
$$\epsilon \sim \mathrm{N}(0, \sigma^2 I)$$



Assumptions

- Errors ϵ_is are independent conditional on data.
- Each error ϵ_i has a fixed variance σ^2.

Bayesian Regression

Likelihood function

$$
\begin{aligned}
L_D(b) &= Pr[D \mid b] \\
&= Pr[y_1 \mid x_1, b] \cdot Pr[y_2 \mid x_2, b] \cdot Pr[y_3 \mid x_3, b] \ldots Pr[y_n \mid x_n, b] \\
&= \prod_i Pr[y_i \mid x_i, b]
\end{aligned}
$$

Due to noisy data assumption,

$$(Y - Xb) \sim \mathrm{N}(0, \sigma^2)$$

$$Pr[y_i \mid x_i, b] \propto exp(\frac{-1}{2\sigma^2}(y_i - b^T x_i)^2)$$

Bayesian Regression

Log-likelihood

Issue. The likelihood is a probability, a number that lie between 0 to 1. In order to compute the likelihood over the dataset, the *i.i.d.* assumption demands the product of the likelihoods of individual datapoints. This gives rise to very small numbers.

Solution. For any monotonically increasing function f, maximising $g(x)$ is same as maximising $f(g(x))$. *Logarithm* is a monotonically increasing function that converts products to addition (which solves the earlier issue).

$$\log(ab) = \log a + \log b$$

$$\ell_D(b) = \log L_D(b) = \sum_i \log Pr[y_i \mid x_i, b]$$

$$\ell_D(b) \propto \frac{-1}{2\sigma^2} \sum_i (y_i - b^T x_i)^2 \quad ...(\log e^x = x)$$

Bayesian Regression

Reduction to OLS

$$
\begin{aligned}
\ell_D(b) &\propto \frac{-1}{2\sigma^2} \sum_i (y_i - b^T x_i)^2 \\
&\propto \sum_i (y_i - b^T x_i)^2 \\
&= -(Y - Xb)^T (Y - Xb)
\end{aligned}
$$

Maximum likelihood estimate is,

$$\hat{b} = \underset{b \in \mathrm{R}^{d+1}}{\arg\max}\ \ell_D(b)$$

which is same as OLS, $\hat{b} = \arg\min_{b \in \mathrm{R}^{d+1}} -\ell_D(b)$.

Bayesian Regression

Linearity

$$y = b_0 + b_1 x_1 + \ldots + b_d x_d$$

Statistical Independence

All error terms ϵ_i are conditionally independent of each other given the data.

Normality

All error terms ϵ_i follow normal (gaussian) distribution.

Homoscadasticity

All error terms ϵ_is follow the distribution with a constant variance σ^2.

Participation in a campaign

Success of a campaign

An offer campaign is run to offer discount to users if they become the member of the platform. Let's assume that

- the offer was sent to n users.
- $n_1 \leq n$ users accepted the offer.

How can we design a statistical model for such a campaign?

Let's assume each $x_i \sim Bernoulli(\theta)$

$$L_D(\theta) = \theta^{n_1}(1 - \theta)^{n - n_1}$$

$$\hat{\theta} = \arg\max \ell_D(\theta) = \frac{n_1}{n}$$

Need of a prior

What if the $n_1 = 0$ *in the earlier example?*

Black swan paradox

If you have not spotted a black swan, would you conclude that they do not exist?

Maximum likelihood estimation suffers from **sampling bias** if the rare events exists. MAP estimation alleviates this problem by employing a prior distribution.

Need of a prior

Conjugate Prior

Conjugate prior is that probability distribution which multiplied with likelihood yields the same posterior distribution.

- $(Gaussian \times Gaussian) \sim Gaussian.$
- $(Poisson \times Exponential) \sim Exponential.$
- $(Binomial \times Beta) \sim Beta.$

Beta-Bernoulli Distribution

Beta-Bernoulli Distribution

Prior

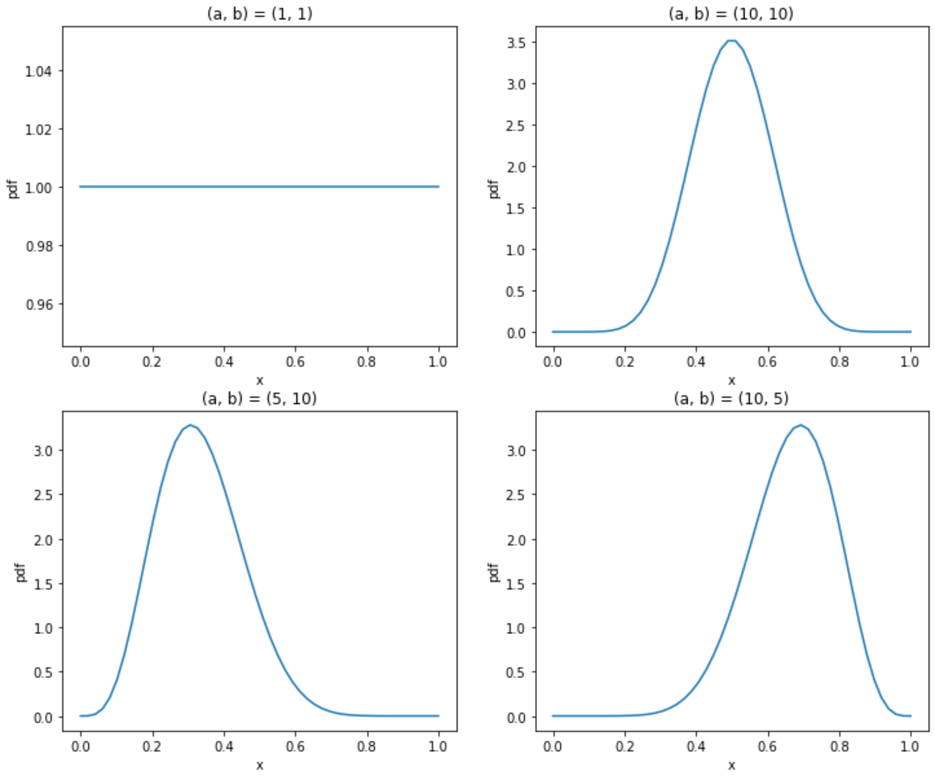
$$Pr[\theta \mid a, b] \propto \theta^{a-1}(1-\theta)^{b-1}$$

Likelihood

$$Pr[D \mid \theta] \propto \theta^{n_1}(1-\theta)^{n-n_1}$$

Posterior

$$Pr[\theta \mid D, a, b] \propto \theta^{a+n_1-1}(1-\theta)^{b+n-n_1-1}$$



Example(cntd)

MAP Estimate

$$\theta_{MAP} = \frac{n_1 + a - 1}{n + a + b - 2}$$

Prior knowledge

Similar campaigns were also run in the past. On average, such campaigns offer 20% conversion rate.

We can put $a = 20$ and $b = 80$ to quantify the prior knowledge.

$$\theta_{MAP} = \frac{n_1 + 19}{n + 98}$$

Summary

Recipe of ML

› Recipe
Bias-Variance

Dataset

$$D = \{(x_i, y_i) \mid x_i \in \mathrm{R}^d, y_i \in R\}$$

Recipe of ML

❯ Recipe
Bias-Variance

Dataset

$$D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^d, y_i \in R\}$$

Classical Recipe

Hypothesis

$$y_i = b^T x_i$$

Minimise MSE

$$\ell_D(b) = \frac{1}{2n} \sum_i (y_i - b^t x_i)^2$$

Prediction

$$\hat{y}_i = \hat{b}^T x_i$$

Bayesian Recipe

Hypothesis

$$y_i = b^T x_i + \epsilon_i, \ \ \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Maximise Likelihood

$$\ell_D(b) = \sum_i Pr[y_i \mid x_i, b]$$

Prediction

$$\hat{y}_i = \hat{b}^T x_i + \epsilon$$

Bias Variance Tradeoff

Recipe
› Bias-Variance

- Our true intention is to find θ^* that truly captures the patterns in the observed data.
- What we learn, in reality, is the parameter $\hat{\theta}$ that captures patterns in the sample.

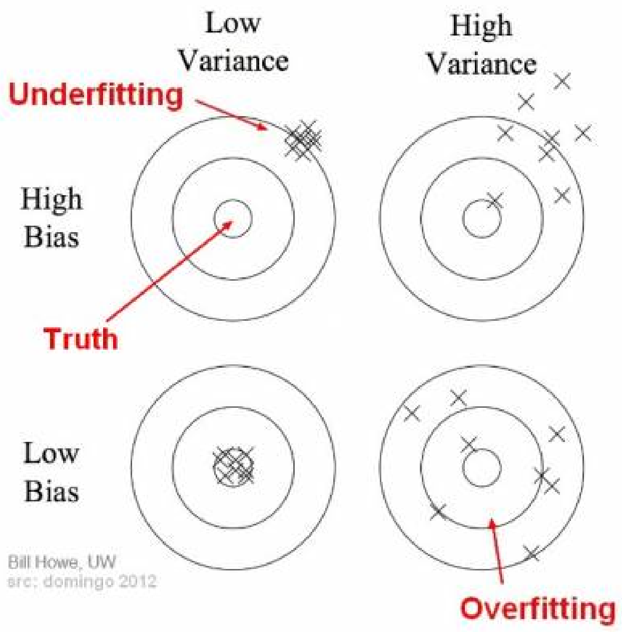
Thus, θ acts as a random variable due to sampling form the population.

Bias

- Defined as $\mathrm{E}[\theta - \theta^*]$
- Quantifies the *goodness of fit*.

Variance

- Defined as $Var[\theta]$
- Quantifies the gap between training and testing error.



Thank you!

Feel free to reach out to me at

dcsashi (at) nus (dot) edu (dot) sg