# ML Supervised Learning 1

by ambedkar@IISc

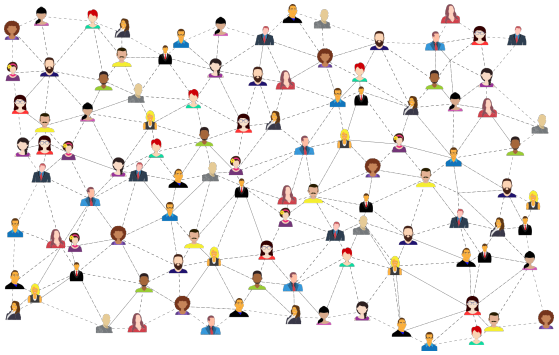
- Let there be AI
- On Machine Learning
- Data and Models
- Distance Based Classifiers

# Let there be AI

# Let there be AI

So that computers can...

- look at our medical reports and provide diagnosis,
- tell us which stocks to buy so that we get higher returns,
- scan social networks and predict terrorist attacks, and
- help police to identify a suspect from CCTV cameras.



# Let there be AI

- The first step is to collect the data and store it in computers.
- Now the computer requires some instructions.
- Can anyone write a program that reads the past history of stock market data and predict the stock market?

# Let there be AI

- The programmer has to be an expert or consult an expert in the stock market who can understand the trends.

- Then, the programmer can write hundreds and thousands of `if-else` statements to emulate an experienced stock broker

# Let there be AI

**What if**

- we can just write a special program that scans through the data, and then itself figures out all `if-else` statements that lead to decisions that can give us profits.

**(In more technical terms)**
**What if**

- we can just build a MACHINE implementable mathematical model that is capable of LEARNING from the data
  and give us all possible optimal decisions.

# Key Words: MACHINE

- Here MACHINE means Turing Machine.

# Key Words: Mathematical Model

- An abstraction that is close to the phenomina or the downstream task that phenomina leads to.
    - If we try to model the undelying phenomina, that leads to generative models, or
    - if we just try to model task that is called discriminative models (in the case of supervised learning)

# Key Words: LEARNING

- While the mathematical model aimed describe the phenomina, not everything about the model cannot be determined upfront.
- Some specifications of the model are kept as variables (called parameters)
- Aim is to "estimate" or "learn" these parameters from data that is observed
- This is central to the field of machine learning

# Key Words: Optimal Decisions

- Optimal means user defined goals or how best one can emulate the underlying phenomina at hand
- Is it always easy to evaluate our models and learning algorithms?
  - No!
  - In some scinarios we do not have "ground truth" to check how well we are performing
  - Here too, generative models plays an important role (like Stochastic Blockmodels)

# Machine Learning...

is a subbranch of computer science that involves

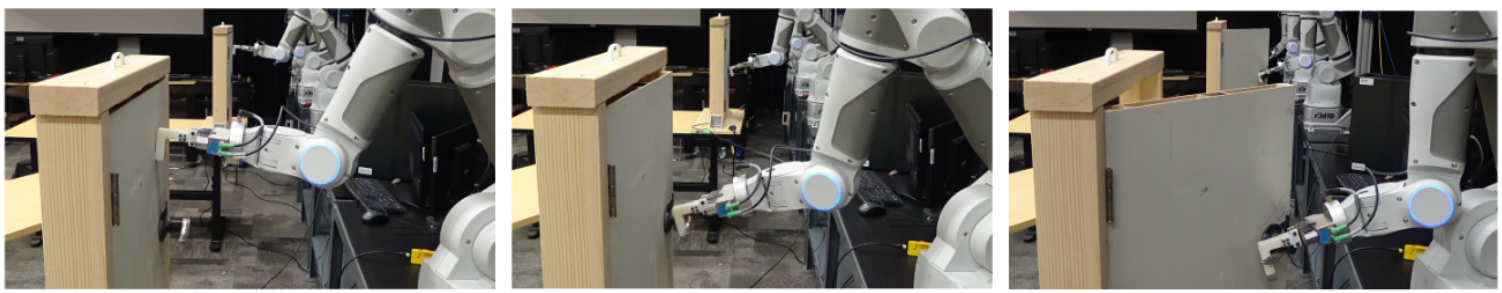
- probabilty
- statistics
- optimization
- matrices
- algorithms
- programming

# Outlook

# The nature of Machine Learning

- Almost all problems in machine learning cannot be solved exactly.
    - For a small network with 100 nodes, the number of different partitions exceed the number of atoms in the universe
- Hence we provide approximate algorithms (identifying the clusters by computing eigenvectors is indeed an approximate solution)
- We rely on mathematical analysis to provide some guarantees that our methods give reasonable solutions.

# Robotic Movements



*Robots opening a door.*[1]

[1] GuHolLilLev17.

# Drone control

*Composite image of a flight of the drone.*[2]

[2] song2021autonomous.

# On Machine Learning

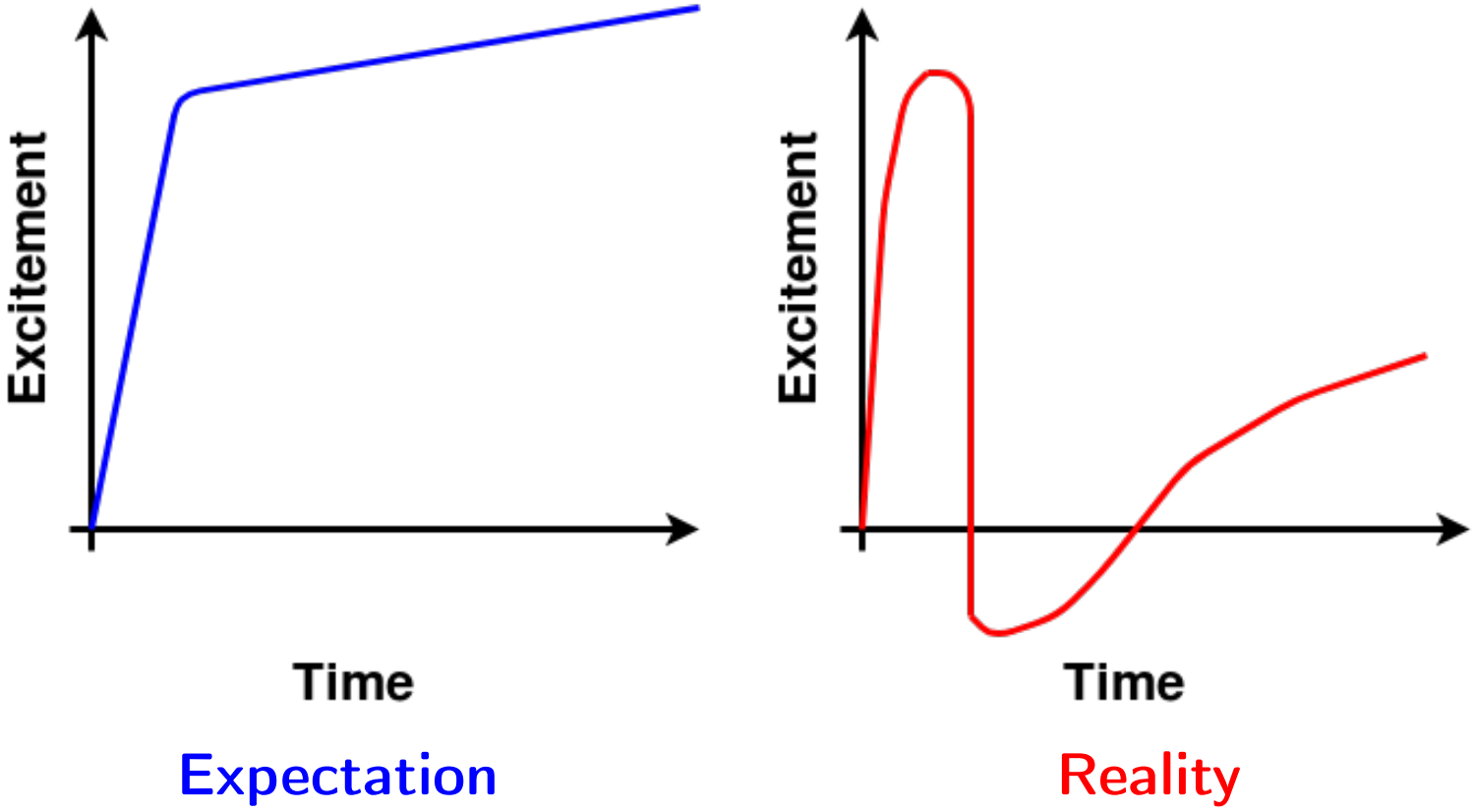
# On ML

- Why getting into or mastering ML need not be very easy?
  - If one is starting fresh, there is an ocean out there
  - If one already knows some concepts, one can be confused about what to learn next
- Machine learning can be viewed as list of models or methods
- Or...solutions to some practical problems based on few foundational principles, that involve probabilistic and statistical concepts.

# The Best Strategy

- One should constantly....
  - strengthen the foundations
  - try to understand relations between different paradigms and methods
  - most importantly, always experiment...

# Machine Learning 101

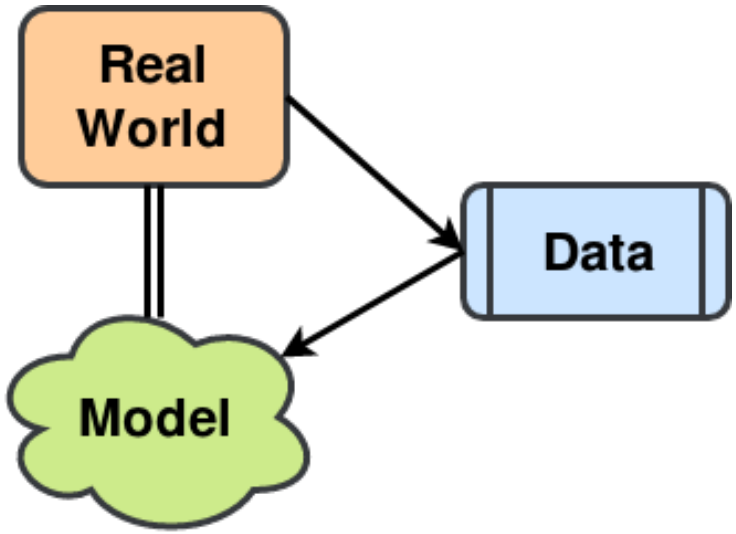



First Machine Learning Class - Expectation vs Reality

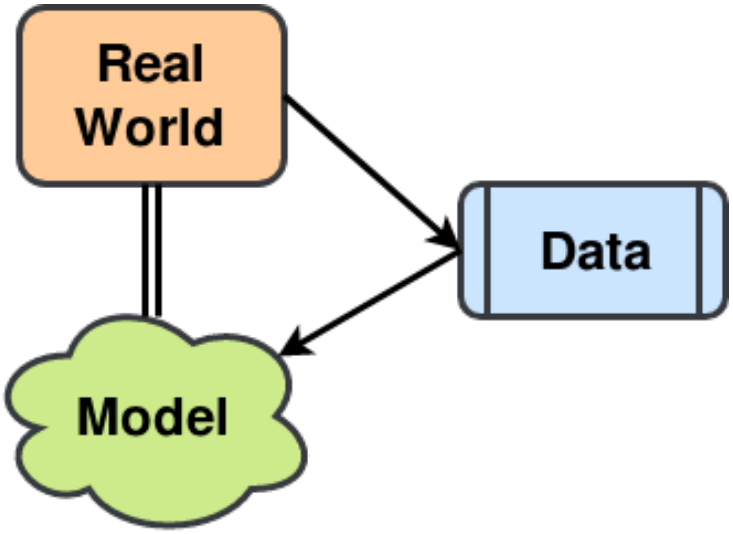
# Data and Models

# Basics: Data and Models

- Various phenomena in real world offers us data
- A model is a representation of these phenomina
- Data obtained from real world is used for finding parameters of the model
- The model is then used for making predictions or gaining insights about the real world



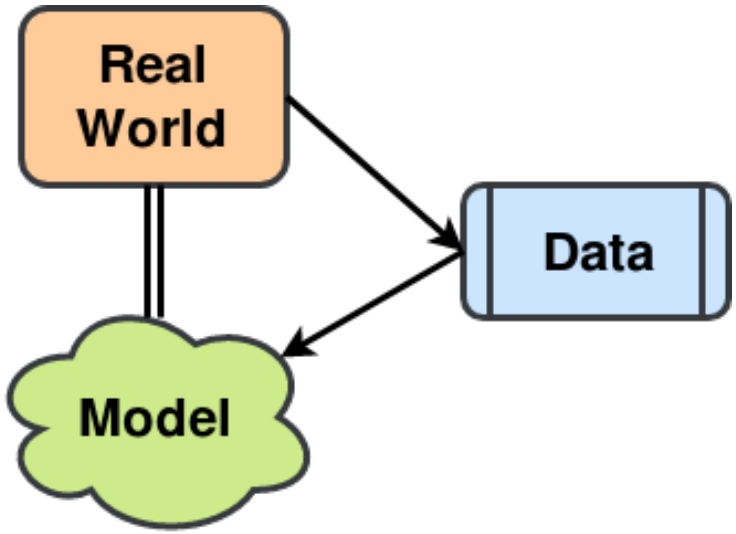
# Basics: Data and Models - Example 1




- **Real World**: Sentences used by people in conversations about machine learning
- **Data**: Sentences uttered during this talk
- **Model**: A probability distribution over all possible sentences of length $\leq 50$ with Markov assumption

# Basics: Data and Models - Example 2

- **Real World**: Students and friendships among them
- **Data**: An observed friendship network involving students from grade one and grade two in a school
- **Model**: Assume people in same grade become friends with probability $p$ and students across grades become friends with probability $q$



# Representation of Data

**Most often we represent data in the form of a vector in real space**

- Feature vector corresponding to a speech signal
- Feature vector corresponding to a region to predict housing prices
- Feature vector corresponding to pixels of an image
- Feature vector corresponding to a word or a sentence in natural language text (Is this possible?)

# Different types of data

- **Spatially Regular Data**
  - Images
- **Sequential Data**
  - Sentences
  - Time series data
- **Relational Data**
  - Tabular data collected during surveys
  - Graph structured data
- **Multimodal Data**
  - Videos
  - Medical records

# Basics: Models

- A model is an abstraction of real world
- Model the aspects of real world that are to be studied
- A very complicated model is usually of no use
  - Should be flexible enough to represent phenomenon of interest
  - Should be tractable

# Basics: Models - Examples

- Linear Gaussian model - Regression
- Naïve Bayes model - Classification
- Gaussian mixture model - Clustering
- Hidden Markov model - Discrete valued time series
- Linear dynamical system - Continuous valued time series
- Restricted Boltzmann machines - Data with latent variables
- Stochastic Blockmodels - Networks

# Data Representation

# On Data

- Remember that aim of machine learning is to model a phenonmenon or solve problems underlying
- Problems can be of three types
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Data provides some partial information about the phenomenon

# On Data

Availability of data depends on the underlying problem.

- Supervised Learning
  - Data is available in this case as set of input-output pairs
  - Data is of the form $(x_1, y_1), (x_2, y_2) \ldots (x_N, y_N)$
  - Here each $x_n$ can be an image or document and $y_n$ is a label or groundtruth
- Unsupervised Learning
  - Data is available without any labels or groundtruth
  - Data is of the form $x_1, x_2, \ldots, x_N$
- Reinforcement Learning
  - Here data is not available upfront
  - The data that is available will be based on agent's "strategy".

Every data point is a vector

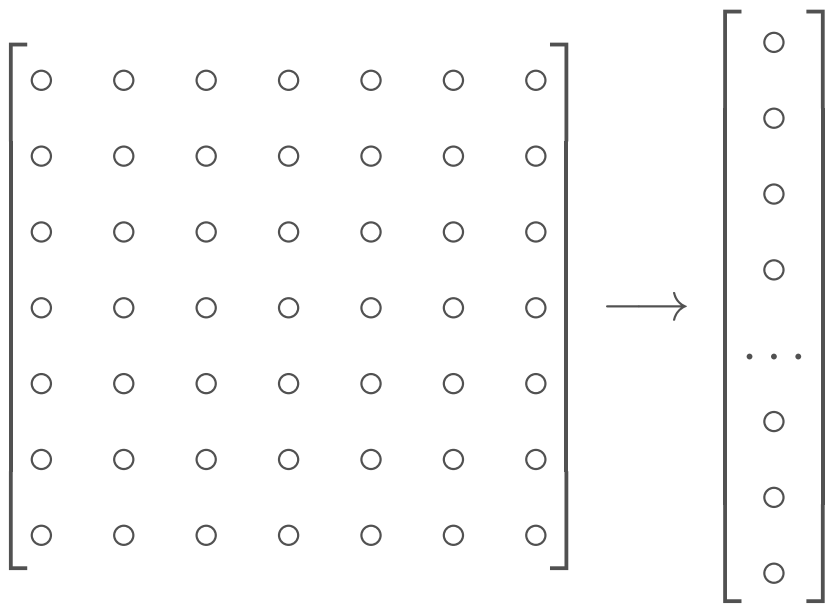
occationally matrices or tensors

# Data Representation

- Each input $x_n$ is usually a $D$ dimensional feature vector that is $x_n$ can be written as

$$x_n = (x_{n1}, x_{n2}, \ldots, x_{nD})$$

  - Support $x_n$ is a $7 \times 7$ image. It can be represented using a vector of size $49$ of pixel intensities



# Data Representation

- Note that in certain applications input $x_n$ need not be a fixed length of vector. For example protein sequences, etc.
- Output $y_n$ can be
  - real values (eg. regression)
  - categorical (eg. classification)
  - structured object (eg. structured output learning)
- The learning task becomes tougher and tougher when the dimensionality of data is very high.

# Data: In what form?

- Data is always "raw".
- Most machine learning models works only when the "nice" and "appropriate" and "useful" features are fed to them.
- So feature can be learned or extracted
  - Learned: The model/algorithms automatically learn the useful features
  - Extracted: Hand-crafted features defined by a domain expert.

# On vector Representation

- Each feature vector $x_n$ is a point in the $D$ dimensional vector space $\mathbb{R}^D$
- By putting data in a vector space we can incorporate all tools that is provided by Linear Algebra in our problem solving
- More importantly matrix computations play an important role in machine learning

# Vector Representation

- Vector space provides us with distance and similarity measures

- Euclidean distance between two data points $x_n, x_m \in \mathbb{R}^D$

$$d(x_n, x_m) = \|x_n - x_m\|_2 = \sqrt{(x_n - x_m)^T (x_n - x_m)}$$

$$= \sqrt{\sum_{d=1}^{D} (x_{n_d} - x_{m_d})^2}$$

# Vector Representation

- Vector space provides us with distance and similarity measures
- Inner product (or cosine similarity) between $x_n, x_m \in \mathbb{R}^D$

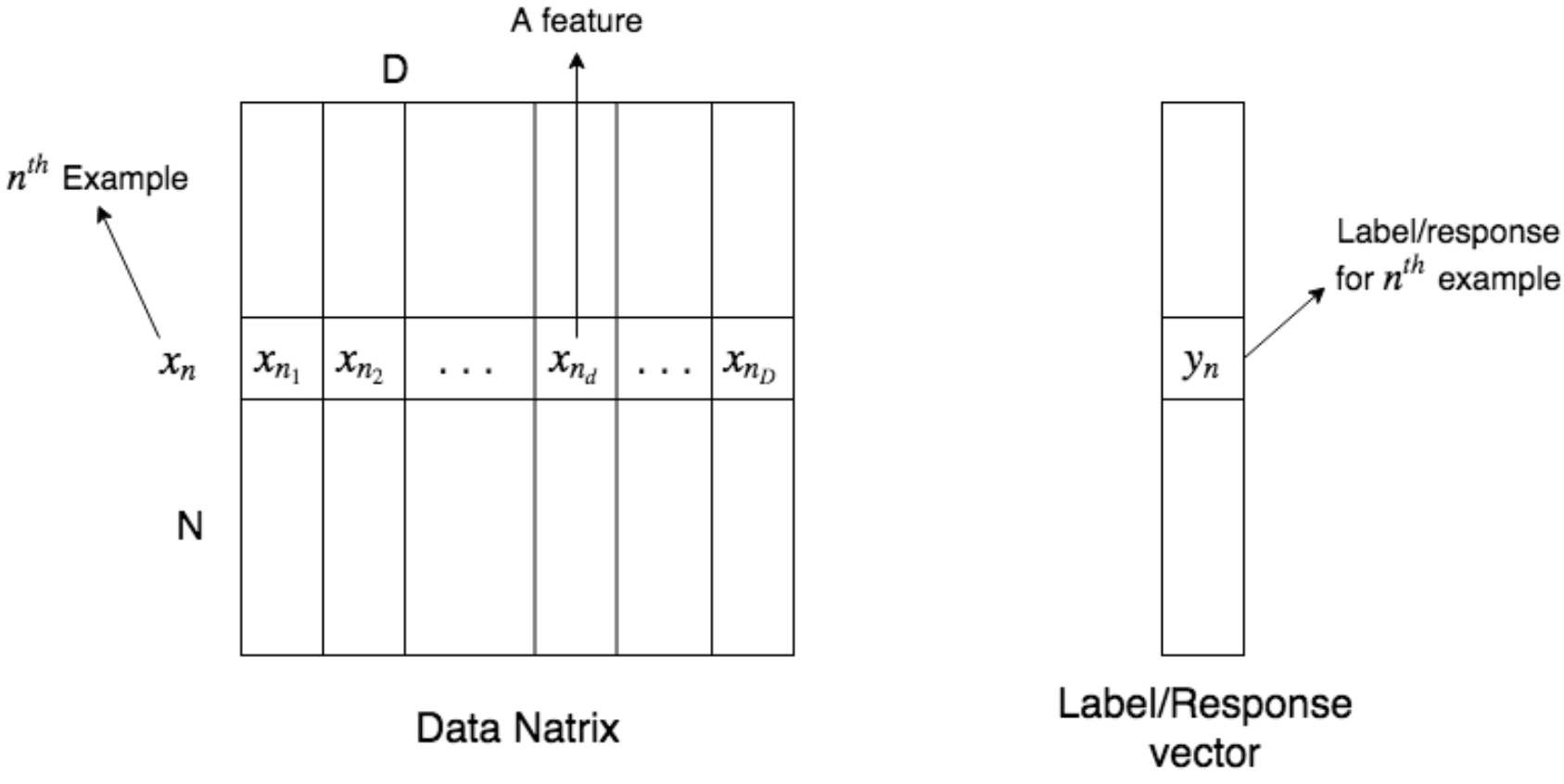
$$\langle x_n, x_m \rangle = x_n^T x_m = \sum_{d=1}^{D} x_{n_d} x_{m_d}$$

- $\ell_1$ distance between $x_n, x_m \in \mathbb{R}^D$

$$\ell_1(x_n, x_m) = \|x_n - x_m\|_1 = \sum_{d=1}^{D} ||x_{n_d} - x_{m_d}||$$

# Data Matrix

- $x = \{x_1, \ldots, x_n\}$ denotes data in form of $N \times D$ feature matrix

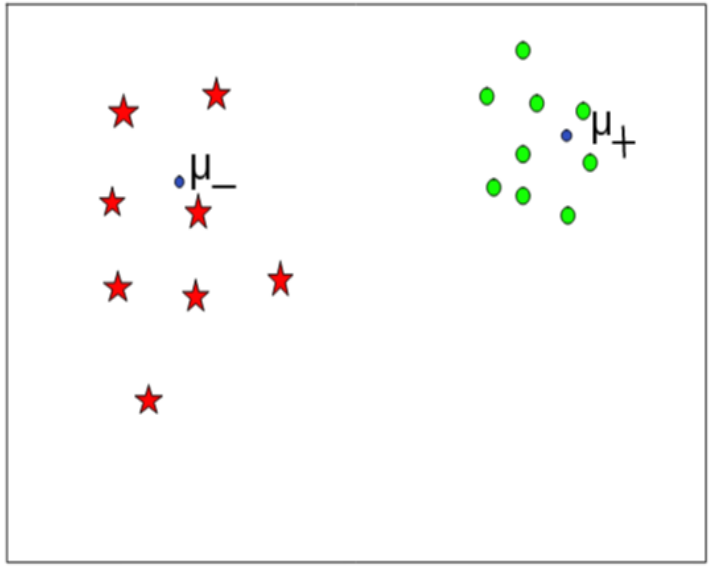



- $y = \{y_1, \ldots, y_N\}$ denotes labels/responses in the form of an $N \times 1$ label/response vector.

# Distance based Classifiers

# Setting

- Given $N$ labelled training examples $\{(x_n, y_n)\}_{n=1}^{N}$ from two classes (+ve and -ve)
  - Assume positive is green and negative is Red.
  - Assume we have $N_+$ examples from +ve class and $N_-$ examples from negative class.
- **Aim:** Learn a model to predict label $y$ for a new test sample.



# A Simple Decision Rule based on Means

**Rule:** Assign test sample to classes with closer mean.

- The mean of each class is given by

$$\mu_- = \frac{1}{N_-} \sum_{y_n=-1} x_n$$

$$\mu_+ = \frac{1}{N_+} \sum_{y_n=+1} x_n$$

- Can we just store the two means and throw away data.

# A Simple Decision Rule based on Means (contd. . . )

- Distances from each mean are given by

$$\|\mu_- - x\|^2 = \|\mu_-\|^2 + \|x\|^2 - 2\langle \mu_-, x \rangle$$

$$\|\mu_+ - x\|^2 = \|\mu_+\|^2 + \|x\|^2 - 2\langle \mu_+, x \rangle$$

- Here
  - $\|a - b\|^2$ denotes squared Euclidean distance between $a$ and $b$.
  - $\langle a, b \rangle$ denotes inner product of two vectors $a$ and $b$.
  - $\|a\|^2 = \langle a, a \rangle$ denotes squared $l_2$ norm of $a$.

# The Decision Rule

- Denote the decision rule by $f : \chi \longrightarrow \{+1, -1\}$

$$\begin{aligned} f(x) &= \|\mu_- - x\|^2 - \|\mu_+ - x\|^2 \\ &= 2\langle \mu_+ - \mu_-, x \rangle + \|\mu_-\|^2 - \|\mu_+\|^2 \end{aligned}$$

- Decision Rule: if $f(x) > 0$ then x in +1
  otherwise x in -1

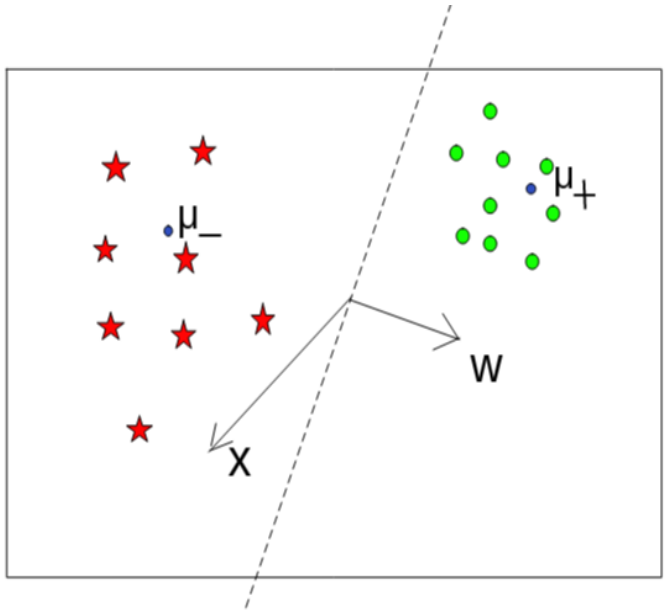
  i.e. y = $\text{sign}[f(x)]$

**What is the significant of introducing function notation?**

In machine learning mostly we will be learning functions or devising approximate methods to estimate functions. Neural Networks are nothing but function approximators.

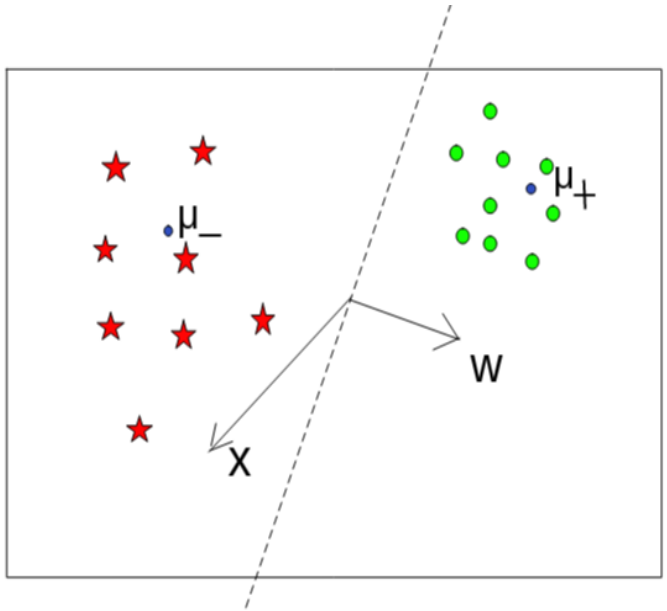
# The Decision Rule

- $f(x)$ denotes a hyperplane based classification rule, where $w = \mu_+ - \mu_-$ represents the direction rule to the hyperplane.




- This specific form of decision rule appears in many supervised algorithms.
- Inner product can be replaced by more general similarity measures.

# Decision Rule Based on Means: Some Comments




- It can be implemented easily.
- Would require plenty of traning data for each class to estimate mean realiably
- If we have class imabalanced data, this will not work

# Decision Rule Based on Means: Some Comments (contd ..)

- It can only learn linear decision boundaries.
  - We need to replace Euclidean distance by nonlinear distance function. Kernels?
- Data: We assume that there is an underlying probability distribution.
  - Mean can be thought of as one characteristic of a distribution.
  - How about modelling each class by a class conditional probability distribution.
  - Then compute distances from these distributions.
  - Linear Discriminant Analysis

# Basic Probability

# On Probability and Statistics

- Probability formalizes and models the concept of "random experiment"
  - Once we have underlying "probabilites" it provides frameworks and tools for futher computations and analysis
- Statistics provide tool to estimate probability distributions given the data.
  - That is the assumption is that given some data we assume that data is sampled from an unknown probability distributions

*Probability and statistical methods help us to build efficient algorithms to learn automatically hidden information from huge chunks of data.*

# Sample Space

**Example**

Consider a coin tossing experiment. We ask the following questions. What is the probability of getting heads ($P(H)$) and what is the probability of getting tails ($P(T)$). For this, without hesitation we say half and half (why?). What else can we ask? Does it make sense to ask what is the probability of getting either $H$ or $T$? Yes.

**Example**

Consider throwing a dice. Apart from asking questions like what is the probability of getting $1$ or $2$, we might be interested in asking questions like what is the probability of getting an even number. That is what is $P(2, 4, 6)$.

# Probability: Mathematical Defintion (cont. . . )

**Definition**

Let $(\Omega, \mathcal{F})$ be a sample space, where $\Omega$ is set of all possible outcomes and $\mathcal{F}$ is set of all events (this is a $\sigma$-algebra and for all practical purposed we can consider this as $2^{\Omega}$). A function $P : \mathcal{F} \to \mathbb{R}$ is said to be probability if it satisfied the following conditions:

1. $P(A) \geq 0$
2. $P(\Omega) = 1$
3. Let $\{A_n \subset \mathcal{F}\}$ a countable collection of disjoint sets, i.e., $A_i \bigcap A_j = \emptyset$, for all $i \neq j$, then

$$P(\bigcup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P(A_n).$$

(This is called countable additivity property)

# Conditionary Probability and Bayes Theorem

- Let $A$ and $B$ are two events. Then conditional probability is defined as

$$P(A|B) = P_B = \frac{P(A \bigcap B)}{P(B)} \quad \text{for all} A \in \mathcal{F}$$

- Note that above, using an event $B$ we have defined a new probability $P(.|B)$ on the sample space $(\Omega, \mathcal{F})$

# Law of probability and Bayes Formula

Let $E_1, \ldots, E_n$ are partition of $\Omega$. Then for any event $A$ we have

$$P(A) = \sum_{k=1}^{n} P(A|E_k)P(E_k)$$

**Bayes Formula**

For any event $A$ we have

$$P(E_k|A) = \frac{P(A|E_k)P(E_k)}{\sum_{i=1}^{n} P(A|E_i)P(E_i)}$$

# Independence

We say two events $A$ and $B$ are independent if

$$P(A \bigcap B) = P(A)P(B)$$

**Result:**

Suppose $A$ and $B$ are independent events then $P(A|B) = P(A)$ if $P(B) > 0$ and $P(B|A) = P(B)$ if $P(A) > 0$

# Random Variable

**Let us see why do we need this concept**

A coin is tossed till the first head appears. Then the set of outcomes

$$\Omega = \{(H), (T, H), (T, T, H), ...\}$$

Let the set of all events $\mathcal{F} = 2^{\Omega}$. Now, instead of treating $\Omega$ as set of sequence of symbols, we can write $\Omega$ as the number of flips required to get a head i.e. $\Omega = \mathbb{N}$ and $\mathcal{F} = 2^{\Omega}$. That is, we map

$$\begin{aligned} H &\longrightarrow 2 \\ TH &\longrightarrow 2 \\ TTH &\longrightarrow 3 \\ &\vdots \end{aligned}$$

# Random Variable (cont. . . )

Let the probability of getting heads is $0 < p < 1$. Now

$$
\begin{aligned}
&P(N = 1) = P(H) = p \\
&P(N = 2) = P(TH) = (1-p)p \\
&P(N = 3) = P(TTH) = (1-p)^2 p \\
&\vdots \\
&P(N = n) = P(\underbrace{TT \ldots T}_{\text{n-1}} H) = (1-p)^{n-1} p,\ n \geq 1
\end{aligned}
$$

We can note that

$$P(\bigcup_{n=1}^{\infty}(N = n)) = \sum_{n=1}^{\infty} P(N = n) = p \sum_{i=1}^{\infty}(1-p)^{n-1} = \frac{p}{1-(1-p)} = 1$$

Random variable is a function that assigns some number to each of the element in $\Omega$

# Expectation

- Suppose a random variable takes values $x_1, x_2, \ldots, x_N$
- And the corresponding probabilities are $p_1, p_2, \ldots, p_N$. That is Probability$(X = x_n) = p_n$

The expectation of $X$ is defined as

$$\mu = \mathsf{E}X = \sum_{n=1}^{N} x_k p_k$$

# Example

- Random variable is nothing but a function that maps outcome to a number
  - Consider a coin tossing experiment: Outcomes are `H` and `T`
  - Random variable $X$ can map `H` to $1$ and can map `T` to $0$
- Now let us assign probabilities
  - Suppose $P(X = 1) = \frac{1}{4}$ and $P(X = 0) = \frac{3}{4}$
  - That is probability mass function of $X$ is $(\frac{1}{4}, \frac{3}{4})$
- Let us calculate expectation of a random variable

$$\mathsf{E}_P X = \sum_{i=1}^{2} x_i p_i = 1\left(\frac{1}{4}\right) + 0\left(\frac{3}{4}\right)$$

# Variance

We call $\mathsf{E}(X-\mu)^2$ the variance of $X$ which is denoted by

$$\sigma^2 = Var(X) = \mathsf{E}(X-\mu)^2$$

where $\sigma$ is standard deviation.