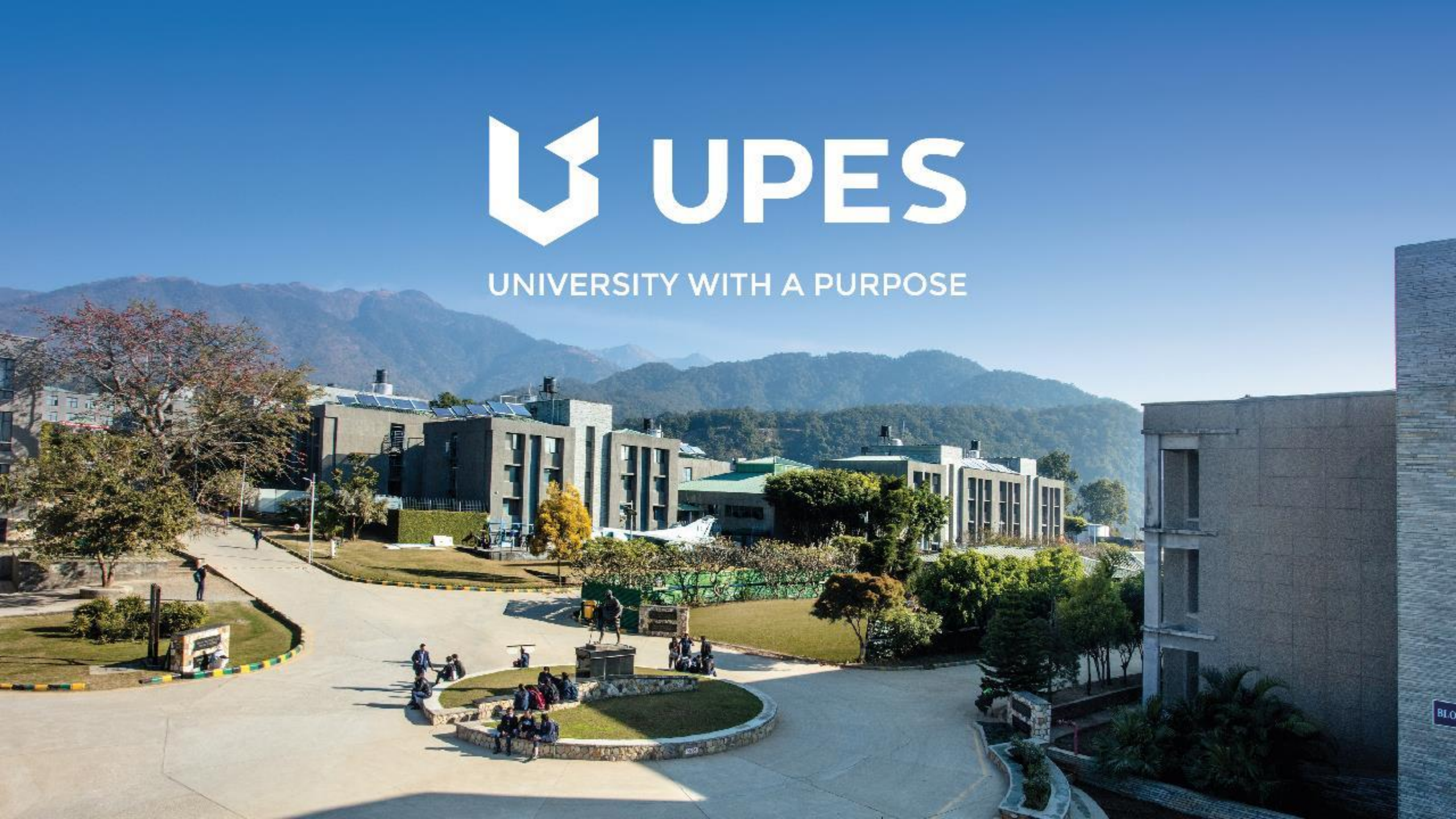
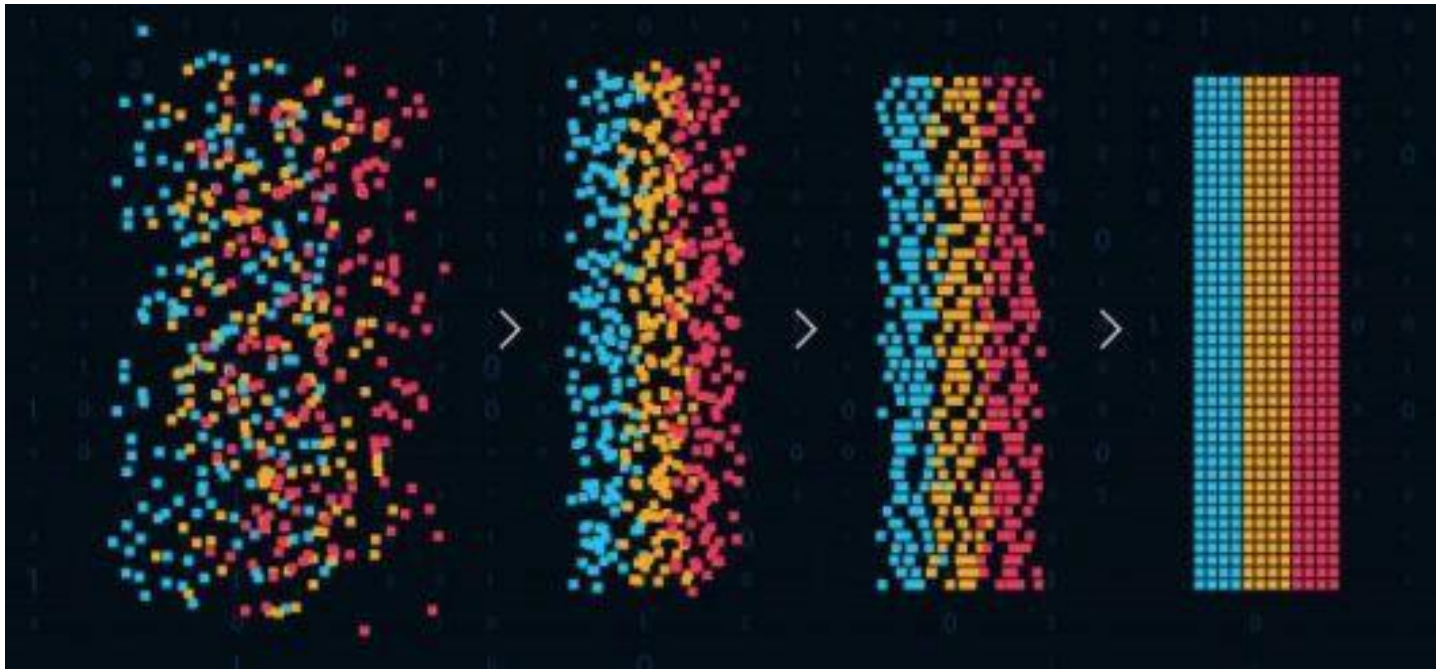

UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

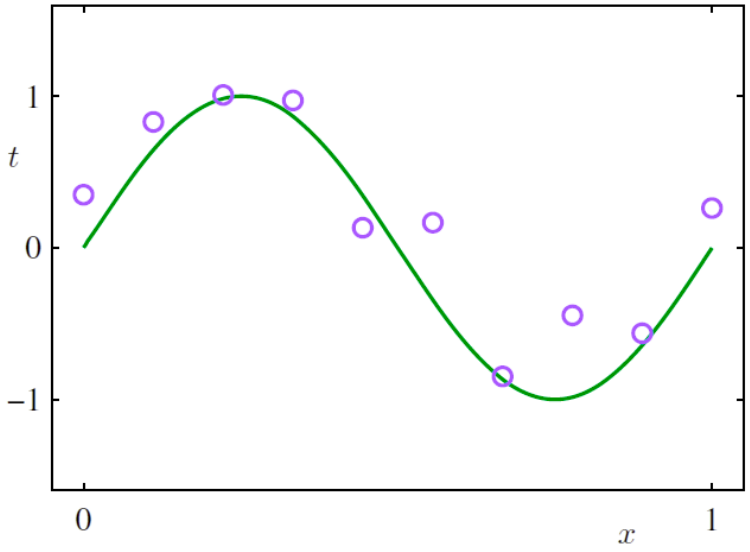
B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

Context: Regression

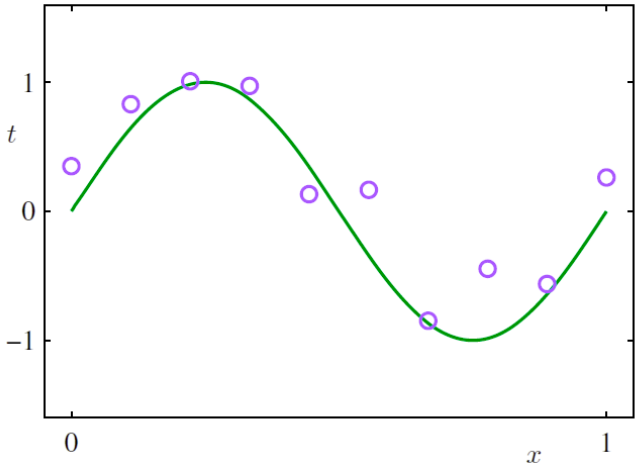
- Observe a real-valued variable (input) *(x)*
- Using this observation to predict the value of a real-valued target variable *(t)*
- Create some data using function Sin(2p*x)*
- Add random noise to target variable
- We have N observations

$$\mathbf{x} = (x_1, x_2, \ldots\ldots x_N)^T$$
$$\mathbf{t} = (t_1, t_2, \ldots\ldots t_N)^T$$



Let say N = 10, x varies from 0 to 10, t is computed using Sin(2p*x)* and then adding Gaussian noise.

Context: Regression



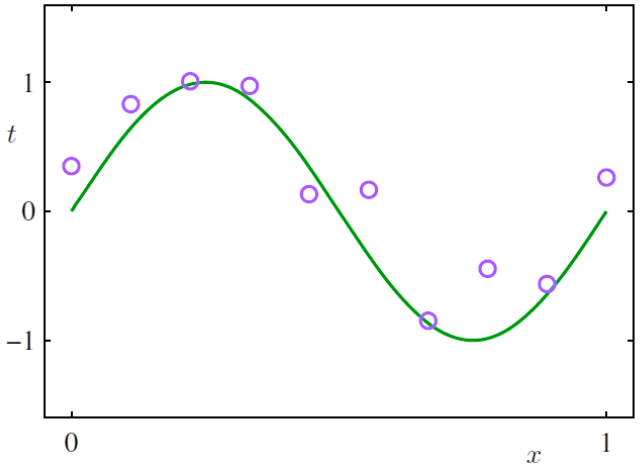
- **Goal**: Use training data to make better predictions of for some new *x*.
- Implicitly trying to discover the underlying function.
- **Problem**: Generalize from finite data, corrupted with noise (nature unknown).
- Solution approach: Curve fitting

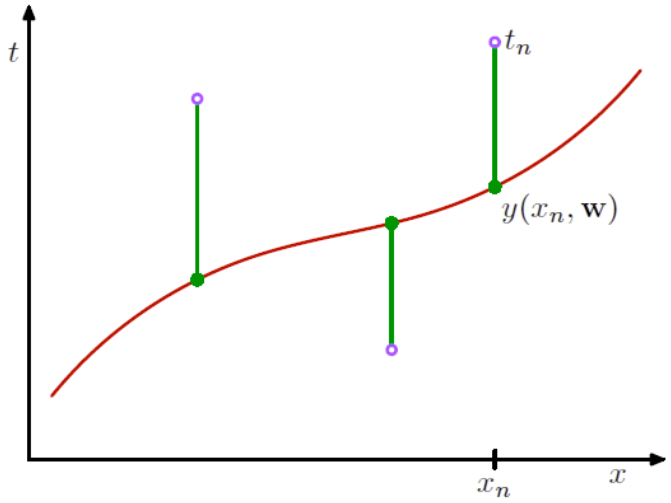
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Function is non-linear in *x* but linear in **w**. Therefore such functions are called **linear models.**

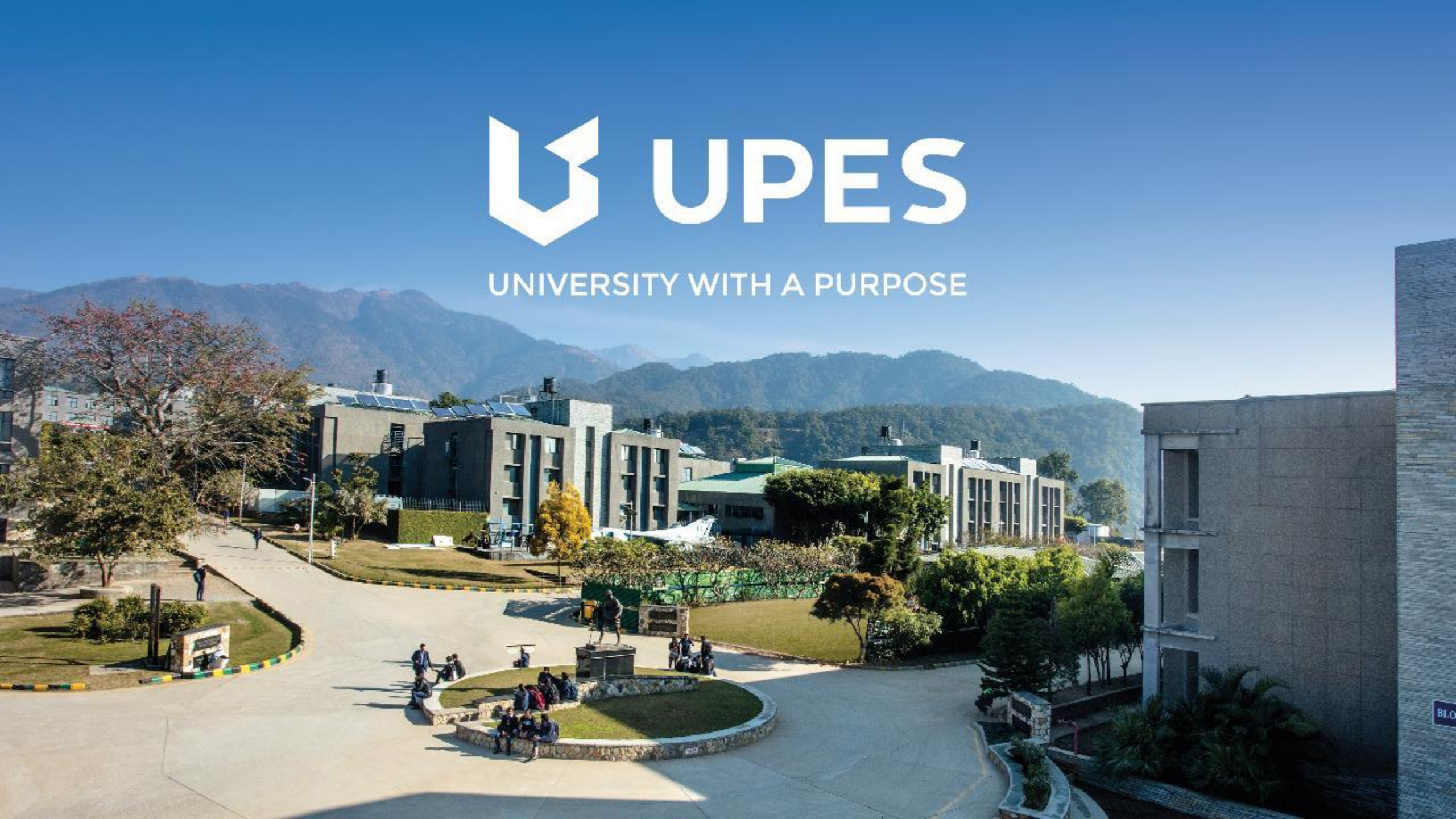
Context: Regression

- Error function $$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$
- Change **w** so as to minimize E(**w**).
- Change M to change model.

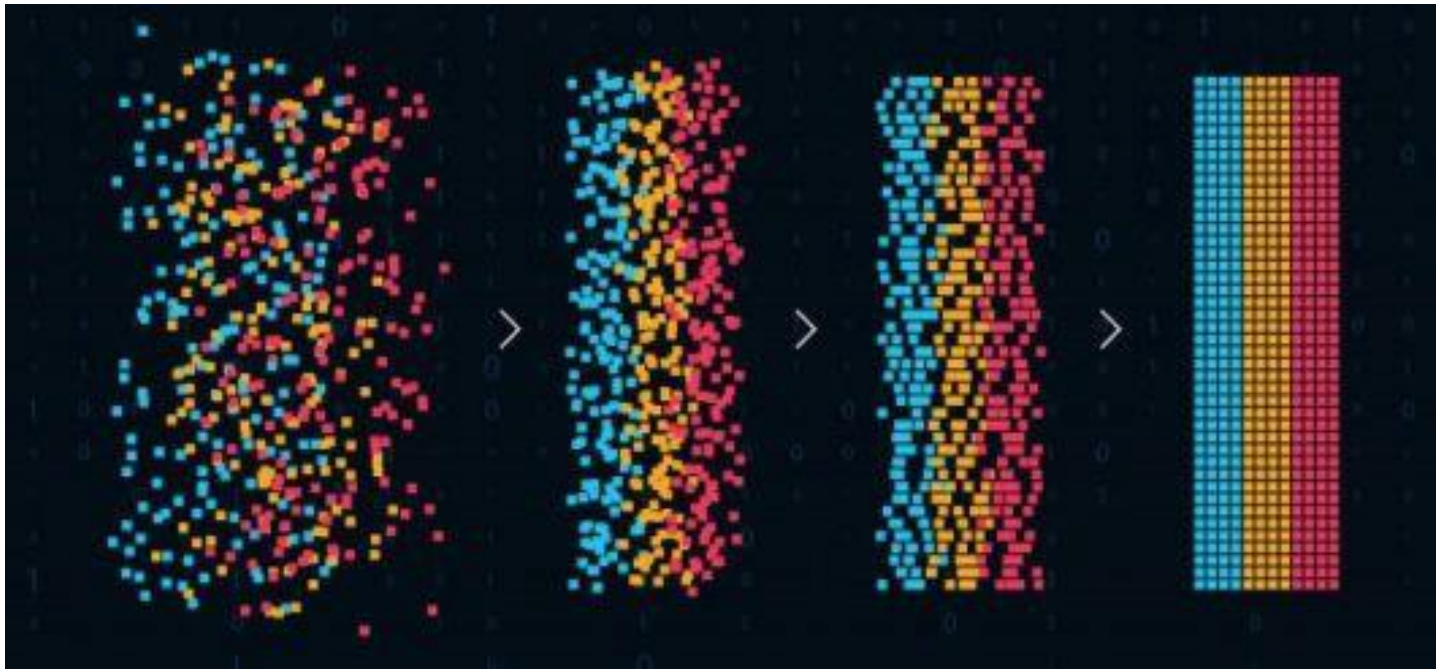




Thank You


UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

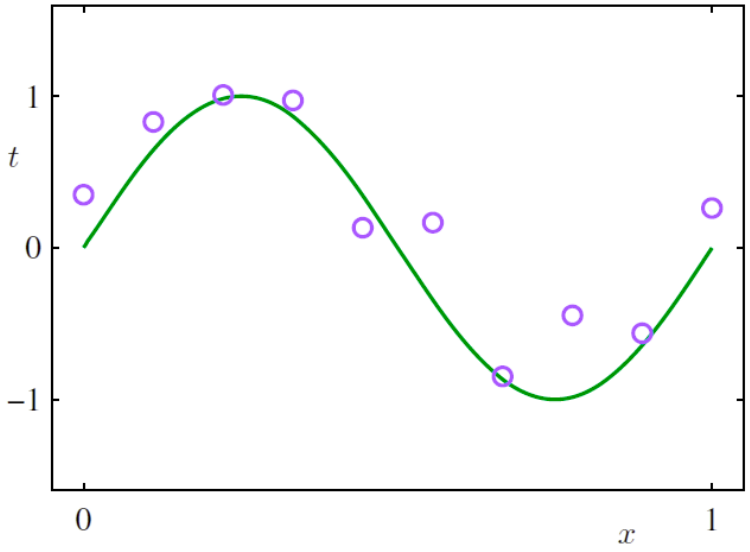
Dr Gopal Singh Phartiyal

9/08/2021

Context: Regression

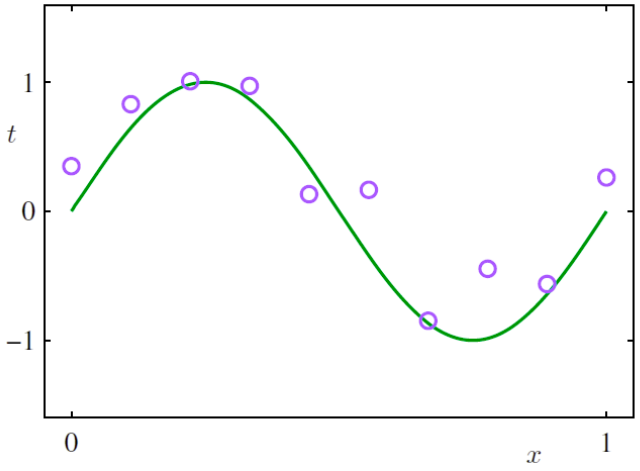
- Observe a real-valued variable (input) *(x)*
- Using this observation to predict the value of a real-valued target variable *(t)*
- Create some data using function Sin(2p*x)*
- Add random noise to target variable
- We have N observations

$$\mathbf{x} = (x_1, x_2, \ldots\ldots x_N)^T$$
$$\mathbf{t} = (t_1, t_2, \ldots\ldots t_N)^T$$



Let say N = 10, x varies from 0 to 10, t is computed using Sin(2p*x)* and then adding Gaussian noise.

Context: Regression



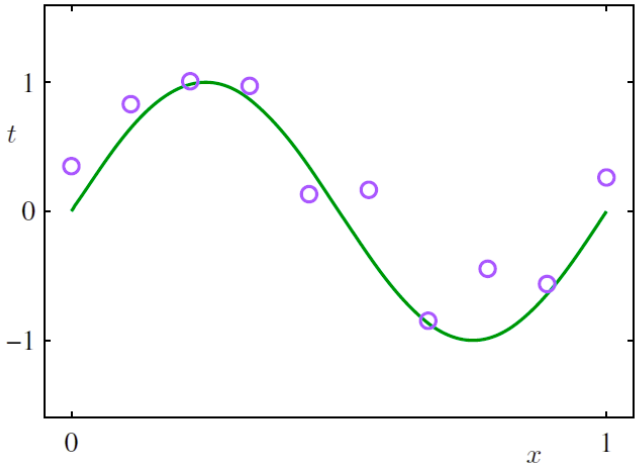
- **Goal**: Use training data to make better predictions of for some new *x*.
- Implicitly trying to discover the underlying function.
- **Problem**: Generalize from finite data, corrupted with noise (nature unknown).
- Solution approach: Curve fitting

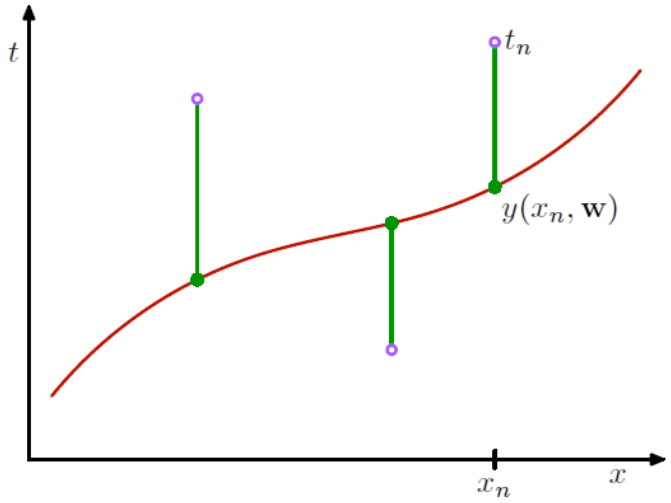
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Function is non-linear in *x* but linear in **w**. Therefore such functions are called **linear models.**

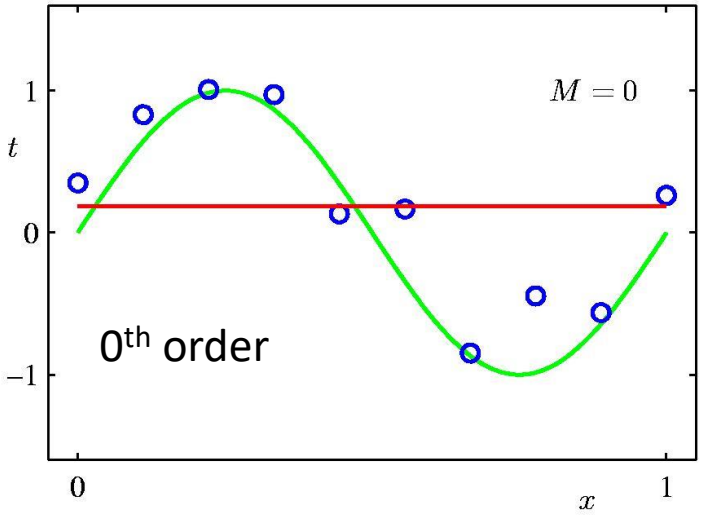
Context: Regression

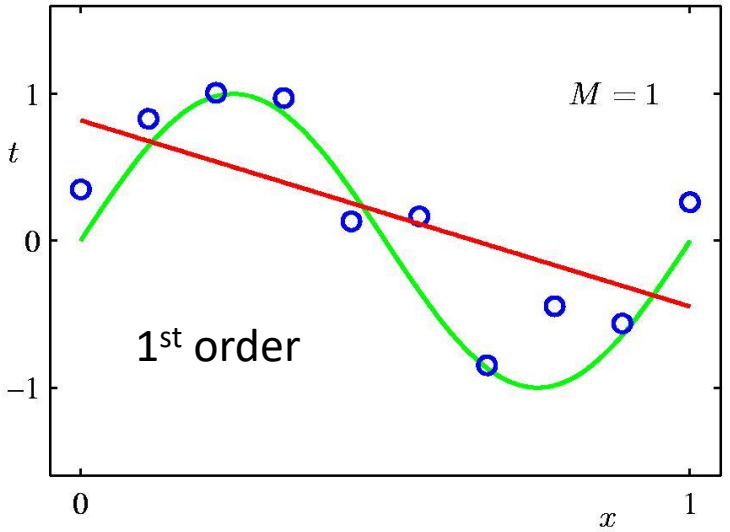
- Error function $$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$
- Change **w** so as to minimize E(**w**).
- Change M to change model.

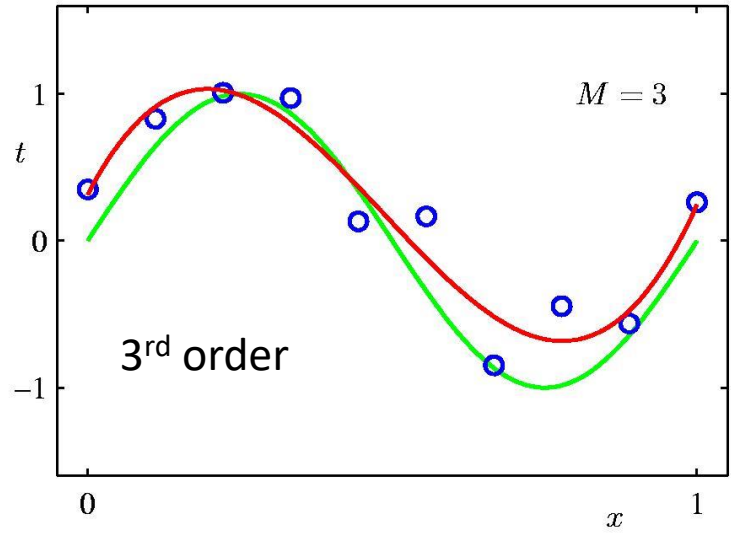


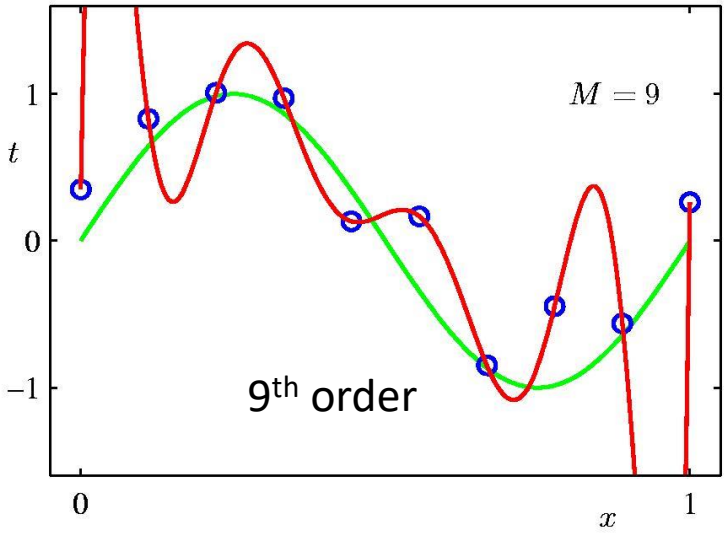


Different models and weight parameters



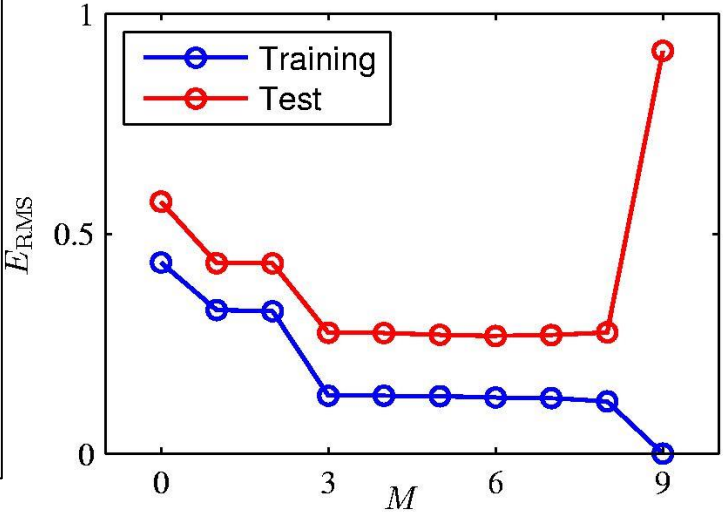




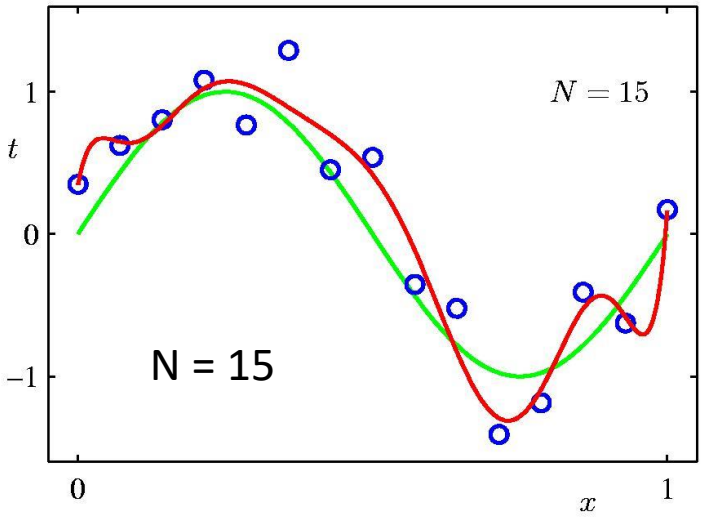


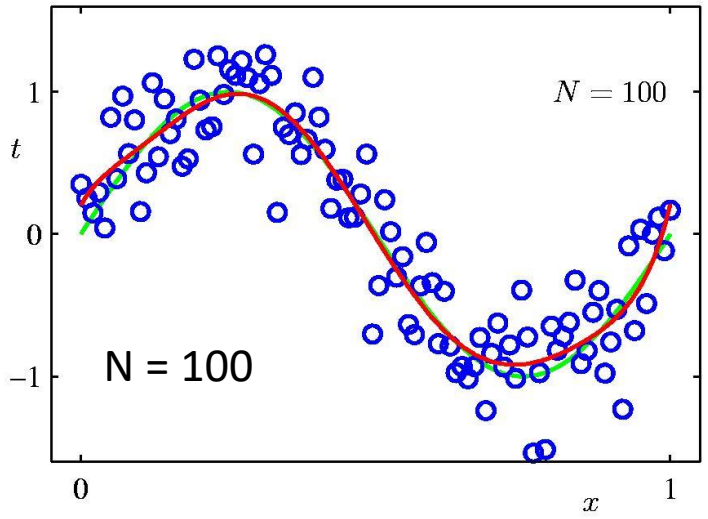
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Overfitting

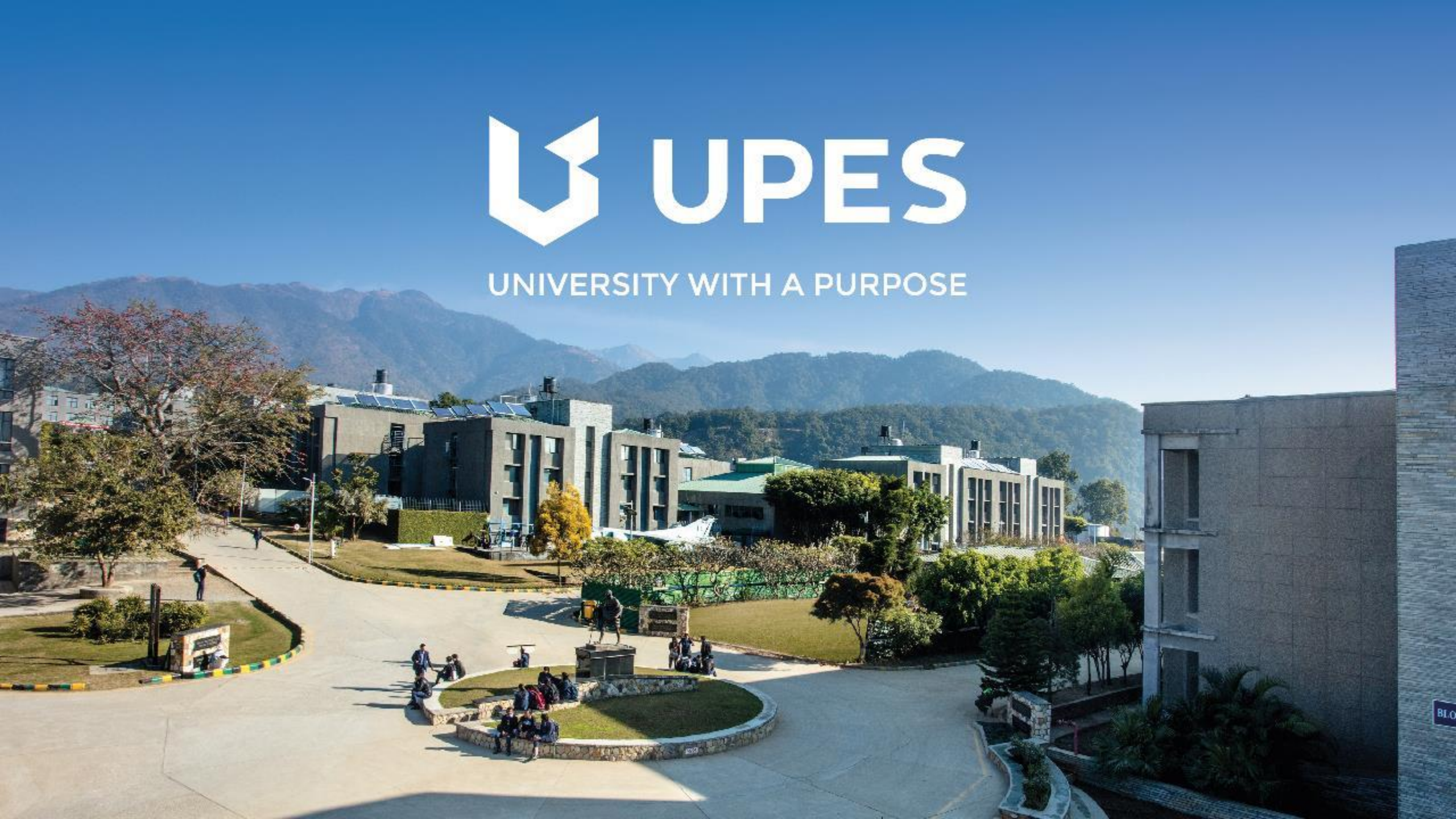


Impact of data samples



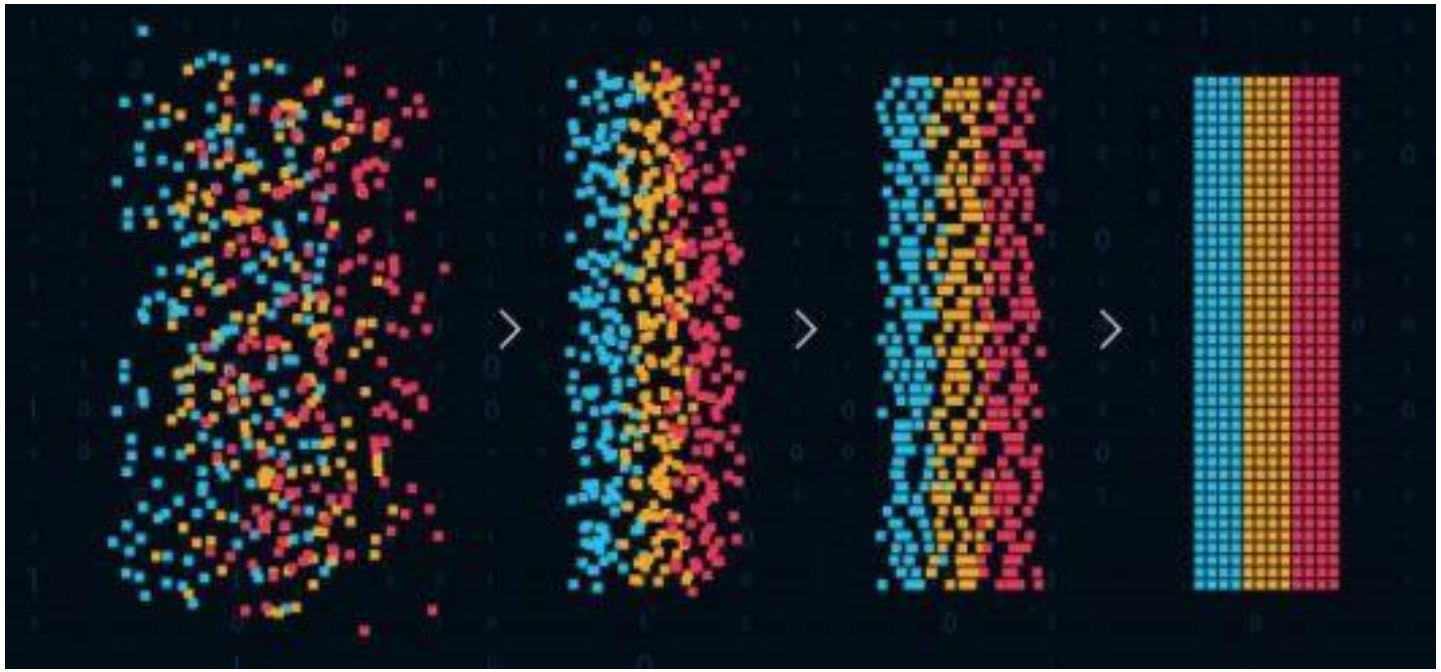


Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



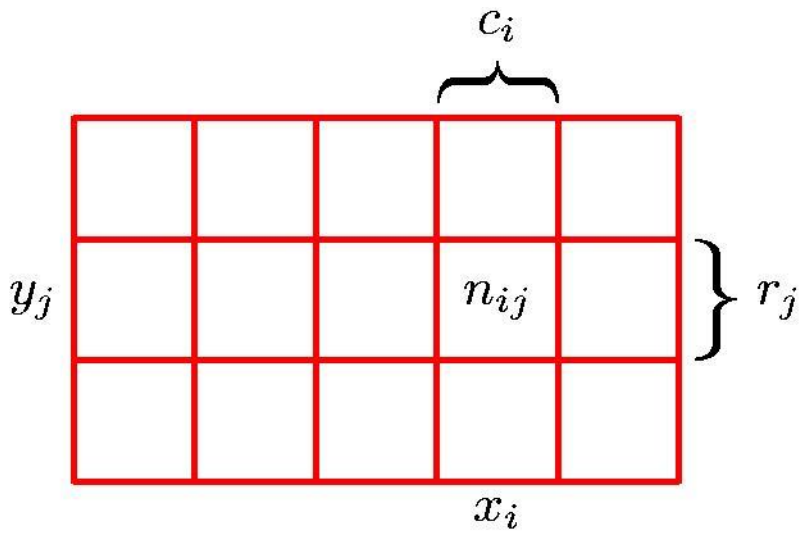
Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/08/2021

Probability Theory



Sum Rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N}\sum_{j=1}^{L} n_{ij}$$

$$= \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Product Rule

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N}$$

$$= p(Y = y_j | X = x_i)p(X = x_i)$$

Probability Theory

Baye's Theorem

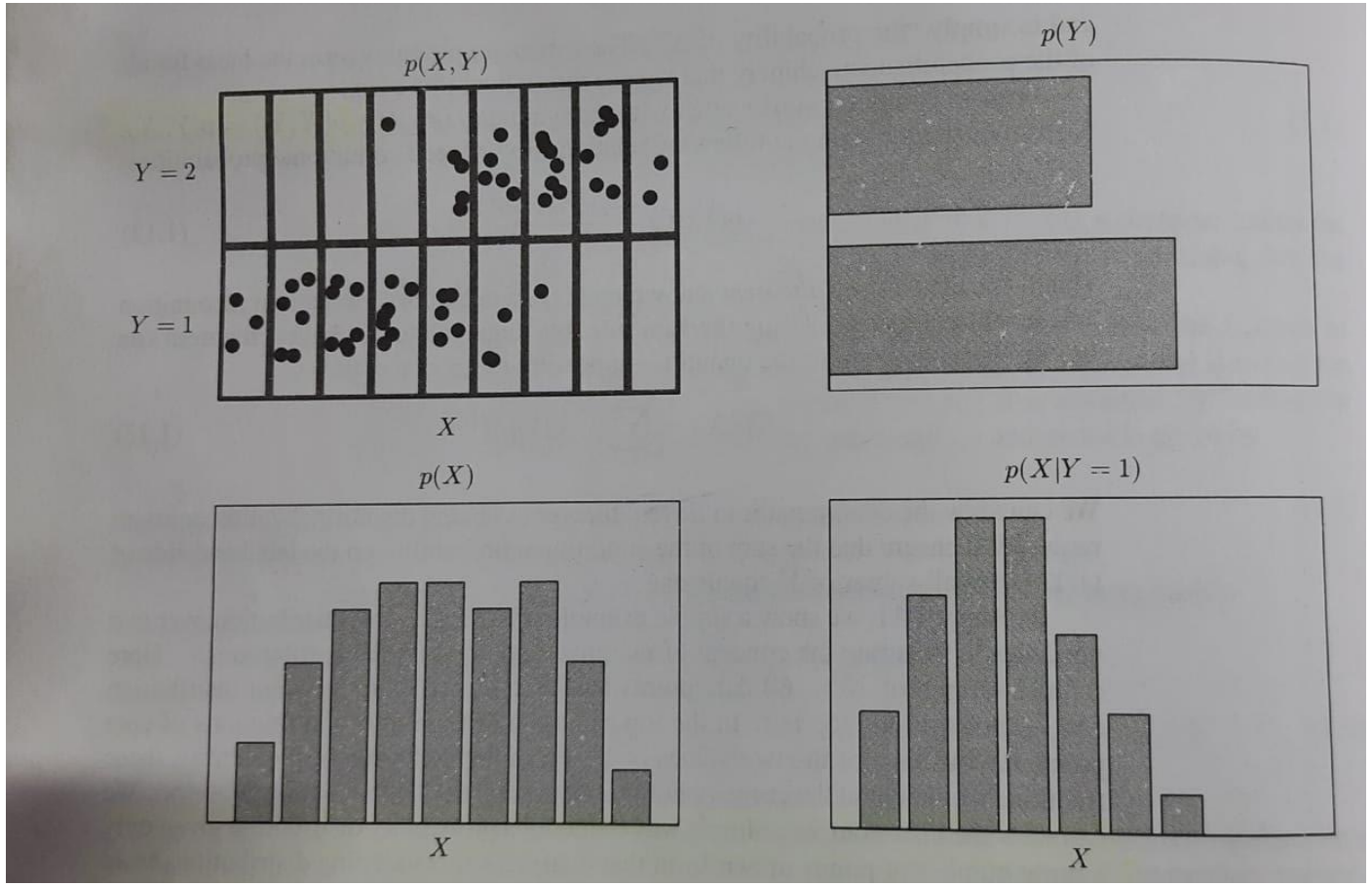
$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

$$p(X) = \sum_{Y} p(X|Y)p(Y)$$

posterior $\propto$ likelihood × prior

- Plays critical role in pattern recognition and machine learning

Example: Y =2, X = 9



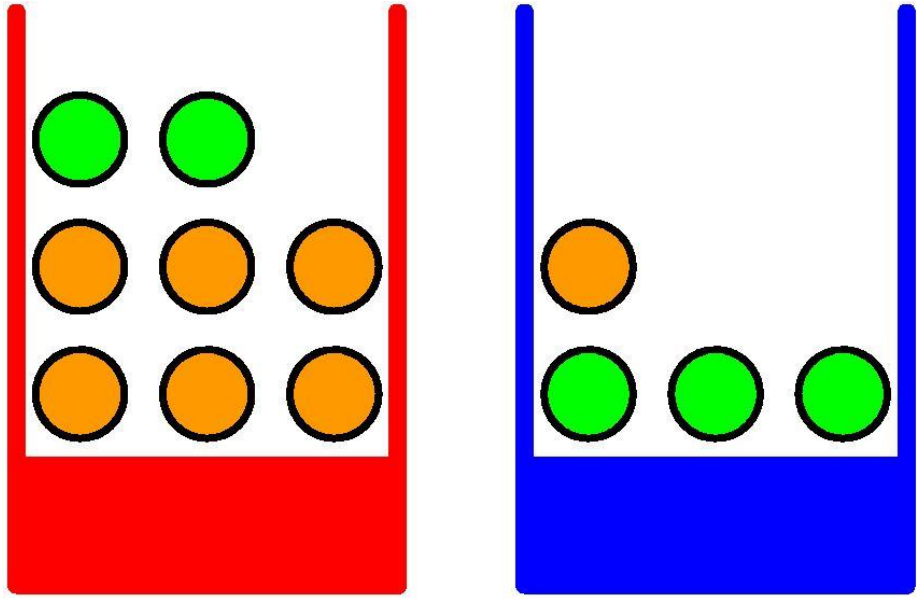
- The histogram depicts the distribution
- It can be interpreted as probability if N tends to infinity

Example: Apples (a) and Oranges (o)

- Given

$$p(B = r) = 4/10$$
$$p(B = b) = 6/10$$
$$p(B = r) + p(B = b) = 1.$$

- What is the probability of picking an apple given box is red?
- Overall probability of choosing an apple?
- What is the probability of picking a a red box given fruit is apple?



Red (r)

2 apples

6 oranges

Blue (b)

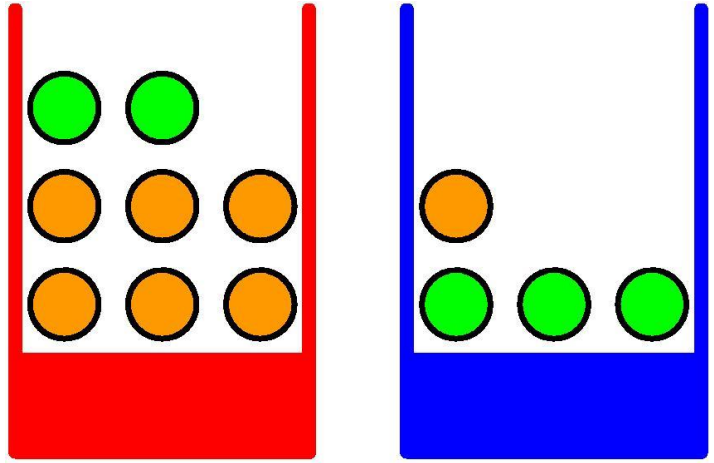
1 orange

3 apples

Example: Apples (a) and Oranges (o)

- Given

$$p(B = r) = 4/10$$
$$p(B = b) = 6/10$$
$$p(B = r) + p(B = b) = 1.$$



- What is the probability of picking an apple given box is red?

$$p(F = o|B = r) = 3/4$$

- Overall probability of choosing an apple?

$$p(F = a) = p(F = a|B = r)p(B = r) + p(F = a|B = b)p(B = b)$$

- What is the probability of picking a a red box given fruit is apple?

$$p(B = r|F = o) = \frac{p(F = o|B = r)p(B = r)}{p(F = o)} = \frac{3}{4} \times \frac{4}{10} \times \frac{20}{9} = \frac{2}{3}.$$

Probability Densities

- Probability Density

It is the probability of a variable x, i.e. p(x) over an interval (x, x+Δx) when Δx tends to zero.

- Based on the above assumption, the probability p(x) over interval (a, b) is given as

$$p(x \in (a,b)) = \int_a^b p(x)\,\mathrm{d}x.$$

- With the consideration

$$\begin{aligned} p(x) &\geqslant 0 \\ \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x &= 1. \end{aligned}$$

Expectations

- The average value of some function f(x) under a probability distribution p(x) is called the expectation of f(x) and denoted as $\mathbb{E}$(f)

- For discrete distribution

$$\mathbb{E}[f] = \sum_{x} p(x) f(x)$$

For continuous

$$\mathbb{E}[f] = \int p(x) f(x) \, \mathrm{d}x$$

Approximate Expectation (discrete and continuous)

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

Conditional Expectation (discrete)

$$\mathbb{E}_x[f|y] = \sum_{x} p(x|y) f(x)$$

Variance and Covariance

- Variance

$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

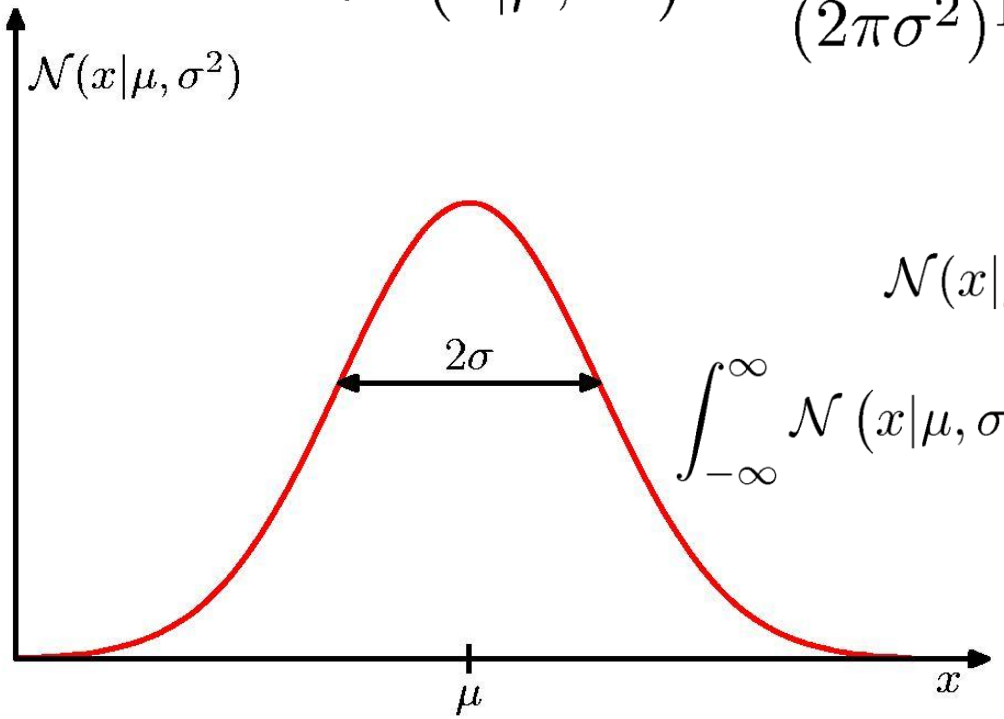
- Covariance

$$\begin{aligned}\mathrm{cov}[x, y] &= \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]\end{aligned}$$

$$\begin{aligned}\mathrm{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^{\mathrm{T}} - \mathbb{E}[\mathbf{y}^{\mathrm{T}}]\}\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\mathrm{T}}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\mathrm{T}}]\end{aligned}$$

The Gaussian Distribution

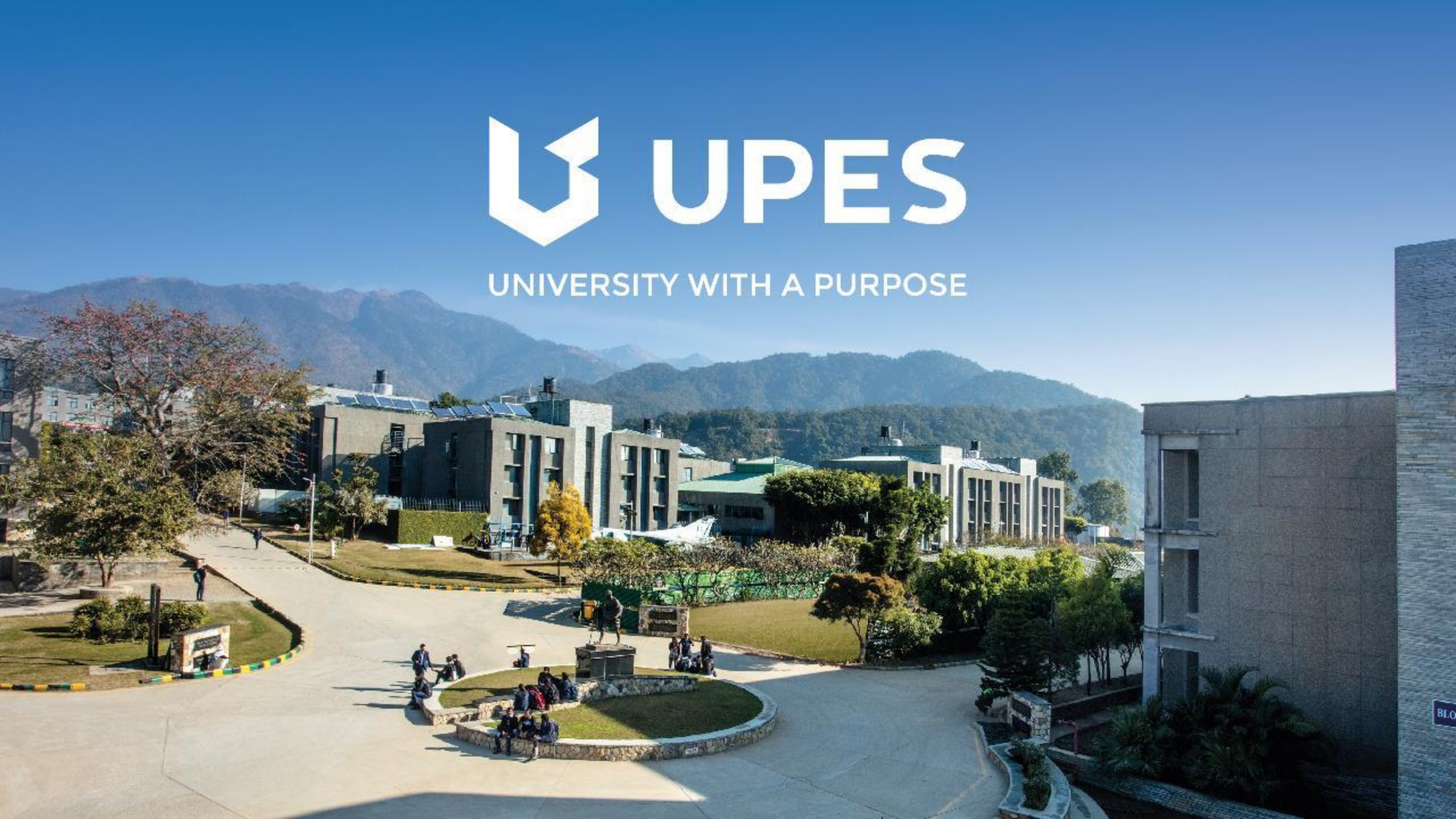
$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

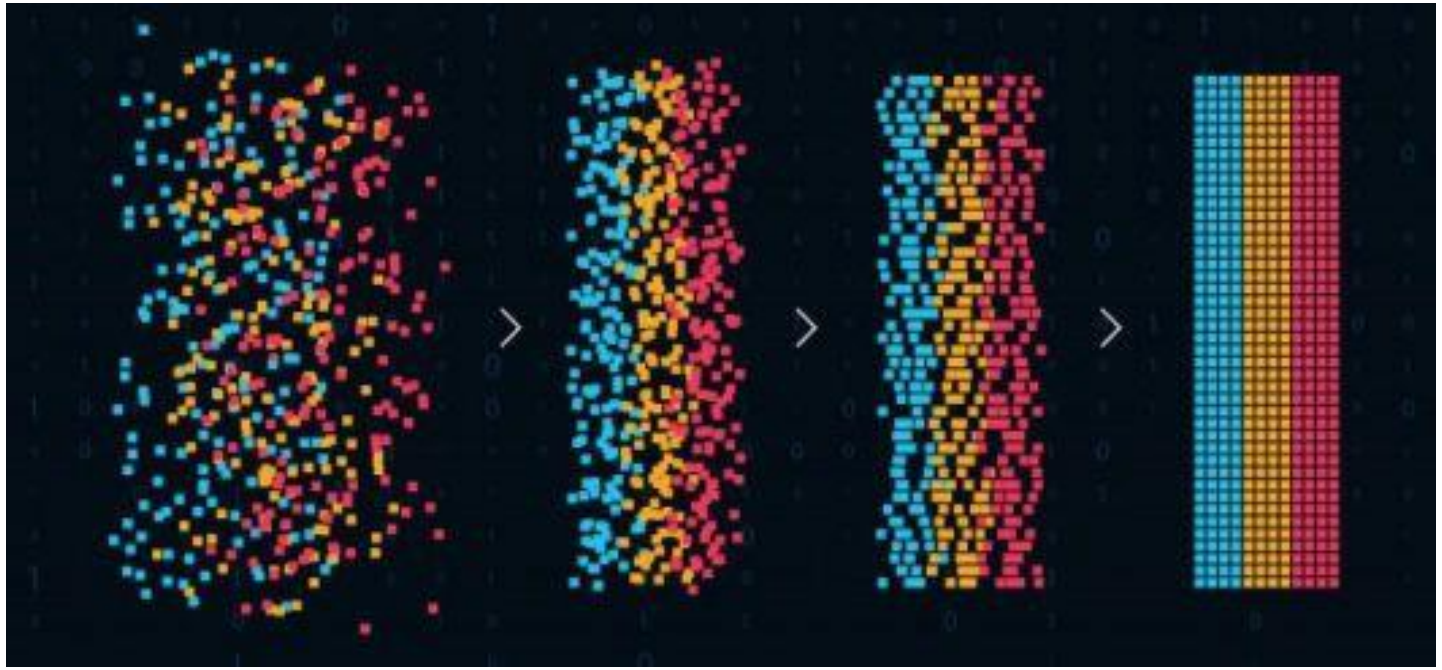
$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

18/08/2021

Bayesian Probabilities

- Sofar in probability: Classical or frequency representation
- Another way: Bayesian view: Probabilities provide a quantification of uncertainty
- Example: Uncertain event: Climate change: Polar ice cap melting
  - Constraint with classical representation: Cannot be repeated to have notion of probability laws.
  - The information on how quickly the ice is melting with help us take corresponding actions.

Bayesian Probabilities

- In such circumstances, we would like to be able to quantify our expression of uncertainty and make precise revisions of uncertainty in the light of new evidence, as well as subsequently to be able to take optimal actions or decisions as a consequence.

- This can all be achieved through the elegant, and very general, Bayesian interpretation of probability.

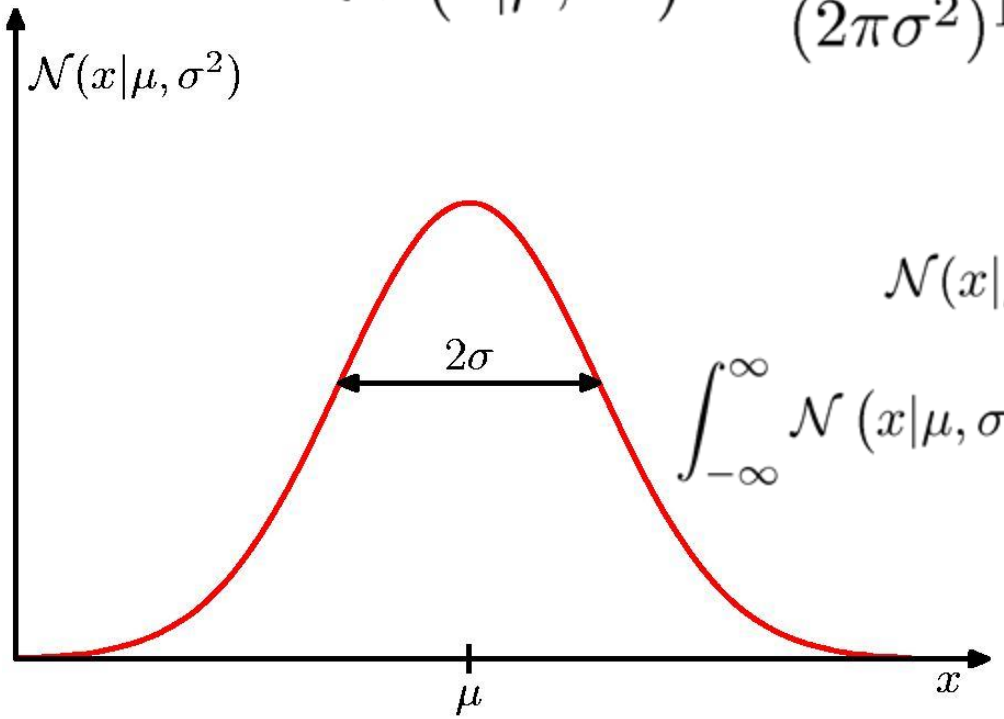
Bayesian Probabilities

Bayesian probabilities are quantities (axioms or properties) that behave precisely according to rules of probability.

Reasonable expectation (quantitative) of a belief, state of knowledge

The Gaussian Distribution

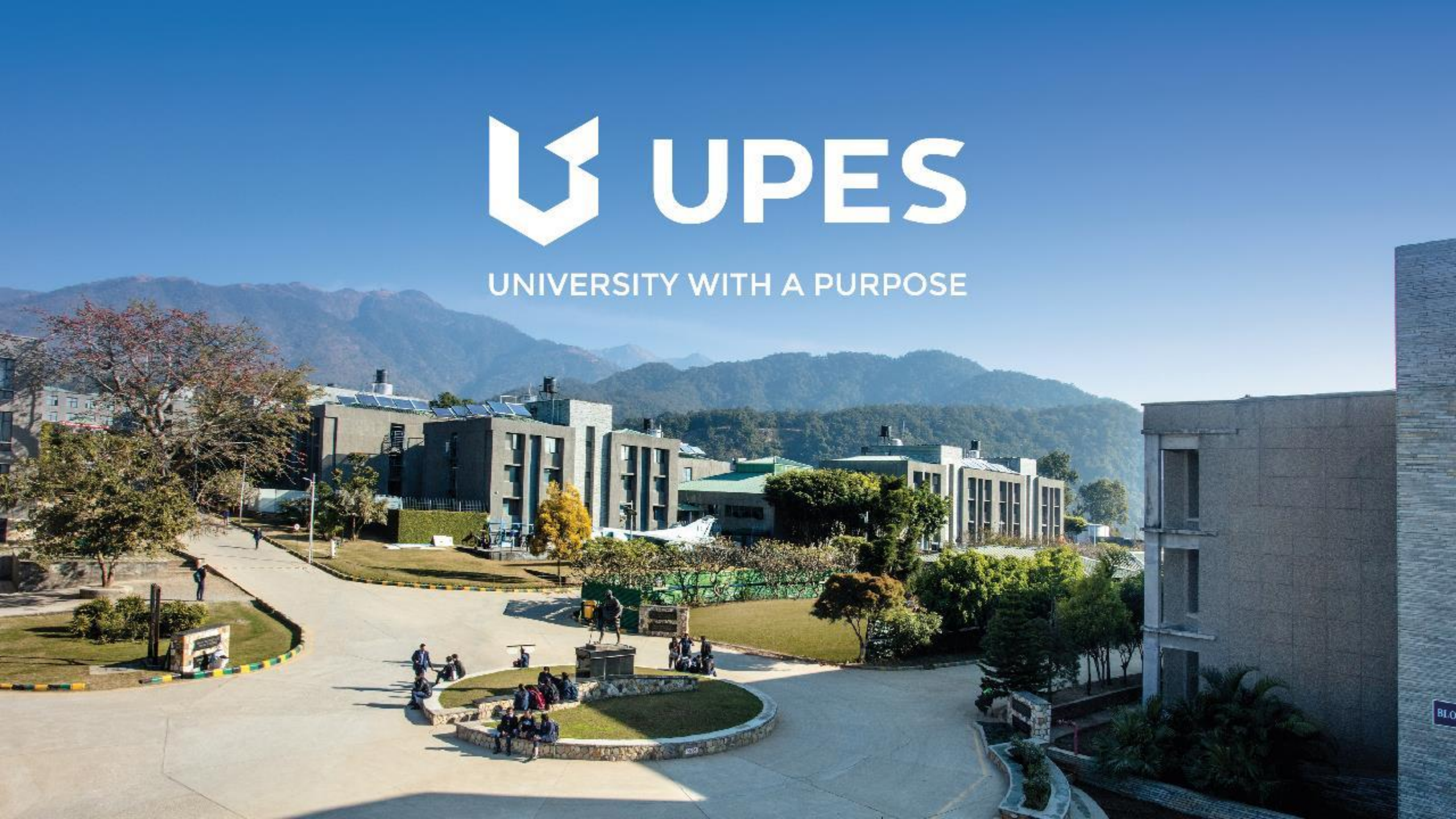
$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

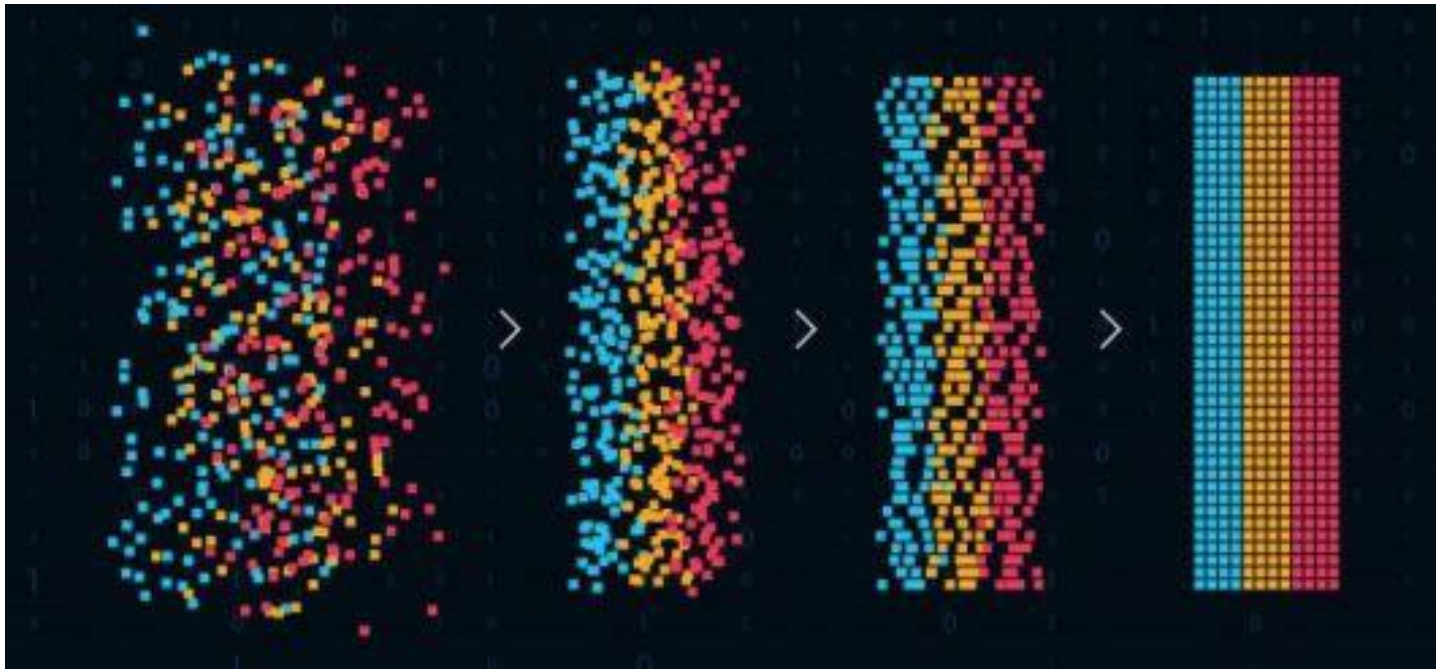
$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

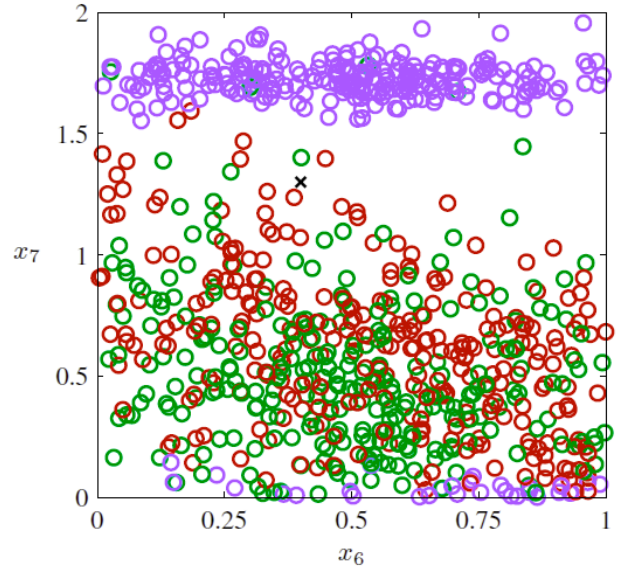
B. Tech., CSE + AI/ML

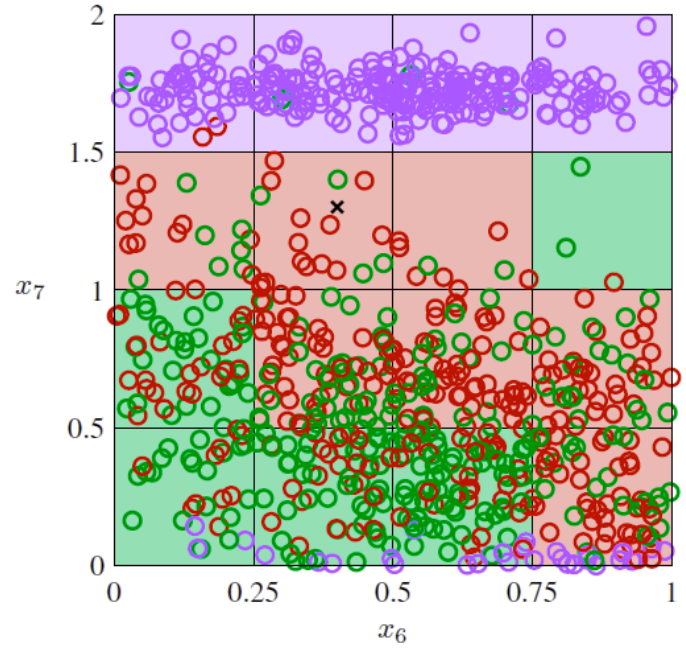
Dr Gopal Singh Phartiyal

26/08/2021

Curse of Dimensionality

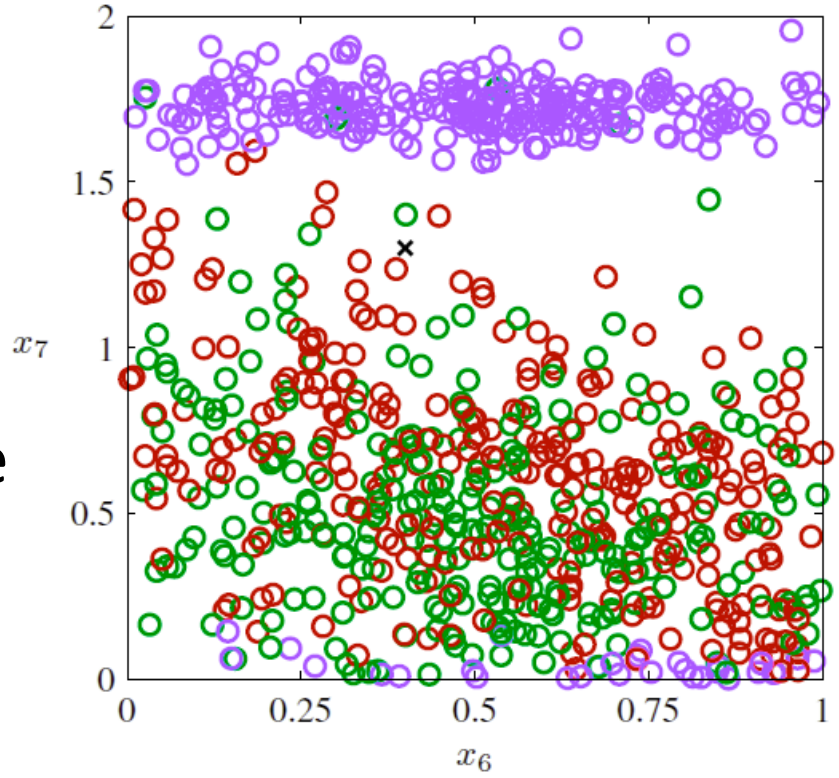
- So far we have considered input with 1-dimension space or scalar or one variable.
  - X was 1- dimensional
- But in practical applications today, x is a high-dimensional input.
- The challenges with high-dimensional data in PR applications?

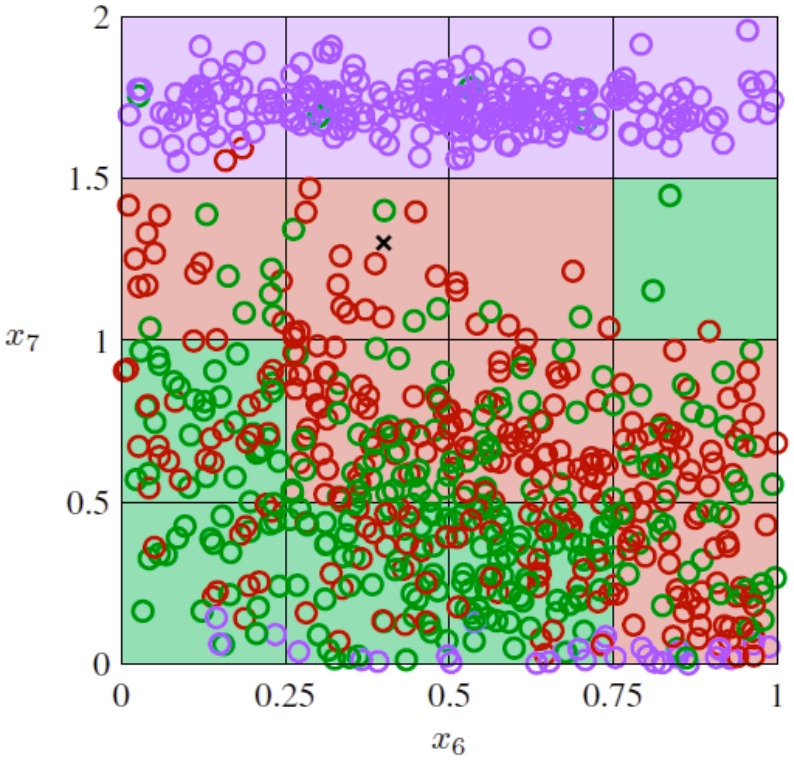




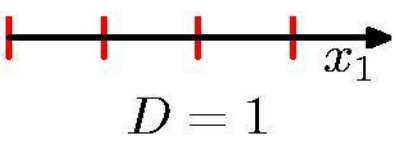
High-dimension data: Example

- Input dimension = 12
- Output = labels or classes
- Goal= To predict the label of test point (marked with x)
- Demo: Classification problem



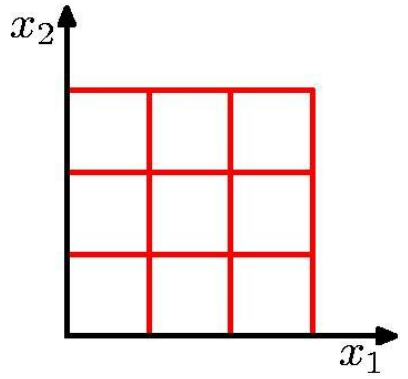


Curse of Dimensionality



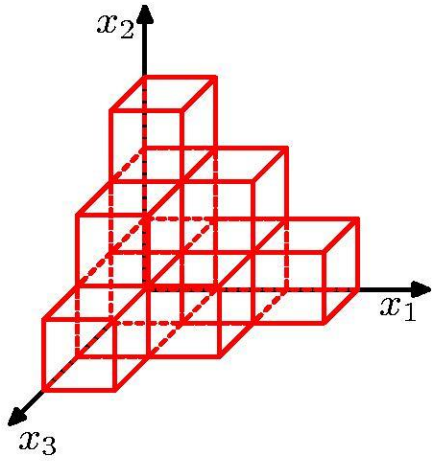
$D = 1$

- Problem with the before mentioned example:



$D = 2$

- As input dimension increases, the number of cells increases exponentially.



$D = 3$

- In turn requires large amounts of training data.

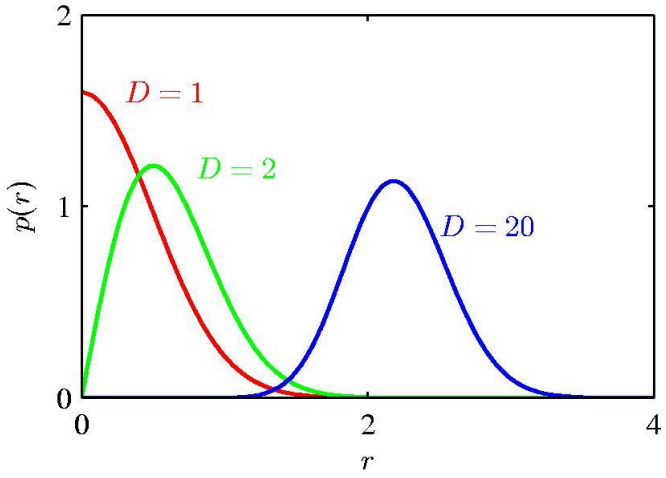
Curse of Dimensionality

- Impact of increased input-dimensionality on polynomial curve fitting
  - M = 3

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

- Number of Coefficients are proportional to **D^M** **(power law function)**

- Gaussian Densities in higher dimensions

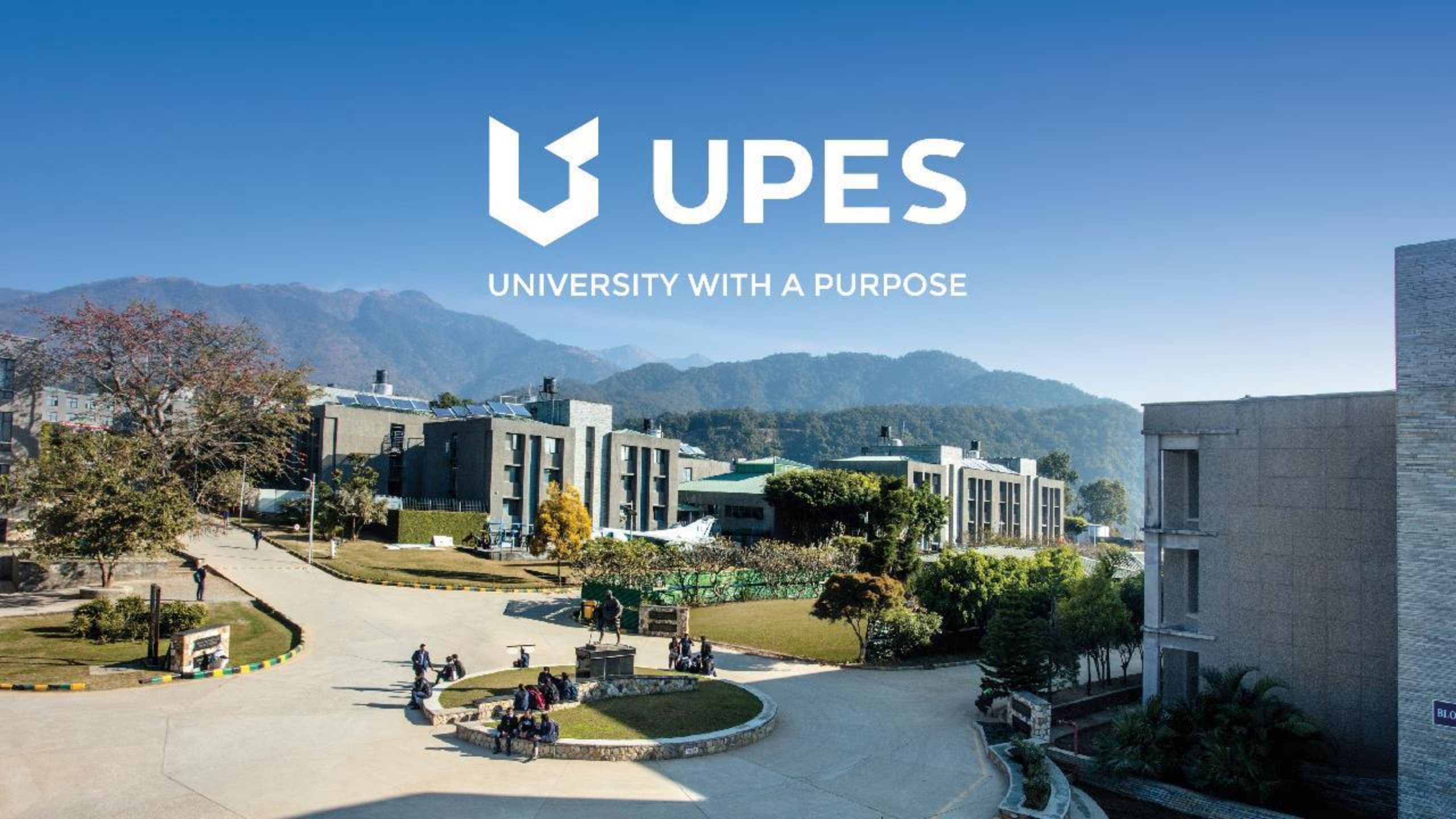


Next time: Decision Theory

- Probability theory provides
  - A consistent mathematical framework for quantifying and manipulating uncertainty
- **Inference**
  - Determination of $p(x, t)$ from a set of training data is an example of *inference* and
- **Decision theory**
  - The subject of decision theory to tell us how to make optimal decisions given the appropriate probabilities.
- Decision theory **plus** Probability theory
  - Help us To make optimal decisions

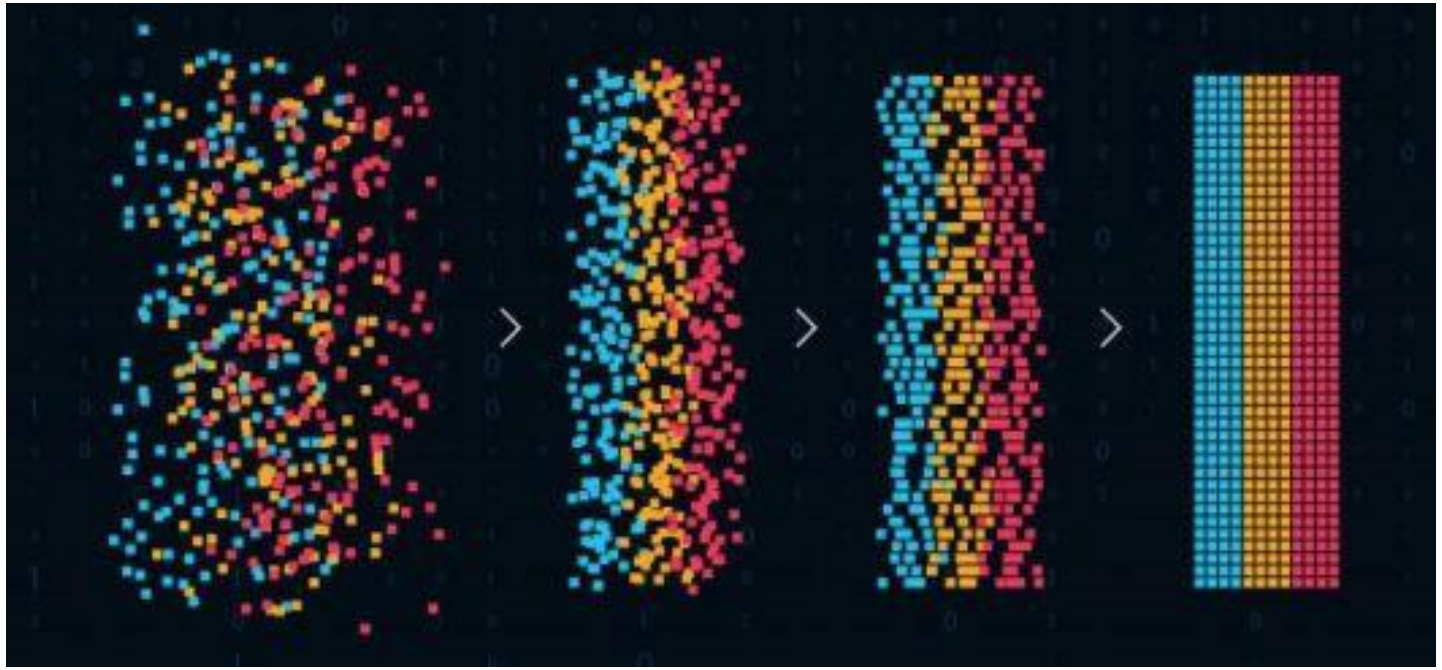
Next time: Curse of Dimensionality

Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

26/08/2021

Decision Theory

- Need: Decision theory establishes the fundamentals on what decision to take once ***inference*** from **data** is mode.

- Inference step
- Determine either $p(t|\mathbf{x})$ or $p(\mathbf{x}, t)$.

- Decision step
- For given X, determine optimal t.

Decision Theory

- Example:
- Goal: To tell whether a person has cancer or not.
- Input: Medical image (X-ray), a vector or matrix
- Target: Two Classes
  - Cancer/No-cancer (C_1/C_2 or True/False or 1/0)
- Inference problem
  - To compute p(**x**,C_1) and further p(**x**, t)
- Decision problem
  - This inferred value will decide whether to diagnose or not. Therefore, we want the decision to be to be optimal
- **In a nutshell:** how to make optimal decisions based on appropriate probabilities

Images Source: url: https://www.shutterstock.com/search/lung+cancer+x+ray

Decision Theory

- In the previous example, if we want to know the class of new patient based on his/her X-ray image. We need $p(C_k|x)$.
- According to Bayes theorem

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{p(\boldsymbol{x})}$$

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N}\sum_{j=1}^{L} n_{ij}$$

$$= \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

- All quantities on the right can be computed from $p(\boldsymbol{x}, C_k)$ i.e. Joint probability
- Remember sum and product rule to relate Joint probability with marginal/prior probability and conditional probability

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N}$$

$$= p(Y = y_j|X = x_i)p(X = x_i)$$

Continued…

- If we wish to minimize the chance of assigning x to wrong class, then **intuitively** we will choose class having higer posterior probability.

- New goal: To make as few misclassifications as possible
- How?
- Distribute the input space into **decision regions** (R_k) such that allo x on R_k are assigned to C_k.
  - Here we have two classes, therefore two decision regions
- The boundaries between these regions is called ***decision boundaries*** or ***decision surface***.

Continued…

- Optimal decision rule via example of cancer patient
- We will try to minimize misclassification.
- Miscalssification: **x** in R_1 assigned to C_2 or **x** in R_2 assigned to C_1

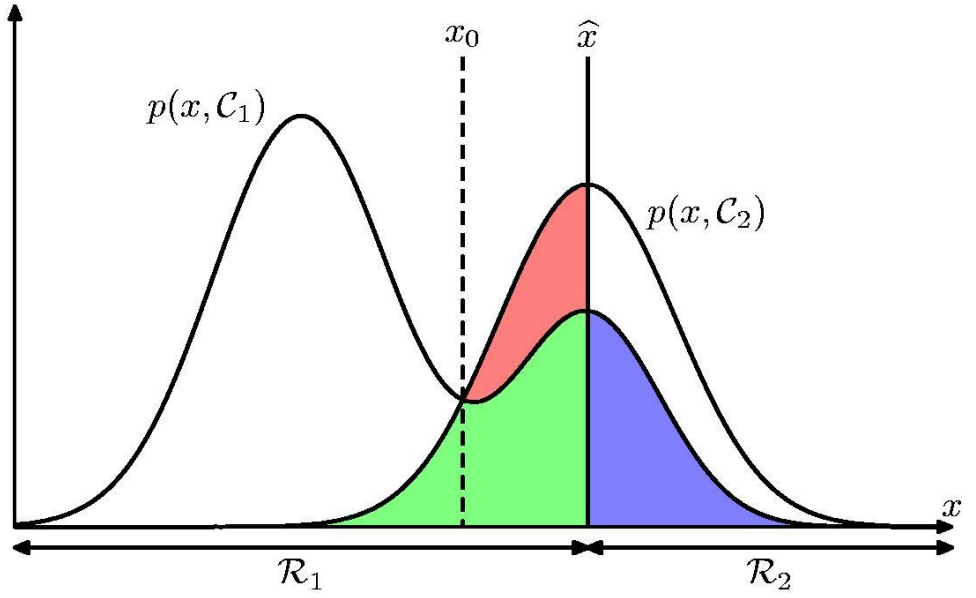
$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, \mathrm{d}\mathbf{x}. \end{aligned}$$

$$\begin{aligned} p(\text{correct}) &= \sum_{k=1}^{K} p(\mathbf{x} \in \mathcal{R}_k, \mathcal{C}_k) \\ &= \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(\mathbf{x}, \mathcal{C}_k)\, \mathrm{d}\mathbf{x} \end{aligned}$$

- The p(mistake) is minimum when x is assigned to class which has the highest posterior probability
- General case: For more number of classes, it is easier to maximize the probability of correct classification minimizing the incorrect classification.

Graphically: Two classes, single variable

Schematic illustration of the joint probabilities $p(x, \mathcal{C}_k)$ for each of two classes plotted against x, together with the decision boundary $x = \widehat{x}$. Values of $x \geqslant \widehat{x}$ are classified as class $\mathcal{C}_2$ and hence belong to decision region $\mathcal{R}_2$, whereas points $x < \widehat{x}$ are classified as $\mathcal{C}_1$ and belong to $\mathcal{R}_1$. Errors arise from the blue, green, and red regions, so that for $x < \widehat{x}$ the errors are due to points from class $\mathcal{C}_2$ being misclassified as $\mathcal{C}_1$ (represented by the sum of the red and green regions), and conversely for points in the region $x \geqslant \widehat{x}$ the errors are due to points from class $\mathcal{C}_1$ being misclassified as $\mathcal{C}_2$ (represented by the blue region). As we vary the location $\widehat{x}$ of the decision boundary, the combined areas of the blue and green regions remains constant, whereas the size of the red region varies. The optimal choice for $\widehat{x}$ is where the curves for $p(x, \mathcal{C}_1)$ and $p(x, \mathcal{C}_2)$ cross, corresponding to $\widehat{x} = x_0$, because in this case the red region disappears. This is equivalent to the minimum misclassification rate decision rule, which assigns each value of x to the class having the higher posterior probability $p(\mathcal{C}_k|x)$.



Minimizing the Loss

- This time, simply minimizing the number of misclassification will not work.
- We have to prioritize misclassifications.
- Example: Diagnosing cancer patient
- Misclassifications
  - No cancer actually but diagnosed with cancer treatment as per model
  - Had cancer actually but diagnosed as healthy
- Are these two misclassifications of equal importance?????
- We need other parameter to measure the impact of misclassifications and then minimize it. ***Loss function***

Minimizing the Loss

- ***Expected loss***

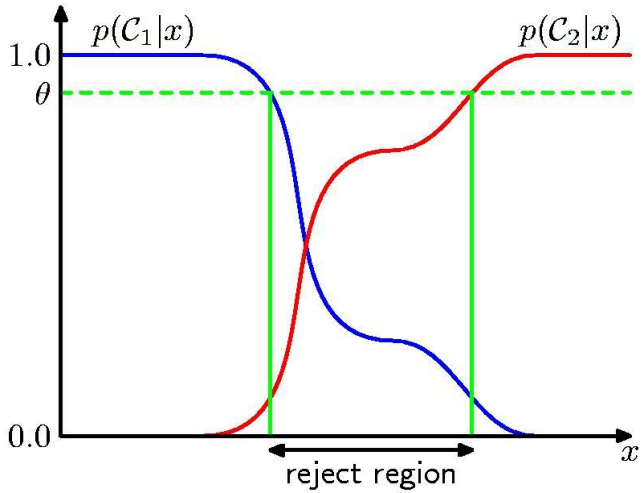
$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k)\, \mathrm{d}\mathbf{x}$$

$$\mathbb{E}[L] = \sum_k L_{kj} p(\mathcal{C}_k|\mathbf{x})$$

Decision

Truth		cancer	normal
	cancer	0	1000
	normal	1	0

Reject option



Like to combine Inference and Decision ?

- ***Inference stage:*** *Computing posterior probabilities via models learning*
- ***Decision stage****: Making decisions or class assignments*

- *Why not do it in one go (combine the stages): one function doing both the stages*
  - *Discriminant function does that*

- *The advantage of keeping inference and decision stages separate are*
  - Minimizing risk (loss matrix may change over time)
  - Reject option
  - Unbalanced class priors
  - Combining models

Types of approaches to solve decision problems

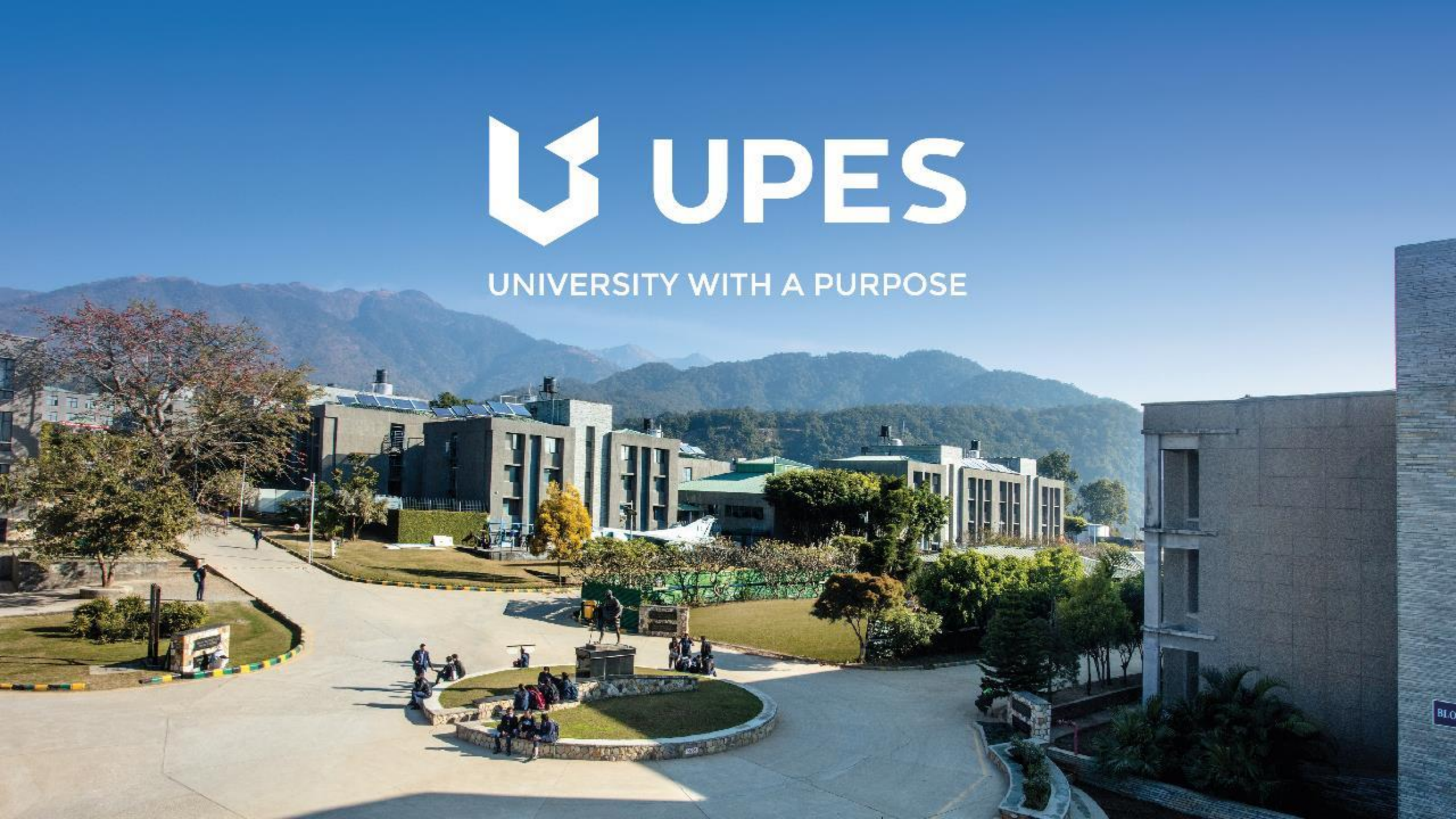
- ***Generative models:*** *First model joint distributions, then decision*
- ***Discriminative models:*** *First model class posterior probabilities then decision*
- ***Discriminant functions:*** *Directly map input to output (probabilities play no role)*

Next time:

Thank You

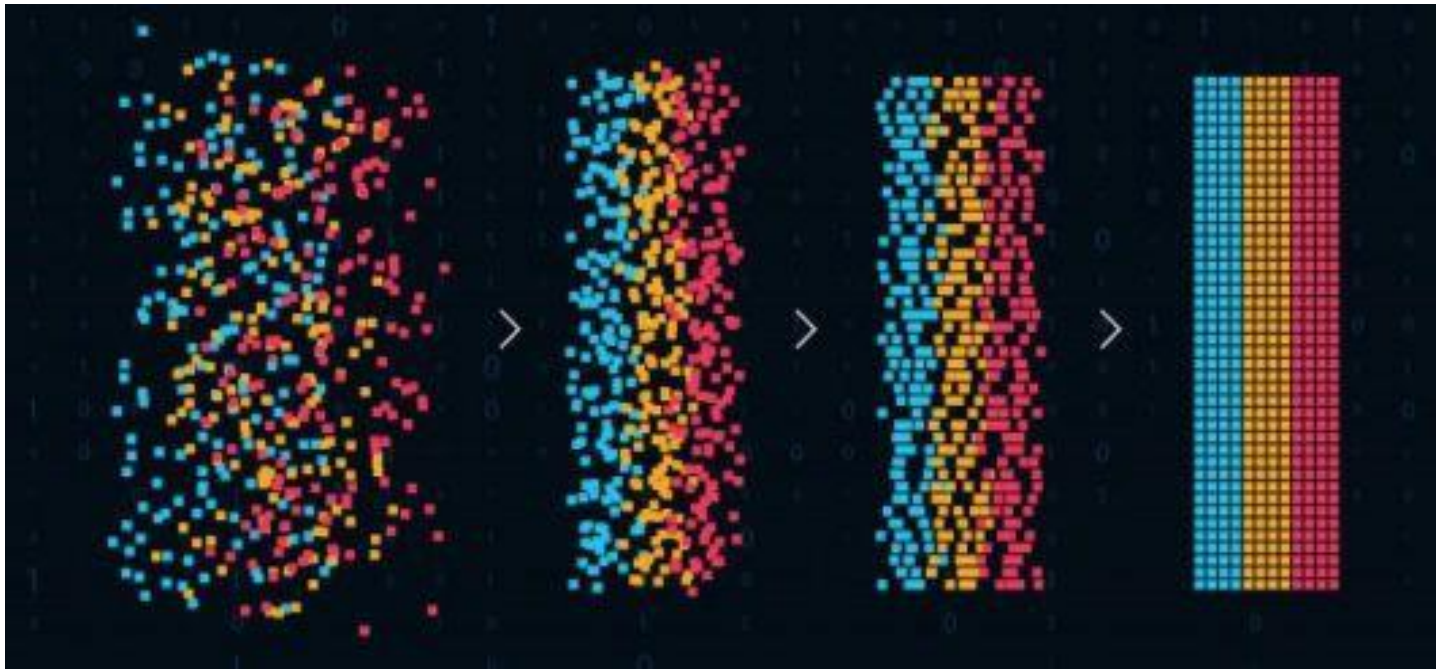
UPES

UNIVERSITY WITH A PURPOSE


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

26/08/2021

Information Theory

- Consider a discrete random variable *x.*
- ***Question****: How much* information is received when we observe a specific value for this variable x.
- Degree of ***surprise:*** On learning the value of x.

- If we are told that a highly improbable event has just occurred, we will have received more information than if we were told that some very likely event has just occurred, and if we knew that the event was certain to happen we would receive no information.

- measure of information content will therefore depend on the probability distribution $p(x)$, and we therefore look for a quantity $h(x)$ that is a monotonic function of the probability $p(x)$ and that expresses the information content

Mathematical formulation

- Consider 2 events x and y that are unrelated

- Net information gain in overserving the two together = sum of information gained on observing them separately

$$h(x, y) = h(x) + h(y).$$

- The two events are statistically independent as well.

$$p(x, y) = p(x)p(y)$$

- Negative sign
- Base 2
- Unit of information = **bits**

$$\boldsymbol{h(x) = -\log_2 p(x)}$$

Mathematical formulation

- The average amount of information transferred between sender and receiver is = expectation of

$$h(x) = -\log_2 p(x)$$

$$\mathrm{H}[x] = -\sum_{x} p(x) \log_2 p(x)$$ ***Entropy !!!***

- Important quantity in
  - coding theory
  - statistical physics
  - machine learning

Example

- Consider a random variable *x* having 8 possible states, each of which is equally likely.
- In order to communicate the value of *x* to a receiver, we would need to transmit a message of length 3.

$$H[x] = ??$$

- Change probabilities

(1/2 , 1/4 , 1/8 , 1/16 , 1/64 , 1/64 , 1/64 , 1/64)

- New H[*x*] ??

Example

x	a	b	c	d	e	f	g	h
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$
code	0	10	110	1110	111100	111101	111110	111111

- New H[*x*] ??

$$\begin{aligned} \mathrm{H}[x] &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} \\ &= 2 \text{ bits} \end{aligned}$$

$$\begin{aligned} \text{average code length} &= \frac{1}{2}\times 1 + \frac{1}{4}\times 2 + \frac{1}{8}\times 3 + \frac{1}{16}\times 4 + 4\times\frac{1}{64}\times 6 \\ &= 2 \text{ bits} \end{aligned}$$

Another Example

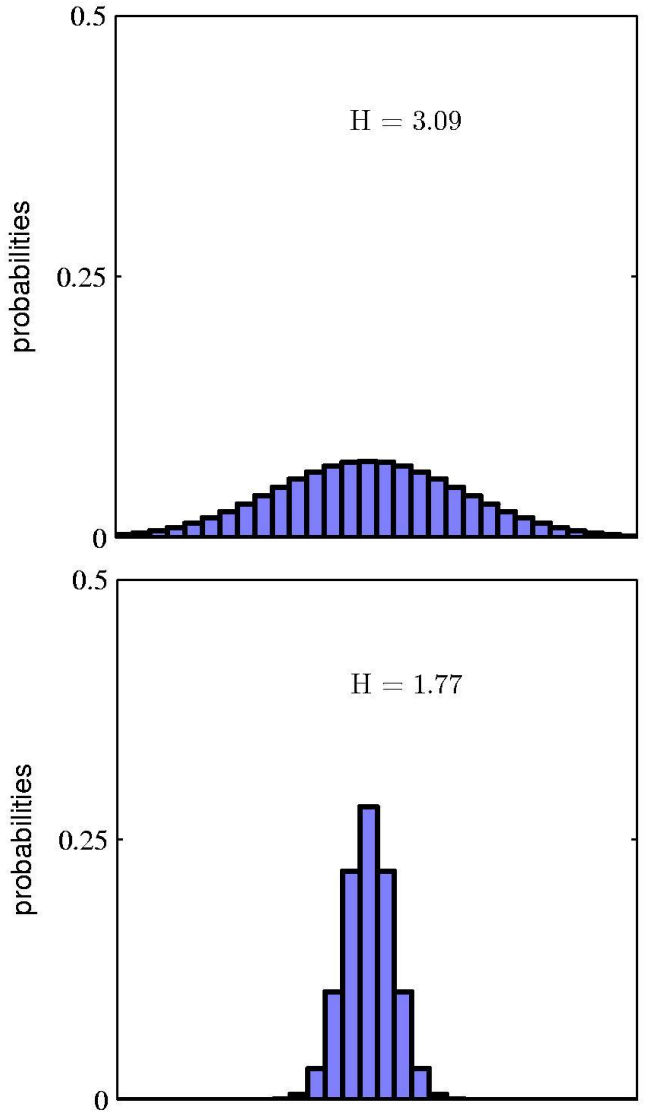
- In how many ways can N identical objects be allocated M bins?

$$W = \frac{N!}{\prod_i n_i!}$$

$$\mathrm{H} = \frac{1}{N} \ln W \simeq - \lim_{N \to \infty} \sum_i \left(\frac{n_i}{N} \right) \ln \left(\frac{n_i}{N} \right) = - \sum_i p_i \ln p_i$$

- Entropy maximized when

$$\forall i : p_i = \frac{1}{M}$$



Important formulas

- Differential Entropy

$$\mathrm{H}[x] = \frac{1}{2}\left\{1 + \ln(2\pi\sigma^2)\right\}.$$

- Conditional Entropy

$$\mathrm{H}[\mathbf{y}|\mathbf{x}] = -\iint p(\mathbf{y},\mathbf{x}) \ln p(\mathbf{y}|\mathbf{x})\, \mathrm{d}\mathbf{y}\, \mathrm{d}\mathbf{x}$$

$$\mathrm{H}[\mathbf{x},\mathbf{y}] = \mathrm{H}[\mathbf{y}|\mathbf{x}] + \mathrm{H}[\mathbf{x}]$$

- Kullback-Leibler Divergence

$$\mathrm{KL}(p\|q) \simeq \frac{1}{N}\sum_{n=1}^{N}\left\{-\ln q(\mathbf{x}_n|\boldsymbol{\theta}) + \ln p(\mathbf{x}_n)\right\}$$

$$\mathrm{KL}(p\|q) \geqslant 0 \qquad \mathrm{KL}(p\|q) \not\equiv \mathrm{KL}(q\|p)$$

- Mutual Information

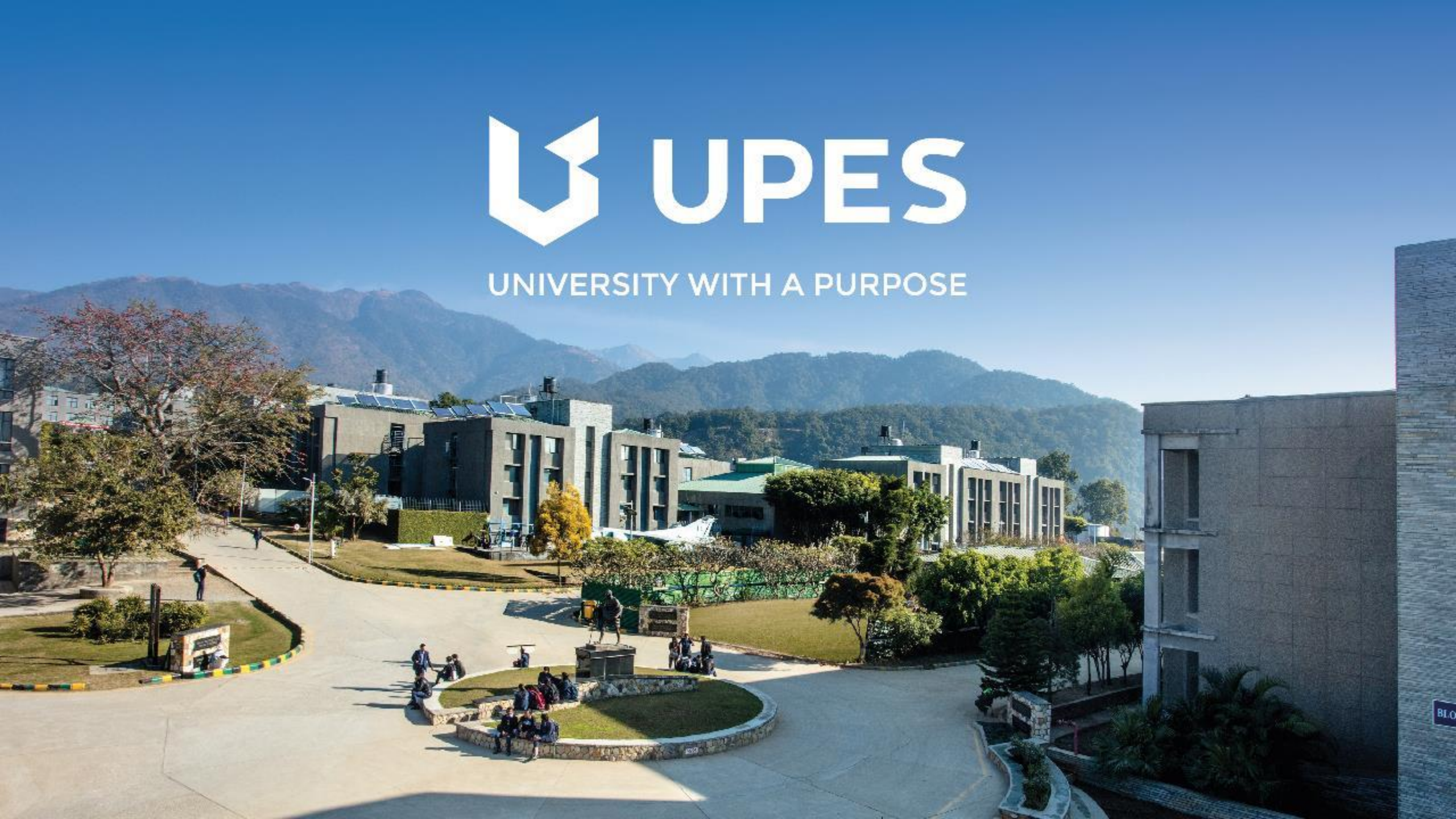
$$\mathrm{I}[\mathbf{x},\mathbf{y}] = \mathrm{H}[\mathbf{x}] - \mathrm{H}[\mathbf{x}|\mathbf{y}] = \mathrm{H}[\mathbf{y}] - \mathrm{H}[\mathbf{y}|\mathbf{x}]$$

Next time:

Thank You

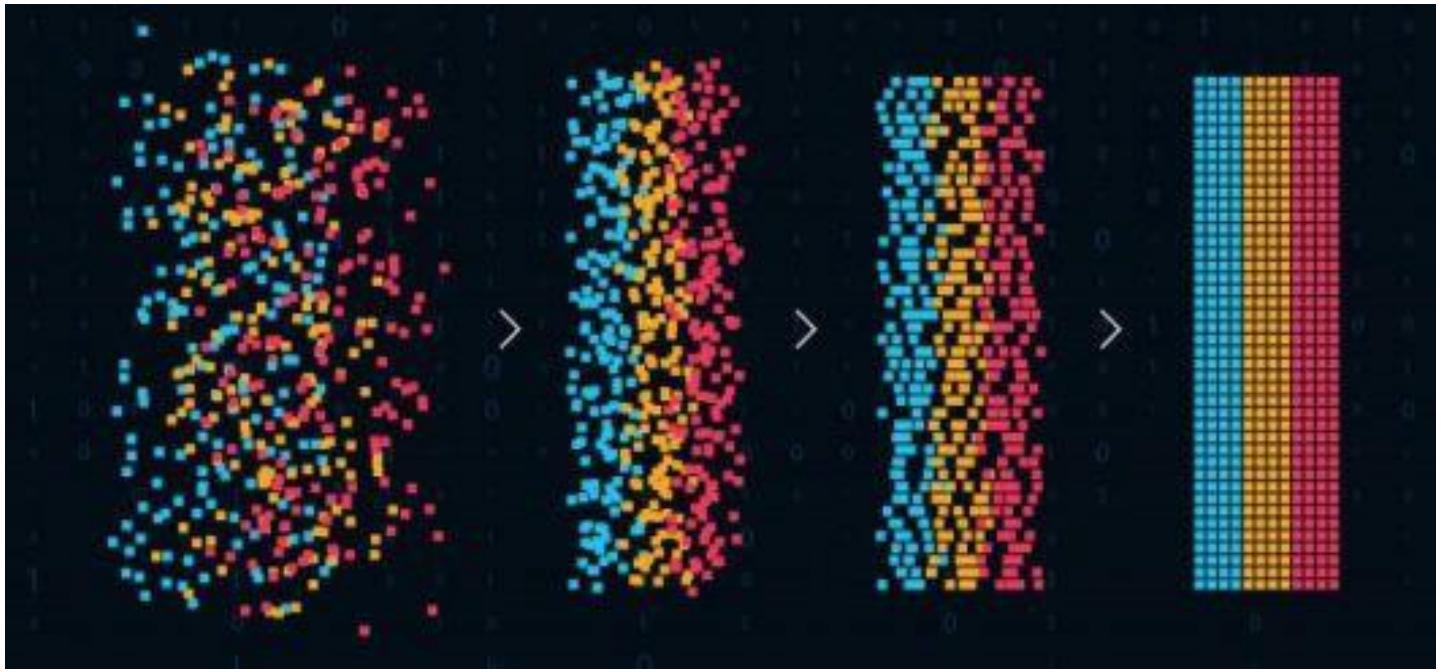
UPES

UNIVERSITY WITH A PURPOSE


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/09/2021

Binary variables

$$\mathbb{E}[f] = \sum_x p(x) f(x)$$

$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

For discrete random binary variable

Coin flipping: heads =1, tails = 0

$$p(x = 1|\mu) = \mu$$

- Probability distribution over x

$$\mathrm{Bern}(x|\mu) = \mu^x (1-\mu)^{1-x}$$ Bernoulli Distribution

- Distribution statistics

$$\mathbb{E}[x] = \mu$$

$$\mathrm{var}[x] = \mu(1-\mu)$$

Bernoulli Distribution

- Likelihood function for Bernoulli
- Given: $\mathcal{D} = \{x_1, \ldots, x_N\},\ m$ heads $(1),\ N - m$ tails (0)

$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n}$$

- This can be interpreted as likelihood function
- In order to find μ that maximizes the likelihood

$$\ln p(\mathcal{D}|\mu) = \sum_{n=1}^{N} \ln p(x_n|\mu) = \sum_{n=1}^{N} \{x_n \ln \mu + (1-x_n)\ln(1-\mu)\}$$

The probability of landing heads is given, in this maximum likelihood framework, by the fraction of observations of heads in the data set.

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{m}{N}$$

Simple mean

$$\mu_{\mathrm{ML}} = \frac{m}{N}$$

Summaries

- We have seen that the joint probability of two independent events is given by the product of the marginal probabilities for each event separately.

$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu,\sigma^2\right).$$

- In terms of parameters, this function is called likelihood function

Summaries

- One common criterion for determining the parameters in a probability distribution using an observed data set is to find the parameter values that maximize the likelihood function.

- This might seem like a strange criterion because, from our foregoing discussion of probability theory, it would seem more natural to maximize the probability of the parameters given the data, not the probability of the data given the parameters..

Challenges with Bernoulli and maximizing likelihood

- The probability of landing heads is given, in this maximum likelihood framework, by the fraction of observations of heads in the data set.

- Example: $$\mathcal{D} = \{1, 1, 1\} \rightarrow \mu_{\mathrm{ML}} = \frac{3}{3} = 1$$

- Prediction: *all* future tosses will land heads up

Overfitting to D

- Possible solution: Prior distribution over parameter

Binary Variables and Binomial Distribution

- N coin flips:

$$p(m \text{ heads}|N, \mu)$$

- Binomial Distribution

$$\text{Bin}(m|N, \mu) = \binom{N}{m} \mu^m (1-\mu)^{N-m}$$

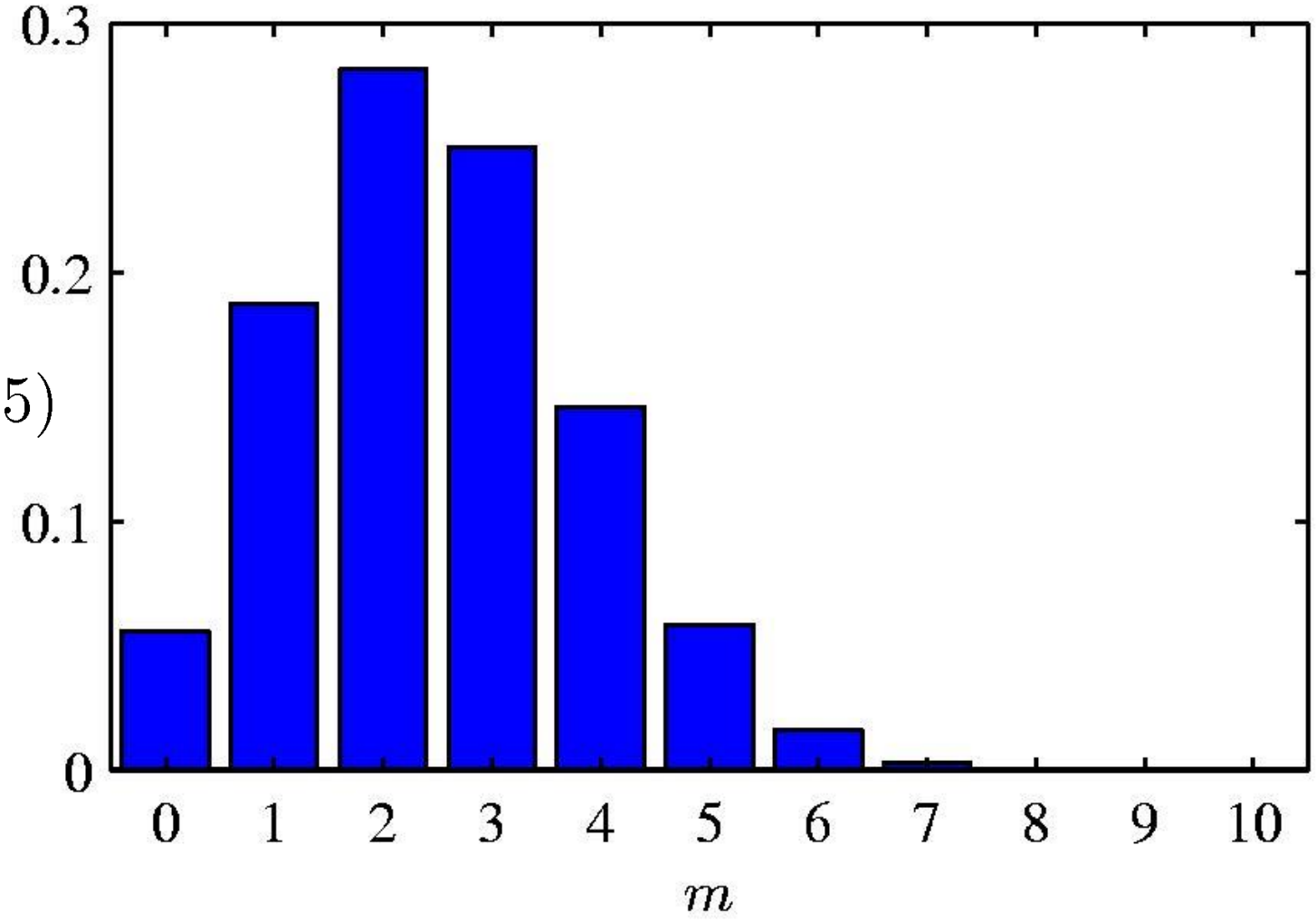
$$\binom{N}{m} \equiv \frac{N!}{(N-m)!m!}$$

$$\mathbb{E}[m] \equiv \sum_{m=0}^{N} m\text{Bin}(m|N, \mu) = N\mu$$

$$\text{var}[m] \equiv \sum_{m=0}^{N} (m - \mathbb{E}[m])^2 \text{Bin}(m|N, \mu) = N\mu(1-\mu)$$

Example: Binary variables and Binomial Distribution

$\mathrm{Bin}(m|10, 0.25)$



Next time: Multinomial Distribution

Thank You

UPES

UNIVERSITY WITH A PURPOSE

Binary Variables and Beta Distribution

- Distribution over

$$\mu \in [0, 1]$$

$$\begin{aligned} \mathrm{Beta}(\mu|a,b) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \\ \mathbb{E}[\mu] &= \frac{a}{a+b} \\ \mathrm{var}[\mu] &= \frac{ab}{(a+b)^2(a+b+1)} \end{aligned}$$

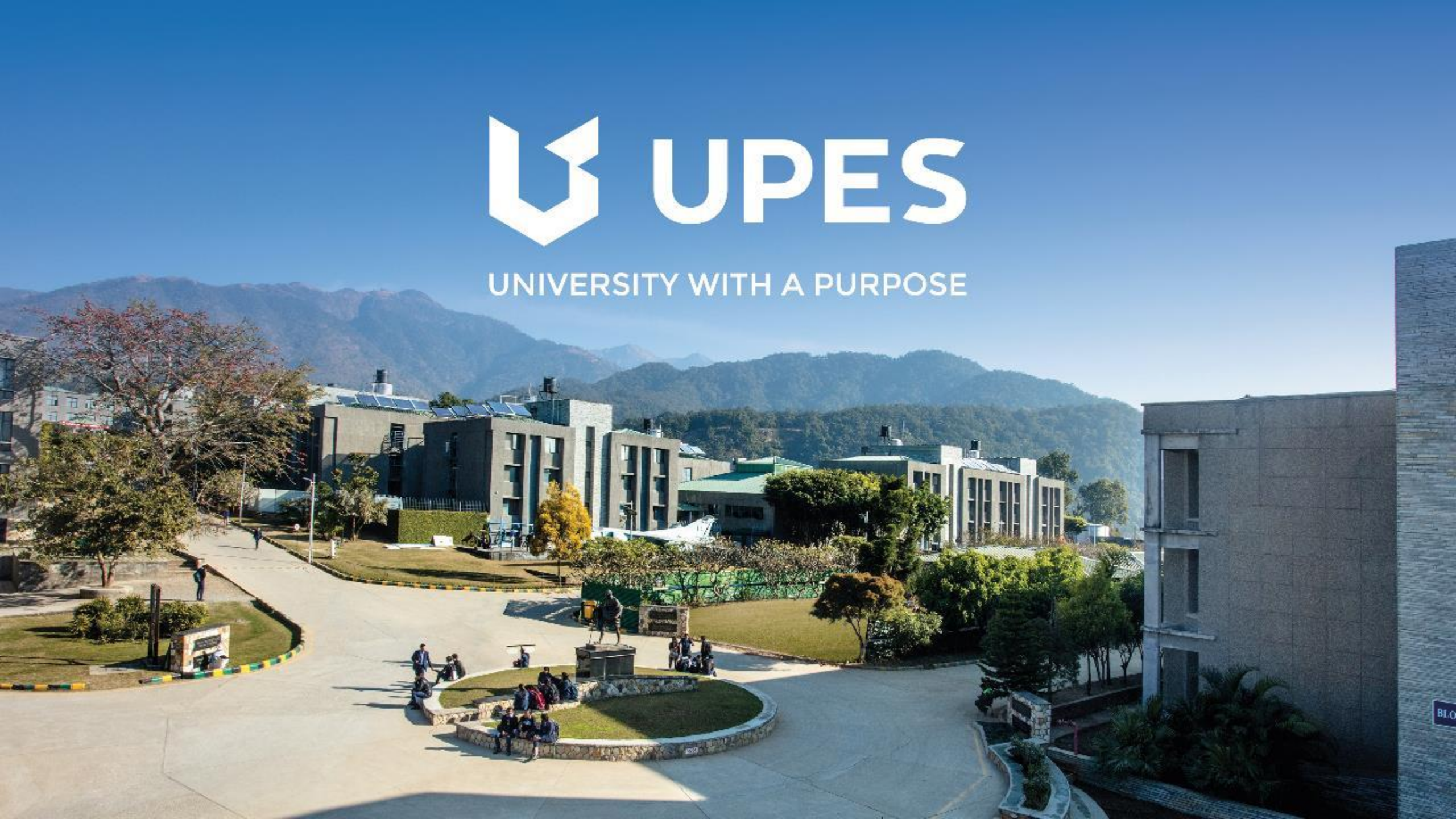
$$\Gamma(x) \equiv \int_0^\infty u^{x-1}e^{-u}\,\mathrm{d}u.$$

Beta function

$$
\begin{aligned}
p(\mu|a_0, b_0, \mathcal{D}) &\propto p(\mathcal{D}|\mu)p(\mu|a_0, b_0) \\
&= \left(\prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n} \right) \text{Beta}(\mu|a_0, b_0) \\
&\propto \mu^{m+a_0-1}(1-\mu)^{(N-m)+b_0-1} \\
&\propto \text{Beta}(\mu|a_N, b_N)
\end{aligned}
$$

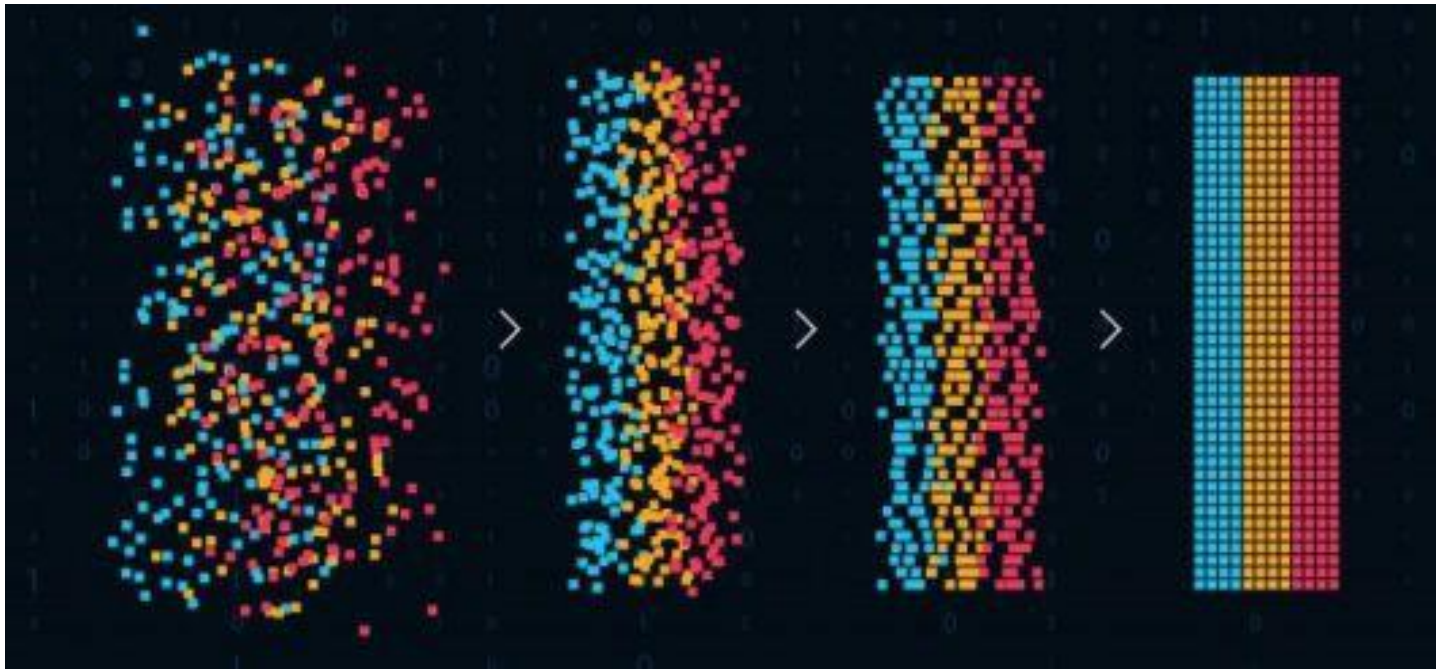
$$a_N = a_0 + m \qquad b_N = b_0 + (N - m)$$

The Beta distribution provides the *conjugate* prior for the Bernoulli distribution.


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/09/2021

Recap: Types

- Discrete random variables: a countable number of possible values.
- continuous random variables: Takes all values in a given interval of numbers.
- Parametric distributions: Governed by adaptive parameters (mean, variance etc.).
  - Binomial and Multinomial distribution for discrete random variables
  - Gaussian distribution for continuous random variables
- In order to apply these distributions for density estimation, need to determine suitable value of these parameters given an observed data set
  - If frequenist treatment: determine parameters by optimizing likelihood function
  - In Bayesian treatment: posterior probabilities of parameters from prior probabilities of parameters conditioned to observed data

Binary Variables: Beta Distribution

- Overfitting with Binomial (frequist treatment)
- Let's go for Bayesian treatment
- We need prior distribution in the form proportional to μ and $(1-\mu)$,
- So that posterior should be a product of factors of form $\mu^x(1-\mu)^{1-x}$.

- Therefore we choose Beta distribution as prior distribution
- It will assure ***conjugacy***

Binary Variables: Beta Distribution

- Distribution over

$$\mu \in [0, 1]$$

$$\int_0^1 \mathrm{Beta}(\mu|a,b)\,\mathrm{d}\mu = 1.$$

$$\begin{aligned}
\mathrm{Beta}(\mu|a,b) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \\
\mathbb{E}[\mu] &= \frac{a}{a+b} \\
\mathrm{var}[\mu] &= \frac{ab}{(a+b)^2(a+b+1)}
\end{aligned}$$

$$\Gamma(x) \equiv \int_0^\infty u^{x-1}e^{-u}\,\mathrm{d}u.$$

Binary Variables: Posterior

Posterior = likelihood*prior

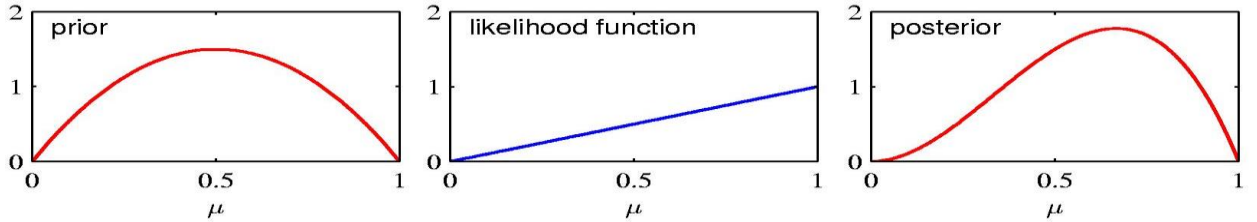
- Proportionality analogous to prior

$$\begin{aligned} p(\mu|a_0, b_0, \mathcal{D}) &\propto p(\mathcal{D}|\mu)p(\mu|a_0, b_0) \\ &= \left(\prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n}\right) \mathrm{Beta}(\mu|a_0, b_0) \\ &\propto \mu^{m+a_0-1}(1-\mu)^{(N-m)+b_0-1} \\ &\propto \mathrm{Beta}(\mu|a_N, b_N) \end{aligned}$$

- Posterior

$$p(\mu|m, l, a, b) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\mu^{m+a-1}(1-\mu)^{l+b-1}.$$

- Effective number of observations
- Sequential method



Binary Variables: Posterior: Properties

As the size of the data set, N, increase

$$
\begin{aligned}
a_N &\rightarrow m \\
b_N &\rightarrow N - m \\
\mathbb{E}[\mu] &= \frac{a_N}{a_N + b_N} \rightarrow \frac{m}{N} = \mu_{\mathrm{ML}} \\
\mathrm{var}[\mu] &= \frac{a_N b_N}{(a_N + b_N)^2 (a_N + b_N + 1)} \rightarrow 0
\end{aligned}
$$

Posterior varies between prior and maximum likelihood

Binary Variables:Prediction under the Posterior

What is the probability that the next coin toss will land heads up?

$$
\begin{aligned}
p(x=1|a_0,b_0,\mathcal{D}) &= \int_0^1 p(x=1|\mu)p(\mu|a_0,b_0,\mathcal{D})\,\mathrm{d}\mu \\
&= \int_0^1 \mu p(\mu|a_0,b_0,\mathcal{D})\,\mathrm{d}\mu \\
&= \mathbb{E}[\mu|a_0,b_0,\mathcal{D}] = \frac{a_N}{b_N}
\end{aligned}
$$

Next time: Multinomial Distribution

Thank You

UPES

UNIVERSITY WITH A PURPOSE

Multinomial Variables

Variable with K states

- 1-of-K coding scheme:

$$\mathbf{x} = (0, 0, 1, 0, 0, 0)^{\mathrm{T}}$$

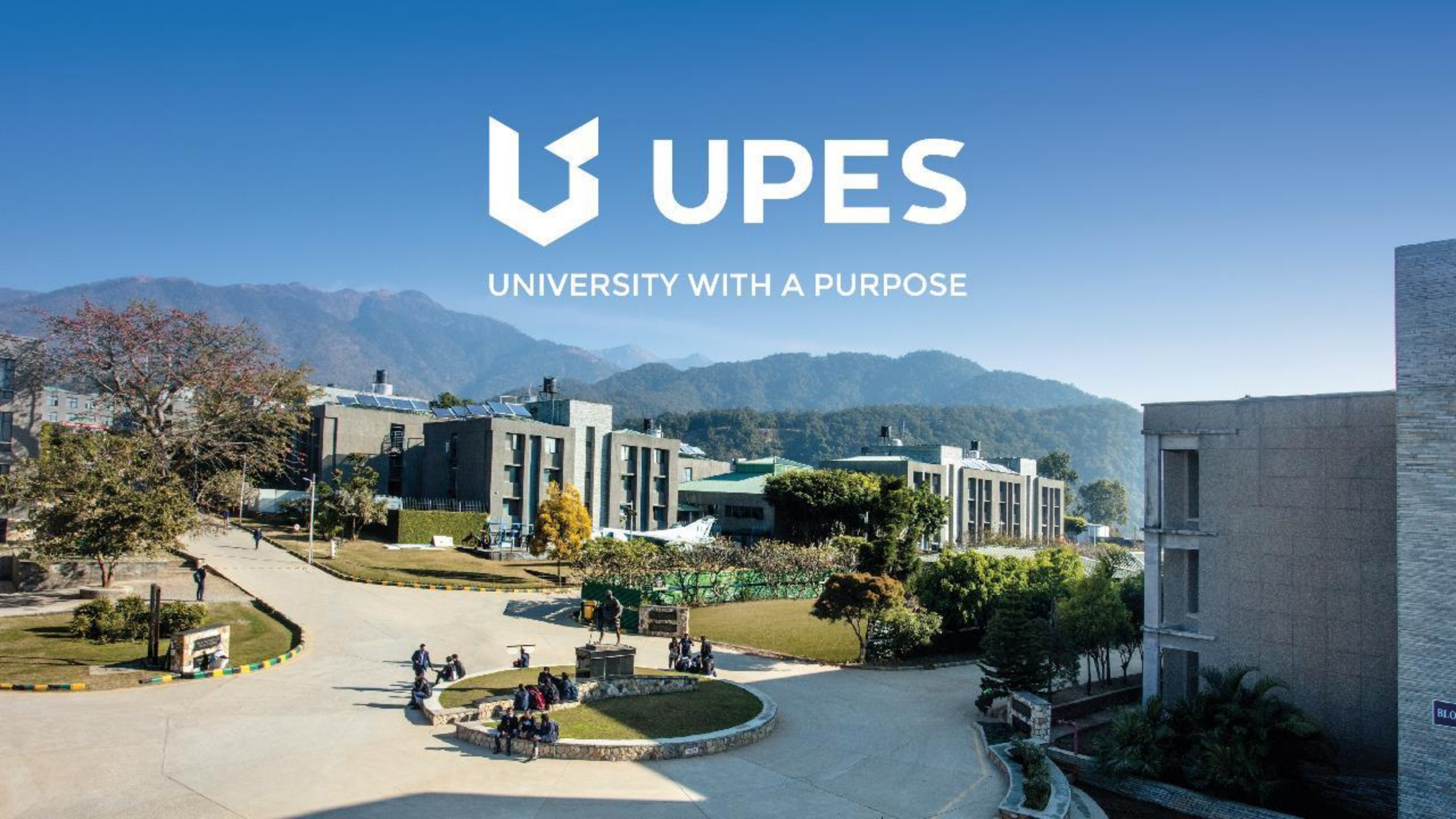
$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$$

$$\forall k : \mu_k \geqslant 0 \quad \text{and} \quad \sum_{k=1}^{K} \mu_k = 1$$

$$\mathbb{E}[\mathbf{x}|\boldsymbol{\mu}] = \sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu})\mathbf{x} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}} = \boldsymbol{\mu}$$

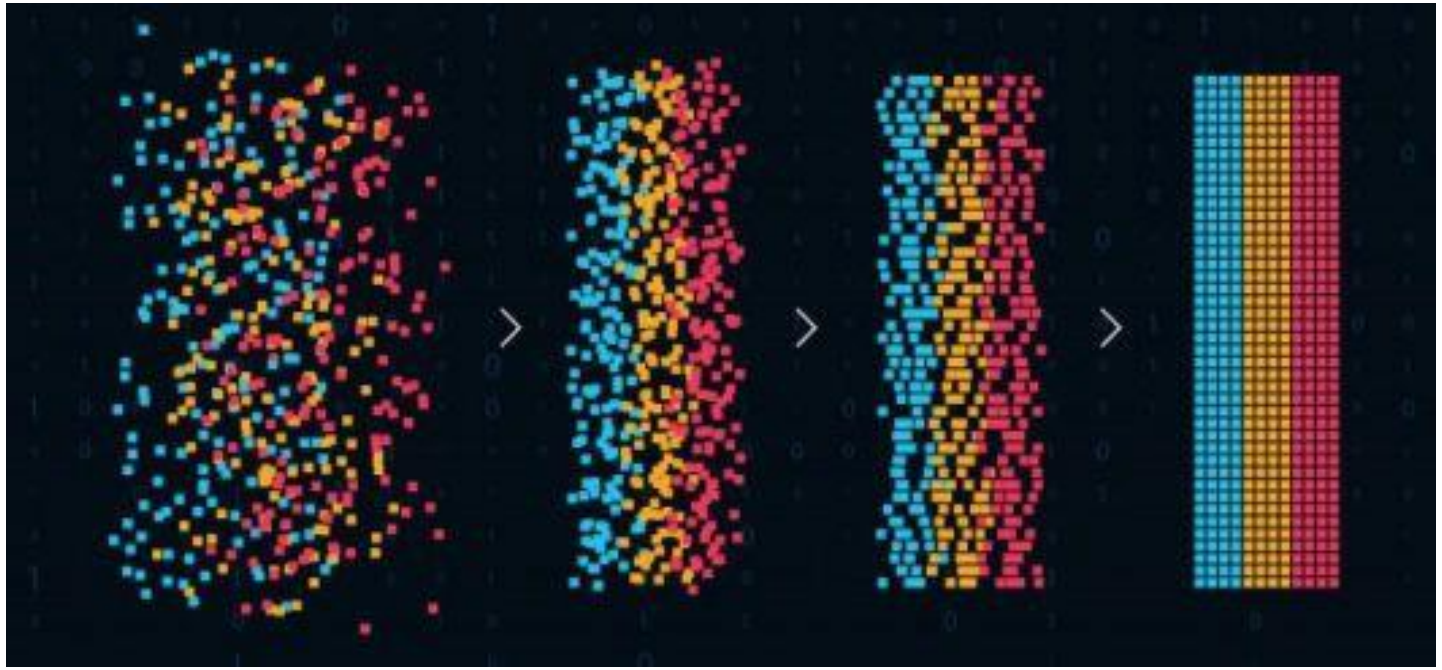
$$\sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu}) = \sum_{k=1}^{K} \mu_k = 1$$

Multinomial Variables


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/09/2021

Recap: Binary Variables

Discrete binary random variable

Two states (True and False)

Popular Distributions

- *Bernoulli*
- *Binomial (Frequenist)*
- *Beta (Bayesian)*

Multinomial Variables

Variable with K states

- 1-of-K coding scheme: $\mathbf{x} = (0, 0, 1, 0, 0, 0)^{\mathrm{T}}$
- Probability distribution $p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$ Generalized version of Bernoulli
- Under constraints $\forall k : \mu_k \geqslant 0 \quad \text{and} \quad \sum_{k=1}^{K} \mu_k = 1$
- Expectation and variance

$$\mathbb{E}[\mathbf{x}|\boldsymbol{\mu}] = \sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu})\mathbf{x} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}} = \boldsymbol{\mu}$$

Multinomial Variables: Parameter Estimation

- Given $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

$$p(\mathcal{D}|\boldsymbol{\mu}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mu_k^{x_{nk}} = \prod_{k=1}^{K}\mu_k^{\left(\sum_n x_{nk}\right)} = \prod_{k=1}^{K}\mu_k^{m_k}$$

- Ensure $\sum_k \mu_k = 1$, use a Lagrange multiplier, λ.

$$\sum_{k=1}^{K} m_k \ln \mu_k + \lambda\left(\sum_{k=1}^{K}\mu_k - 1\right)$$

$$\mu_k = -m_k/\lambda \qquad \mu_k^{\mathrm{ML}} = \frac{m_k}{N}$$

Multinomial Variables: Multinomial Distribution

Joint distribution of $(m_1, m_2, \ldots, m_K$

$$\begin{aligned}
\mathrm{Mult}(m_1, m_2, \ldots, m_K|\boldsymbol{\mu}, N) &= \binom{N}{m_1 m_2 \ldots m_K} \prod_{k=1}^{K} \mu_k^{m_k} \\
\mathbb{E}[m_k] &= N\mu_k \\
\mathrm{var}[m_k] &= N\mu_k(1-\mu_k) \\
\mathrm{cov}[m_j m_k] &= -N\mu_j\mu_k
\end{aligned}$$

$$\binom{N}{m_1 m_2 \ldots m_K} = \frac{N!}{m_1! m_2! \ldots m_K!}.$$

$$\sum_{k=1}^{K} m_k = N.$$

Multinomial Variables: Dirichlet Distribution

Bayesian treatment

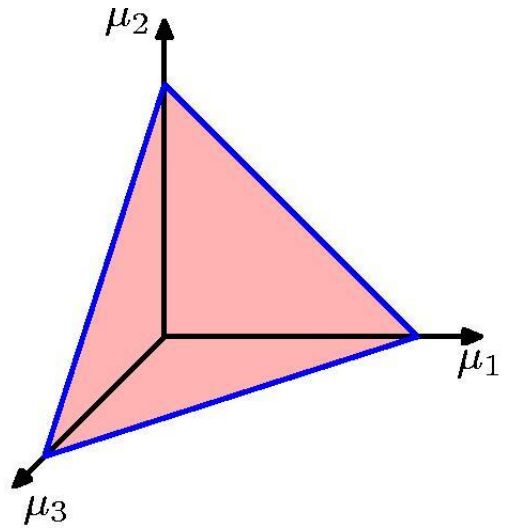
$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)}\prod_{k=1}^{K}\mu_k^{\alpha_k-1} \qquad \alpha_0 = \sum_{k=1}^{K}\alpha_k$$

A family of priors for $\{\mu_k\}$

Parameters of the distribution

$$\alpha_1, \ldots, \alpha_K$$

Conjugate prior for the multinomial distribution.



Simplex (bounded linear mainfold) of dimensionality K-1.

Multinomial Variables: Dirichlet Distribution

Multiplying likelihood with prior

Posterior =

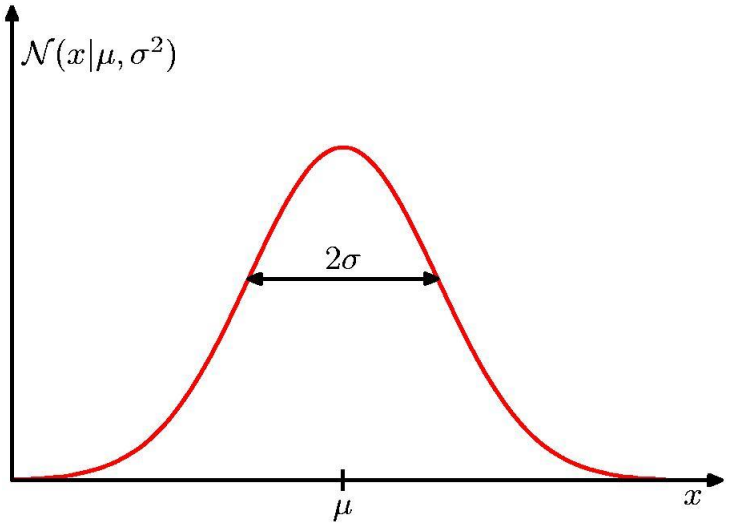
$$p(\boldsymbol{\mu}|\mathcal{D},\boldsymbol{\alpha}) \propto p(\mathcal{D}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k+m_k-1}$$

$$\begin{aligned} p(\boldsymbol{\mu}|\mathcal{D},\boldsymbol{\alpha}) &= \mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}+\mathbf{m}) \\ &= \frac{\Gamma(\alpha_0+N)}{\Gamma(\alpha_1+m_1)\cdots\Gamma(\alpha_K+m_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k+m_k-1} \end{aligned}$$

Two-state (Binary) variables: Either via binomial and Beta or
With Multinomial and Dirichlet (1 of 2 scenario).

Continuous Variables: Gaussian Distribution

- Widely used model for the distribution of continuous variables
- Also known as Normal distribution

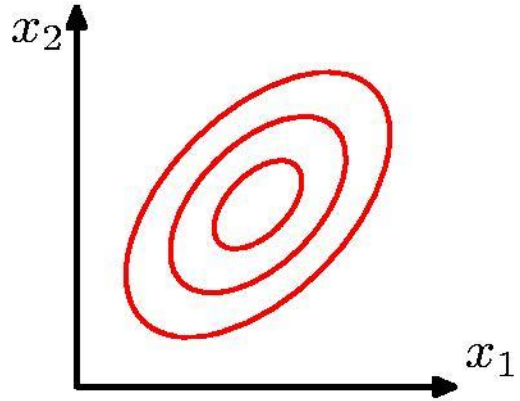


- For single variable

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

- Multivariate Gaussian distribution

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

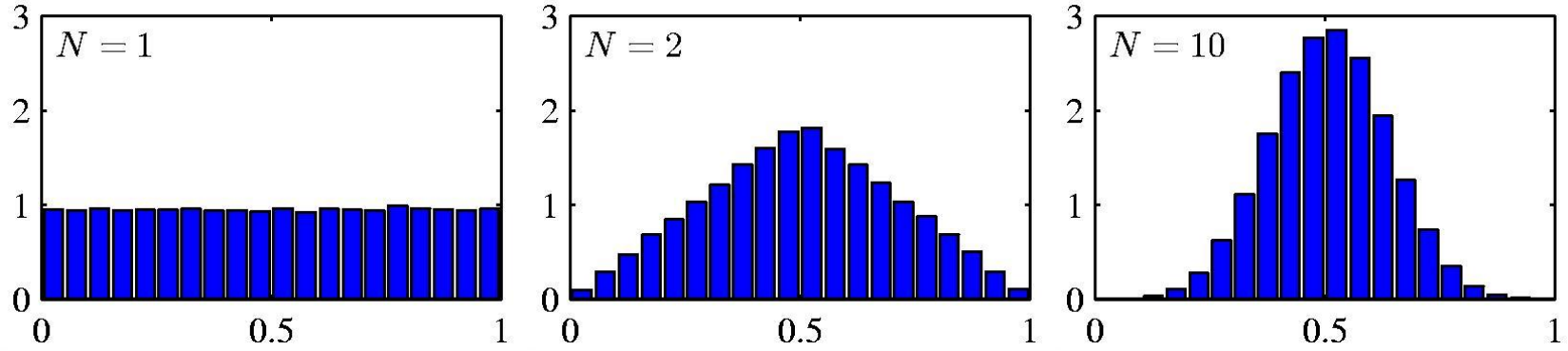


Continuous Variables: Gaussian Distribution

- For singe or multiple continuous variable, the distribution that maximizes the entropy is the **Gaussian.**

- The distribution of a random variable (which itself is a sum of multiple random variables) tends to be Gaussian as the number of number of variables summing up increases.

Example: N uniform [0,1] random variables



Continuous Variables: Gaussian Distribution

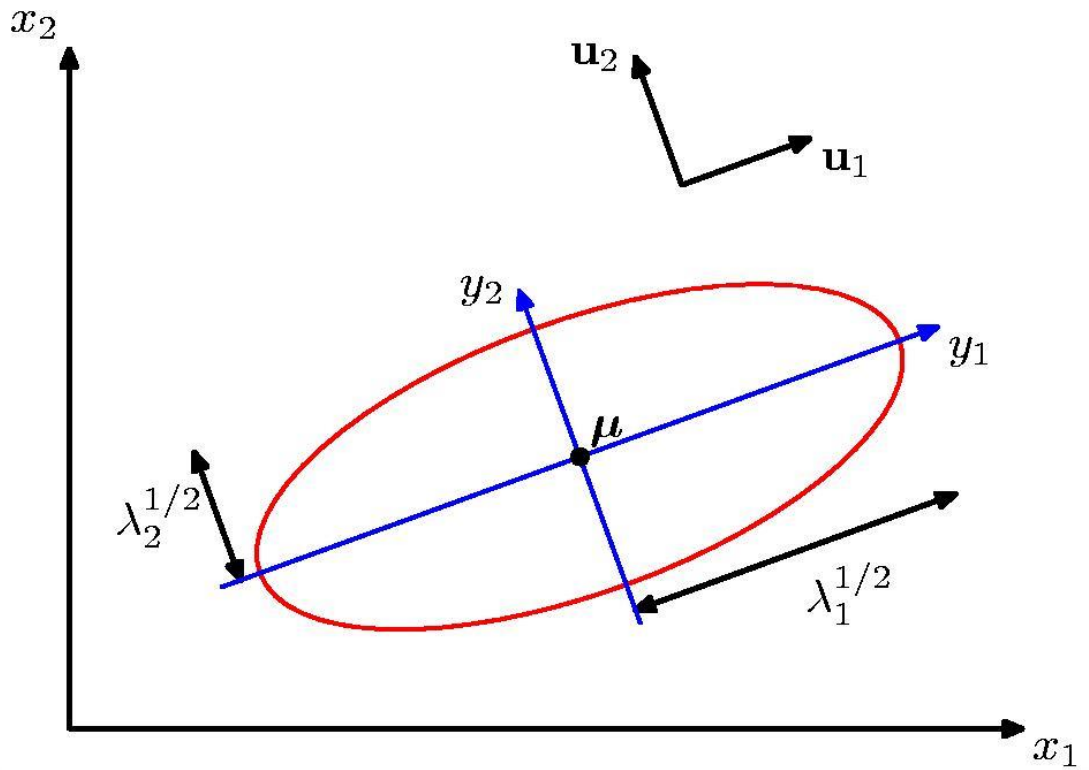
- Gaussian distribution has important analytical properties
  - Geometrical form interpretation: Δ is ***Mahalonobis distance***

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{D} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}}$$

$$\Delta^2 = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

$$y_i = \mathbf{u}_i^{\mathrm{T}} (\mathbf{x} - \boldsymbol{\mu})$$



Non-Parametric

- Parametric distribution models are restricted to specific forms, which may not always be suitable; for example, consider modelling a multimodal distribution with a single, unimodal model.

- Nonparametric approaches make few assumptions about the overall shape of the distribution being modelled.

- We will focus on frequentist treatment however Bayesian treatment is also interesting.

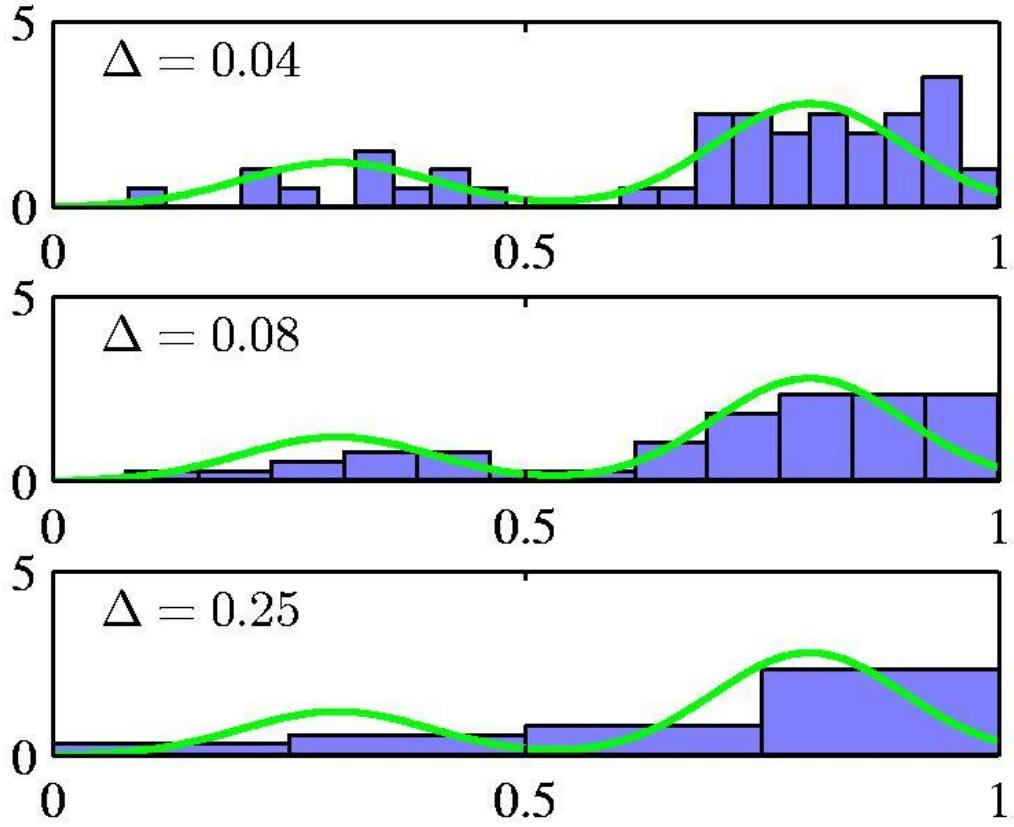
Histogram methods(histogram density models)

Single continuous variable x

Histogram methods partition the data space into distinct bins with widths $\dot{\Delta}_i$ and count the number of observations, n_i, in each bin.

$$p_i = \frac{n_i}{N\Delta_i}$$

- Often, the same width is used for all bins, $\dot{\Delta}_i = \dot{\Delta}$.
- ¢ acts as a smoothing parameter.



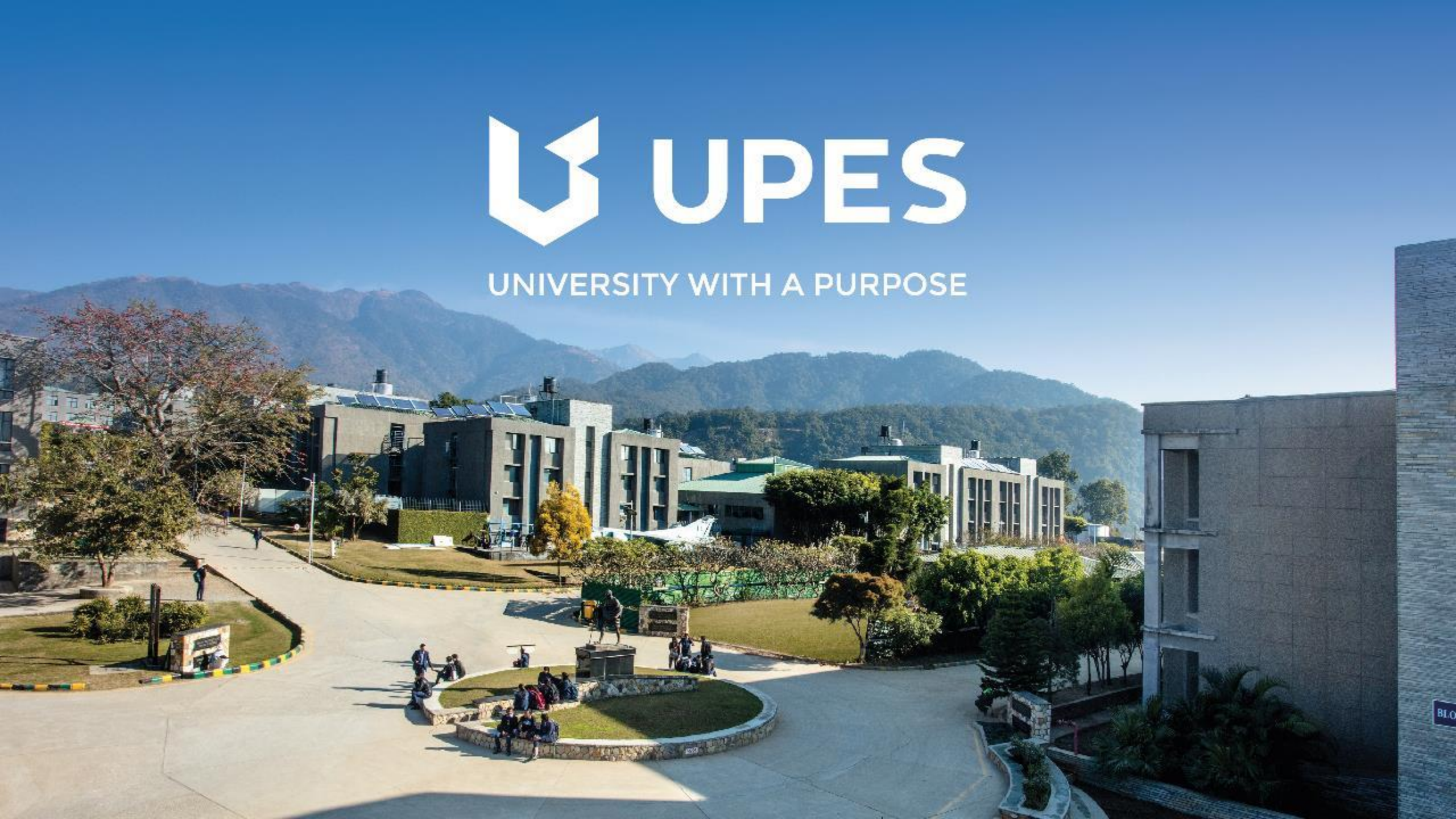
•In a D-dimensional space, using M bins in each dimen-sion will require M^D bins!

Next time: Non-parametric methods

Thank You

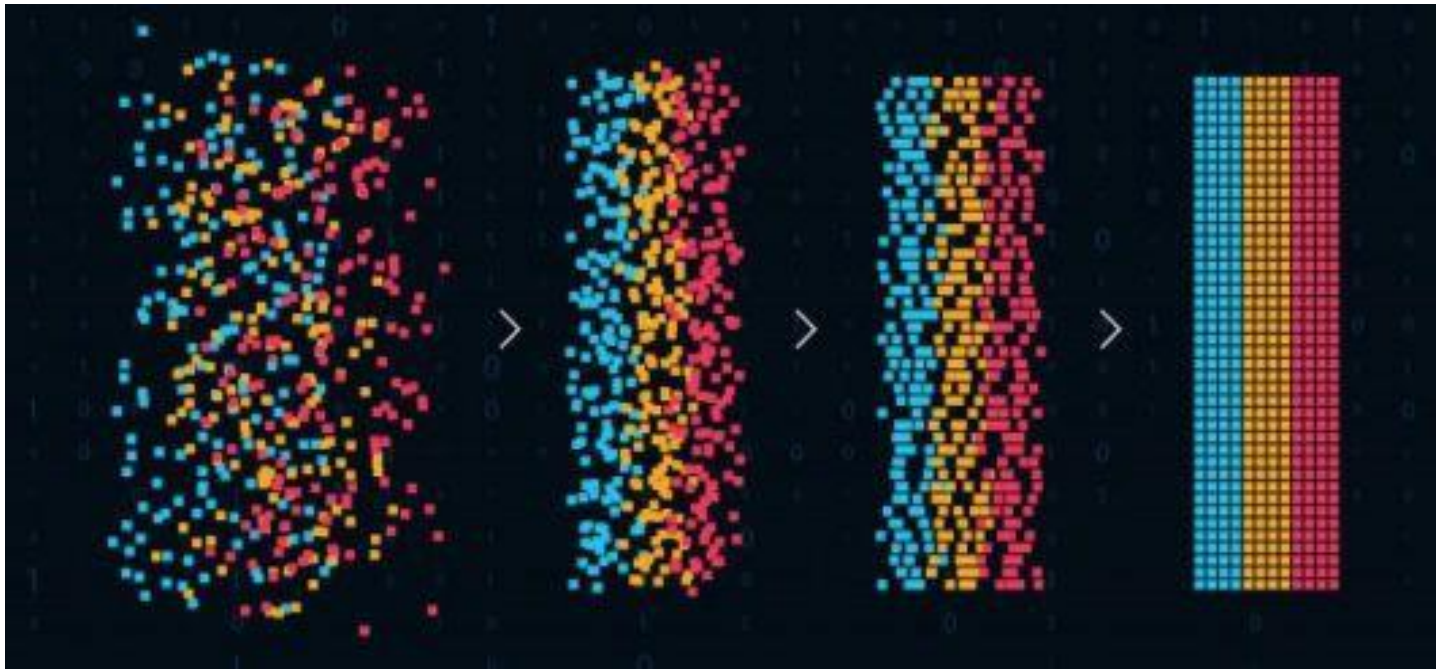
UPES

UNIVERSITY WITH A PURPOSE


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/09/2021

Linear Models

- What are linear models ? Linear functions of adjustable parameters

- Linear models:
  - Linear function of input
  - Non-linear function of input

- Can be used for
  - Regression
  - Classification

Linear Models for Regression

- **Given**: Dataset with N observations {**x**, t}
- **Goal**: Predict the value of t (real valued) for new value of **x**.
- Intuition: formulate y = f(**x**, w) such that for a given **x**, y coincides with t.

- **Simplest form of linear regression**: Linear function of both input variables and adjustable parameters

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$$

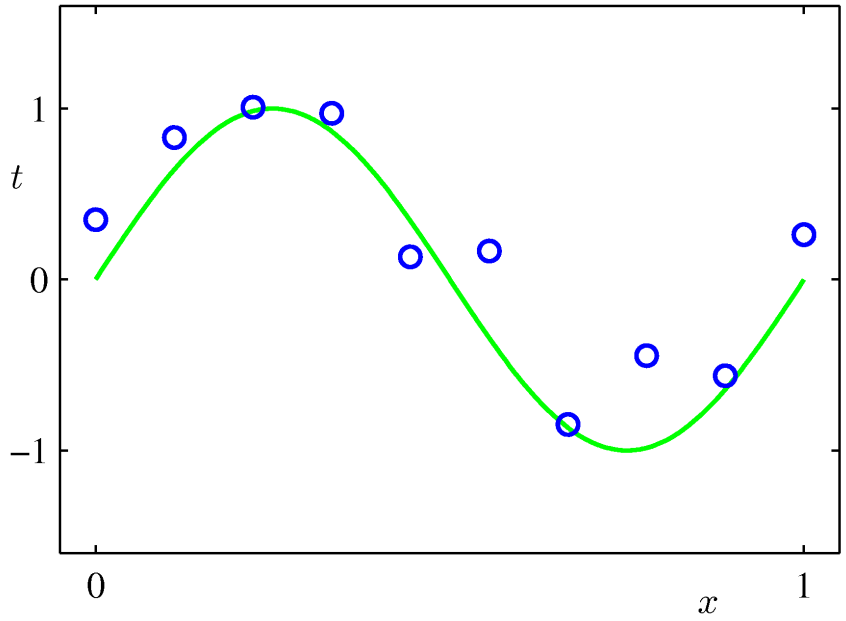
- In vector form?

Basis Functions based Linear Models for Regression

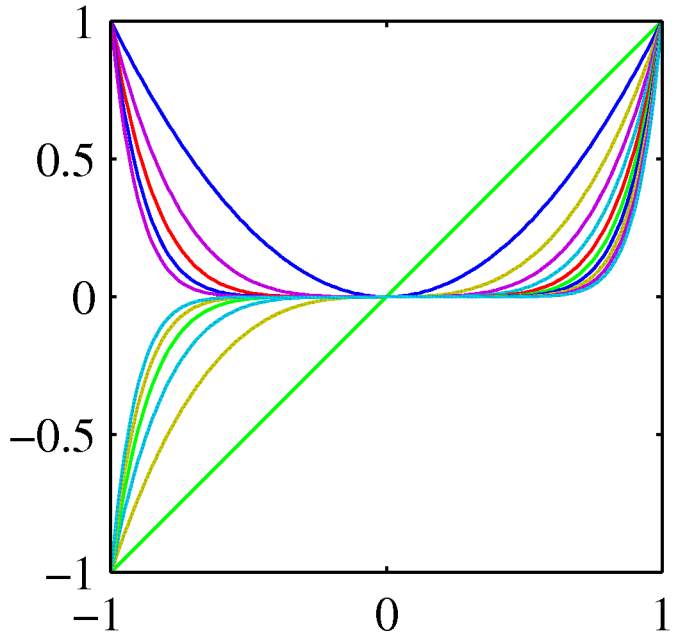
- Example: Polynomial Regression (Non-linear basis functions)
- Polynomial basis functions

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Limitation of polynomial basis function:
  - Global impact
- Spline functions: Dividing input space into regions and using different polynomials for different regions

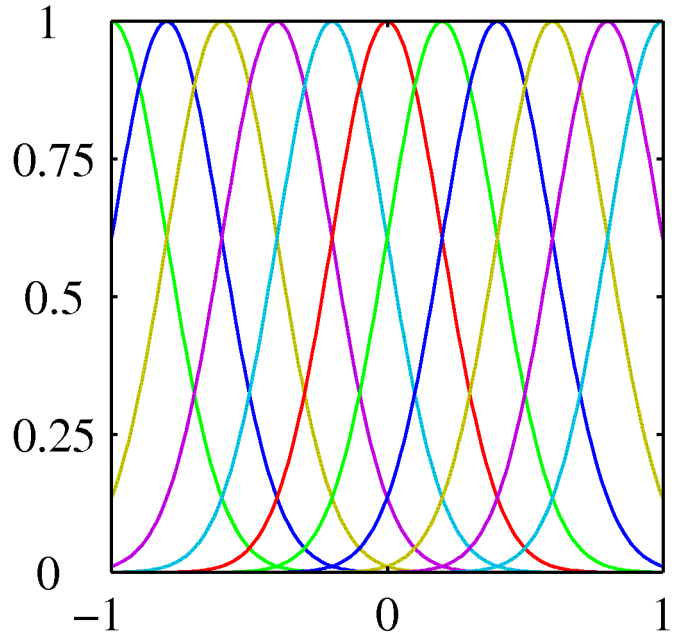


Basis Function: Examples



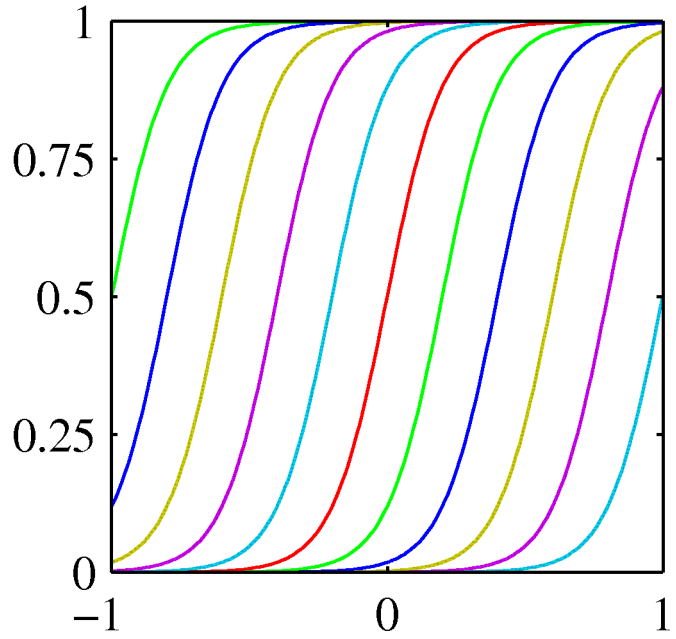
$$\phi_j(x) = x^j.$$

Polynomial basis functions



$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

Gaussian basis functions



$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$$

where $\sigma(a) = \dfrac{1}{1+\exp(-a)}.$

Sigmoidal basis functions

Basis Function: More Examples

- Fourier basis: Expansion in sinusoidal functions: Each basis function represents a specific frequency and has infinite spatial extent. By contrast, basis functions that are localized to finite regions of input space necessarily comprise a spectrum of different spatial frequencies

- Wavelets: Class of basis functions that are localized in both space and frequency. Mutually orthogonal

…….and many more……

LiMod for Reg: Ways to build model

- **Standard**: Sum-of-squares error

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

- **Goal**: Minimize it
  - Differentiate it wrt **w** and equate to 0 to find **w**

LiMod for Reg: Ways to build model: Maximum Likelihood and Least Squares

- Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \text{where} \qquad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

- which is the same as saying,

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}).$$

- Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets, $\mathbf{t} = [t_1, \ldots, t_N]^{\mathrm{T}}$, we obtain the likelihood function

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

LiMod for Reg: Ways to build model: Maximum Likelihood and Least Squares

- Taking the logarithm, we get

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{w},\beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1}) \\ &= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w})\end{aligned}$$

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi).$$

Refer equation 1.54 of section 1.2.4 of Bishop book

- where

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$ is the sum-of-squares error.

Maximization of the likelihood function under a conditional Gaussian noise distribution for a linear model is equivalent to minimizing a sum-of-squares error function

LiMod for Reg: Ways to build model: Maximum Likelihood and Least Squares

- Computing the gradient and setting it to zero yields

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

- Solving for w, we get

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}$$

The Moore-Penrose pseudo-inverse, $\boldsymbol{\Phi}^{\dagger}$.

Helpful for non-square matrix

- This equation is also known as normal equations for least squares problem

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Design Matrix

LiMod for Reg: Ways to build model: Maximum Likelihood and Least Squares

- Maximizing with respect to the **bias**, w_0, alone, we see that

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \overline{\phi_j}$$

$$= \frac{1}{N}\sum_{n=1}^{N} t_n - \sum_{j=1}^{M-1} w_j \frac{1}{N}\sum_{n=1}^{N} \phi_j(\mathbf{x}_n).$$

- We can also maximize with respect to ß (**precision**), giving

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

Bias-Variance trade-off

Sequential Learning

- **Context**: batch techniques (ex. ML) takes all the training data in one go.
- This increases the computational cost and also dependency on presence of whole data at once.

- In sequential learning: Data items considered one at a time (a.k.a. on-line learning); Ex.: **stochastic (sequential) gradient descent**:

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\ &= \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n))\boldsymbol{\phi}(\mathbf{x}_n). \end{aligned}$$

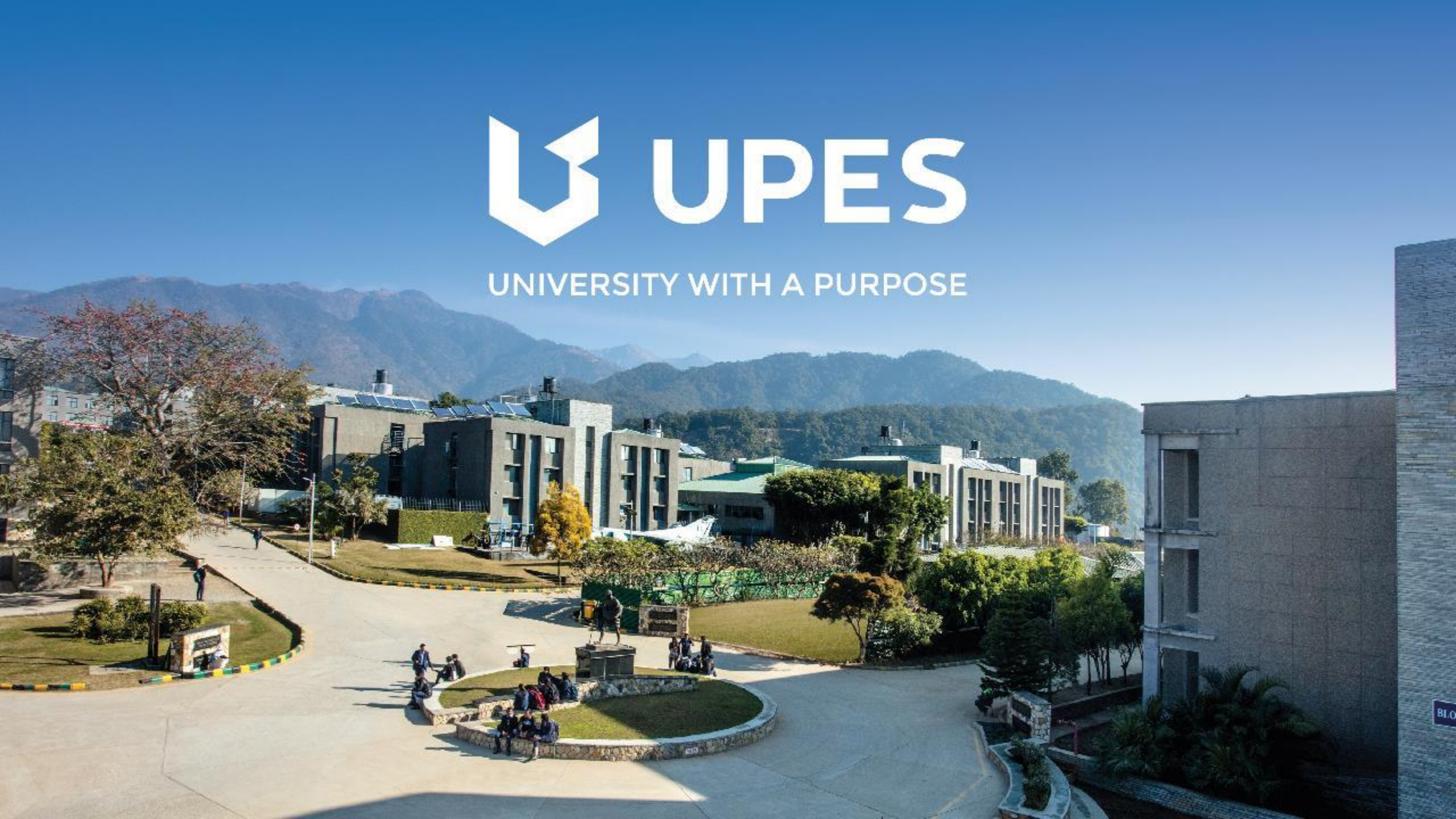
τ denotes the iteration number, and

η is a learning rate parameter

- For sum-of-squares error $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}_n)\boldsymbol{\phi}_n$
- This is known as the *least-mean-squares (LMS) algorithm*.

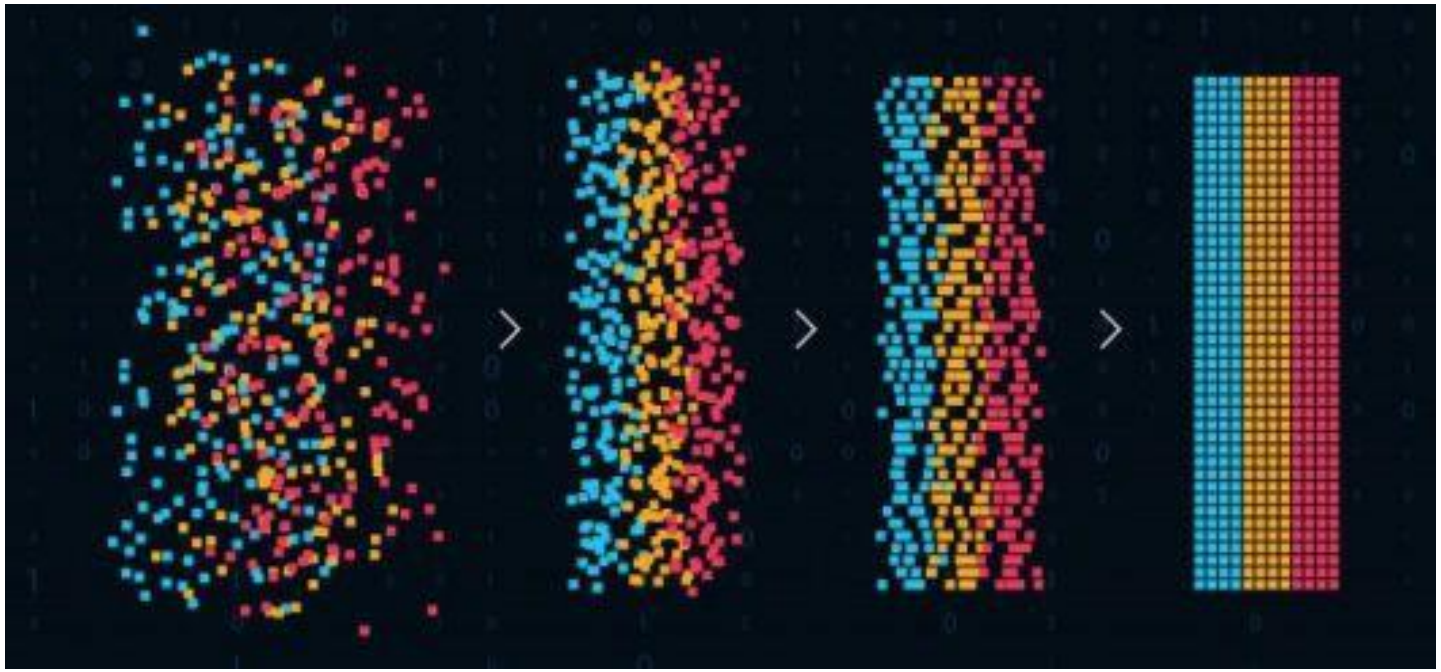
Next time: Regularized least squares

Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/09/2021

Sequential Learning

- **Context**: batch techniques (ex. ML) takes all the training data in one go.
- This increases the computational cost and also dependency on presence of whole data at once.

- In sequential learning: Data items considered one at a time (a.k.a. on-line learning); Ex.: **stochastic (sequential) gradient descent**:

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\ &= \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n))\boldsymbol{\phi}(\mathbf{x}_n). \end{aligned}$$

τ denotes the iteration number, and

η is a learning rate parameter

- For sum-of-squares error $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}_n)\boldsymbol{\phi}_n$
- This is known as the *least-mean-squares (LMS) algorithm*.

Regularized Least Squares

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- **Remember**: Weights being very large
- The idea of adding regularization term

- In general

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Weight Decay regularizer

In statistics: parameter shrinkage method

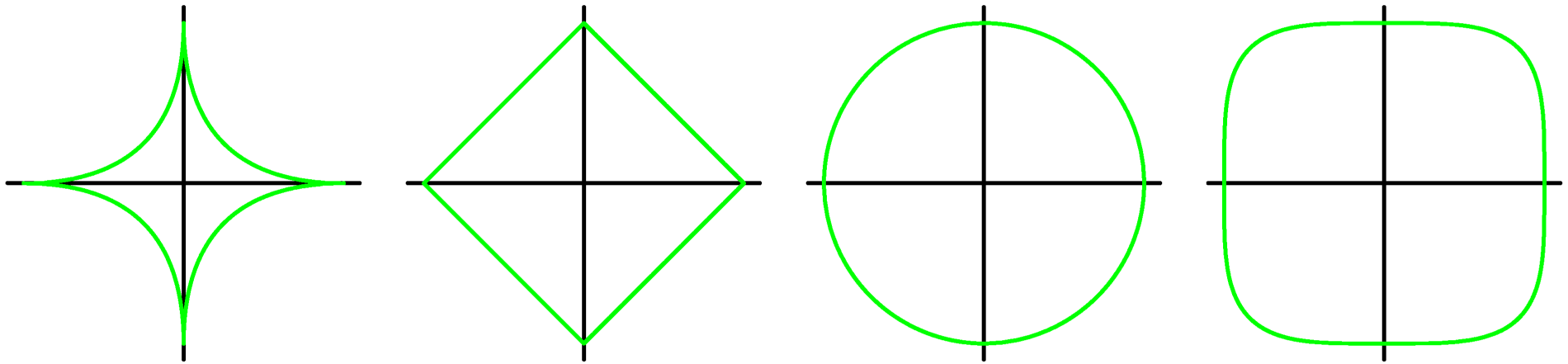
- Minimizing this yields

$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

Regularized Least Squares

- **More generic regularizer form**

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Lasso

Multiple Outputs

- Analogously to the multiple output case we have:

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})$$

$$\begin{aligned} p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) &= \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{W}, \mathbf{x}), \beta^{-1}\mathbf{I}) \\ &= \mathcal{N}(\mathbf{t}|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\mathbf{I}). \end{aligned}$$

- Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets, $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^{\mathrm{T}}$, we obtain the log likelihood function

$$\begin{aligned} \ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2}\ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2}\sum_{n=1}^{N}\left\|\mathbf{t}_n - \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\|^2. \end{aligned}$$

Multiple Outputs

- Maximizing with respect to W, we obtain

$$\mathbf{W}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{T}.$$

- If we consider a single target variable, t_k, we see that

$$\mathbf{w}_k = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}_k = \mathbf{\Phi}^{\dagger}\mathbf{t}_k$$

- where $\mathbf{t}_k = [t_{1k}, \ldots, t_{Nk}]^{\mathrm{T}}$, which is identical with the single output case.

Bias- Variance

- So far in linear models for regression: We have assumed that the form and number of basis functions are both fixed

Limiting the number of basis functions in order to avoid over-fitting

Vs

limiting the flexibility of the model to capture interesting and important trends in the data

- The introduction of regularization terms can control over-fitting for models with many parameters
- New question: Suitable values of these parameters

Bias- Variance

- Recall the *expected squared loss*,

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

- where

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

- The second term of E[L] corresponds to the noise inherent in the random variable t.
- What about the first term?

Bayesian Linear Regression

- Define a conjugate prior over w

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0).$$

•Combining this with the likelihood function and using results for marginal and conditional Gaussian distributions, gives the posterior

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

- where

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta \mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}. \end{aligned}$$

Bayesian Linear Regression

- A common choice for the prior is (zero-mean isotropic Gaussian)

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

for which

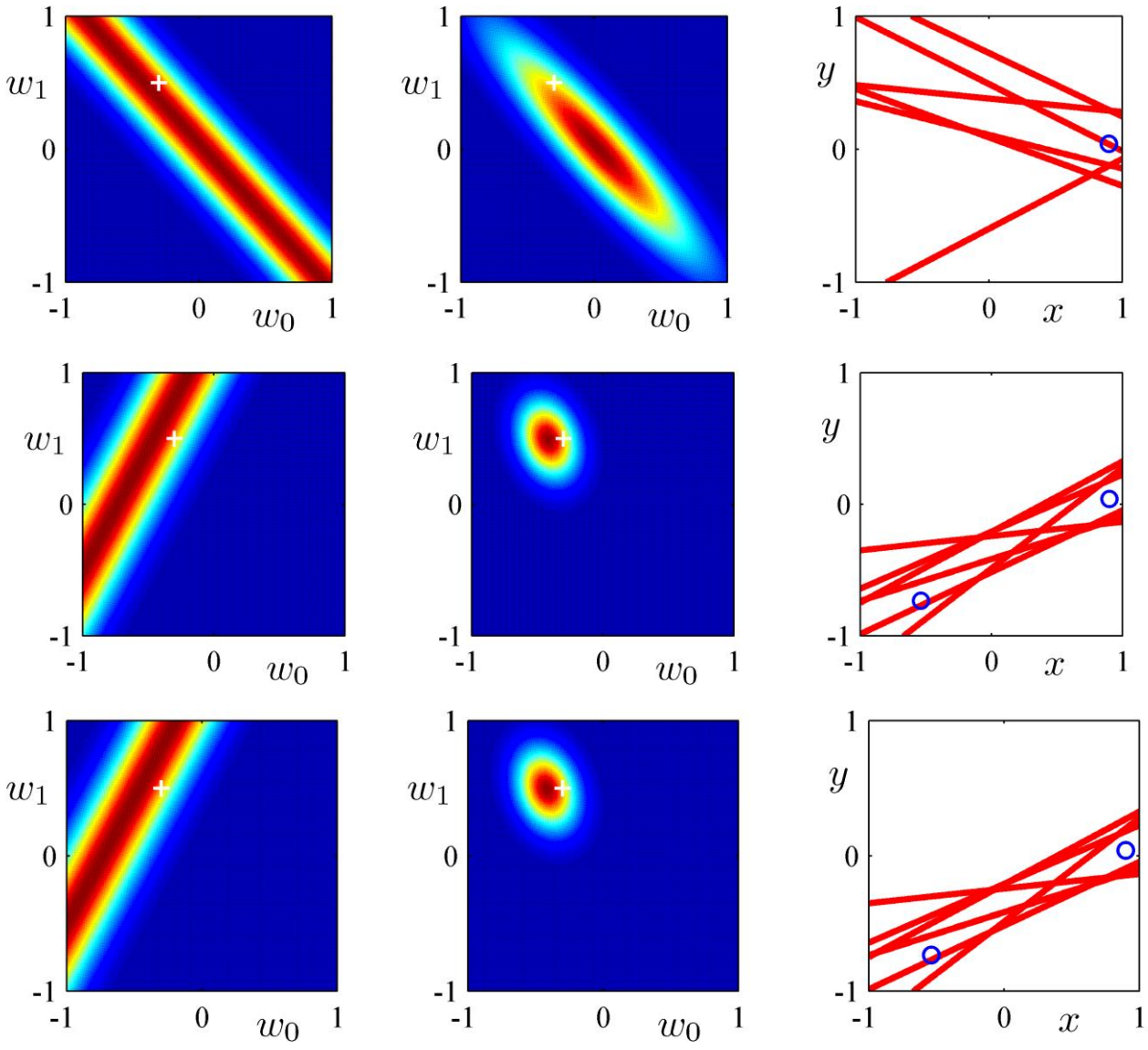
$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}. \end{aligned}$$

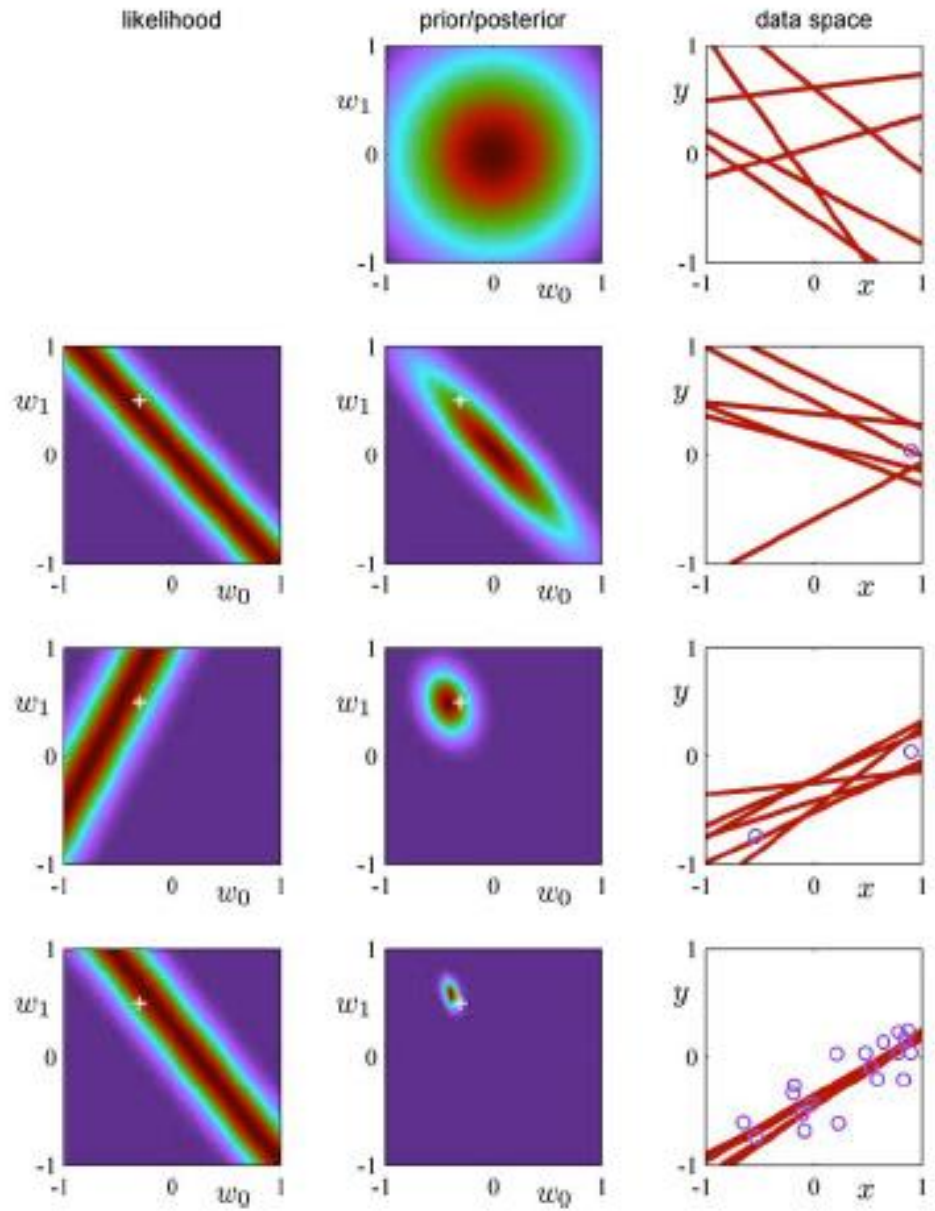
- Next we consider an example …

Likelihood

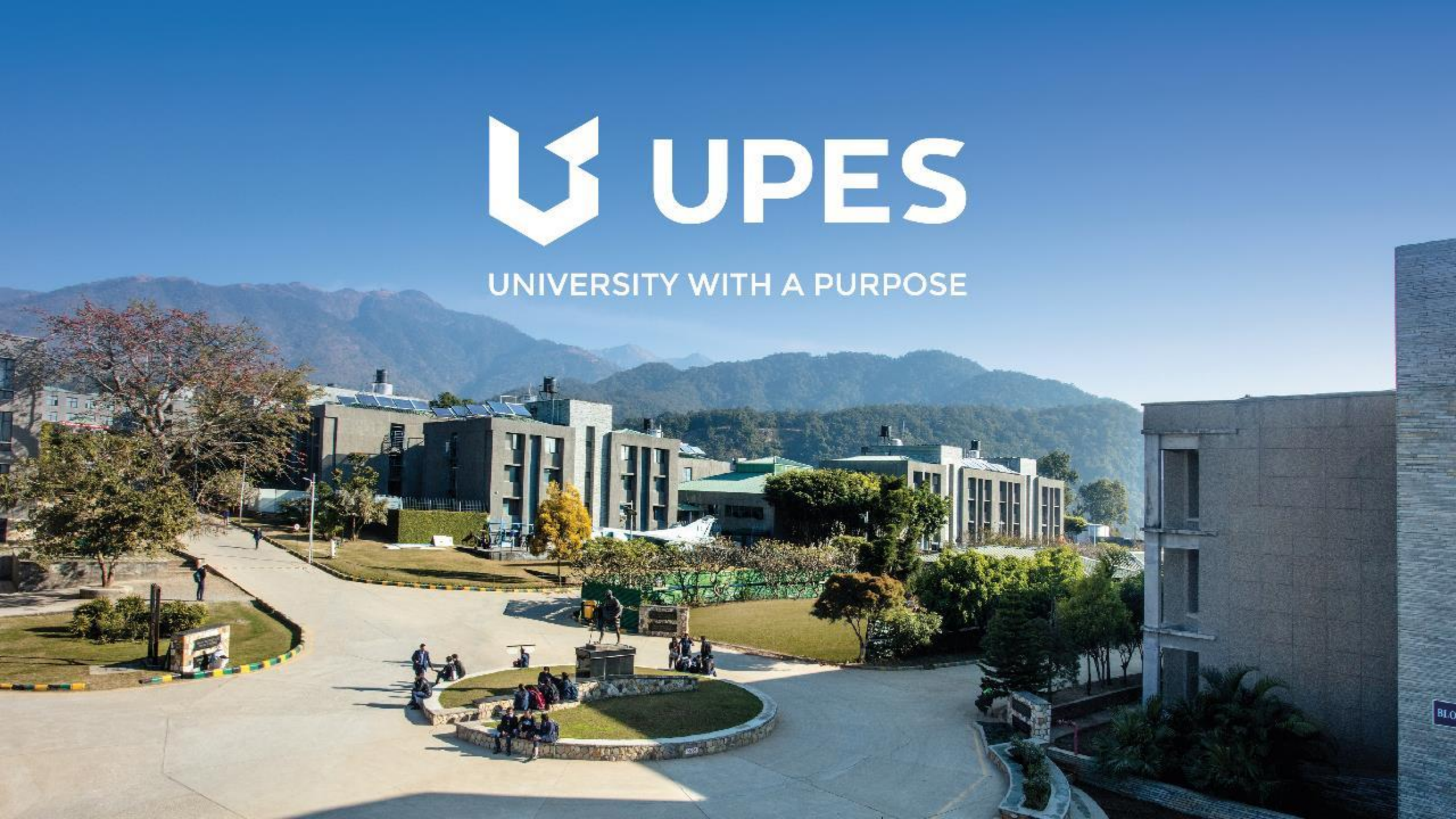
Posterior

Data Space



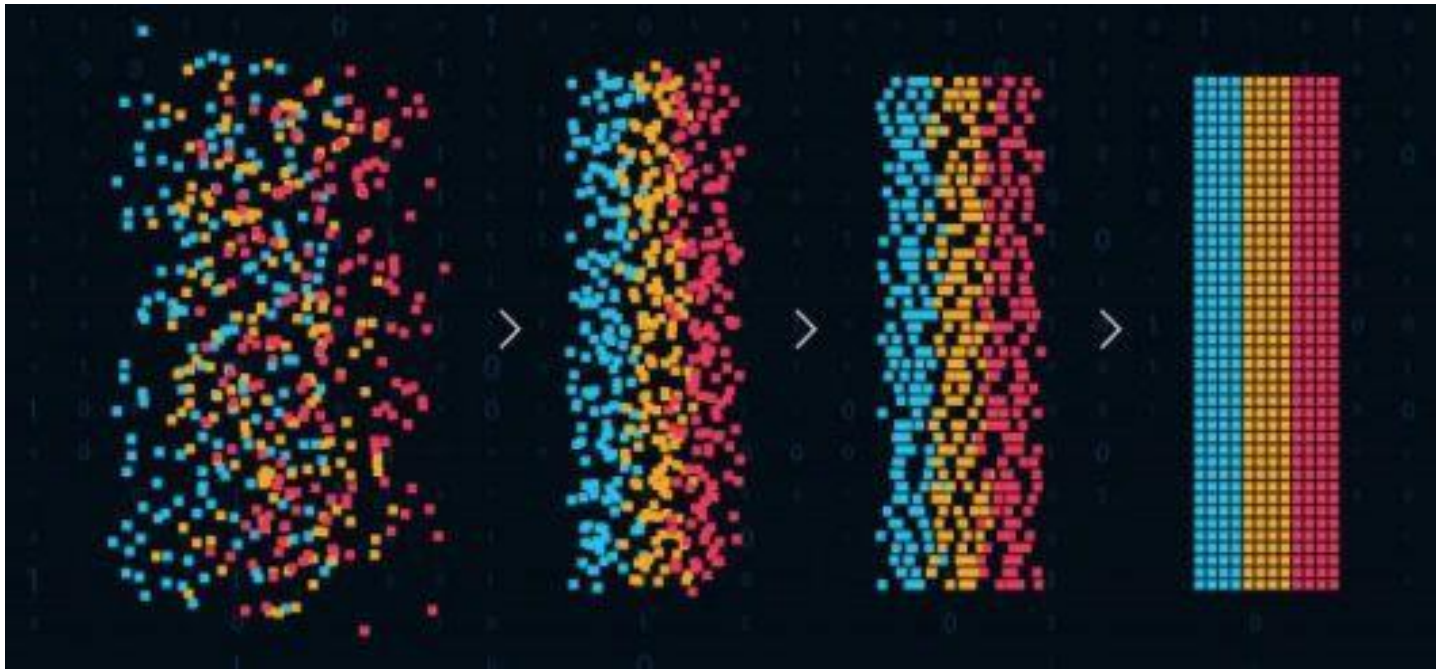


Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/09/2021

Bias- Variance

- So far in linear models for regression: We have assumed that the form and number of basis functions are both fixed

Limiting the number of basis functions in order to avoid over-fitting

Vs

limiting the flexibility of the model to capture interesting and important trends in the data

- The introduction of regularization terms can control over-fitting for models with many parameters
- New question: Suitable values of these parameters

Bias- Variance

- Recall the ***expected squared loss***,(not sum of squared error)

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

- where

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

- The second term of E[L] corresponds to the noise inherent in the random variable t.
- What about the first term?

Bias- Variance

$$\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-h(\mathbf{x})\}^2\right]$$
$$= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]-h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}.$$

- Thus we can write

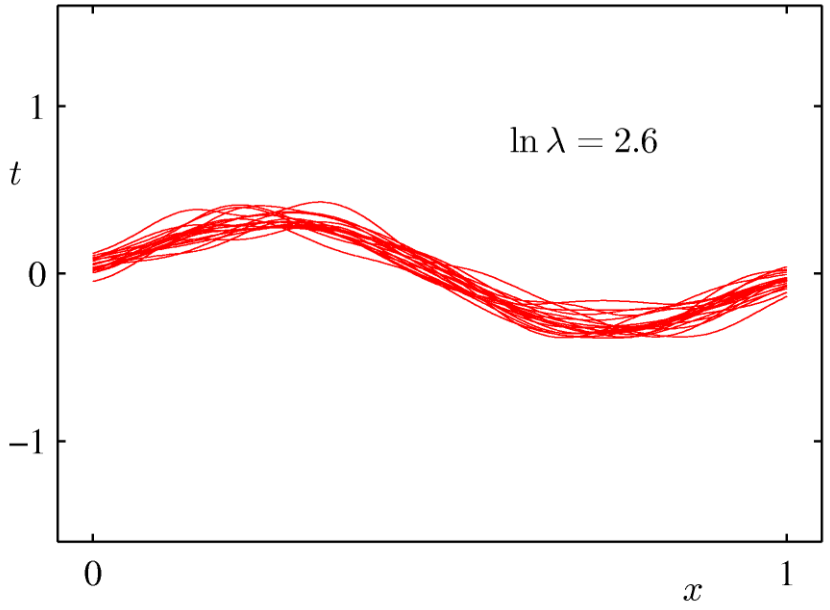
$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

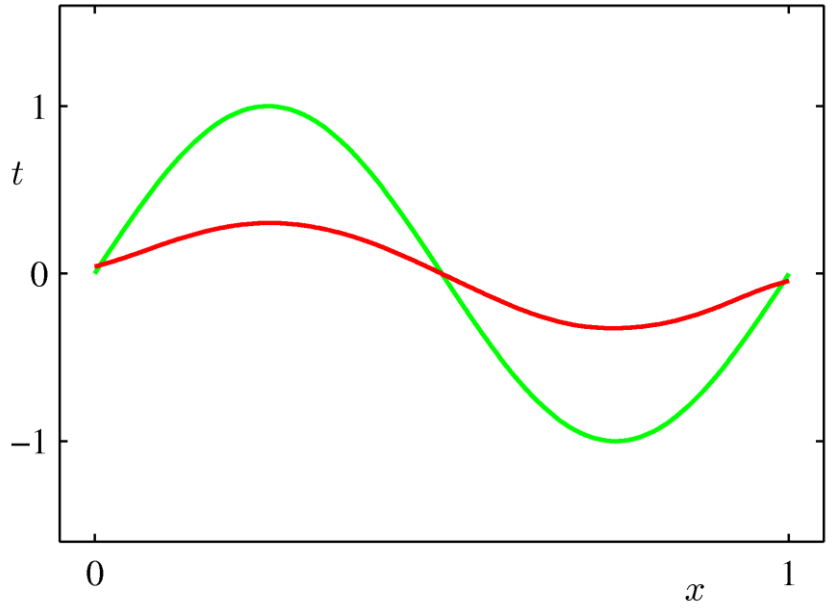
- where

$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]-h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$
$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$
$$\text{noise} = \iint \{h(\mathbf{x})-t\}^2 p(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

Bias- Variance : Example

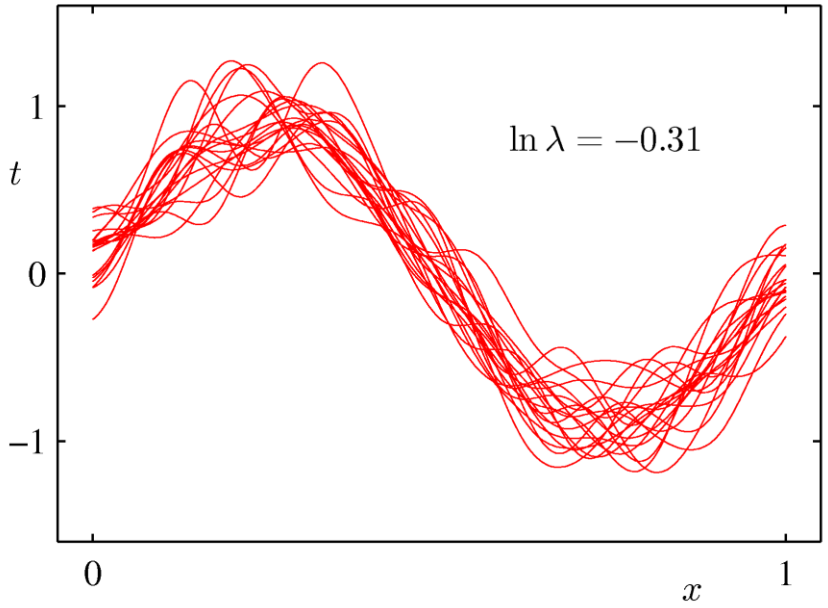
- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ

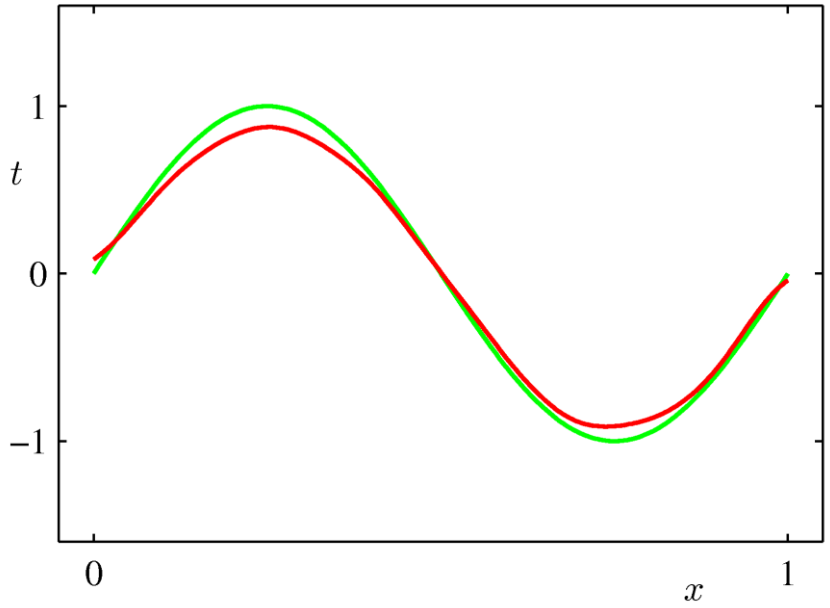




Bias- Variance : Example

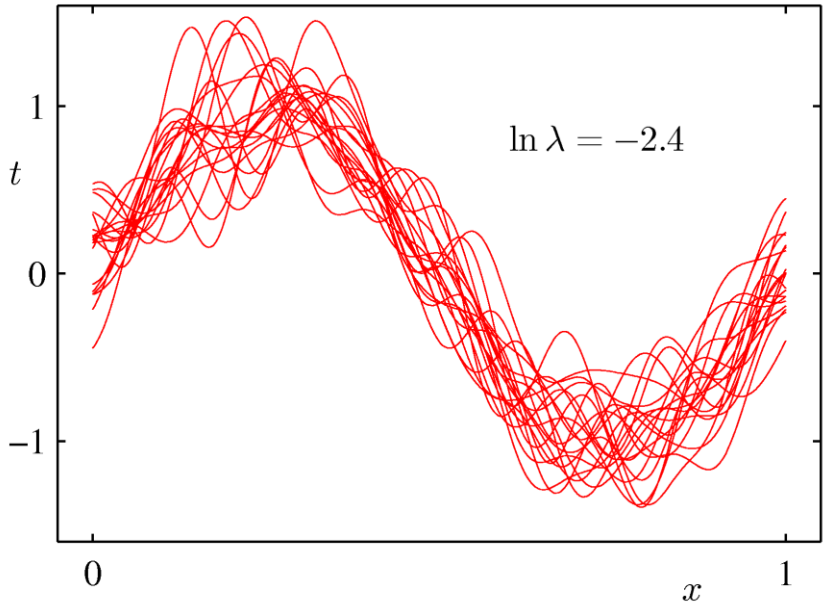
- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ

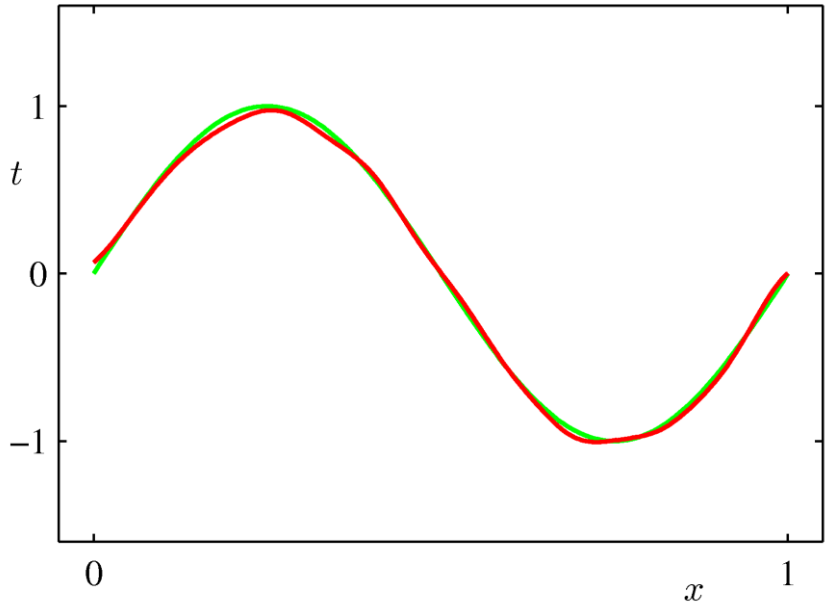




Bias- Variance: Example

- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ





Bias- Variance: Trade-off

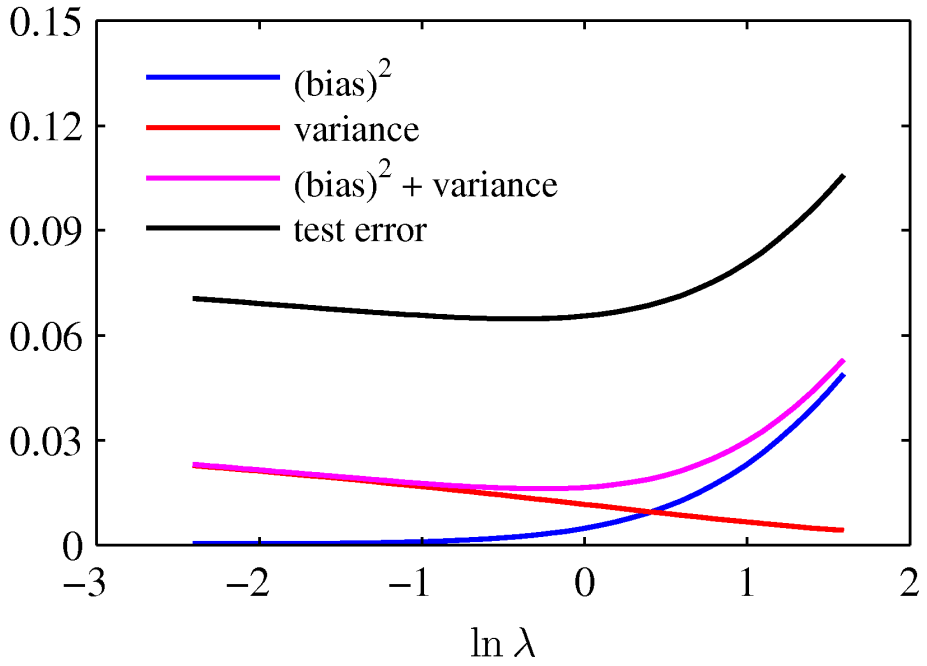
- Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ

•From these plots, we note that an over-regularized model (large λ) will have a high bias, while an under-regularized model (small λ) will have a high variance.

$$\overline{y}(x) = \frac{1}{L}\sum_{l=1}^{L} y^{(l)}(x)$$

$$(\text{bias})^2 = \frac{1}{N}\sum_{n=1}^{N}\{\overline{y}(x_n) - h(x_n)\}^2$$

$$\text{variance} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\{y^{(l)}(x_n) - \overline{y}(x_n)\}^2$$



So far in Linear Regression Models

Goal: Find **w?**

- Why linear model?
- Simple linear regression
- Basis functions
- Solving for w using maximum likelihood and least squares
- For multiple output
- Regularize the model (different regularizers)
- Managing over-fitting via the concept of bias-variance decomposition

So which model to choose?

Remember *model selection* and *cross-validation*

Next: Bayesian Linear Regression

- We know - **Posterior = likelihood x prior**
- Likelihood

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

- Define a conjugate prior over w

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0).$$

•Combining and using results for marginal and conditional Gaussian distributions, gives the posterior *(Refer Chapter-2, Bishop)*

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

- where

$$\mathbf{m}_N = \mathbf{S}_N\left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}\right)$$

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}.$$

Bayesian Linear Regression

- A common choice for the **prior** is (zero-mean isotropic Gaussian)

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

for which

$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}. \end{aligned}$$

- The w may vary from w_{ML} to w_{prior}

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{const.}$$

- Maximization of this posterior distribution with respect to w is therefore equivalent to the minimization of the sum-of-squares error

Example

$$y(x, \mathbf{w}) = w_0 + w_1 x.$$

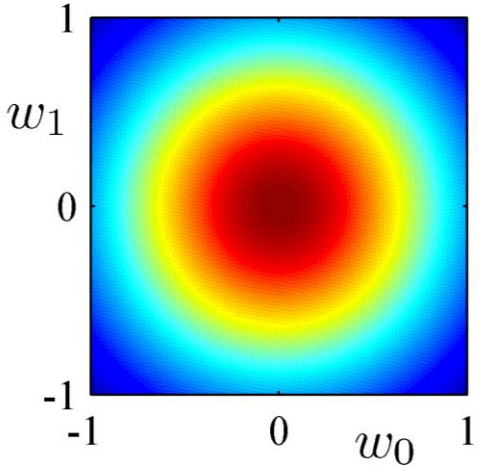
X = U(x| (-1,1), 20 observations

W_0 = 0.3, w_1 = 0.5

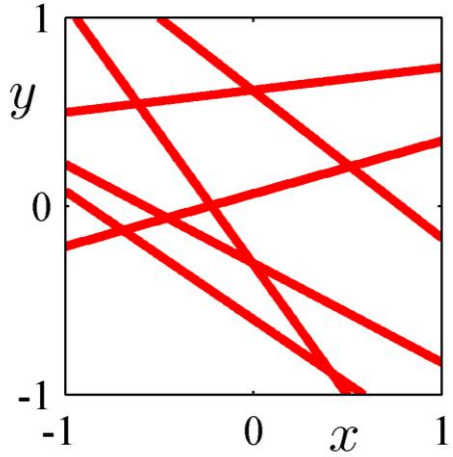
Noise = Gaussian (sigma = 0.2)

Alpha = 2 for prior

Prior

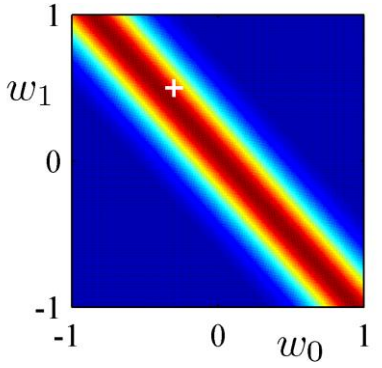


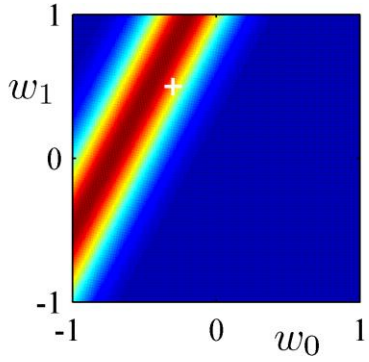
Data Space

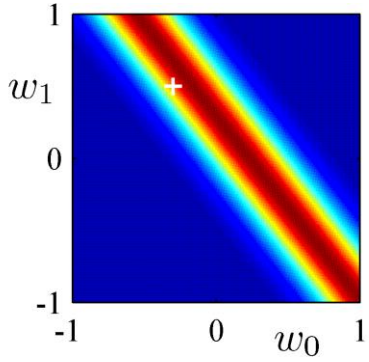


0 data points observed

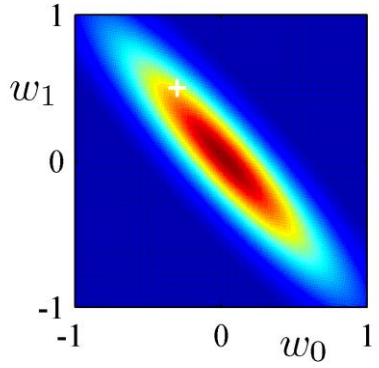
Likelihood

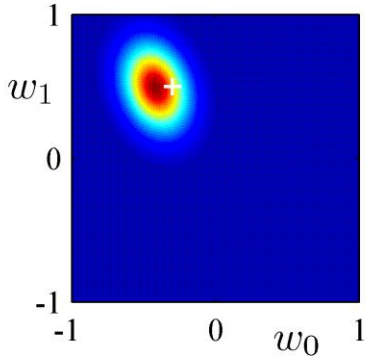


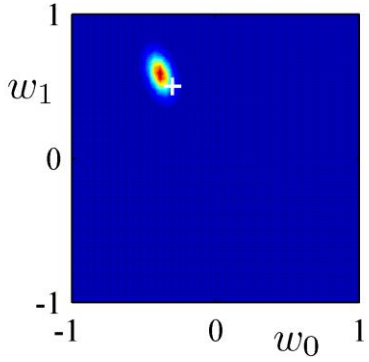




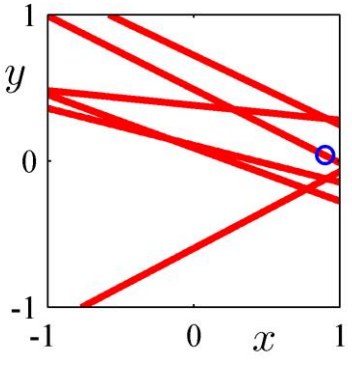
Posterior

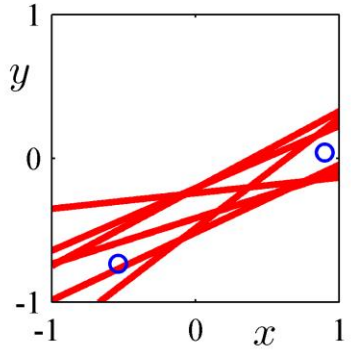


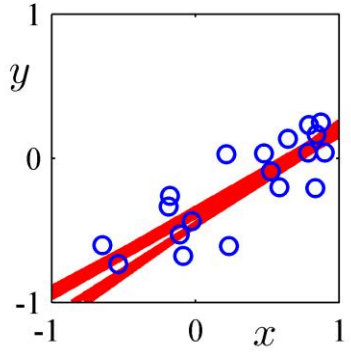




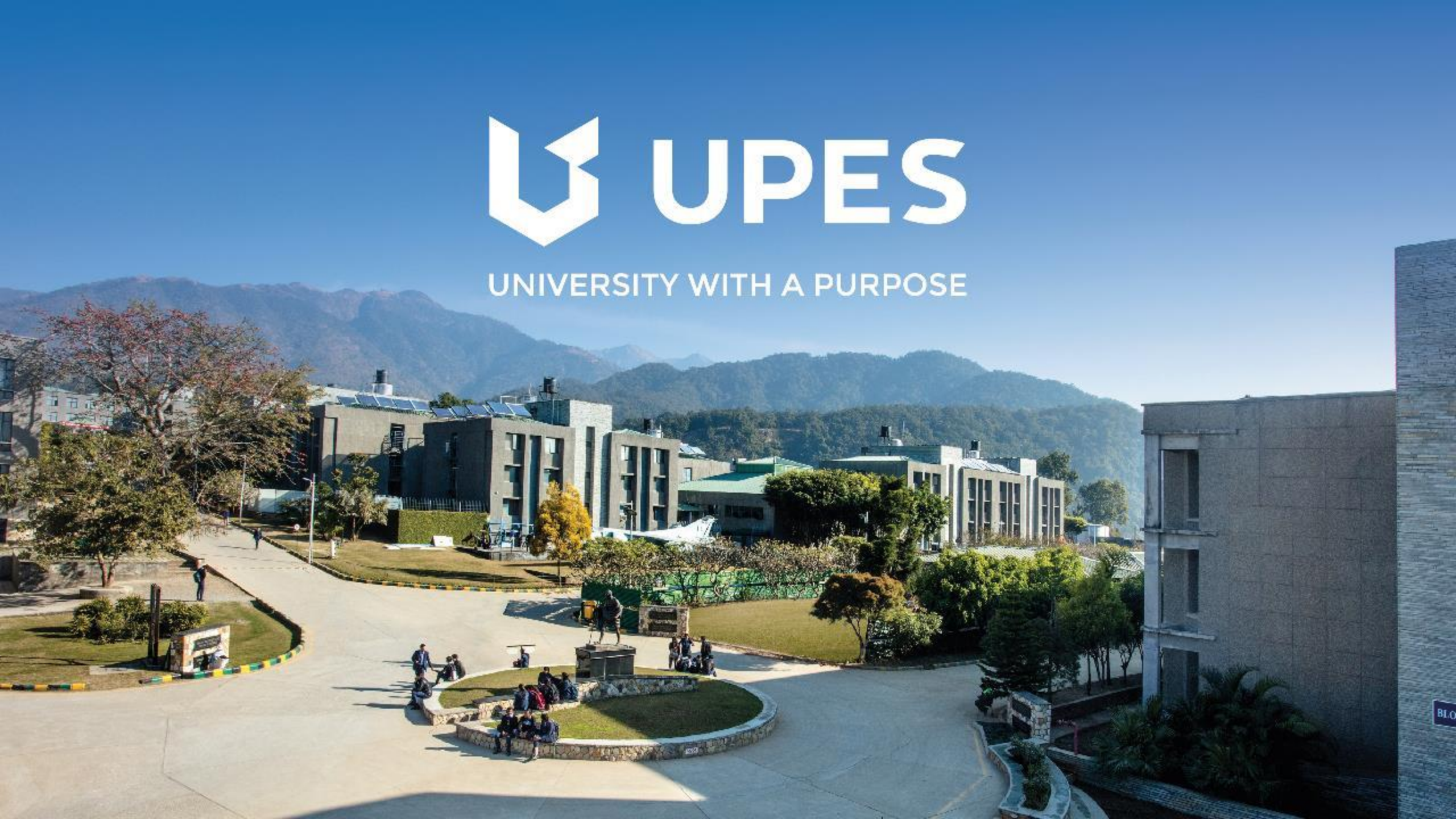
Data Space



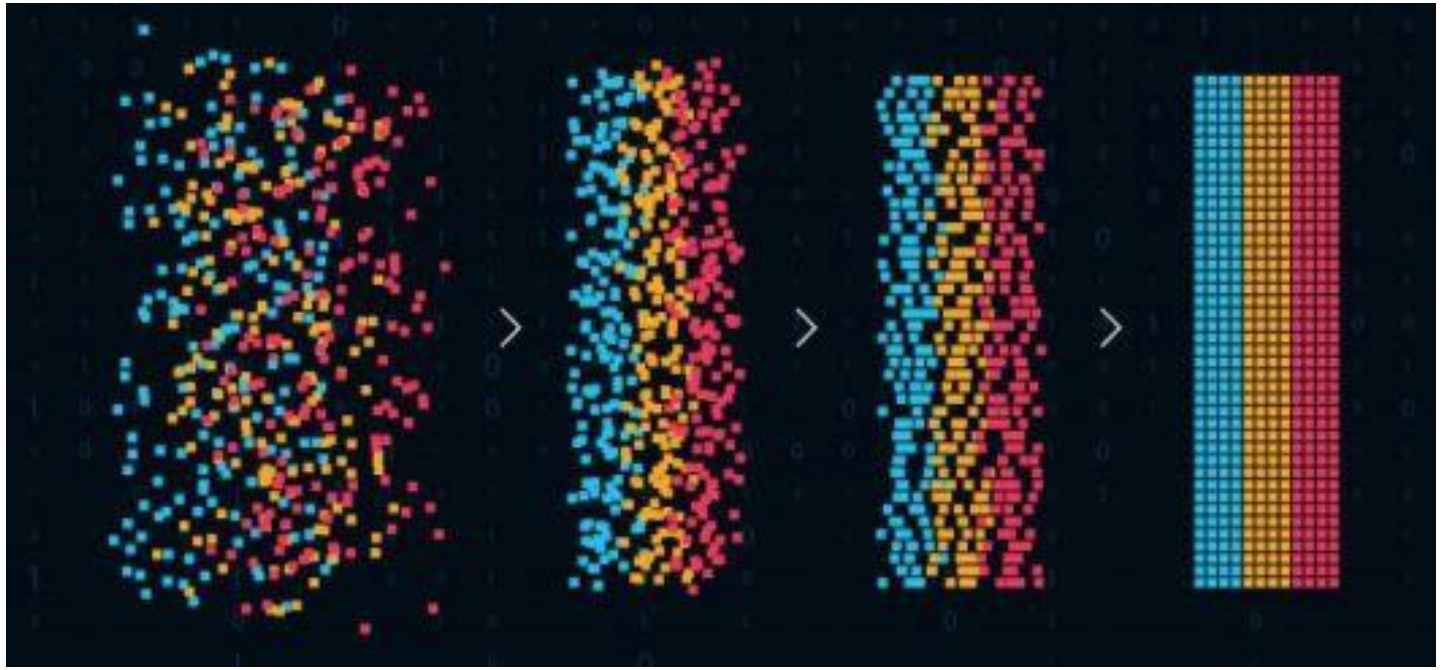




Thank You


UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

18/10/2021

Recap: Linear Regression Models

Goal: Find **w?**

- Why linear model?
- Simple linear regression
- Basis functions
- Solving for w using maximum likelihood and least squares
- For multiple output
- Regularize the model (different regularizers)
- Managing over-fitting via the concept of bias-variance decomposition

So which model to choose?

Remember *model selection* and *cross-validation*

Recap: Bayesian Linear Regression

- Posterior distribution $$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

- With a specific prior $$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$
- Mean and covariance of posterior **w**

$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}. \end{aligned}$$

Example

$$y(x, \mathbf{w}) = w_0 + w_1 x.$$

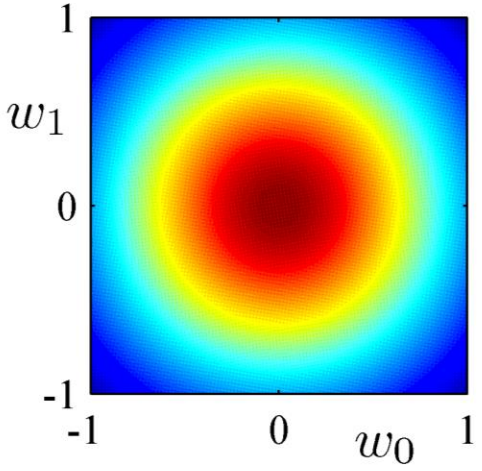
X = U(x| (-1,1), 20 observations

W_0 = 0.3, w_1 = 0.5

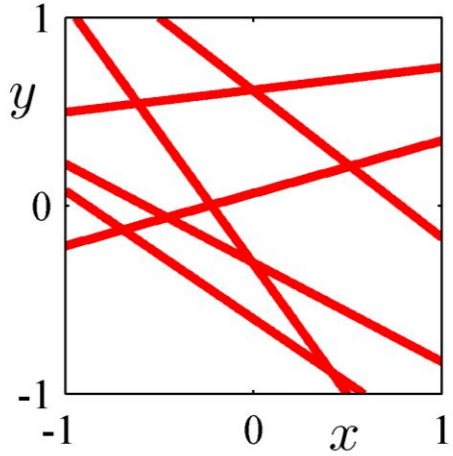
Noise = Gaussian (sigma = 0.2)

Alpha = 2 for prior

Prior

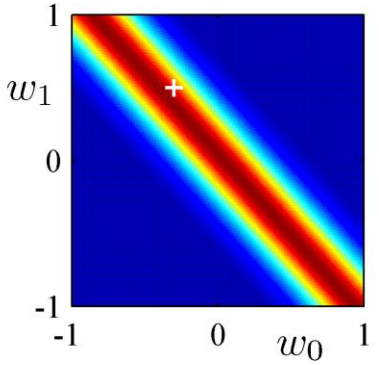


Data Space

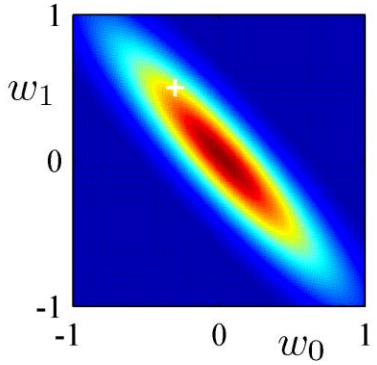


0 data points observed

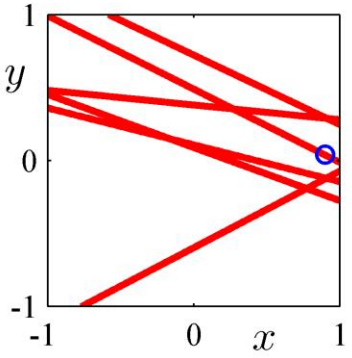
Likelihood

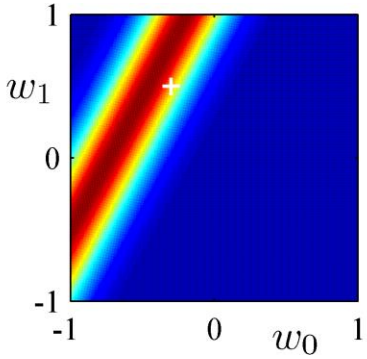


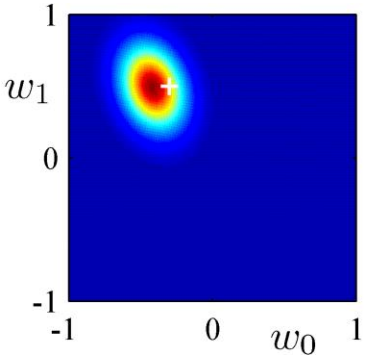
Posterior

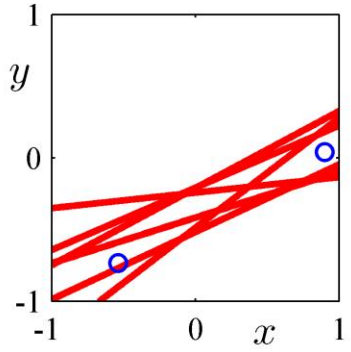


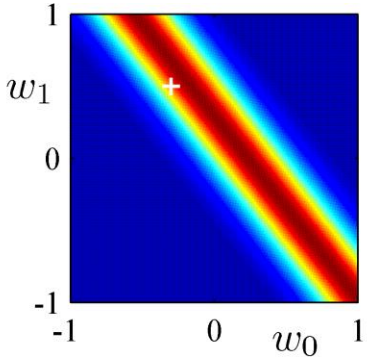
Data Space

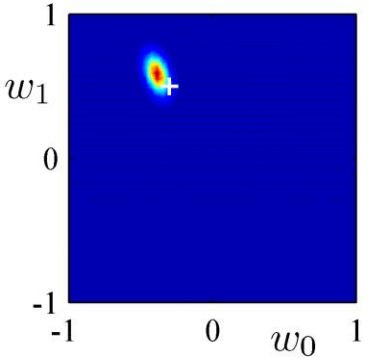


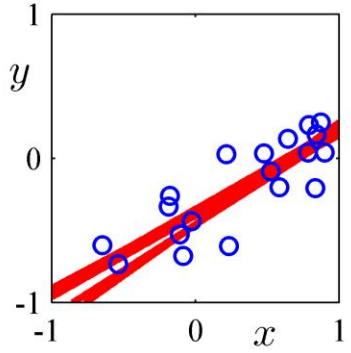












Bayesian Linear Regression: Predictive Distribution

- Predict t for new values of x by integrating over w:

$$p(t|\mathbf{t},\alpha,\beta) = \int p(t|\mathbf{w},\beta)p(\mathbf{w}|\mathbf{t},\alpha,\beta)\,\mathrm{d}\mathbf{w}$$
$$= \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\sigma_N^2(\mathbf{x}))$$

Posterior

Conditional

where

noise on the data

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}).$$

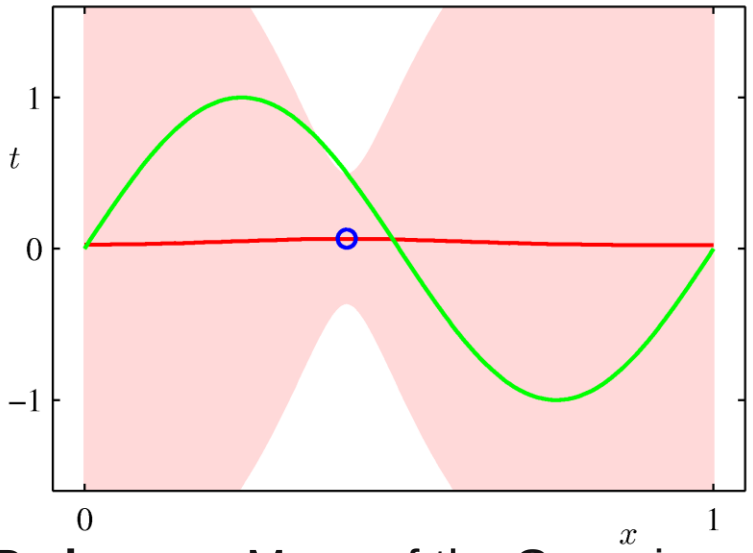
uncertainty associated with the parameters **w**.

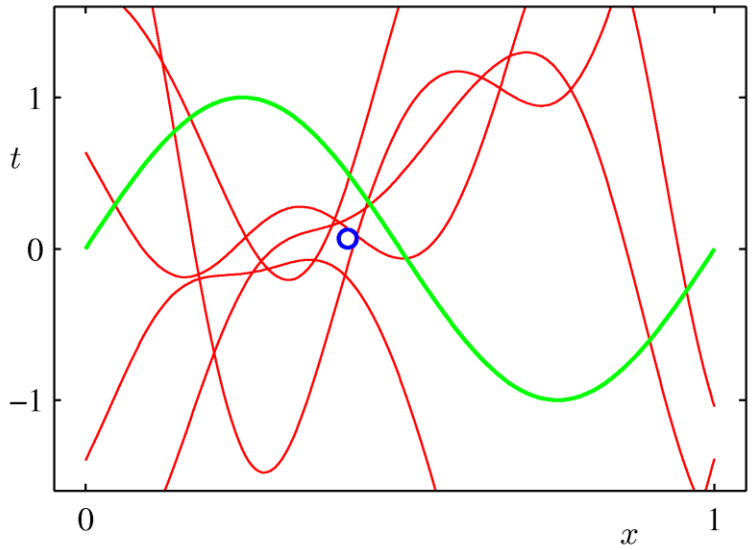
Bayesian Linear Regression: Predictive Distribution

- Example: Sinusoidal data, 9 Gaussian basis functions, 1 data point

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

Model : Comprising a linear combination of Gaussian basis functions



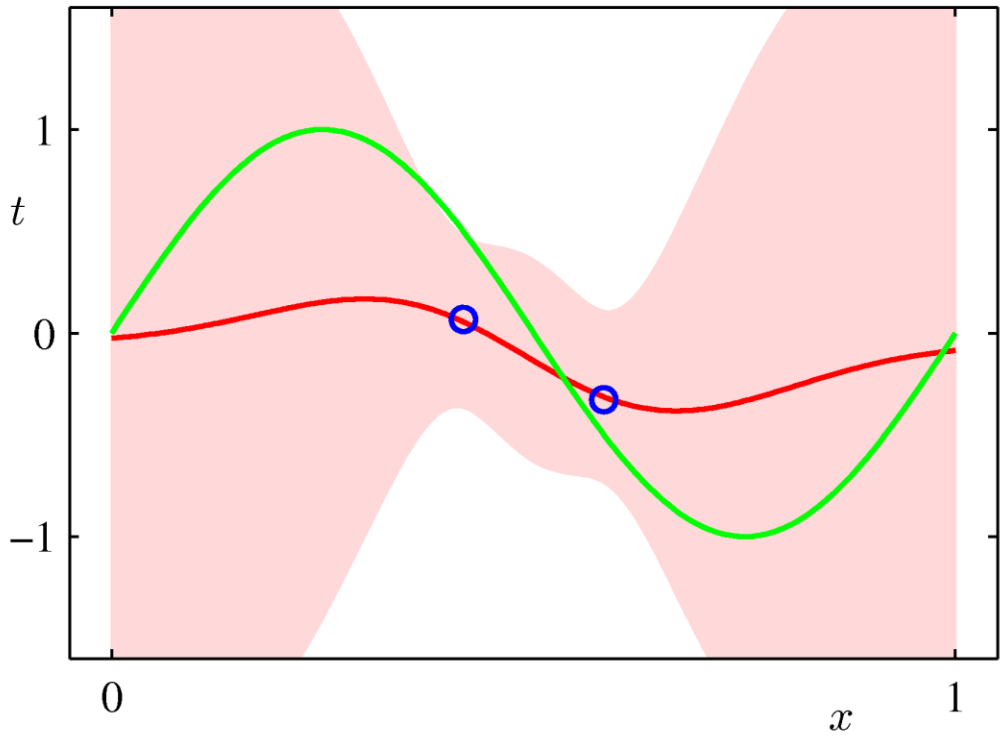


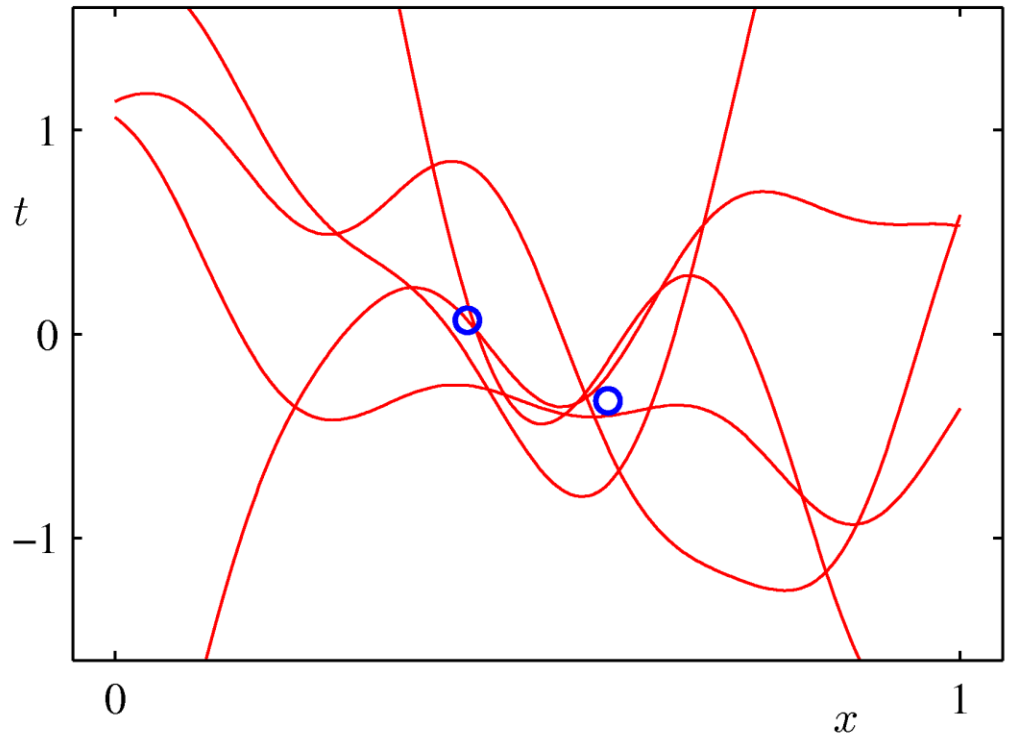
Red curve: Mean of the Gaussian predictive distribution

Shaded region: one standard deviation either side of the mean

Bayesian Linear Regression: Predictive Distribution

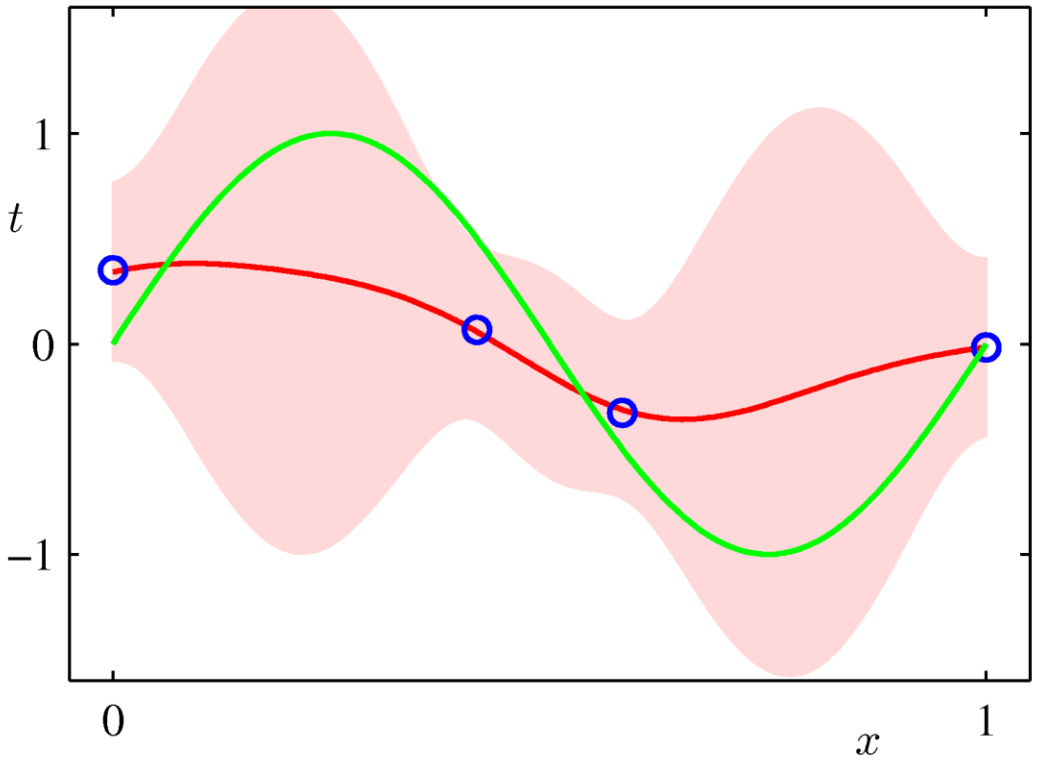
- Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points

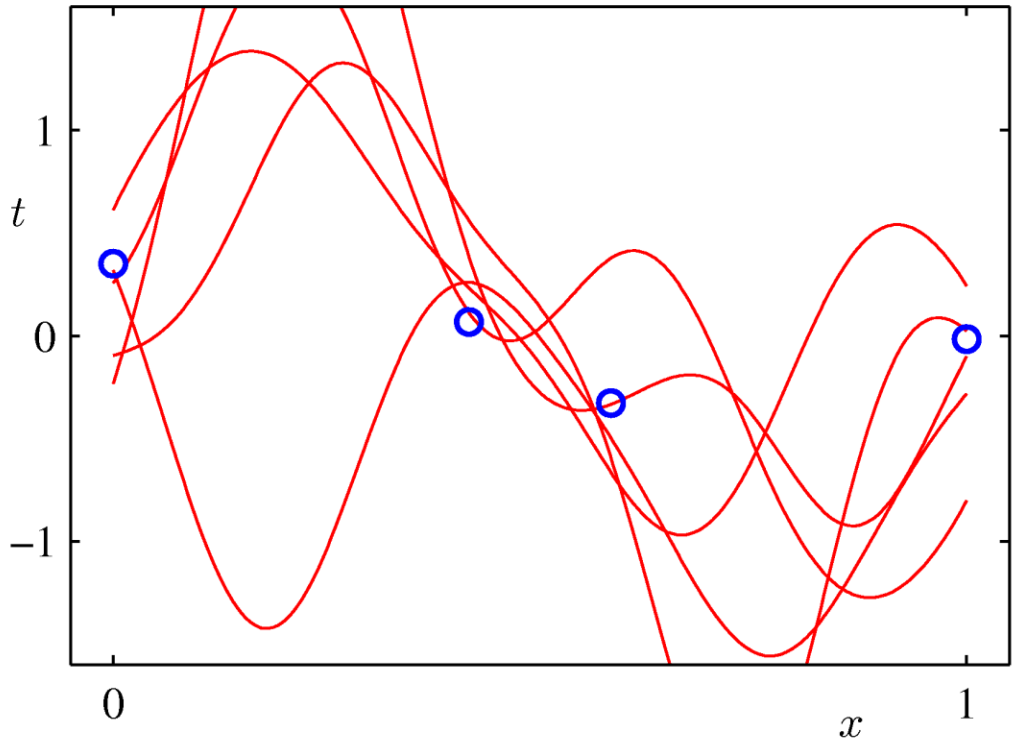




Bayesian Linear Regression: Predictive Distribution

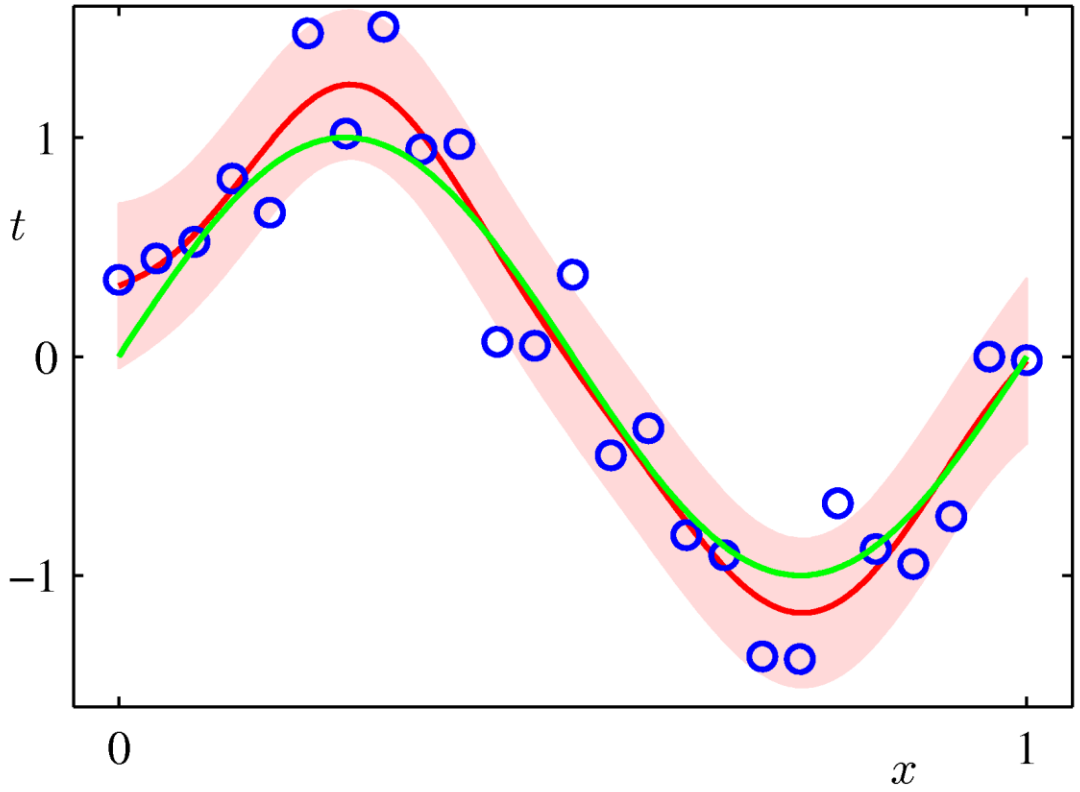
- Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points

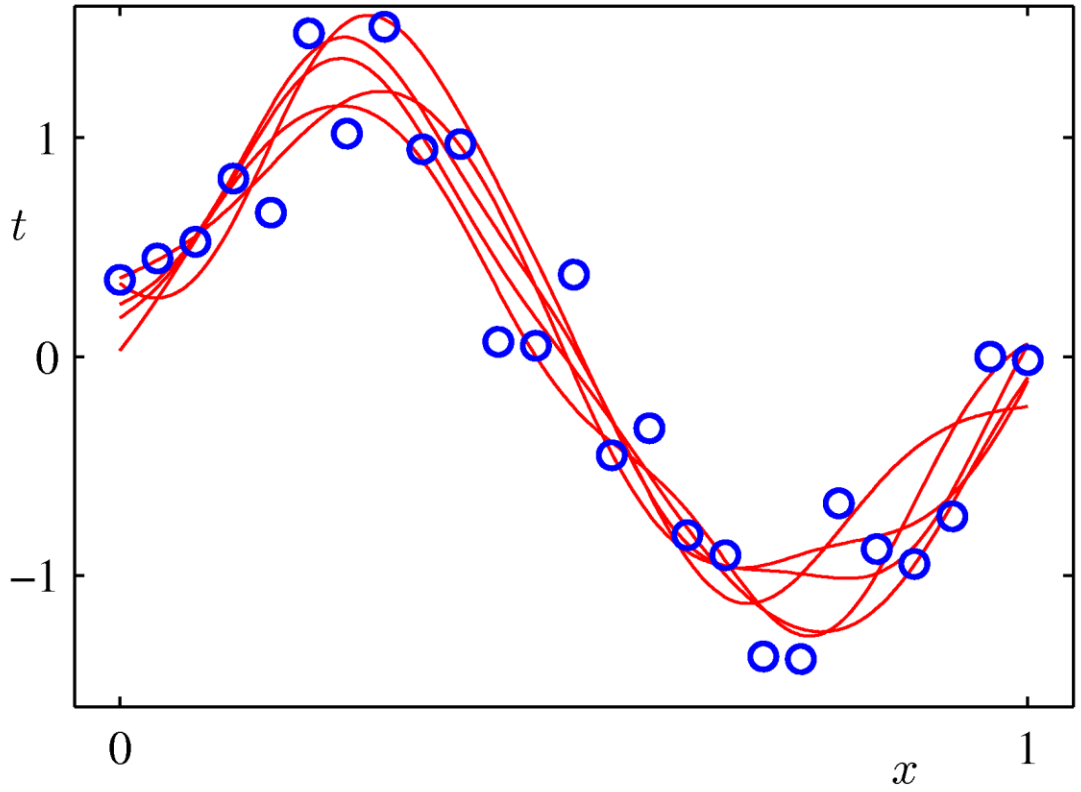




Bayesian Linear Regression: Predictive Distribution

- Example: Sinusoidal data, 9 Gaussian basis functions, 25 data points





Equivalent Kernel

- Remember

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots$$

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

- The predictive mean can be written

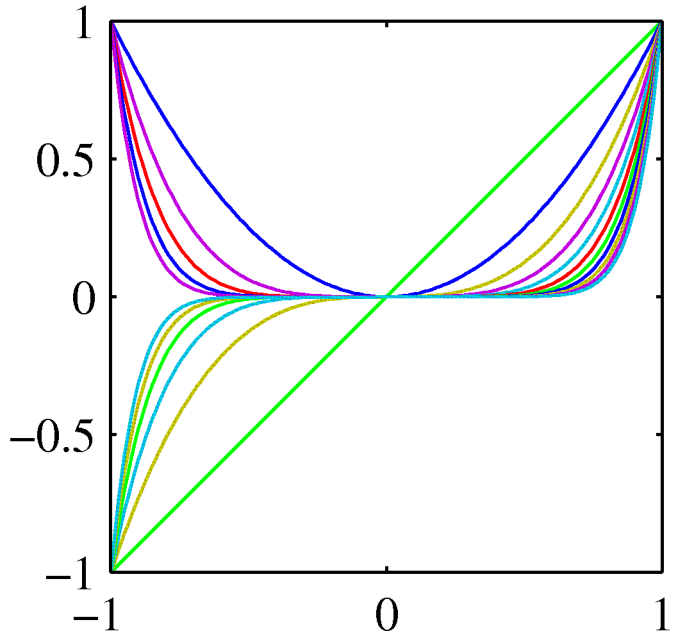
$$\begin{aligned} y(\mathbf{x}, \mathbf{m}_N) &= \mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \\ &= \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}_n) t_n \\ &= \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n. \end{aligned}$$

$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}. \end{aligned}$$

Equivalent kernel or *smoother matrix*.

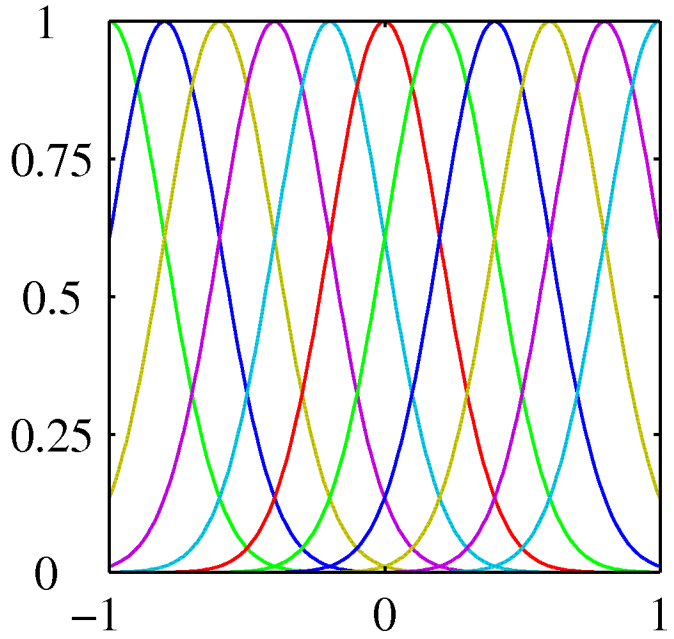
- This is a weighted sum of the training data target values, t_n.

Basis Functions



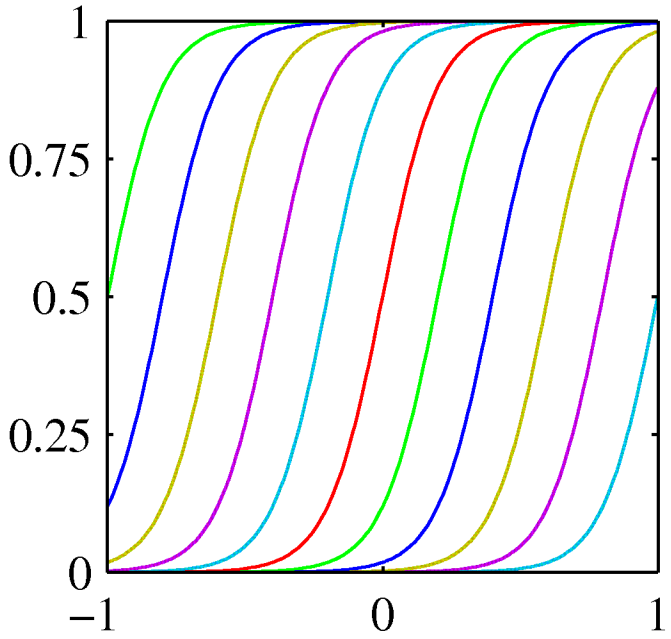
$$\phi_j(x) = x^j.$$

Polynomial basis functions



$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

Gaussian basis functions



$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$$

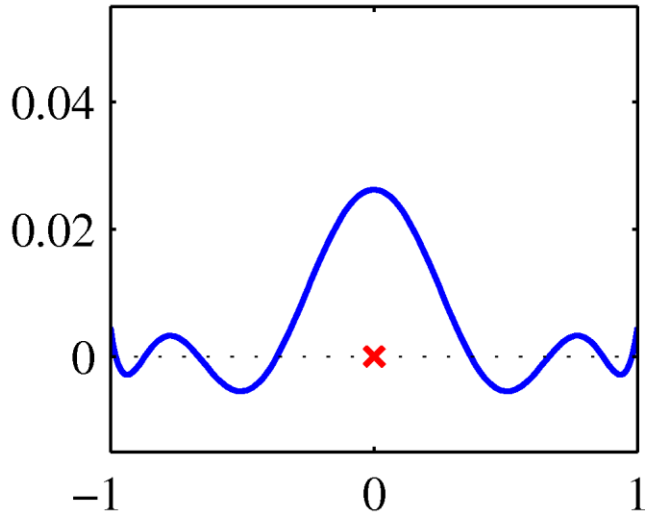
where $$\sigma(a) = \frac{1}{1+\exp(-a)}.$$

Sigmoidal basis functions

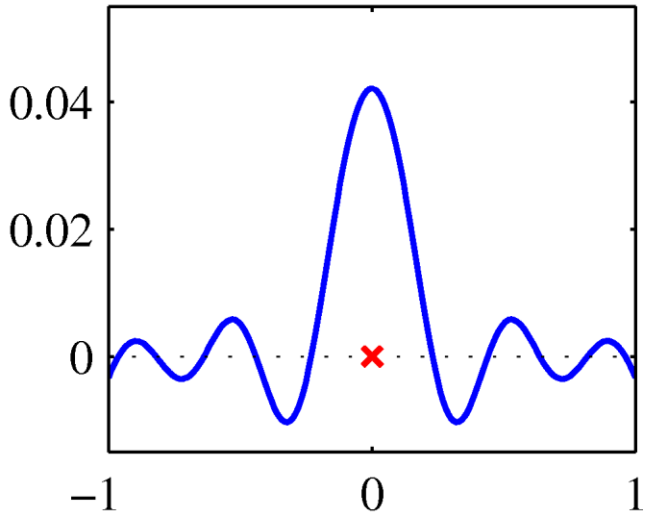
Equivalent Kernel

- Non-local basis functions have local equivalent kernels:

$$k(\mathbf{x}, \mathbf{x}') = \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}')$$

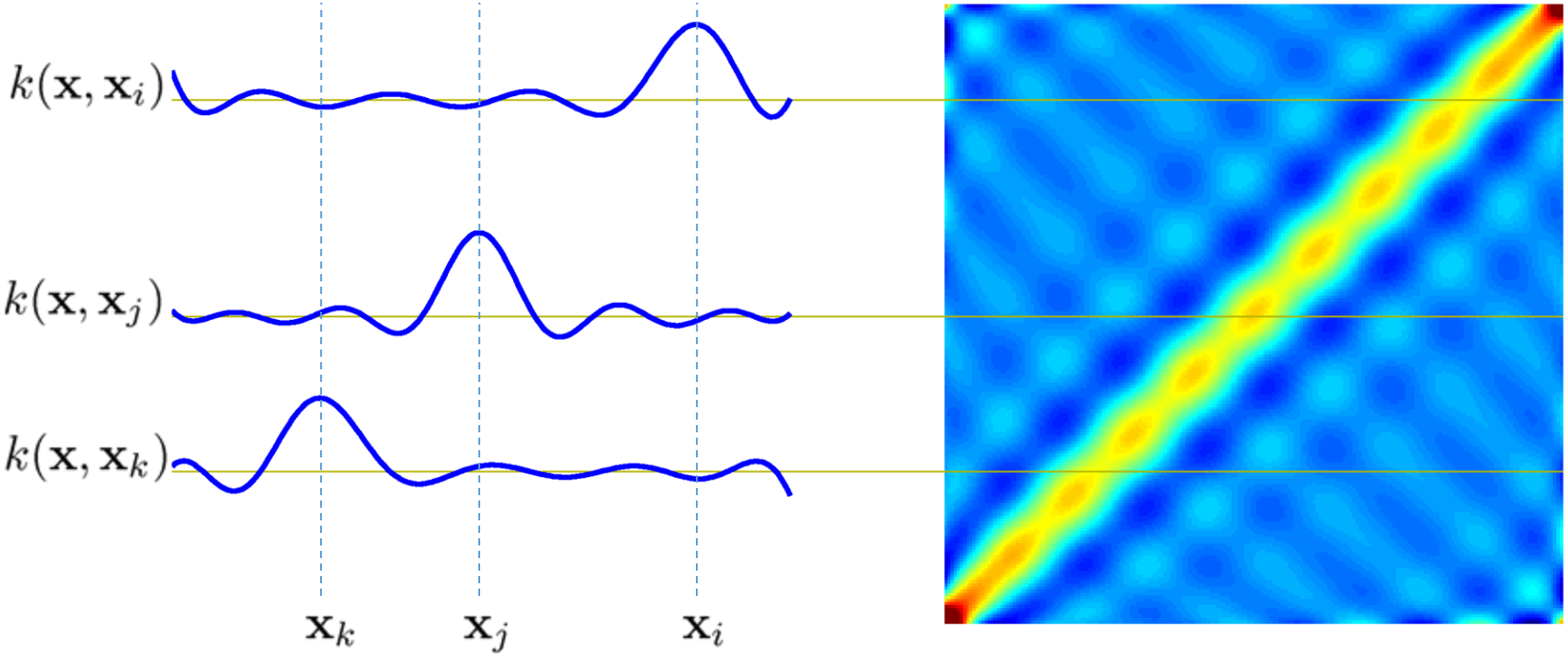


Polynomial



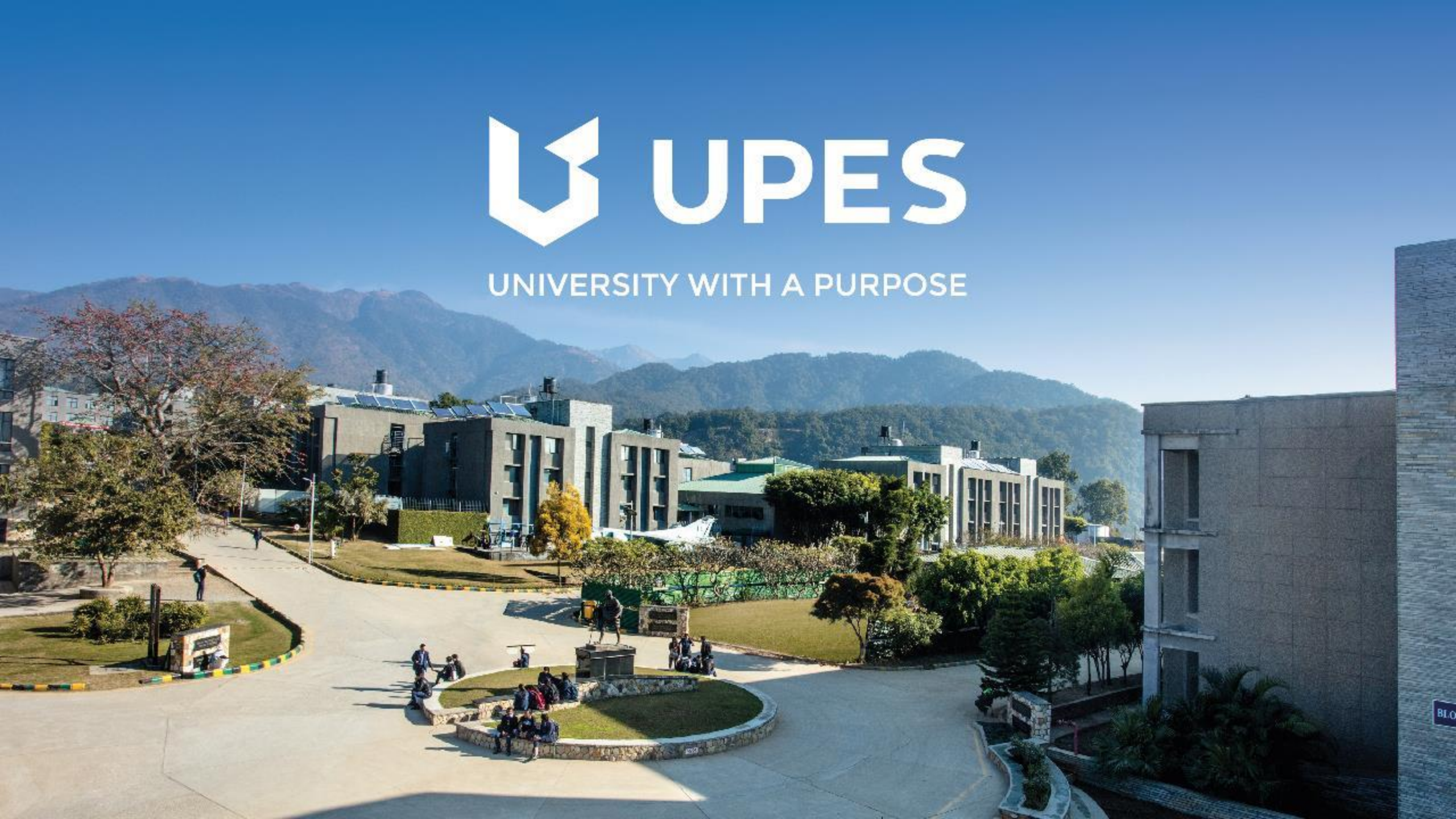
Sigmoidal

Equivalent Kernel

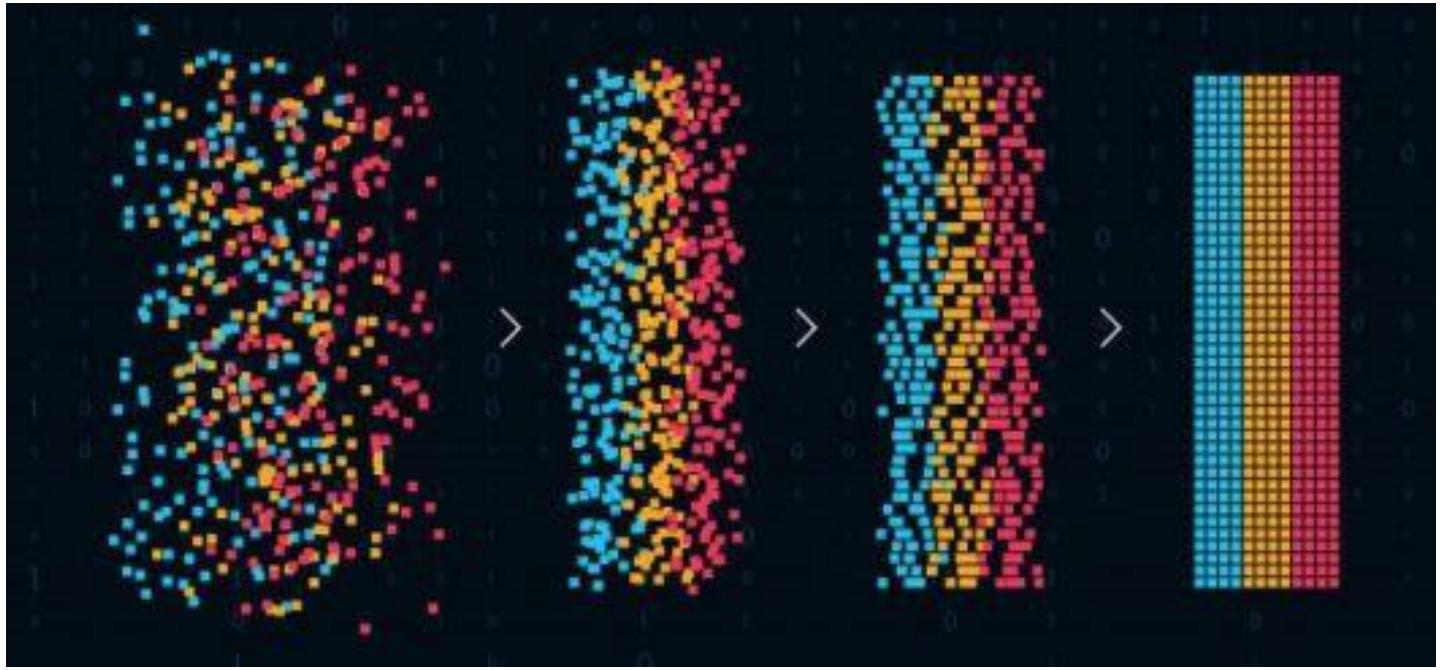


Weight of t_n depends on distance between x and x_n; nearby x_n carry more weight.

Thank You


UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

18/10/2021

Recap: Linear Models for Regression

Goal: Find **w?**

- Why linear model?
- Simple linear regression
- Basis functions and multiple output
- Solving for w using maximum likelihood and least squares or sequential
- Regularize the model (different regularizers)
- Bayesian linear regression models: Parameter and predictive distributions (prior and posterior)
- *Equivalent kernels (output as a linear combination of training data directly)*

Linear Models for Classification

- Goal of classification: Take input (let say x) and assign it to one of the K discrete classes C_k a classes where k = 1, 2, 3, …….K.
- Generic assumption: Classes are disjoint (an input can be assigned to one and only one class, no more no less)
- Models analogous to regression models but for classification problems
- The input space is divided into decision regions whose boundaries are termed as **decision boundaries** or **decision surfaces**.
- At first we will discuss linear models for classification? Decision surface.
- (D-1) dimensional Hyperplane is a linear function of D dimensional input .
- Datasets whose classes can be separated by linear decision surfaces are called linearly separable.

Linear Models for Classification

- For regression problems, the target variable t was simply the vector of real numbers whose values we wish to predict
- In the case of classification, there are various ways of using target values to represent class labels
- **Example**: Two-class problem solved by probabilistic models
- Most convenient is the binary representation

$$t \in \{0, 1\}$$

- Where, t = 1 represents class C_1 and t = 0 represents class C_2. Interpret the value of t as probability of class C_1.

Linear Models for Classification

- For more than two class: one hot encoding or one-of-K coding is used.

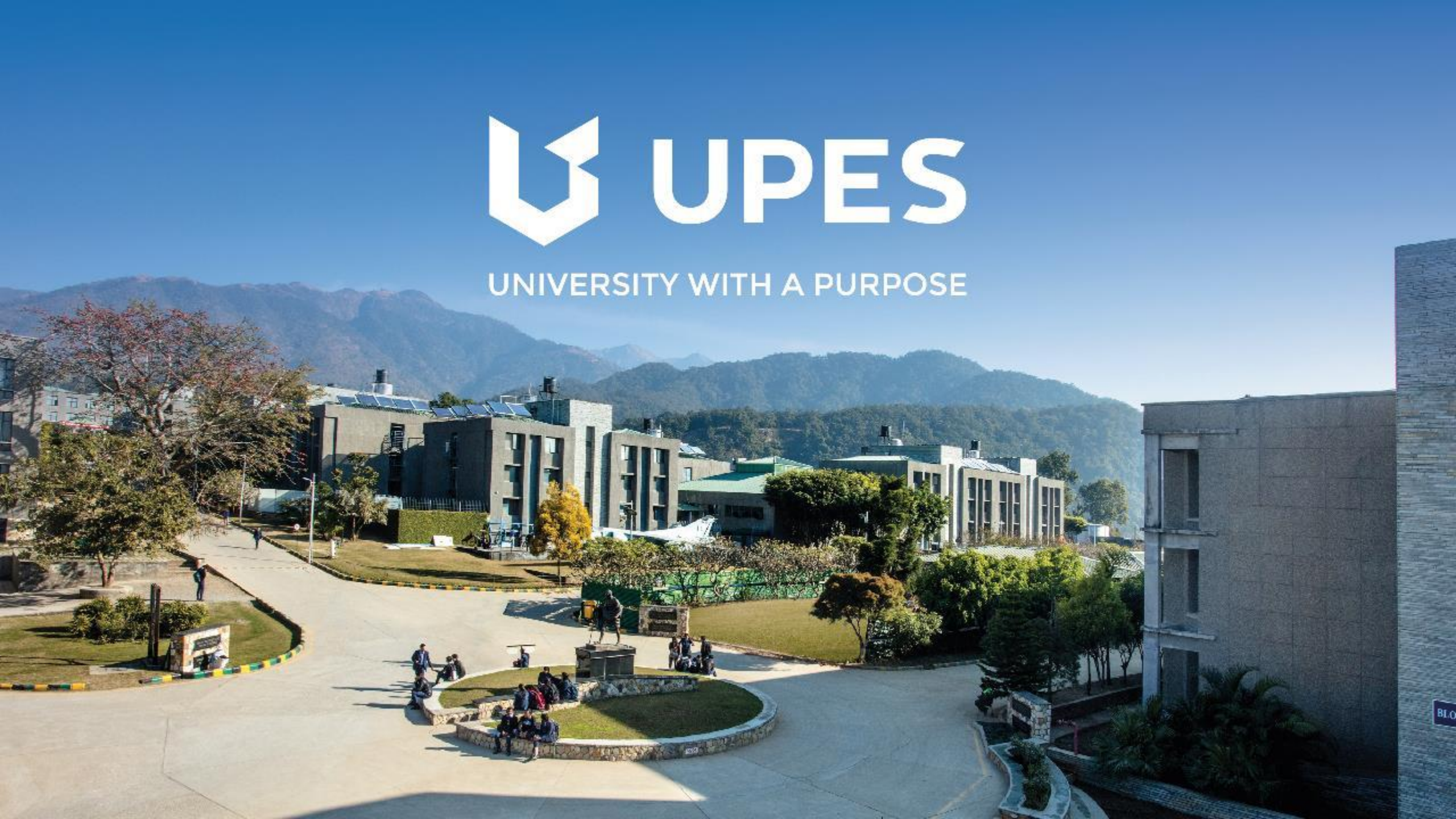
$$\mathbf{t} = (0, 1, 0, 0, 0)^{\mathrm{T}}$$

- For non-probabilistic models, alternative choices of target variable representation can be opted.
- Categories:
  - Discriminant functions
  - Generative
  - Deterministic

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$$

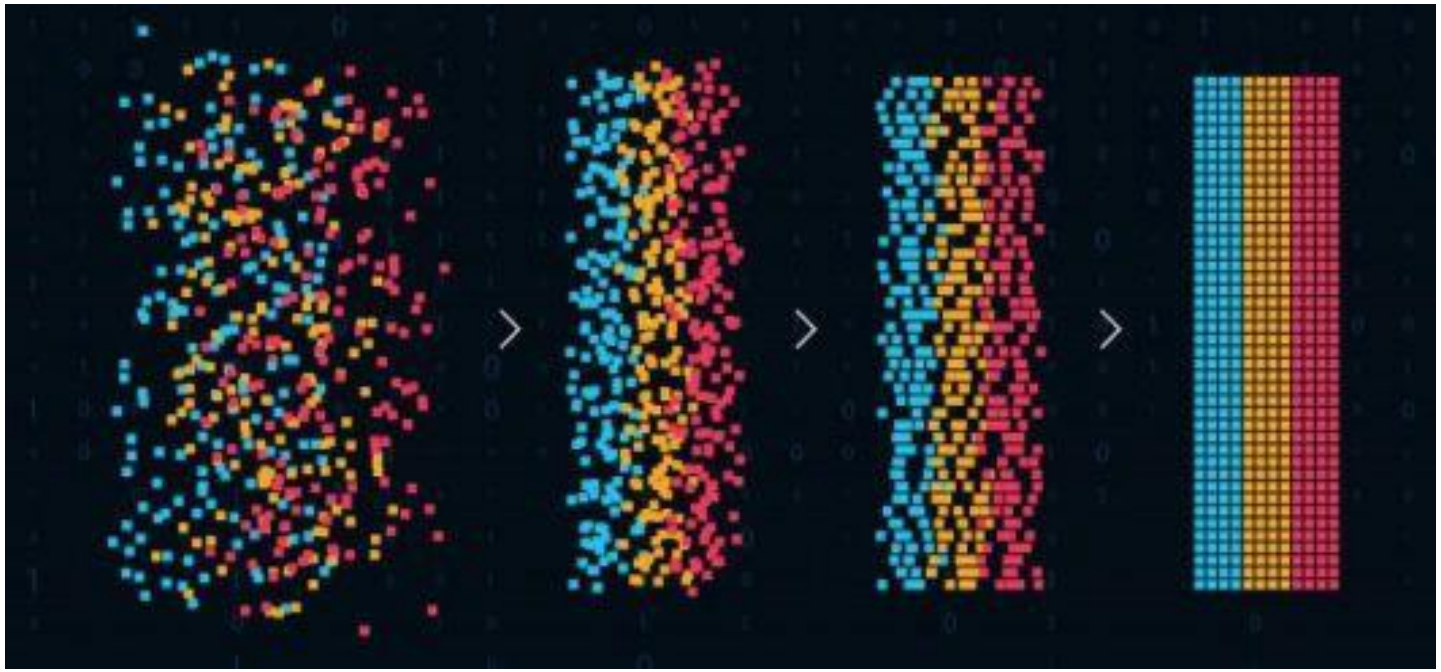
$$y(\mathbf{x}) = \text{constant}$$

Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

25/10/2021

Recap: Linear Models for Classification

- Goal of classification: Take input (let say x) and assign it to one of the K discrete classes C_k a classes where k = 1, 2, 3, …….K.
- Generic assumption: Classes are disjoint (an input can be assigned to one and only one class, no more no less)
- Models analogous to regression models but for classification problems
- The input space is divided into decision regions whose boundaries are termed as **decision boundaries** or **decision surfaces**.
- At first we will discuss linear models for classification? Decision surface.
- (D-1) dimensional Hyperplane is a linear function of D dimensional input .
- Datasets whose classes can be separated by linear decision surfaces are called linearly separable.

Recap: Linear Models for Classification

- For regression problems, the target variable t was simply the vector of real numbers whose values we wish to predict
- In the case of classification, there are various ways of using target values to represent class labels
- **Example**: Two-class problem solved by probabilistic models
- Most convenient is the binary representation

$$t \in \{0, 1\}$$

- Where, t = 1 represents class C_1 and t = 0 represents class C_2. Interpret the value of t as probability of class C_1.

Recap: Linear Models for Classification

- For more than two class: one hot encoding or one-of-K coding is used.

$$\mathbf{t} = (0, 1, 0, 0, 0)^{\mathrm{T}}$$

- For non-probabilistic models, alternative choices of target variable representation can be opted.
- Categories:
  - Discriminant functions
  - Generative
  - Deterministic

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$$

Linear Discriminant Functions

Example: Two-class classification problem

- Input x is assigned to class C_1 if $y(\mathbf{x}) \geqslant 0$ otherwise to class C_2
- The decision boundary therefore is $y(\mathbf{x}) = 0$
- Consider x_A and x_B. Both are on the decision surface. This means

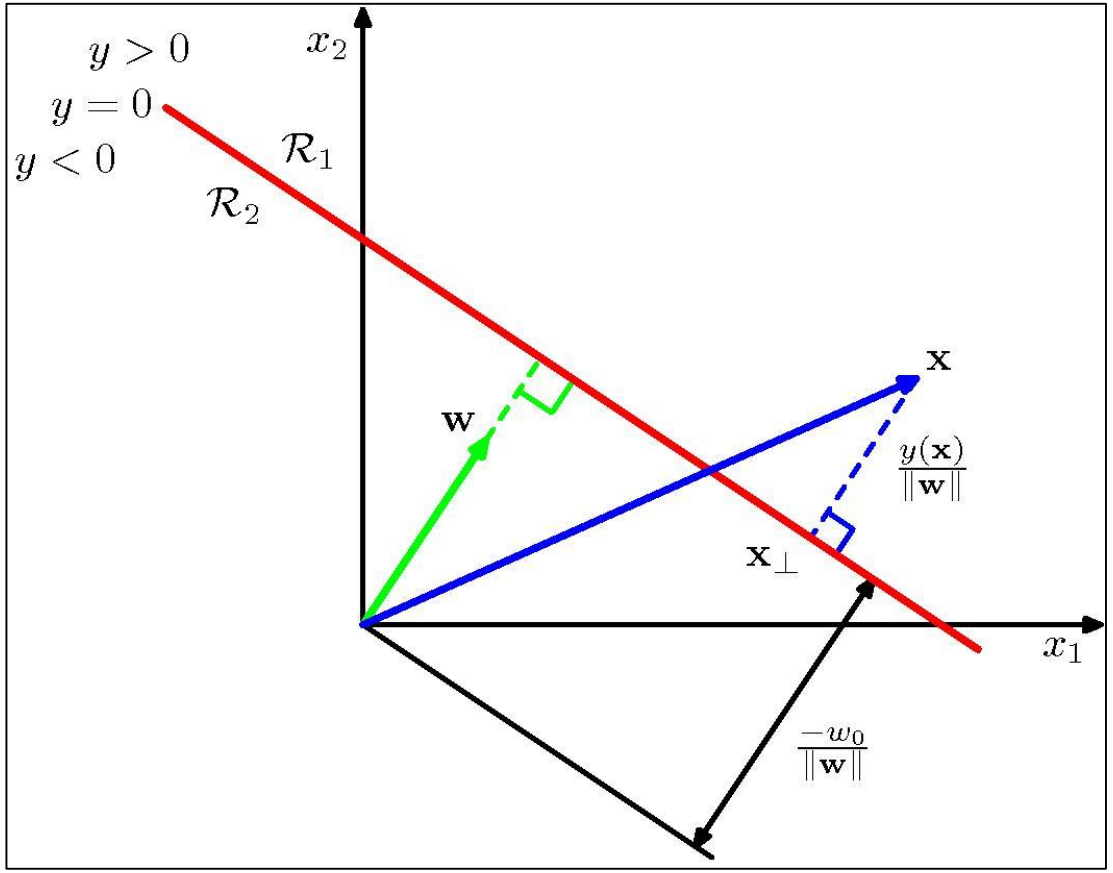
$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_{\mathrm{A}} - \mathbf{x}_{\mathrm{B}}) = 0$$

- Which in turn implies w is perpendicular to every point x which lies on the decision surface.
- Therefore w can determine the orientation of the decision surface.

Linear Discriminant Functions

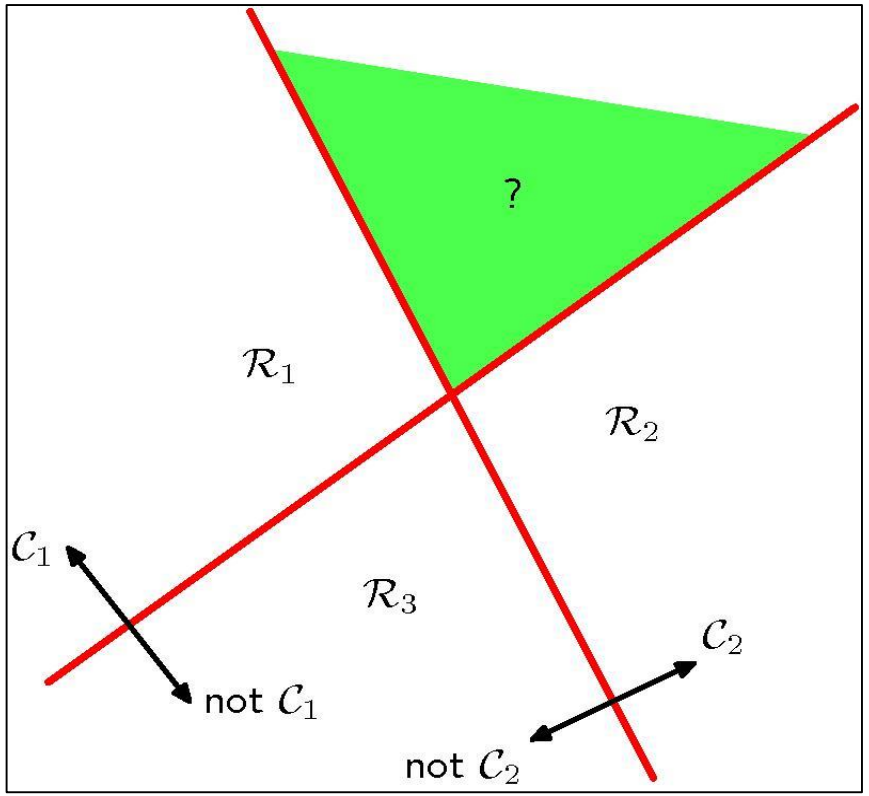
- Also the following holds true if x lies on the decision surface.

$$\frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}.$$



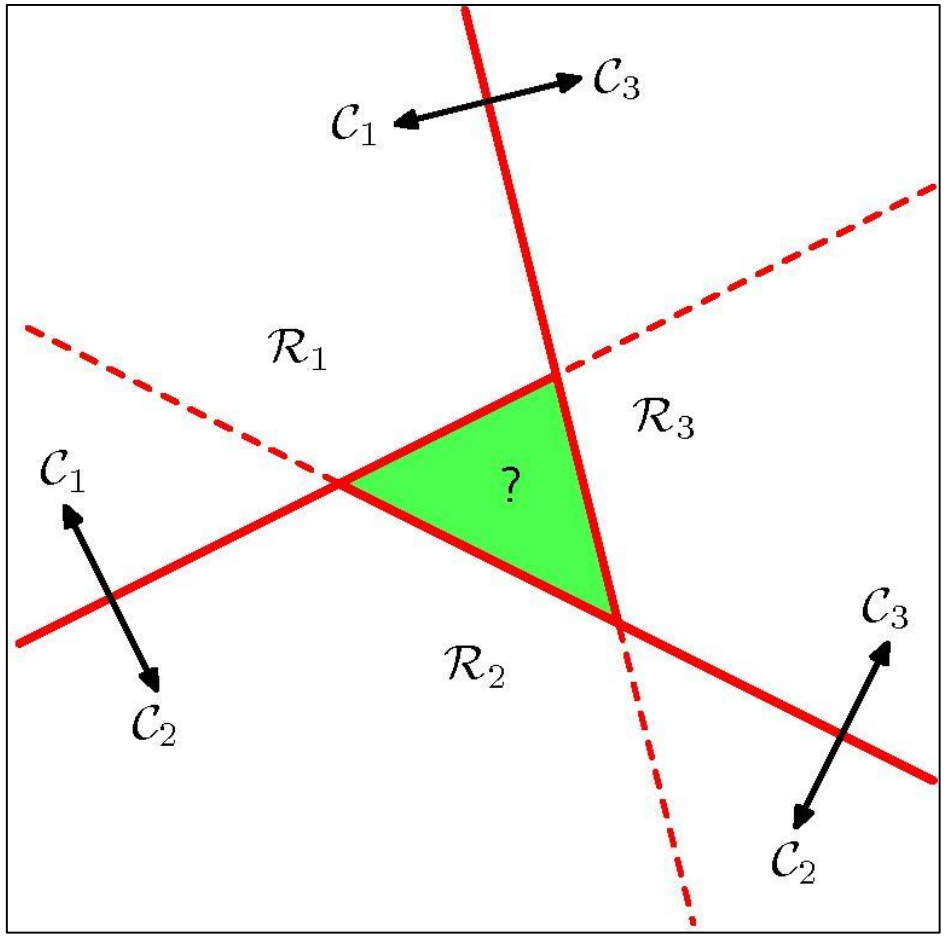
Linear Discriminant Functions

- More than two classes (say K classes)
- Multiple classes: Same approach (K-1 classifiers)
  - Each classifier acts as “One-versus-rest” classifier
  - Ambiguous regions



Linear Discriminant Functions

- Multiple classes
- Alternate approach: K(K-1)/2 classifiers
  - Binary classifiers for each pair of classes.
  - One-versus-one classifier
  - X assigned according to majority vote
  - Ambiguous regions are still present



Linear Discriminant Functions

- Multiple class problem: Alternate approach
- K class discriminant function or classifier defined a

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

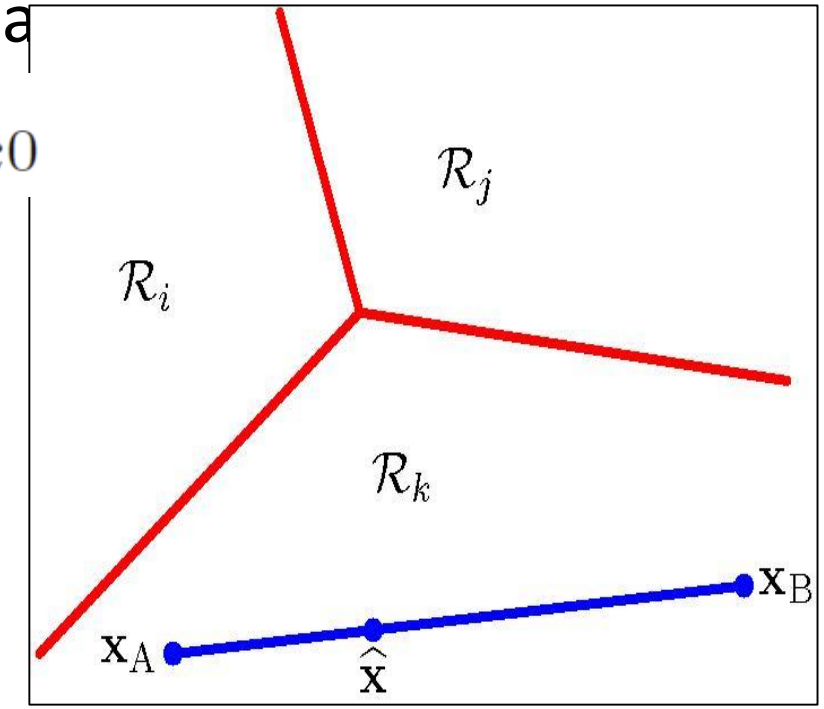
- Assign point x to class C_k it

$$y_k(\mathbf{x}) > y_j(\mathbf{x}) \text{ for all } j \neq k$$

- Decision boundary between class k and class j is

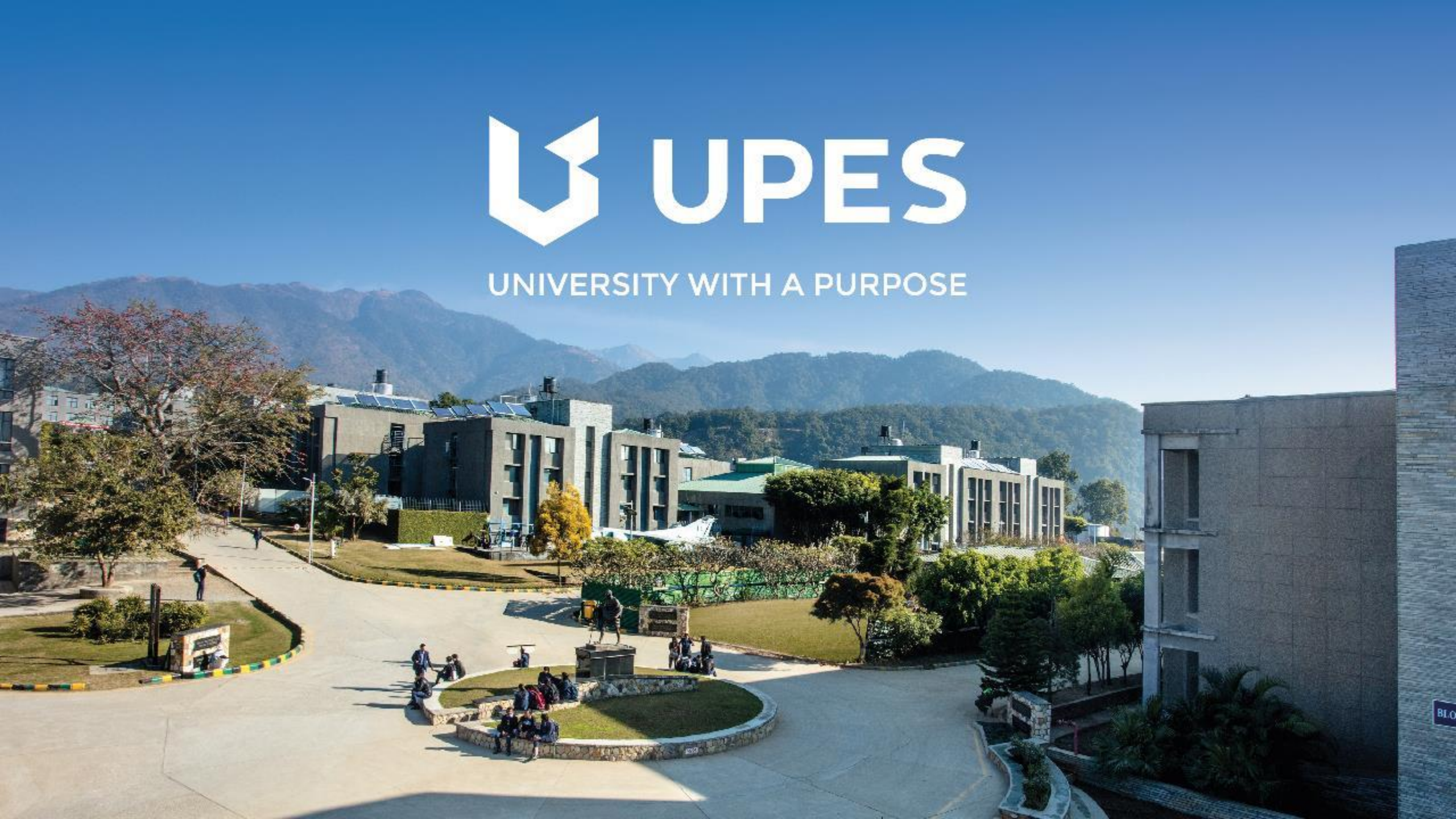
$$y_k(\mathbf{x}) = y_j(\mathbf{x})$$

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0$$



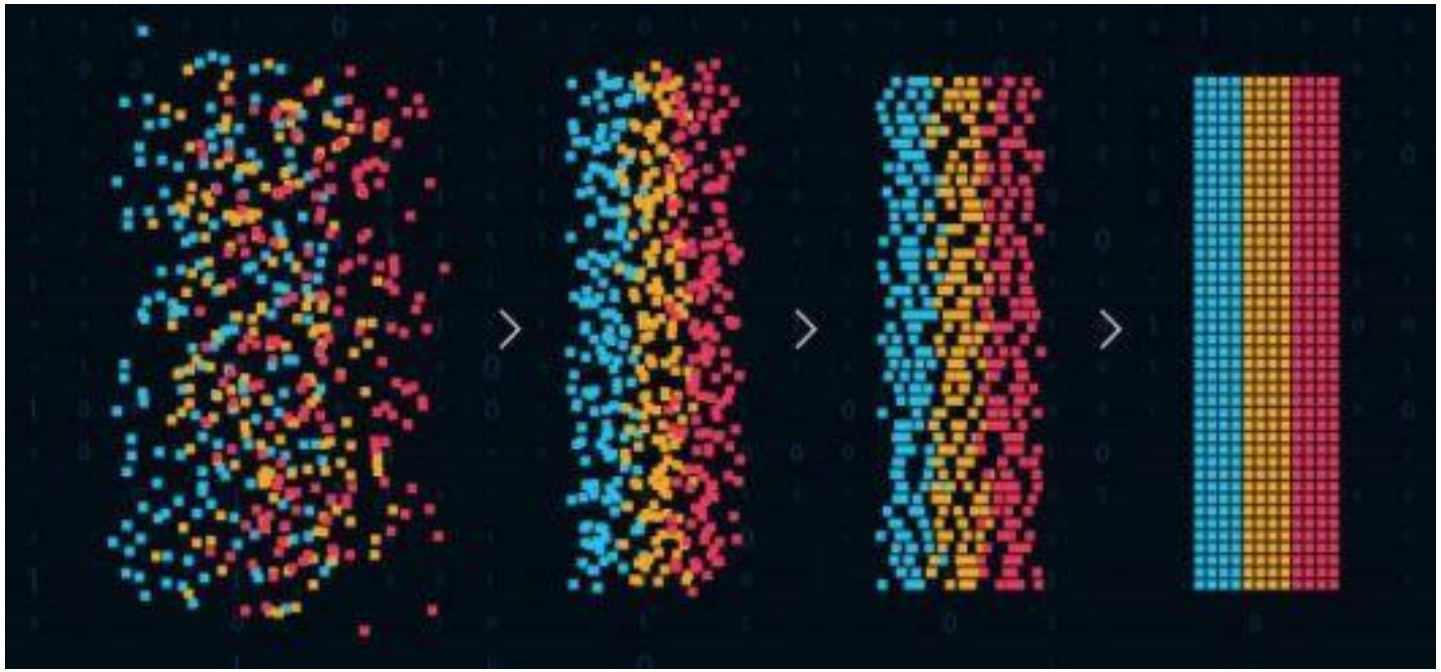
- Singly connected and convex discriminant function

Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

25/10/2021

Recap: Linear Models for Classification

- **Classification:** Assign input x to one of the class labels
- Why we call it linear?
- Linearly separarble dataset and Hyperplanes

- **Linear Discriminant Functions**
  - **Two class**
  - **Multiclass**
    - **3 approaches:** one-versus-the-rest, one-versus-one, and k classifiers
    - Decision region ambiguities

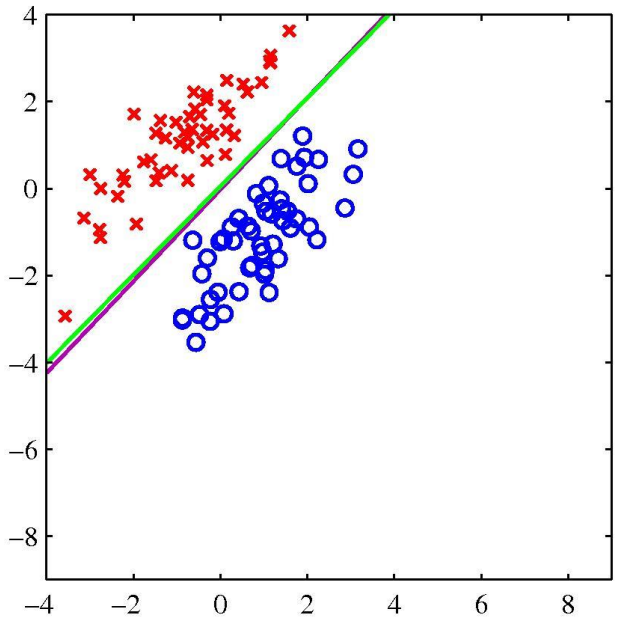
Li-Mod for Classification: Compute W

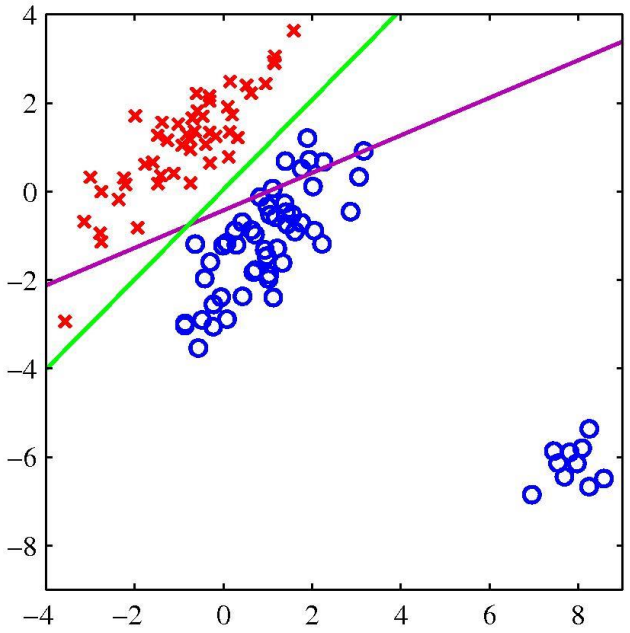
- First approach: Least squares
  - (tildeh when w_0 is included)

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T}$$

- Same issue with least squares

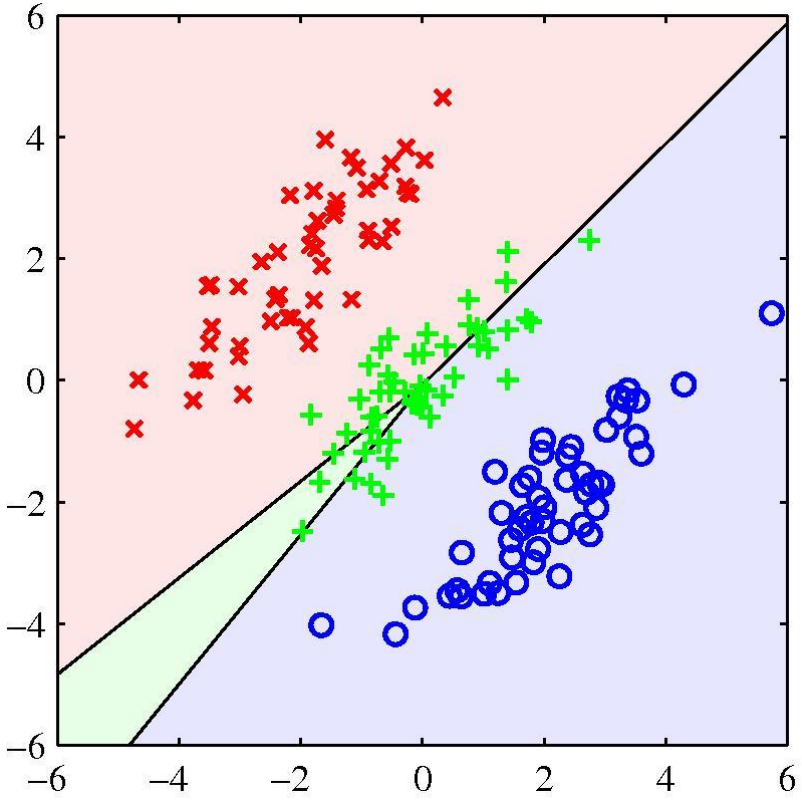
$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}$$



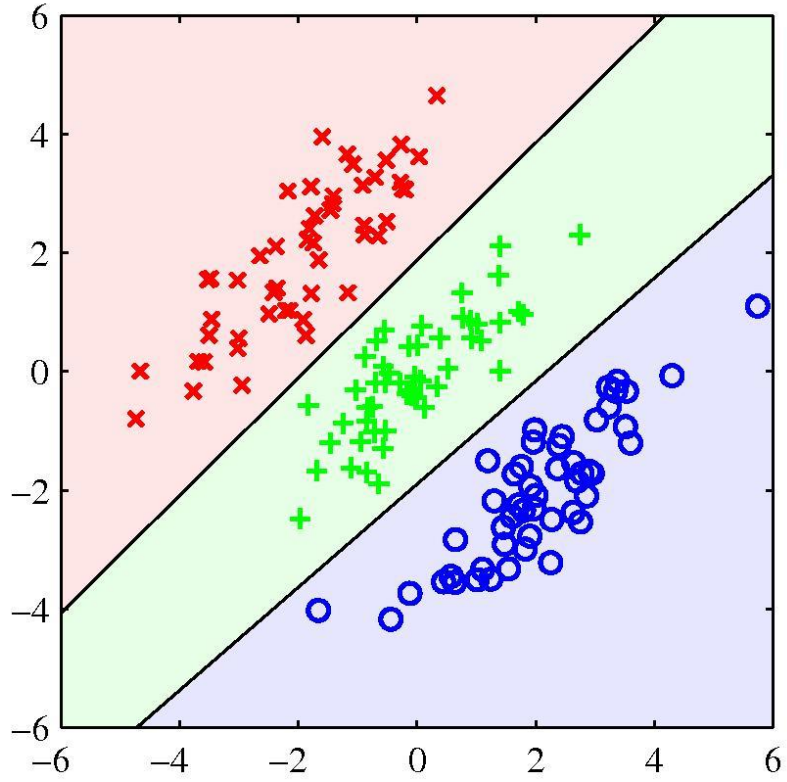


Li-Mod for Classification: Compute W

- Multiple class



- Least squares



- Logistic regression

Li-Mod for Classification: Fisher's Linear Discriminant

- **Concept**: Imagine linear classification models in terms of dimensionality reduction.

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x}.$$

$y \geqslant -w_0$ for C_1

otherwise C_2

- Loss of information in the process of projecting data into lower dimensions.
- Find a projection that **maximizes** the **class separation.**
- FLD is one such projection method (therefore also used in dimension reduction)

Li-Mod for Classification: Fisher's Linear Discriminant

- Consider two-class problem: N_1 points in C_1 and N_2 points in C_2
- Compute mean of both.

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n.$$

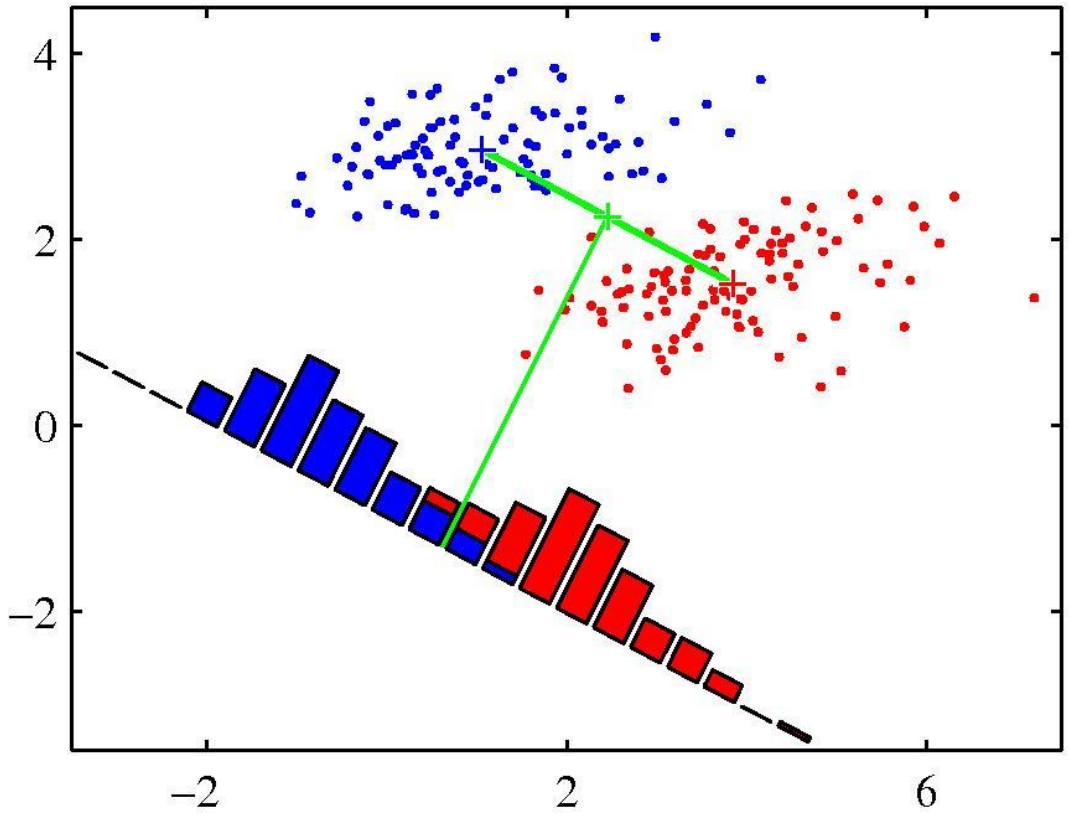
- Maximize the difference in projected space i.e.

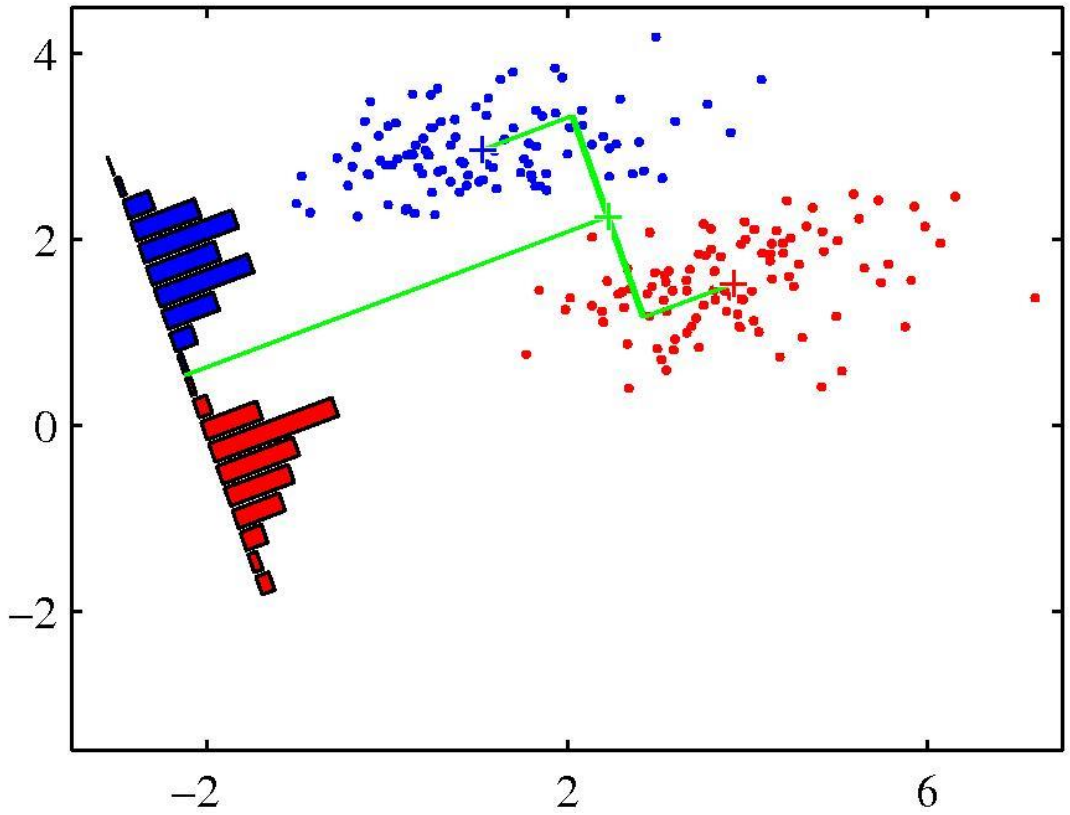
- Maximize $m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1)$

- The within class variance

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$$

Li-Mod for Classification: Fisher's Linear Discriminant





Fisher criterion $J(\mathbf{w}) = \dfrac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1).$$

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$

FLD

Li-Mod for Classification: Fisher's Linear Discriminant

- Multi-class

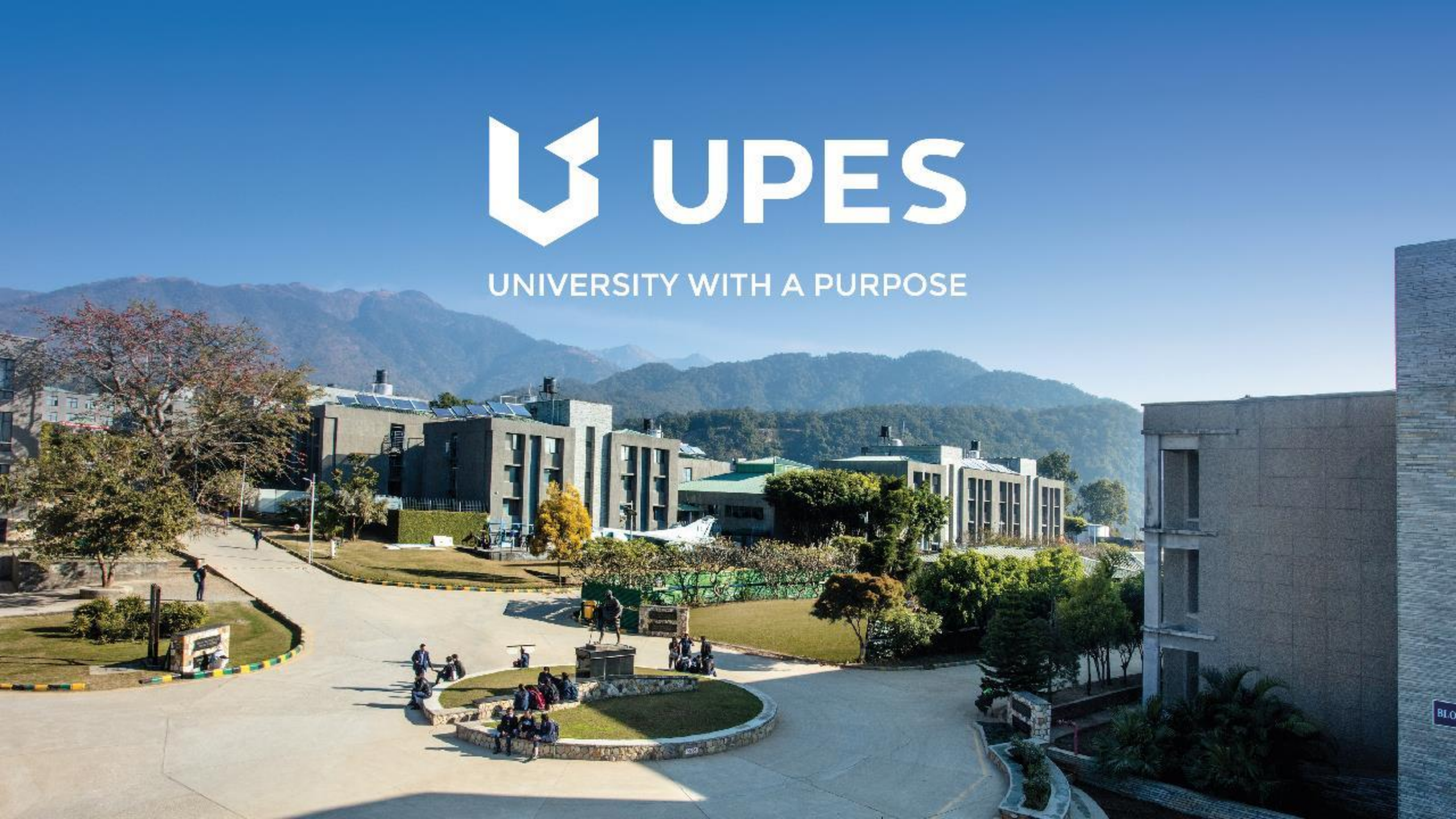
$$\mathbf{s}_{\mathrm{W}} = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

$$\mathbf{s}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^{\mathrm{T}}$$

$$= \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{y}_n, \qquad \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k.$$

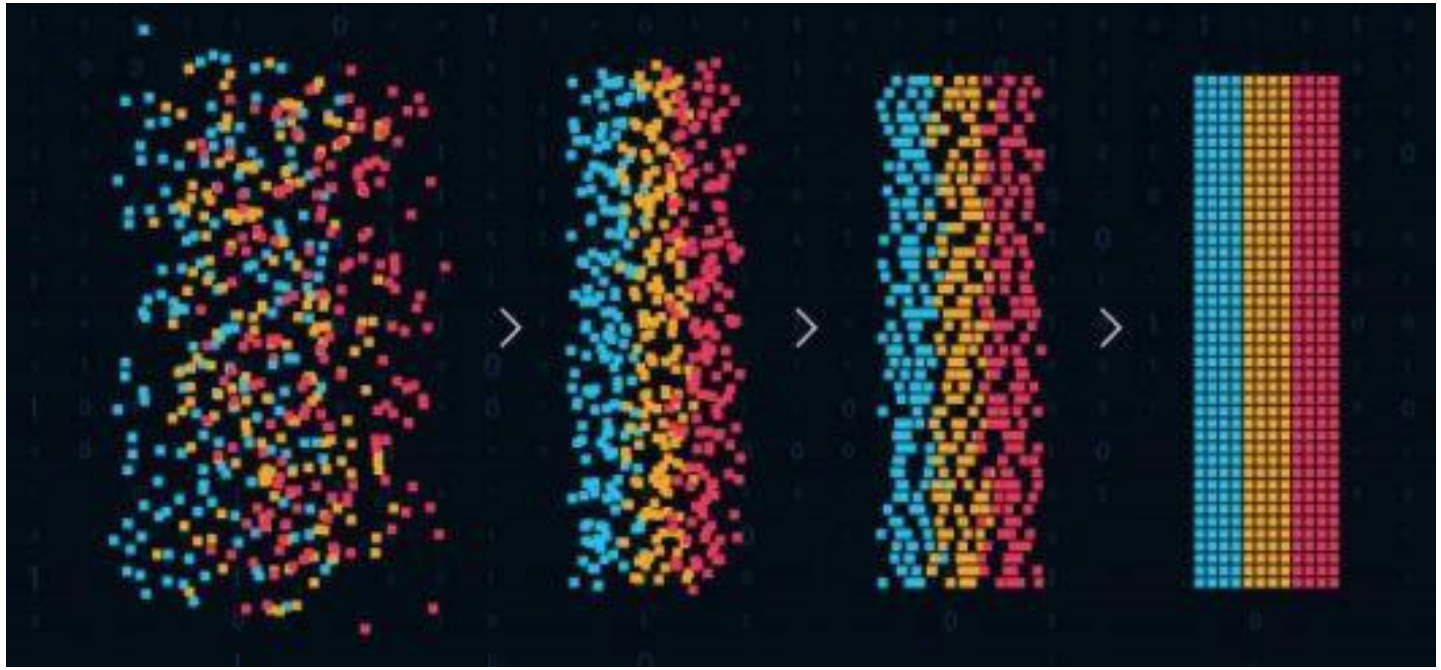
$$J(\mathbf{w}) = \mathrm{Tr}\left\{ (\mathbf{W}\mathbf{S}_{\mathrm{W}}\mathbf{W}^{\mathrm{T}})^{-1} (\mathbf{W}\mathbf{S}_{\mathrm{B}}\mathbf{W}^{\mathrm{T}}) \right\}$$

Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

9/11/2021

Recap: Linear Models for Classification

- **Classification:** Assign input x to one of the class labels
- Why we call it linear?
- Linearly separarble dataset and Hyperplanes
- **Linear Discriminant Functions**
  - **Two class**
  - **Multiclass**
    - **3 approaches:** one-versus-the-rest, one-versus-one, and k classifiers
    - Decision region ambiguities
- **Fisher's Linear Discriminant**
  - Projection algorithm

Li-Mod for Classification: Perceptron Algorithm

- The perceptron of Rosenblatt corresponds to a two-class model in which the input vector **x** is first transformed using a fixed nonlinear transformation to give a feature vector $\boldsymbol{\phi}(\mathbf{x})$
- Then construct a generalized linear model of the form

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})\right)$$

- Where the nonlinear activation function *f*(.) is a step function of the form
- Non-probabilistic model

$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0 \end{cases}$$

Li-Mod for Classification: Perceptron Algorithm

- How to compute w?
- Standard error function doesn't work
- Alternate error function: **Perceptron criterion**
- Goal: if xn belongs to class C1 then $\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) > 0$,

iElse f xm belongs to class C2, then $\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) < 0$

- Also, $t \in \{-1, +1\}$
- In both conditions $\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)t_n > 0$ holds true.
- Therefore minimizing the error of the form would be beneficial

$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}_n t_n$$

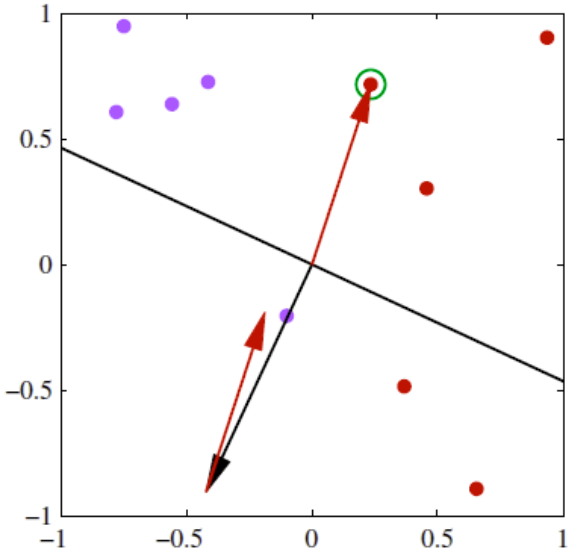
Li-Mod for Classification: Perceptron Algorithm

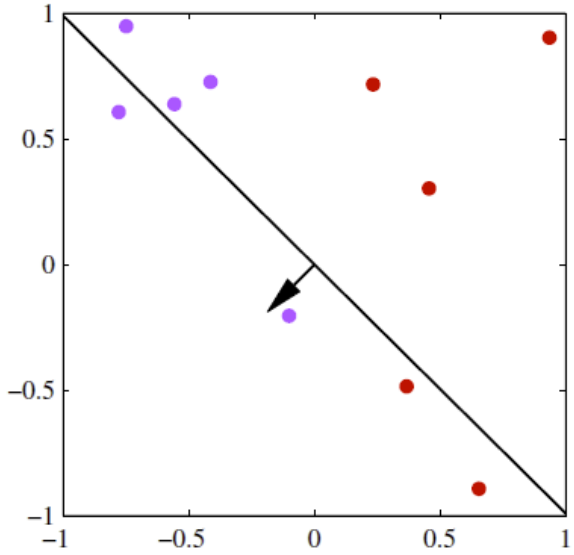
- How to compute w?
- Therefore minimizing the error of the form $E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}_n t_n$ to obtain w would be beneficial
- Applying stochastic gradient descent w takes

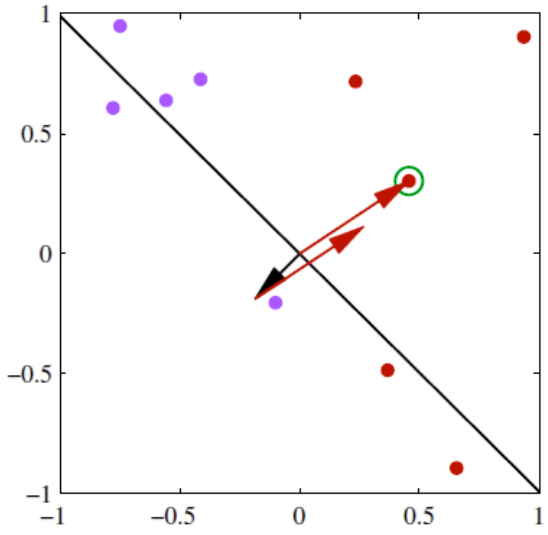
$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \boldsymbol{\phi}_n t_n$$

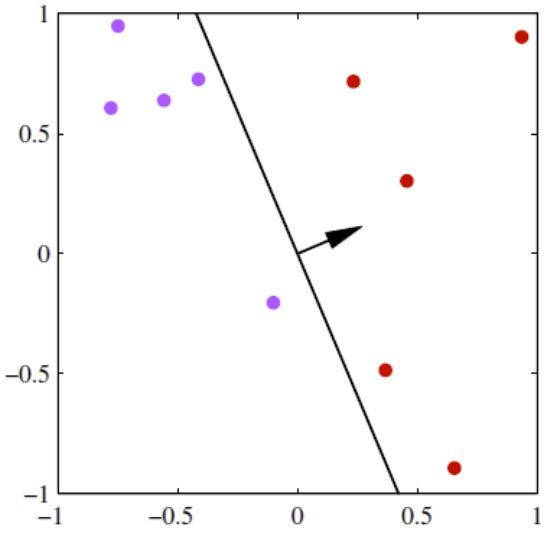
- This is the perceptron learning algorithm.

Li-Mod for Classification: Perceptron Algorithm

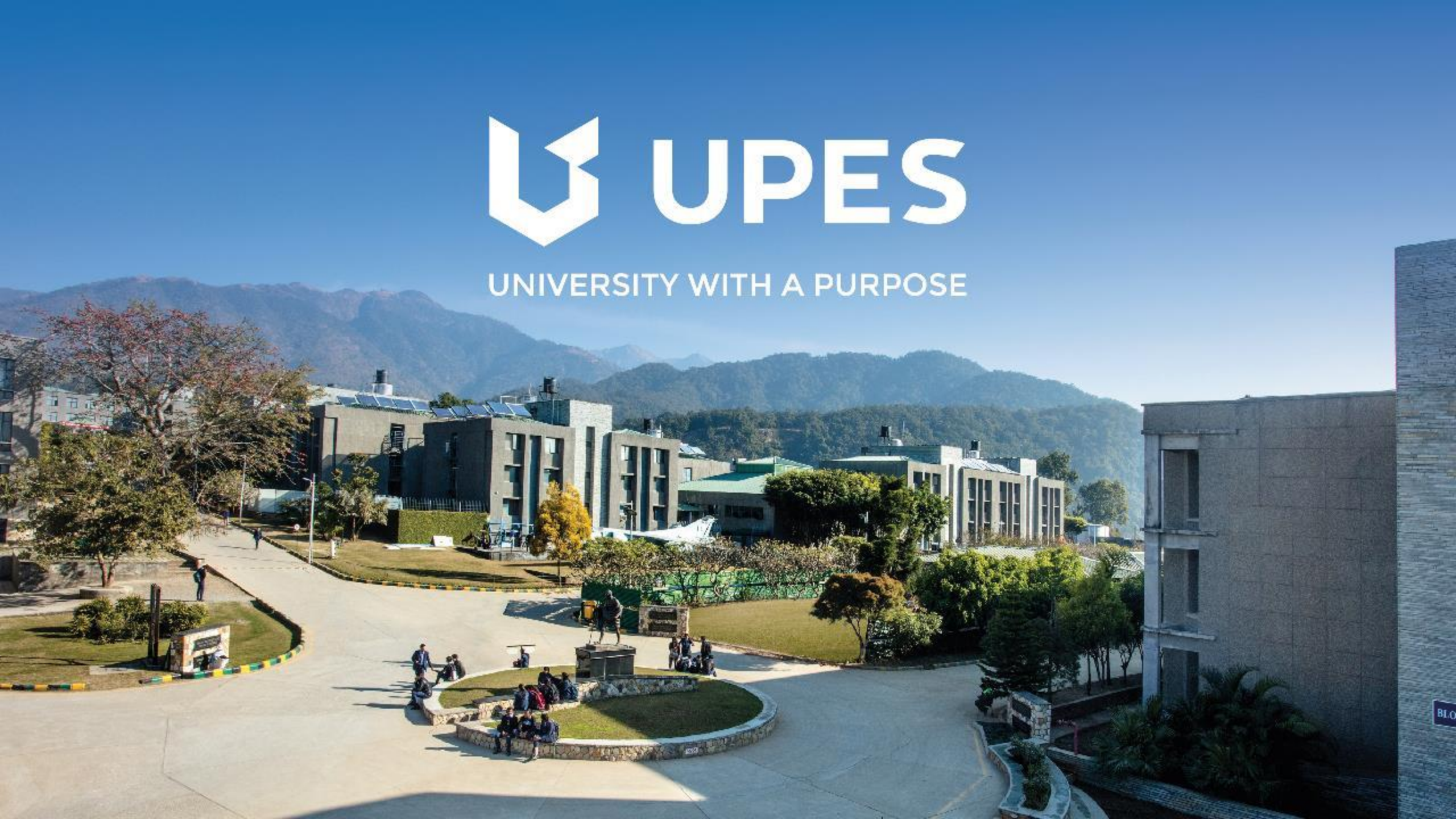






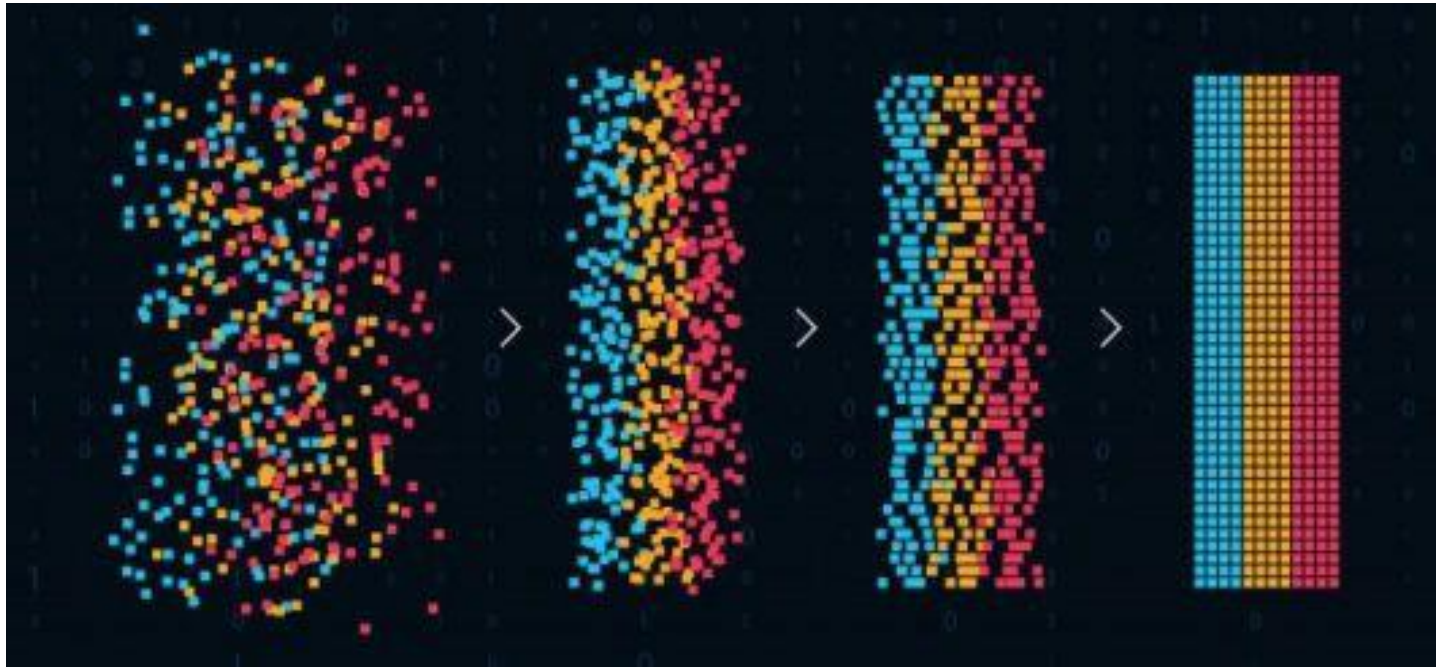


Thank You


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



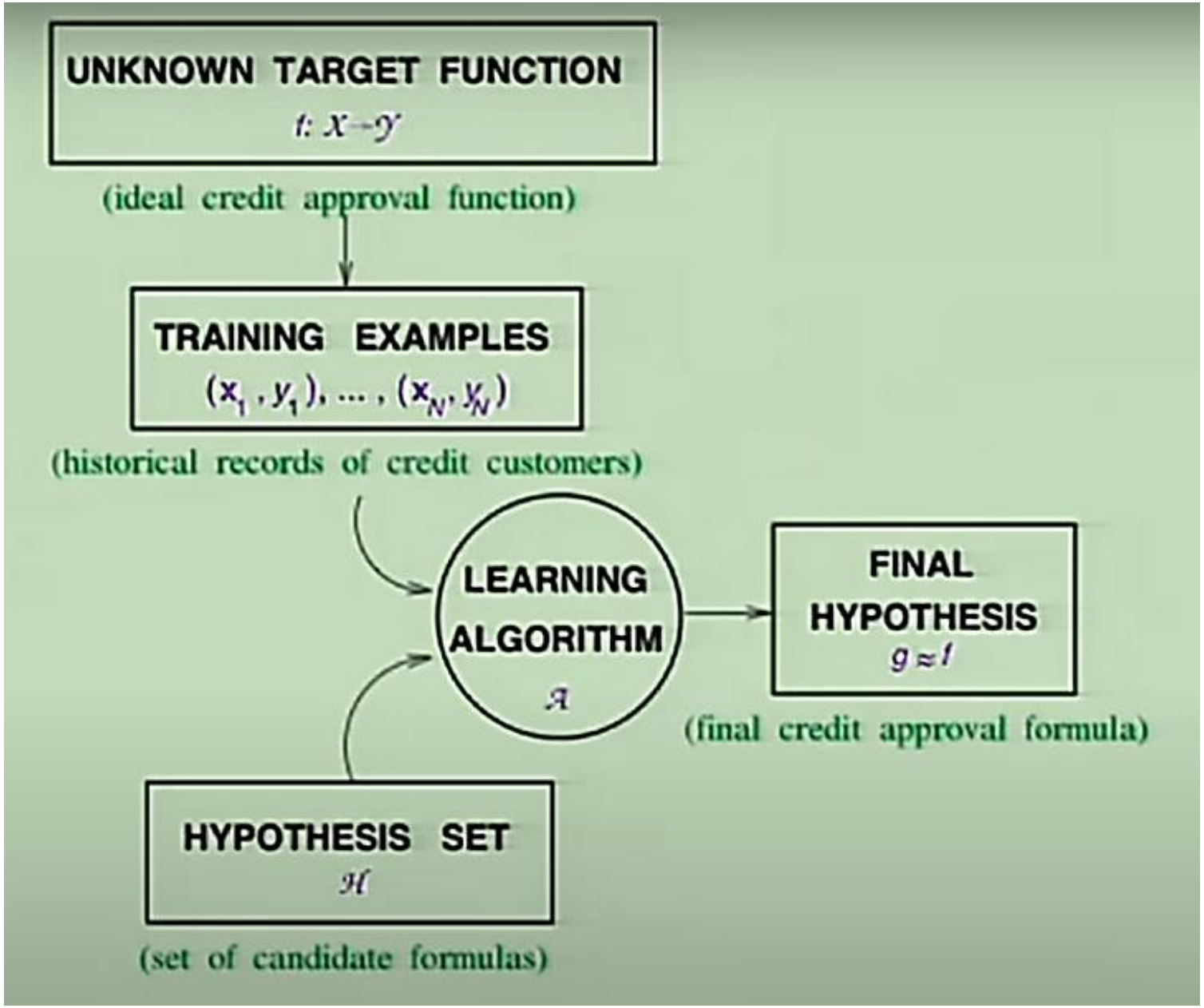
Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

9/11/2021

Pattern recognition: In a Perspective



Neural Networks

- What's wrong with models so far?
  - Curse of Dimensionality
- Solution: Adapt basis function approach
- Neural Networks provide such a approach
- Key concept in NNs: Fix the number of basis functions in advance but keeping them adaptive.
- Parametric basis functions with parameters are set during training.
  - Sigmoid, Gaussian
- Feed-forward Nnsor multilayer perceptrons are discussed here

Neural Networks

- Misnomer: Multilayer perceptron is not multiple layers of perceptron models but multiple layers of logistic regression models

- Advantage: Compact and fast
- Disadvantage: likelihood function is Non-convex function of model parameters (w).

Next time: Neural Networks

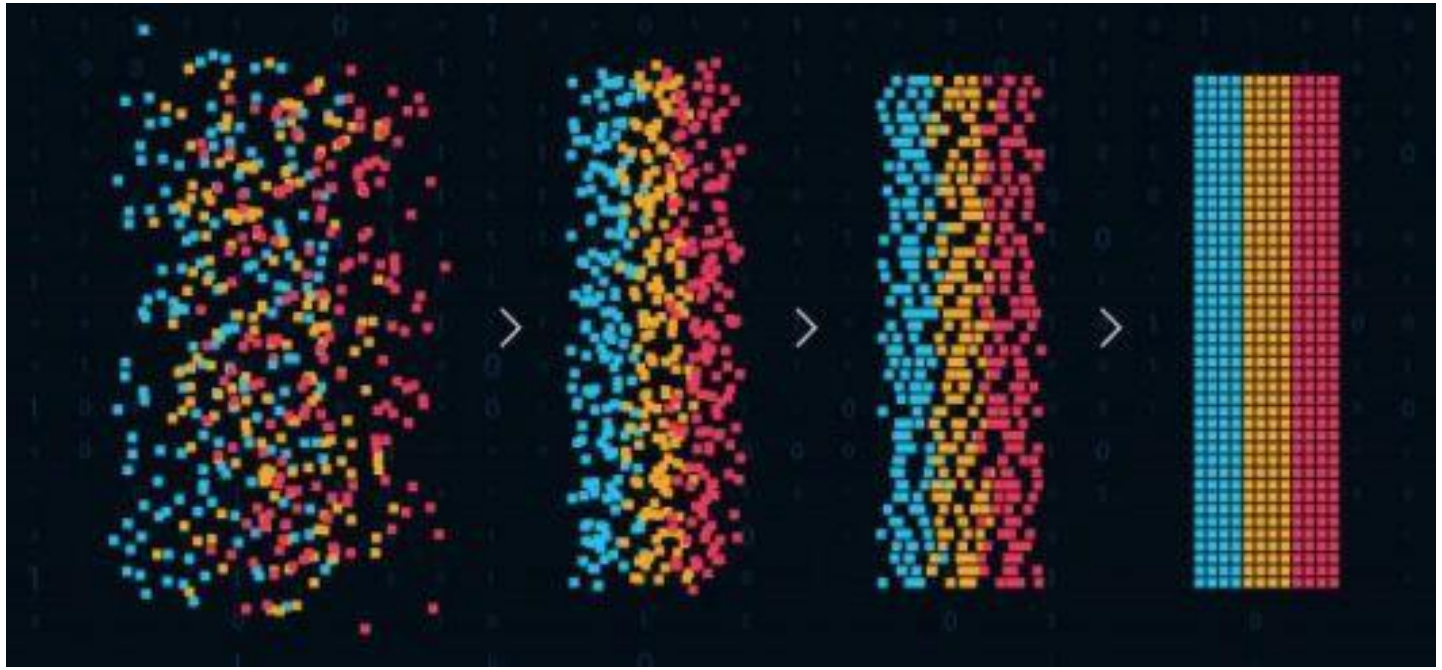
Thank You

UPES

UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

11/11/2021

Neural Networks

- Output of form $$y(\mathbf{x}, \mathbf{w}) = f\left(\sum_{j=1}^{M} w_j \phi_j(\mathbf{x})\right)$$
- In linear models of classification,
  - fixed number of basis functions
  - f is non-linear function (decision function)
- In linear models for regression
  - Fixed number of basis functions
  - f is identity function
- Neural networks
  - Basis functions are parameter based and these parameters are learned during training f is non-linear function (decision.
  - Many ways to form parametric basis functions

Neural Networks

- M different linear combinations for D dimensional input.
- ‘a’ is activation and w is weights

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

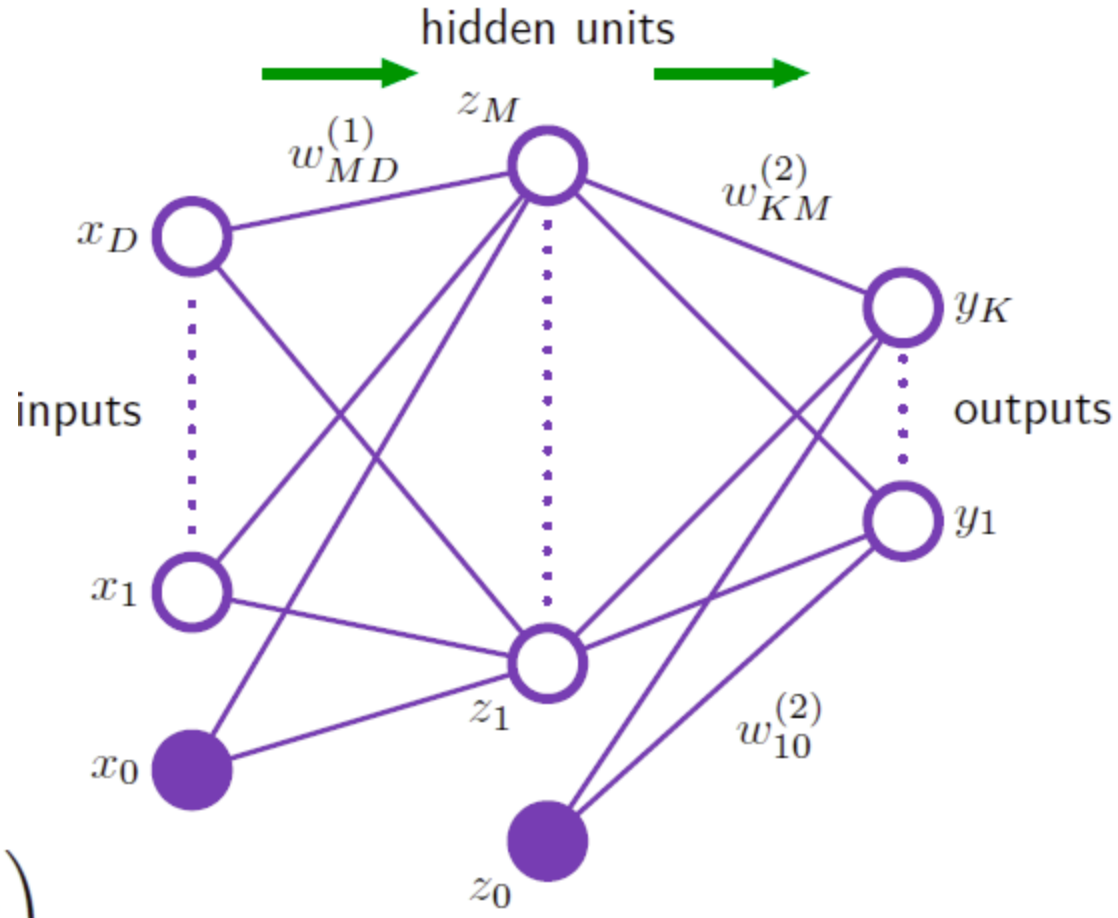
- ‘h’ is activation function.
- z is the output after activation function (hidden unit)

$$z_j = h(a_j).$$

- Repeat

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

Neural Networks

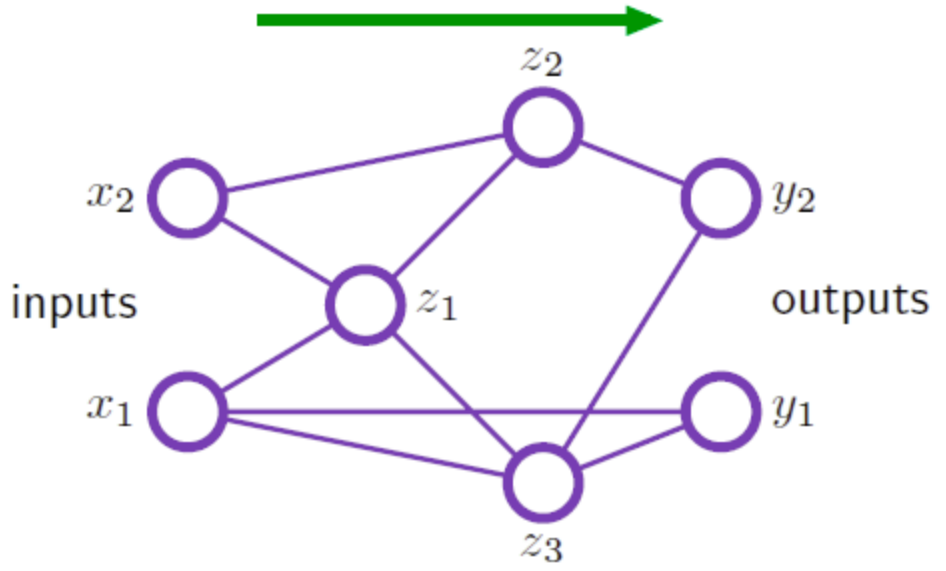


$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$

Neural Networks

- MLP: continuous sigmoidal non-linear function
- Perceptron: step-function non-linear function
- NNs can have other activation functions
- General architecture

$$z_k = h\left(\sum_j w_{kj} z_j\right)$$



Neural Networks: Network Training

- Find w?????
- Intuitive approach: Sum-of-squares error and its minimization with least squares.
- Alternative and more general approach: Probabilistic interpretation
  - Consider Gaussian
  - Create maximum likelihood
  - Either maximize it or minimize the negative of it.
  - We also know relationship between maximum likelihood and sum-of-squares error
  - Use this relationship to create error functions

Neural Networks: Network Training

- Error Functions in NN based models
- 1. Regression: When the output layer of the NN has linear or identity activation function

$$\frac{\partial E}{\partial a_k} = y_k - t_k$$

- 2. Classification
  - Binary: Two class

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

  - Multiple two-class

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} \{t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln(1 - y_{nk})\}$$

Neural Networks: Network Training: Errors

- Error Functions in NN based models
- 1. Regression: When the output layer of the NN has linear or identity activation function

$$\frac{\partial E}{\partial a_k} = y_k - t_k$$

- 2. Classification
  - Binary: Two class

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

  - Multiple two-class

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} \{t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln(1 - y_{nk})\}$$

  - Multi-class

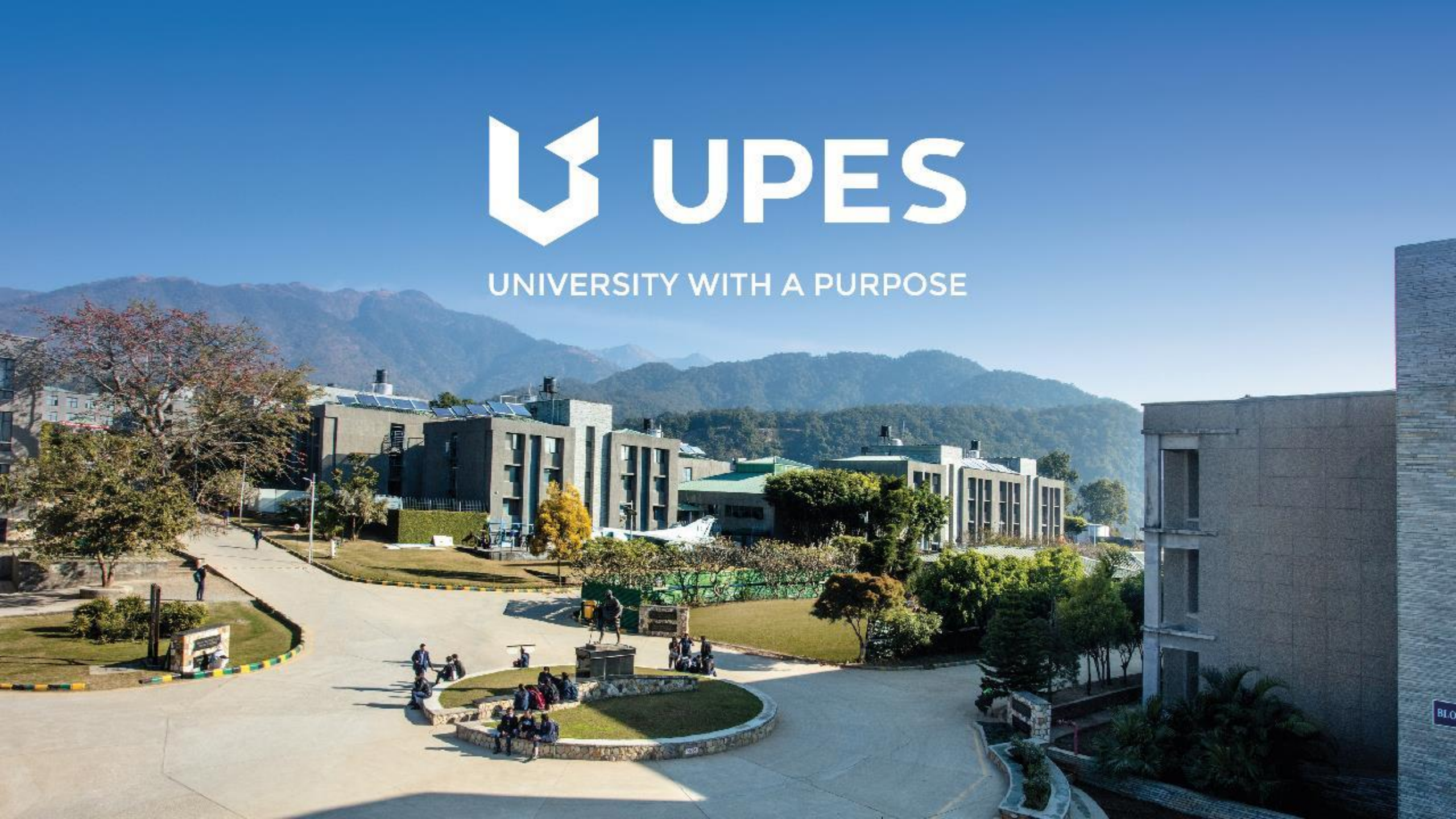
$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}).$$

Next time: Neural Networks

Thank You

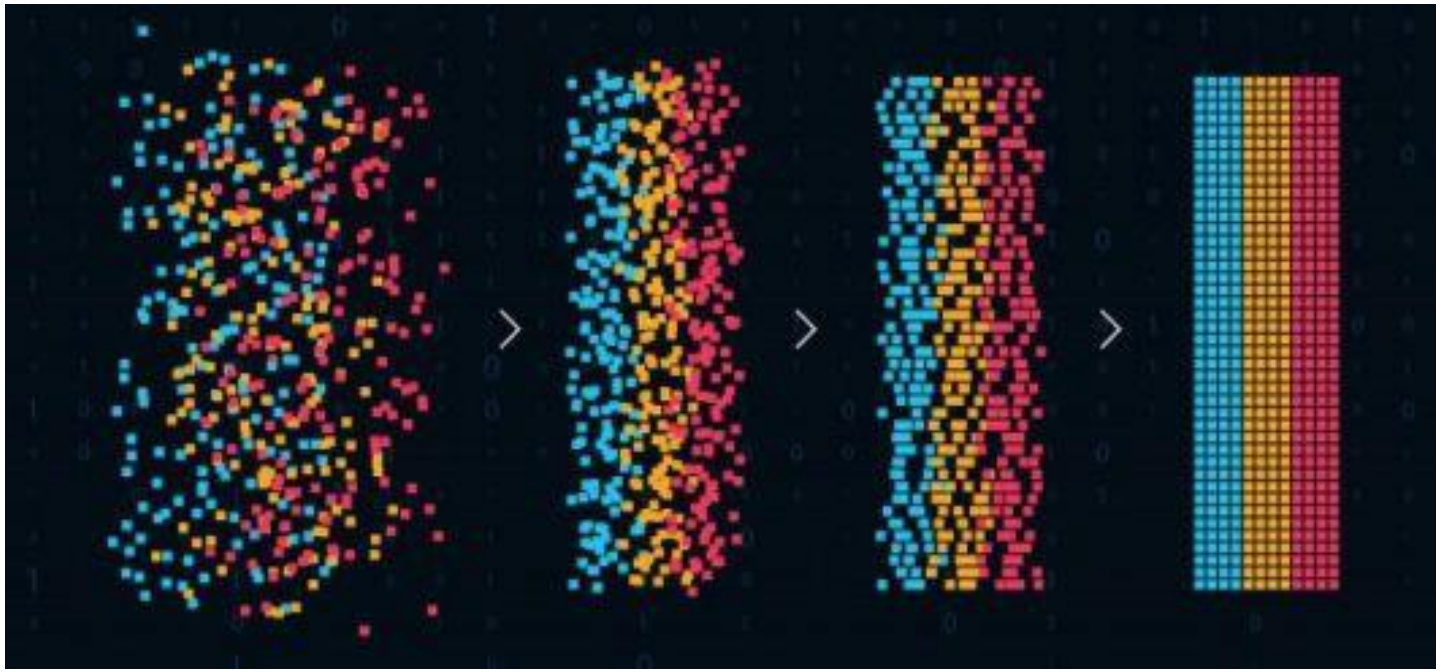
UPES

UNIVERSITY WITH A PURPOSE


UPES
UNIVERSITY WITH A PURPOSE

UPES
UNIVERSITY WITH A PURPOSE

Pattern and Anomaly Detection



Source: Edureka

B. Tech., CSE + AI/ML

Dr Gopal Singh Phartiyal

16/11/2021

Neural Networks: Network Training: Finding Weights

- Use of gradient information

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)})$$

- **Gradient descent** algorithm : change w in a way that a small step is taken in the direction of negative gradient.
  - At each step, the weight vector is moved in the direction of the greatest rate of decrease of the error function
  - After each step, the gradient is re-evaluated and it goes on again and again until termination criteria s met.
  - In batch type gradient descent methods: weights are updated only after the model has seen all training samples once and only once.
  - Conjugate gradients and quasi-Newton are variants of batch gradient descent.
  - Issue: Local minima, all points are required, computationaly expensive

Neural Networks: Network Training: Finding Weights

- **Stochastic GD:**
- **Motivation:-**Updating weight on all training samples in one go is equivalent to updating weights after training with one sample at a time.

$$E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w}).$$

- Advantages: Easier to implement
- Computationally cheap
- Randomization leads to generalization

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)}).$$

Neural Networks: Network Training: Finding Weights

- **Error Backpropagation**
- Method of evaluating the gradient of error function in a feedforward neural network
- Local message passing scheme (forward and backword propagation of information)
- To understand
  - Forward pass (output based on updated weights)
  - Backward pass (backpropagation of error)
  - Weight update
- In general, backpropagation can be used elsewhere.

Neural Networks: Network Training: Errors

- Error Functions in NN based models
- 1. Regression: When the output layer of the NN has linear or identity activation function

$$\frac{\partial E}{\partial a_k} = y_k - t_k$$

- 2. Classification
  - Binary: Two class

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

  - Multiple two-class

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} \{t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln(1 - y_{nk})\}$$

  - Multi-class

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}).$$

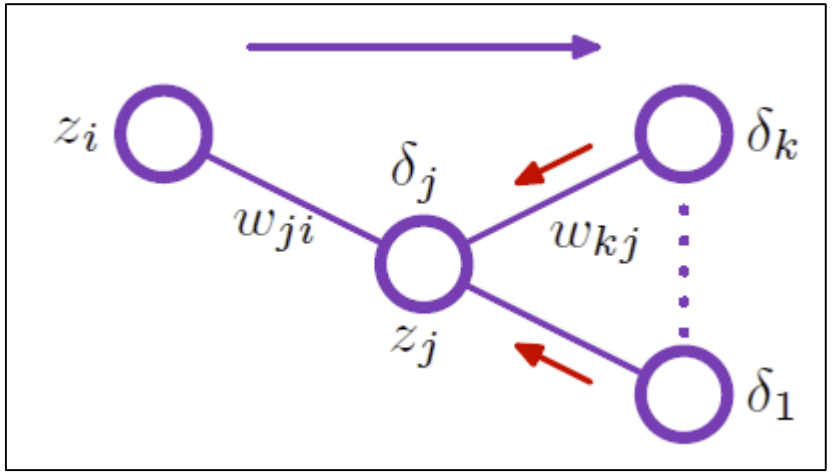
Neural Networks: Network Training: Finding Weights

- **Error Backpropagation:**
- For one sample

$$E_n = \frac{1}{2}\sum_k (y_{nk} - t_{nk})^2$$

$$\frac{\partial E_n}{\partial w_{ji}} = (y_{nj} - t_{nj})x_{ni}$$

$$a_j = \sum_i w_{ji} z_i$$



$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}}.$$

$$\frac{\partial a_j}{\partial w_{ji}} = z_i.$$

$$\boxed{\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i.}$$

Neural Networks: Network Training: Finding Weights

- **Error Backpropagation**
- .

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j}$$

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k$$

- For batch methods

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \frac{\partial E_n}{\partial w_{ji}}.$$

Next time: Neural Networks

Thank You

UPES

UNIVERSITY WITH A PURPOSE