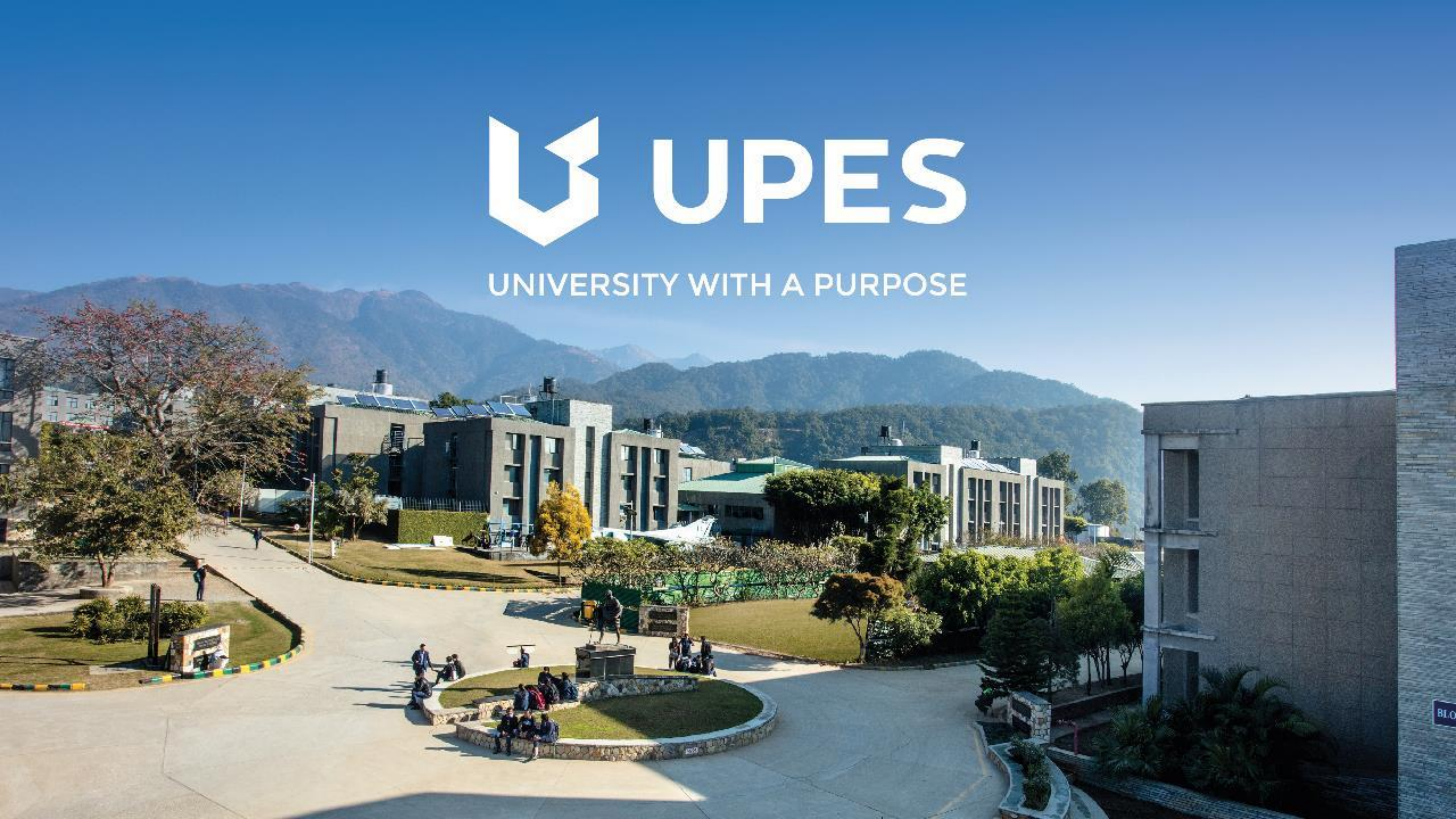
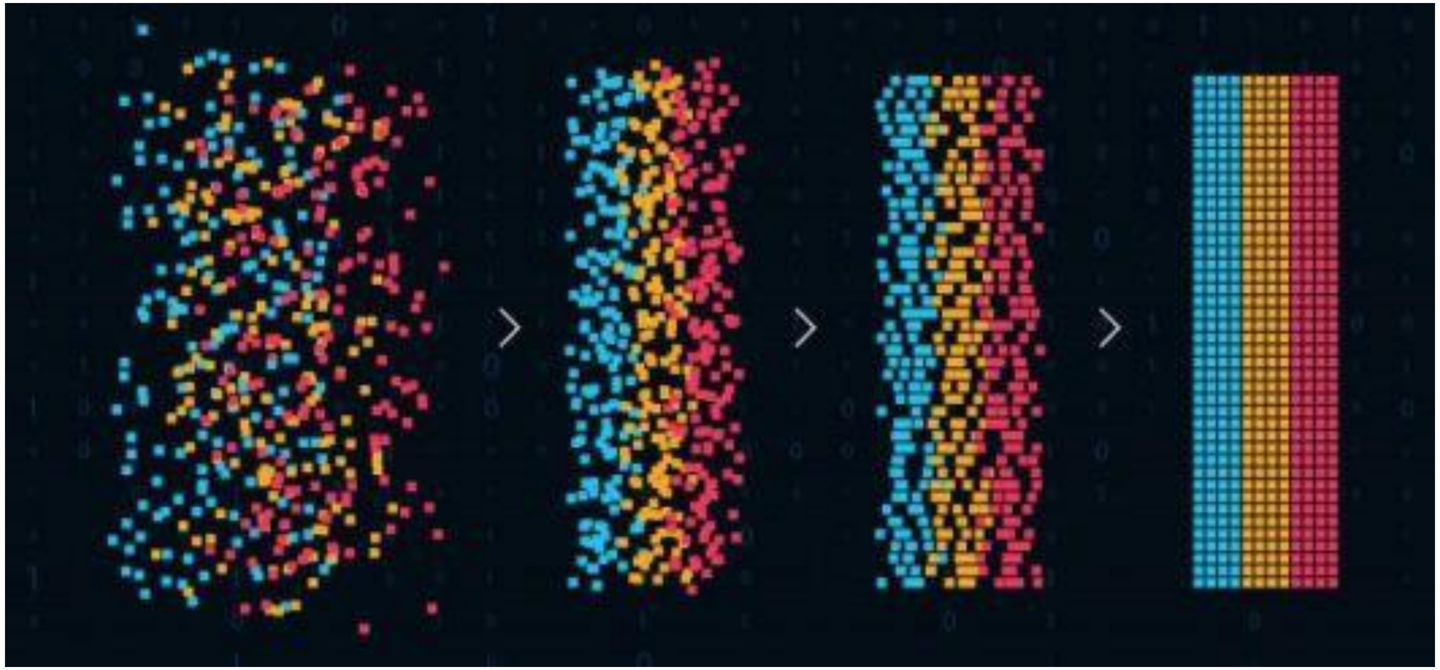

UPES
UNIVERSITY WITH A PURPOSE

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

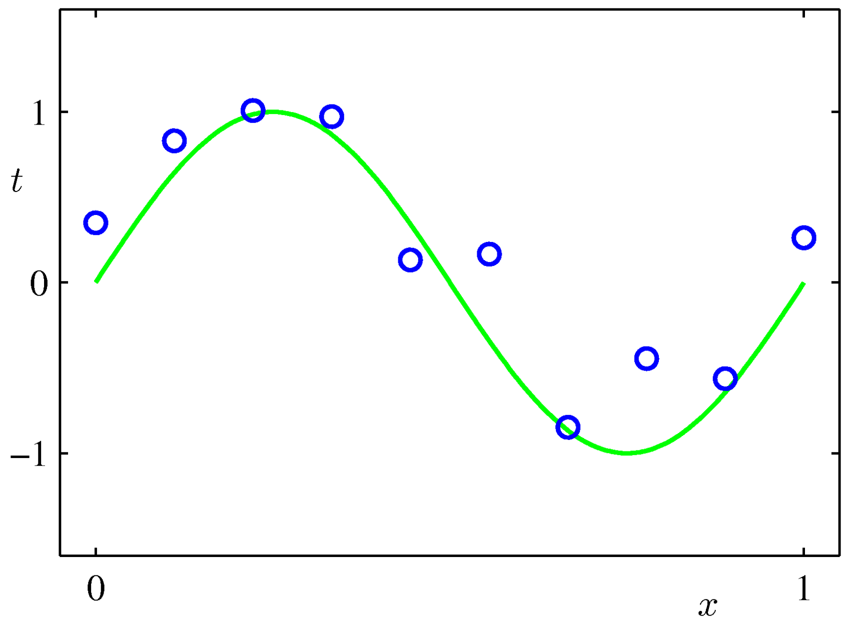
$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$$

- In vector form?

Basis Functions based Linear Models for Regression

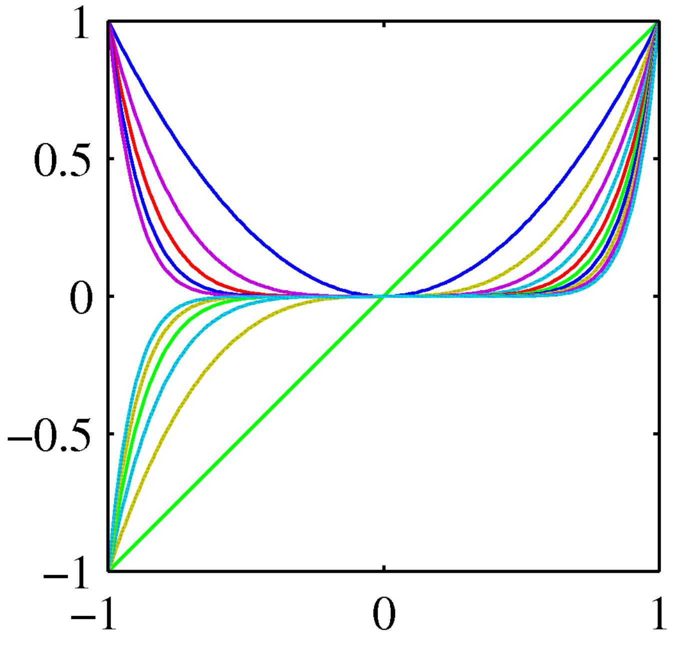
- Example: Polynomial Regression (Non-linear basis functions)
- Polynomial basis functions

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



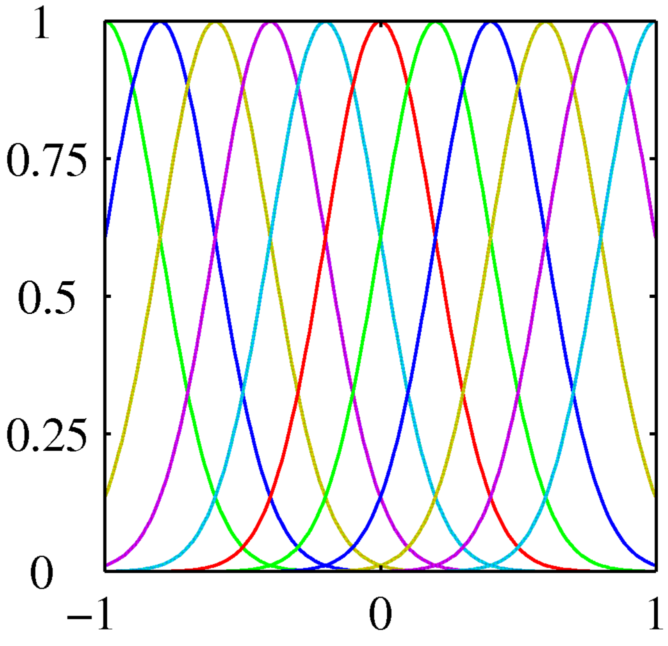
- Limitation of polynomial basis function:
  - Global impact
- Spline functions: Dividing input space into regions and using different polynomials for different regions

Basis Function: Examples



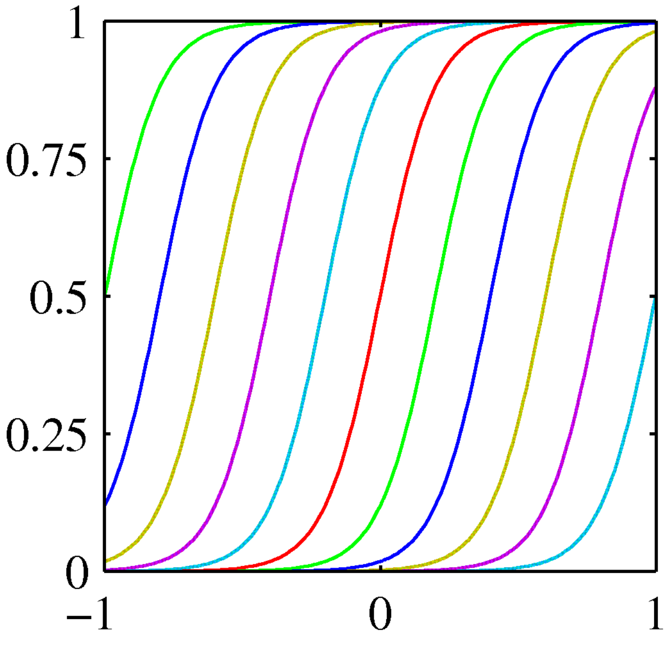
$$\phi_j(x) = x^j.$$

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

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:

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

- 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

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

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

$$
\begin{aligned}
w_0 &= \bar{t} - \sum_{j=1}^{M-1} w_j \overline{\phi_j} \\
&= \frac{1}{N}\sum_{n=1}^{N} t_n - \sum_{j=1}^{M-1} w_j \frac{1}{N}\sum_{n=1}^{N} \phi_j(\mathbf{x}_n).
\end{aligned}
$$

- 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