What we have seen so far are Frequentist Methods.

Unknown Constant $\rightarrow \theta$

$$\ell(\theta) = \log P(data; \theta)$$

$$-\ell(\theta) = \text{loss Function}$$

$$\hat{\theta} = \arg\min_{\theta} -\ell(\theta)$$

$$= \arg\max_{\theta} \ell(\theta)$$

Bayesian

θ - unobserved Random Variable

$\theta \sim$ Prior Distribution

X (Data) $\sim P(x|\theta)$

$$\underbrace{P(\theta|x)}_{\text{Posterior}} = \frac{P(x|\theta)\,P(\theta)}{P(x)}$$

$$= \frac{P(x|\theta)\,P(\theta)}{\int_{\theta} P(x|\theta)\,P(\theta)\,d\theta}$$

In Supervised ML

$\theta \sim \text{Prior}$

$y \sim P(y|x,\theta)$

$\theta \perp x$

$$P(\theta|x,y) = \frac{P(y|x,\theta)\,P(\theta)}{P(y|x)}$$

$\boxed{\text{Posterior}}$

Proof

$$P(\theta|x,y) = \frac{P(\theta,x,y)}{P(x,y)}$$

tells us how likely θ is given x, y

$$= \frac{P(\theta,y|x)\,P(x)}{P(x,y)}$$

$$= \frac{P(\theta,y|x)\,\cancel{P(x)}}{P(y|x)\,\cancel{P(x)}}$$

$$= \frac{P(y|x,\theta)\,P(\theta|x)}{P(y|x)}$$

$$= \frac{P(y|x,\theta)\,P(\theta)}{P(y|x)}$$

Knowing about x tells us nothing.

Posterior Predictive

$*$ → Test set

about θ, we need both x, y to update θ

$$P(y_*|X,y,X_*) = \int P(y_*|x_*,\theta)\, p(\theta|x,y)\,d\theta$$

$$= \mathop{E}_{\theta\sim p(\theta|x,y)}\left[P_*|x_*,\theta\right]$$

(for point estimates $\hat{y}$ $E(E(P_*|x_*,\theta))$

M1: $\theta^{(1)}$

M2: $\theta^{(2)}$

M3: $\theta^{(3)}$

tells us how confident is our model

We are taking predictions from all models & take weighted average

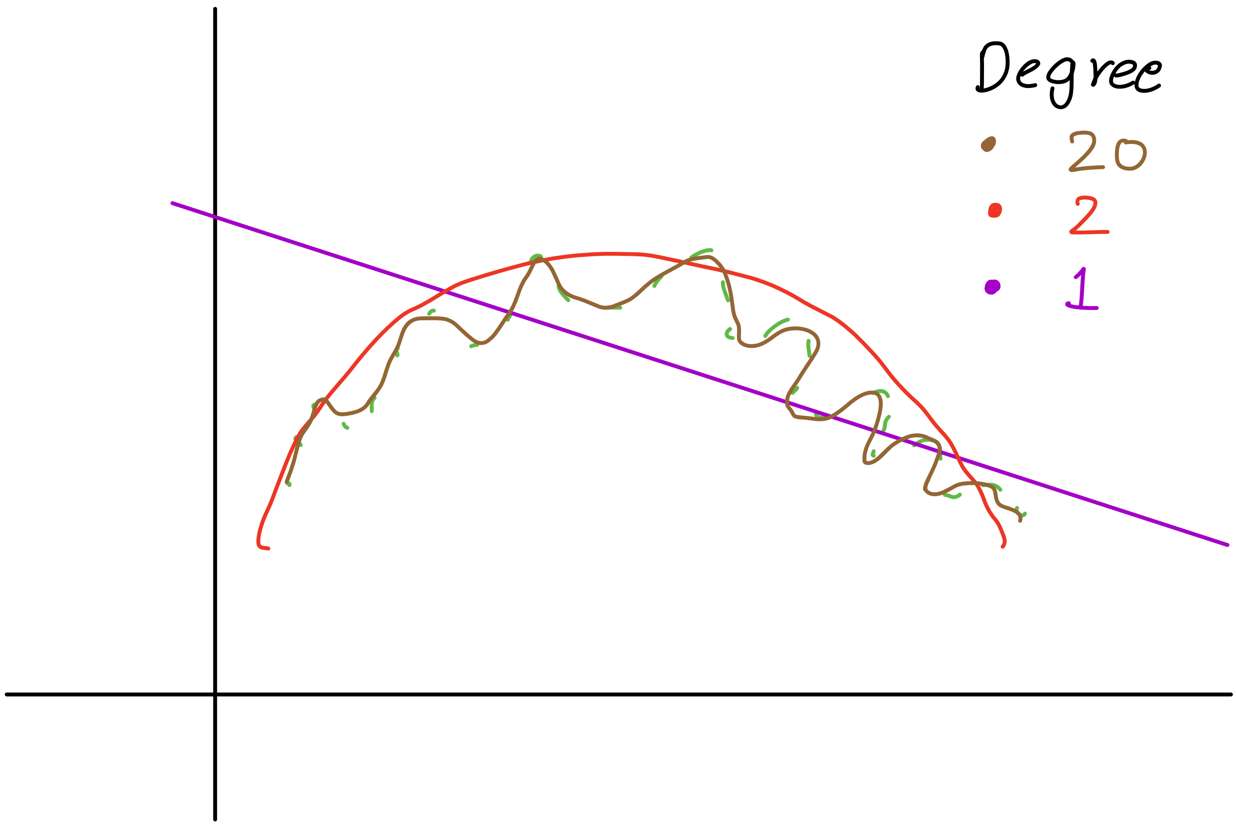
overall models acc.
to posterior

The above approach is parametric because $p(y|x;\theta)$ has some functional form of which θ is parameter

eg. $y|x;\theta = \dfrac{1}{1+e^{-\theta^T x}}$

So, we have less degree of freedom as we can vary only θ & are bound by the function
Functional form prevents us from adapting to other family

of function no matter how much data is given. Gives us a low degree of variance



So, we may have data which requires degree 20 polynomial to fit but we choose parametric approach where linear regression is functional form then we can't fit data well.

Bayesian Linear Regression

$S = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ $\qquad \theta \in \mathbb{R}^d$

$\qquad\qquad\qquad\qquad\qquad x^{(i)} \in \mathbb{R}^d$

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

$$\varepsilon^{(i)} \sim N(0, \sigma^2)$$

$$\theta \sim N(\vec{0}, \underbrace{\tau^2 I}_{\text{diagonal covariance matrix}})$$

$$\theta | S \sim N\left(\frac{1}{\sigma^2} A^{-1} X^T \vec{y}, A^{-1}\right)$$

$$\text{where } A = \frac{1}{\sigma^2} X^T X + \frac{1}{\tau^2} I$$

This term has regularising effect. We are basically adding +ve values to diagonal so as to ↑ make all eigenvalues +ve.

$$\theta | S \sim N\left(\left(X^T X + \boxed{\frac{\sigma^2}{\tau^2} I}\right)^{-1} X^T y, \left(\frac{1}{\sigma^2} X^T X + \frac{1}{\tau^2} I\right)^{-1}\right)$$

$\bar{R}^2$

$$\boxed{(X^T X)^{-1} X^T y} \leftarrow \text{Normal Equation}$$

The mean is quite similar to N·E.

posterior [Model fitting]

[Bayesian approach gives us measure of uncertainity.
If we have a lot of data, posterior is peaked, else flat.
You get uncertainity estimate for free]

$$y_* | X_*, S \sim N\left(\frac{1}{\sigma^2} x_*^T A^{-1} X^T \vec{y},\ x_*^T A^{-1} x_* + \sigma^2\right)$$

$$\left[\ \theta \sim N(\mu, \Sigma)\right.$$

$$\hat{y} = x_*^T \theta \sim N(\mu^T x_*, x_*^T \Sigma x) \rightarrow \text{This}$$

$y^* = \hat{y} + \hat{\varepsilon}$

is what we do for prediction]

This σ^2 is added to account for variance for y^* (test example) due to i.i.d. assumption

GAUSSIAN PROCESSES

Vector: functions!! MV Gaussian:

Gaussian Process

Properties of MV Gaussian

① Normalization

$$\int_X P(X;\mu,\Sigma)\,dx = 1$$

② $$x = \begin{bmatrix} x_A \\ x_B \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}, \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \right)$$

$$P(x_A) = \int_{x_B} p(x;\mu,\Sigma)\,dx$$

$$= N(\mu_a, \Sigma_{AA})$$

Marginalized
[still Normal!!]

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix}, \begin{array}{cc} & 2 \\ 2 & \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1d} \\ \vdots & & \vdots \\ \Sigma_{d1} & \cdots & \Sigma_{dd} \end{bmatrix} \end{array} \right)$$

Removed

(3) Conditioning

$$x_A \mid x_B \sim N\left(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B),\ \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA} \right)$$

See in scalar setting

$$\begin{bmatrix} a \\ b \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_a \\ \mu_B \end{bmatrix}, \begin{bmatrix} \sigma_a^2 & \rho\sigma_a\sigma_b \\ \rho\sigma_a\sigma_b & \sigma_b^2 \end{bmatrix} \right)$$

$$a|b \sim N\left(\underline{\mu_A + \sigma_a}\boxed{\rho}\boxed{\left(\frac{b-\mu_b}{\sigma_b}\right)},\ \underline{\sigma_a^2(1-\rho^2)}\right)$$

Z-value for b

Corelation coefficient

Rescaling back to A

Knowing B, decreases variance

④ Summation

$x \sim N(\mu_1, \Sigma_1)$

$y \sim N(\mu_2, \Sigma_2)$

$$x+y \sim N(\mu_1+\mu_2, \Sigma_1+\Sigma_2)$$

M.V. Gaussian

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix}, \begin{bmatrix} \quad \end{bmatrix} \right)$$

Gaussian Process

$$\begin{bmatrix} | \\ f \\ | \end{bmatrix} \sim GP\left(\begin{bmatrix} m \end{bmatrix}, \begin{bmatrix} \text{Kernel} \end{bmatrix} \right)$$

$$\Vert$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \end{bmatrix}$$

mostly 0

Marginalize out all irrelevant examples
(e.g. NOT in training & NOT in test)

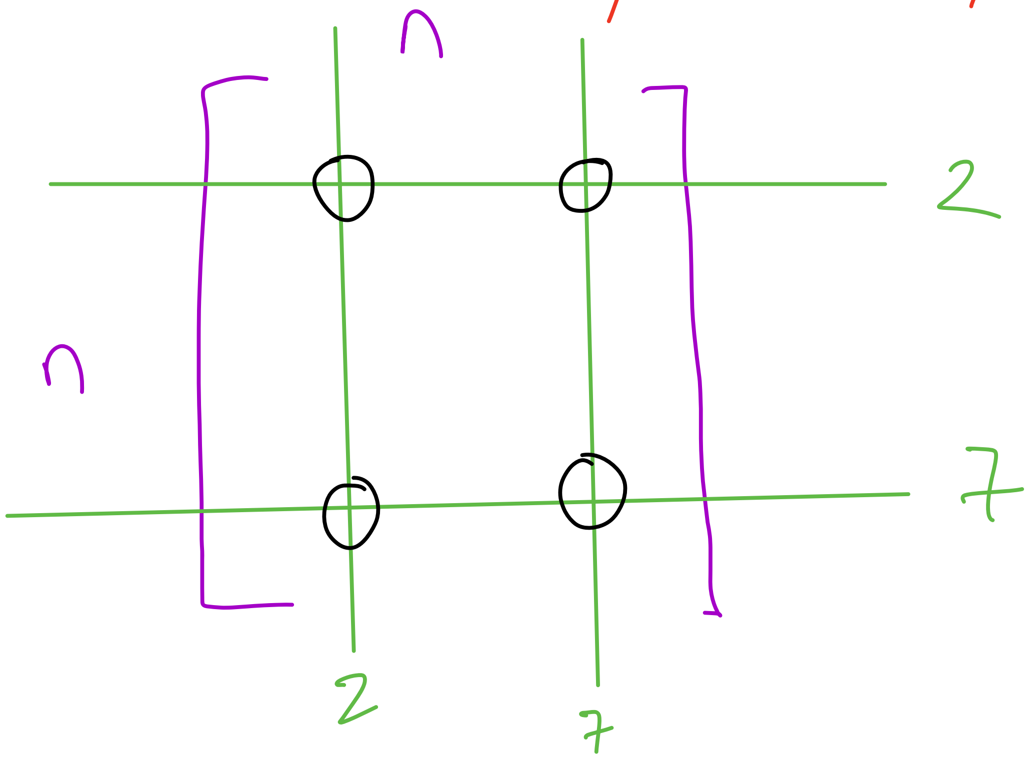
$$\begin{bmatrix} f(x^{(1)}) \\ \vdots \\ f(x^{(m)}) \\ \hline f(x_*^{(1)}) \\ \vdots \\ f(x_*^{(n_*)}) \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} 0 \end{bmatrix}, \begin{bmatrix} K(x^{(1)}, x^{(1)}) & \cdots & K(x^{(1)}, x_*^{(n_*)}) \\ & & \\ - & - & - \end{bmatrix} \right)$$

↓

PSD mercer Theorem

[What is a submatrix?

→ It is not any arbitary block!!



The four circled element make up submatrix. Index of rows & columns should be same

$$\begin{bmatrix} \vec{f} \\ \vec{f}_* \end{bmatrix} = N\left(0, \begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

Any Kernel can be used in covariance since they will us similarity metric. They are dot product in higher dimension

$$y = f(x) + \varepsilon$$

$$\begin{bmatrix} y \\ y_* \end{bmatrix} = \begin{bmatrix} f \\ f_* \end{bmatrix} + \begin{bmatrix} \varepsilon \\ \varepsilon_* \end{bmatrix}$$

$$\sim N\left(\begin{bmatrix}0\end{bmatrix}, \begin{bmatrix} K(X,X)+\sigma^2 I & K(X,X_*) \\ K(X,X_*) & K(X_*,X_*)+\sigma^2 I \end{bmatrix}\right)$$

use conditioning rule

$$Y_* \mid Y, X, X_* \sim N(\mu_*, \Sigma_*)$$

$$\mu_* = K(X_*,X)\left[K(X,X)+\sigma^2 I\right]^{-1} Y$$

$$\Sigma_* = K(X_*,X_*)+\sigma^2 I - K(X_*,X) \times \left[K(X,X)+\sigma^2 I\right]^{-1} K(X,X_*)$$

Σ_* is similar to one in conditioning

[Note: One to one relation bet^n cov & kernel. Ess. same]

[You just need to choose the right kernel, everything else is set in stone]

Gaussian Process is Non Parametric

See lec for more visualization