

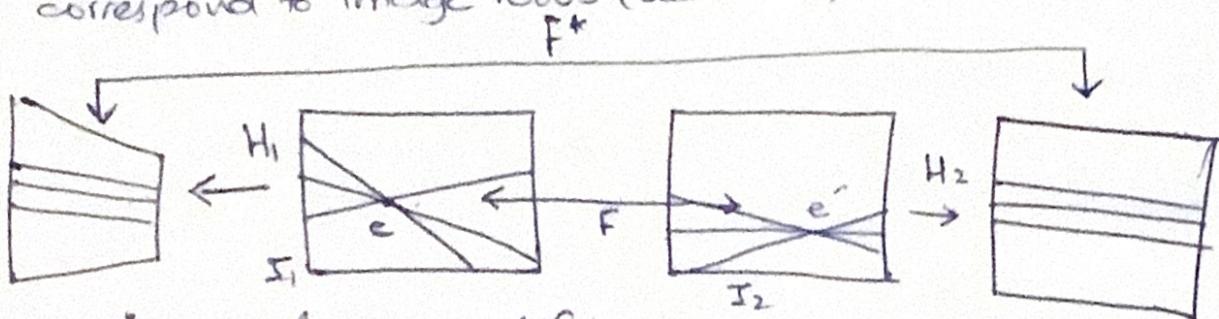
5) Let F be last column of V ($q \times 1$).
Reshape into 3×3 matrix \hat{F} .

8-point algorithm

C) Compute SVD of $\hat{F} = \hat{U}\hat{D}\hat{V}^T$. Zero out lowest singular value of D .
Then recompute $F = \hat{U}\hat{D}'\hat{V}^T$ by 3×3 diagonalization.

D) Renormalization F .

Rectification → warp images so that conjugate epipolar lines correspond to image rows (scanlines).



→ Each image has a rectifying projective transformation H applied to it.

→ What will be F^* ?

$$F^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ y' \end{bmatrix} = 0$$

$$y' - y = 0 \Rightarrow y' = y$$

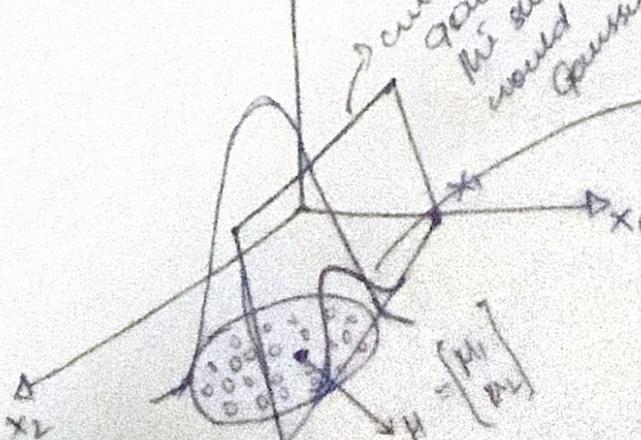
$$P_{x_1 x_2} = \frac{\text{covariance}(x_1, x_2)}{\sigma_1(x_1) \sigma_2(x_2)}$$

$$\sigma_1(x_1) = \sqrt{E(x_1^2) - \mu_1^2}$$

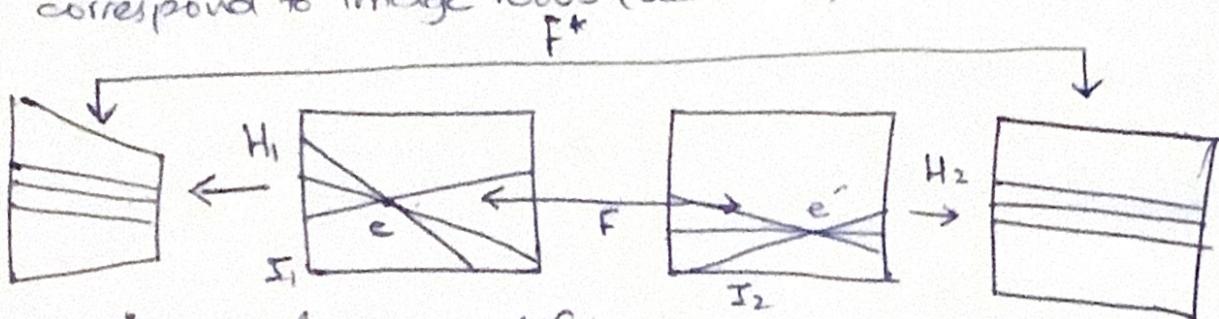
→ dot product measures similarity.
↳ if two points are similar, the dot product will be similar.

$P(x_1 x_2)$ Joint distribution

Joint distribution
We want to
be able to
model Gaussian



8-point algorithm



→ Each image has a rectifying projective transformation H applied to it.

→ What will be F^* ?

$$F^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ y' \end{bmatrix} = 0$$

$$y' - y = 0 \Rightarrow y' = y$$

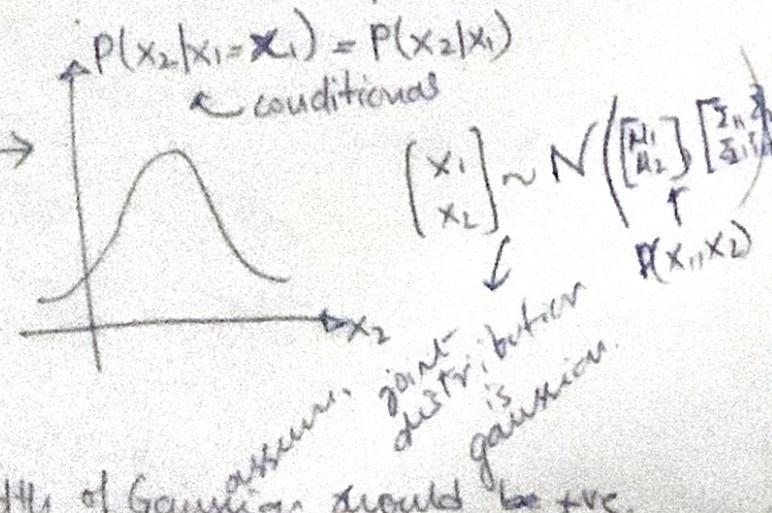
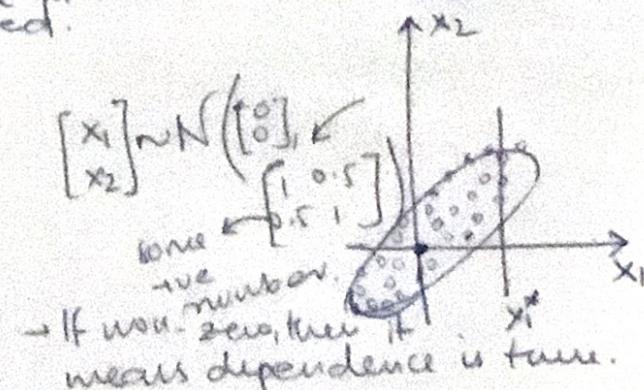
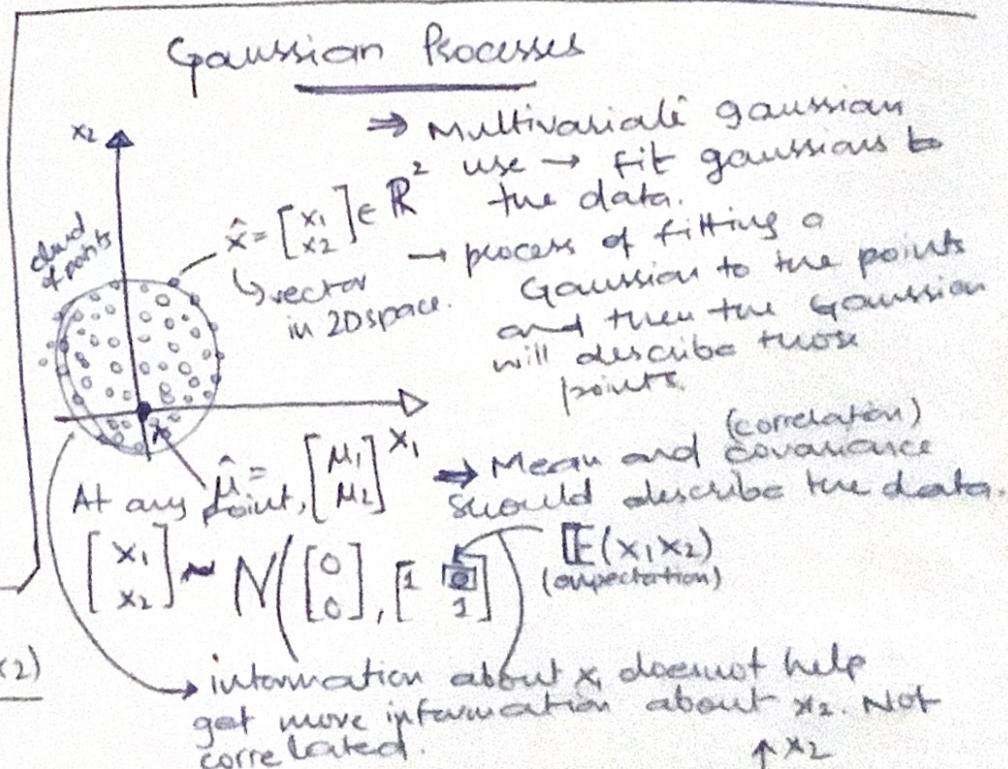
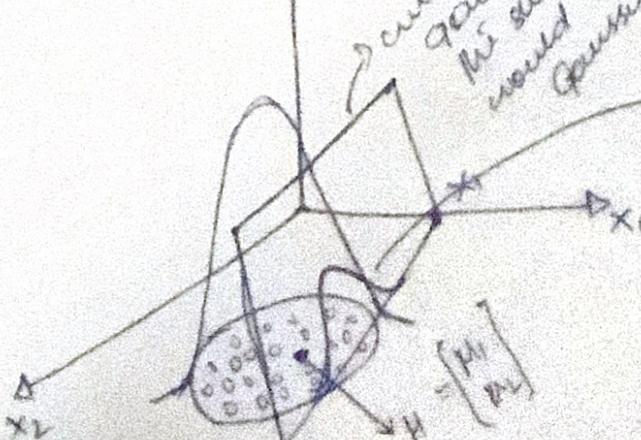
$$P_{x_1 x_2} = \frac{\text{covariance}(x_1, x_2)}{\sigma_1(x_1) \sigma_2(x_2)}$$

$$\sigma_1(x_1) = \sqrt{E(x_1^2) - \mu_1^2}$$

→ dot product measures similarity.
↳ if two points are similar, the dot product will be similar.

$P(x_1 x_2)$ Joint distribution

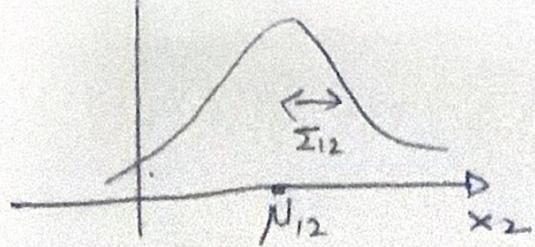
Joint distribution
We want to
be able to
model Gaussian



→ The covariance matrix should be symmetric and +ve.

→ $\mathbf{x}^T \Sigma \mathbf{x}$ has to be +ve.
→ symmetric - positive definite.

$$P(x_2 | x_1 = x_1) = P(x_2 | x_1)$$



CH#4 → Kevin Murphy ML book

→ short complement or matrix inversion lemma.

Given a Gaussian, get the conditional.

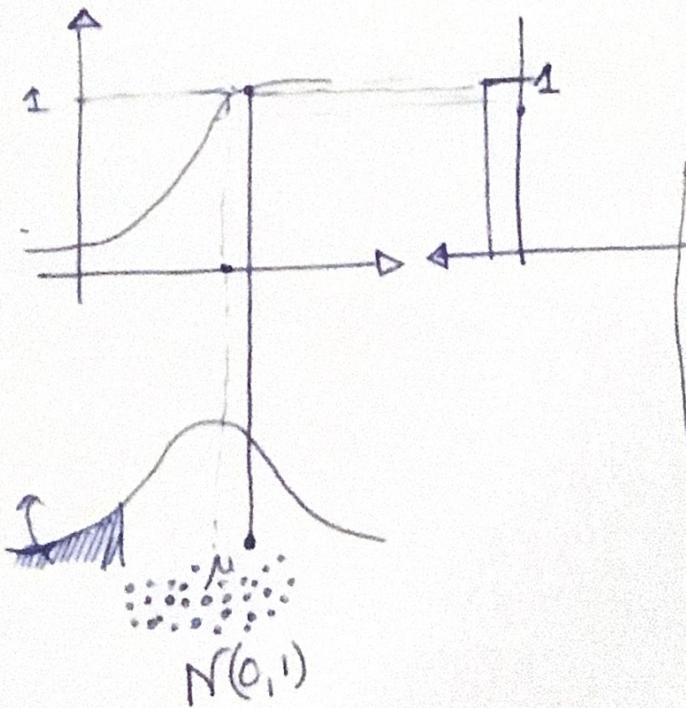
$$\mu_{12} = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (x_2 - \mu_2) \rightarrow \text{Multivariate Gaussian theorem.}$$

PROOF

$$\Rightarrow \Sigma_{12} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$$

A joint distribution can be used to get conditional distribution.

~~Cumulative of Gaussian~~ ⇒ if you start summing the area under the curve of the Gaussian as you move from left.



$x_i \sim N(0, 1)$ → univariate.
 $x_i \sim N(\mu, \sigma^2)$
 $\sim \mu + \sigma N(0, 1)$
↳ change the height

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

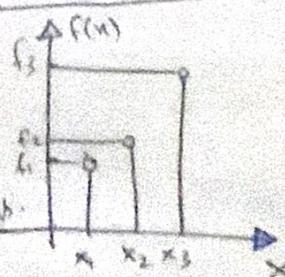
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

$$\hat{x} \sim N(\mu, \Sigma) \quad x = \mu + L N(0, I)$$

→ Cholesky decomposition

$$\Rightarrow \Sigma = LL^T \rightarrow low \text{ matrices}$$

→ $x \in$ given, want to model $f(x)$.



→ Learning the parameters.

$$\checkmark \text{ maximum likelihood} \Rightarrow K_{ij} = e^{-\frac{1}{2} \|x_i - x_j\|^2}$$

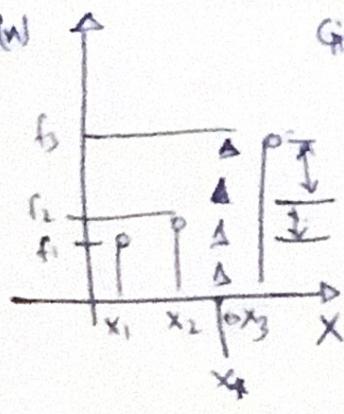
→ This we get kernel $K_{ij} = e^{-\frac{1}{2} \|x_i - x_j\|^2}$

→ If we have data $\{x_i\}$ then $K_{ij} = \begin{cases} 0 & \|x_i - x_j\| \rightarrow \infty \\ 1 & x_i = x_j \end{cases}$

$$\text{exponential } \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

Kernel. (similarity curve)

x_1 and x_2 are correlated (0.7)
 x_1 and x_3 are not (0.2)
 x_2 and x_3 are (0.6).

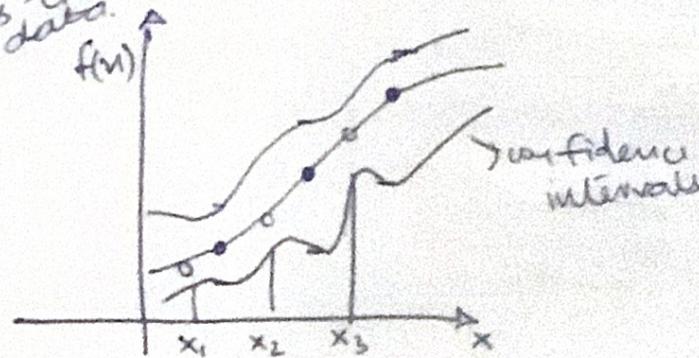


Given $\Rightarrow D = \{(x_1, f_1), (x_2, f_2), (x_3, f_3)\} \Rightarrow f_{\star} = ?$

$$\hat{f}^* = K_{\star}^T \tilde{K}_{\star}^{-1} f$$

$$H_K = E(f^*) = K_{\star}^T \tilde{K}_{\star}^{-1} f$$

$$C^* = K_{\star} K_{\star}^T K_{\star}^{-1} K_{\star \star}$$



GP

Gaussian distribution over functions

$$f(x) \sim GP(m(x), K(x, x'))$$

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

$$K(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^\top] \quad k(x, x') = c$$

GP Posterior

$$\text{GP Posterior} \rightarrow \text{training}$$

$$D = \{(x_i, f_i), i=1:N\}$$

$$p(f|D) = \frac{p(D|f)p(f)}{p(D)}$$

$D = \{(x_i, f_i), i=1:N\}$, where $f_i = f(x_i)$

$$X \rightarrow N \times D$$

$$\begin{pmatrix} f \\ f_r \end{pmatrix} \sim N \left(\begin{pmatrix} \mu \\ \mu_r \end{pmatrix}, \begin{pmatrix} K & K_r \\ K_r & K_{rr} \end{pmatrix} \right)$$

$$\Rightarrow K = K(x, x)$$

$$\Rightarrow K^* = K(x, x_*)$$

$$\Rightarrow K_{\text{ext}} = K(x_*, k_x)$$

$$K(x, x') = \frac{6^2}{4\pi} e^{-\frac{1}{4\pi}(x-x')^2}$$

$$p(f_* | x_*, x, f) = N(f_* | \mu_f, \Sigma_f)$$

$$\Rightarrow \mu_r = \mu(x_r) + K_r^T K^{-1}(e - \mu(x))$$

$$\Rightarrow Z_{\text{fit}} = K_{\text{fit}} - k_{\text{fit}}^T K^{-1} k_{\text{fit}}$$

Algorithm GP regression (non linear res.)

$$L = \text{Cholesky}(K + \sigma^2 I)$$

$$f_k = k_* K g^* g$$

$$d = Ky = LL^T Y$$

$\alpha = \hat{L}^{-1}/(\hat{L}y) \rightarrow$ linear operator y solve it

$$\mathbb{E}[G_t] = k_p \alpha$$

$$V = L/k_*$$

$$\text{var}[A] = K(x_0, x_0) - \hat{v}^T V$$

$$\log p(\hat{y}|\hat{x}) = -\frac{1}{2} y^T \alpha - \sum_i \log l_i + \frac{N}{2} \log(2\pi)$$

$$\Rightarrow L^T x = m$$

→ where there is data, believe on the data otherwise believe the prior knowledge.

Noiseless GPR

$$(f_{\star}) \sim N\left(\begin{pmatrix} N \\ \mu_{\star} \end{pmatrix}, \begin{pmatrix} K & K_{\star} \\ K_{\star}^T & K_{\star} K_{\star} \end{pmatrix}\right)$$

$$p(f_{\star} | X_{\star}, X, f) = N(f_{\star} | \mu_{\star}, \Sigma_{\star}) \quad \text{Gaussian dist.}$$

$$\mu_{\star} = \mu(X_{\star}) + K_{\star} K^{-1}(f - \mu(X))$$

$$\Rightarrow \Sigma_{\star} = K_{\star} K_{\star}^T - K_{\star} K^{-1} K_{\star}$$

$$K(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x-x')^2\right)$$

✓ If l very small, the RBF kernel, then the function would be more wiggly function.

✓ If l is increased, it becomes smooth.

✓ If it is still increased then it will become a straight line.

→ How to deal with noise?

To acknowledge uncertainty, assume that the noise is Gaussian then add a noise parameter (Gaussian variable) then get noisy f in i.e. y . → marginalize the f .

$$p(y|X) = \int p(y|f, X) p(f|X) df$$

→ To get rid of the variable, you need to integrate over it. → basic operation of probability

$$p(f|X) = N(f|0, K)$$

$$p(y|f) = \prod_i N(y_i|f_i, \sigma_y^2)$$

$$\text{cov}[y|X] = K + \sigma_y^2 I \stackrel{\text{noise}}{\triangleq} K_y$$

→ the kernel is now modified and has extra diagonal entry that was noise.

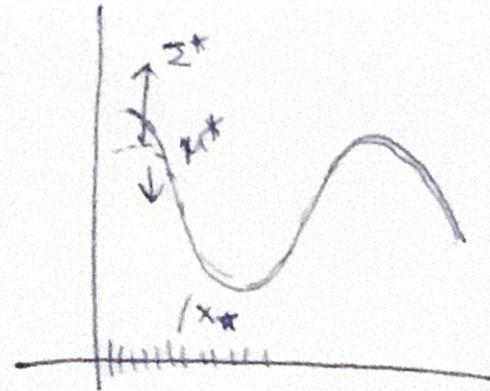
$$\left(\begin{array}{c} y \\ f_{\star} \end{array}\right) \sim N\left(0, \begin{pmatrix} K_y & K_{\star} \\ K_{\star}^T & K_{\star} K_{\star} \end{pmatrix}\right) \quad p(f_{\star}|X_{\star}, X, y) = N(f_{\star} | \mu_{\star}, \Sigma_{\star})$$

$$\mu_{\star} = K_{\star}^T K_y^{-1} y$$

$$\Sigma_{\star} = K_{\star} K_{\star}^T - K_{\star}^T K_y^{-1} K_{\star}$$

⇒ If you have to fit GP to data, you just construct a matrix K using the similarity kernel and if you have noise then add variance of noise to diagonals.

→ For the prior, then check the hypothesis. Verify that the prior is robust. The prior must be insensitive.



⇒ To learn the Kernel parameters ⇒ cross-validation, maximum likelihood, Bayesian learning. One more approach is to write the likelihood.

$$\text{marginal likelihood} \quad p(y|x) = \int p(y|f, x) p(f|x) df$$

$$\log p(y|x) = \text{WSN}(y|\mu, K_y) = -\frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} \log |K_y| - \frac{N}{2} \log(2\pi)$$

$$\frac{\partial \log p(y|x)}{\partial \theta_j} = \frac{1}{2} y^T K_y^{-1} \frac{\partial K_y}{\partial \theta_j} K_y^{-1} y - \frac{1}{2} \text{tr}(K_y^{-1} \frac{\partial K_y}{\partial \theta_j}) \leftarrow \begin{matrix} \text{optimal } \theta \\ \text{max. likelihood.} \end{matrix}$$

$$p(f_*|x_*, x, y) = N(f_* | k_*^T K_y^{-1} y, k_{**} - k_*^T K_y^{-1} k_*)$$

$$k_* = [k(x_*, x_1), \dots, k(x_*, x_N)]_{1 \times n} \quad k_{**} = K(x_*, x_*) \quad \text{kernel}$$

$$\begin{aligned} f_* &= \underbrace{k_*^T K_y^{-1} y}_{\text{mean}} = \underbrace{\alpha^T x}_{\text{RBF}} = \underbrace{\sum_i \alpha_i e^{-\gamma_i \|x_i - x_*\|^2}}_{\text{basis function.}} \quad \begin{matrix} \alpha = \underbrace{K_y^{-1} y}_{\text{answers}} \\ \text{training} \end{matrix} \\ &\Rightarrow \text{linear combination of basis functions.} \quad \begin{matrix} \text{subset of vector space } V \\ \text{linear combination of basis functions.} \end{matrix} \end{aligned}$$

Ridge Regression

$$\min_{\theta \in \mathbb{R}^d} \|y - X\theta\|_2^2 + \delta^2 \|\theta\|_2^2$$

$$\Rightarrow (X^T X + \delta^2 I_d) \theta = X^T y$$

$$\theta = X^T \alpha \text{ where } \alpha = \delta^{-2} (y - X\theta)$$

↳ solution can be written as:

$$\#1 \quad X^T X \theta + \delta^2 \theta = X^T y$$

$$\#2 \quad \delta^2 \theta = X^T (y - X\theta)$$

$$\theta = X \delta^{-2} (y - X\theta)$$

$$\theta = X^T \alpha$$

Solution #1 ↗

$$X \in \mathbb{R}^{n \times d}, \quad x_i \in \mathbb{R}^d, \quad y \in \mathbb{R}^n$$

$$\begin{aligned} \#1 \quad & \delta^2 \alpha = y - X\theta \quad \theta = X^T \alpha, \alpha = \delta^{-2} (y - X\theta) \\ \#2 \quad & \delta^2 \alpha = y - X X^T \alpha \\ & X X^T \alpha + \delta^2 I_d \alpha = y \quad \text{identity} \\ & \Rightarrow \alpha = (X^T X + \delta^2 I_n)^{-1} y \end{aligned}$$

Depends on d and n you can parameterize the one. If $n \gg d$

→ use solution #2 else solution #1.

→ e.g. 20 patients, 20000 features ($d \times d$). So, parameterize α

($n \times n$) rather than θ ($X^T X \alpha$) or ($X^T \theta$).

↳ α is parameterization.

$$\Rightarrow Y^* = X^T \theta = X^T X \alpha$$

If $y^* = X^T \theta$ (prediction for a new point)

$$y^* = X^T X^T (X X^T + \delta^2 I_n)^{-1} y$$

$$= K_*^T K_y^{-1} y$$

kernel matrix
measure of similarity

$$K = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}_{n \times d}$$

$$X X^T = \begin{bmatrix} x_1^T & \cdots & x_n^T \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$$

$$X^T X = \begin{bmatrix} x_1^T & \cdots & x_n^T \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

- ⇒ Dual form of Ridge regression is a GP.
- ⇒ If you let a single hidden layer neural network and let the number of neurons go to infinity, that leads to Gaussian process.
- $K_{\star}^T = [x^* x_1^T \ x_1^* x_2^T \ \dots \ x_1^* x_n^T]$
- ⇒ cross validation, max-likelihood \Rightarrow works only when we have lots of data.
- ⇒ Ridge regression \rightarrow Tikhonov regularization \rightarrow adding noise to our data.

Murphy's Book (CH# 04)