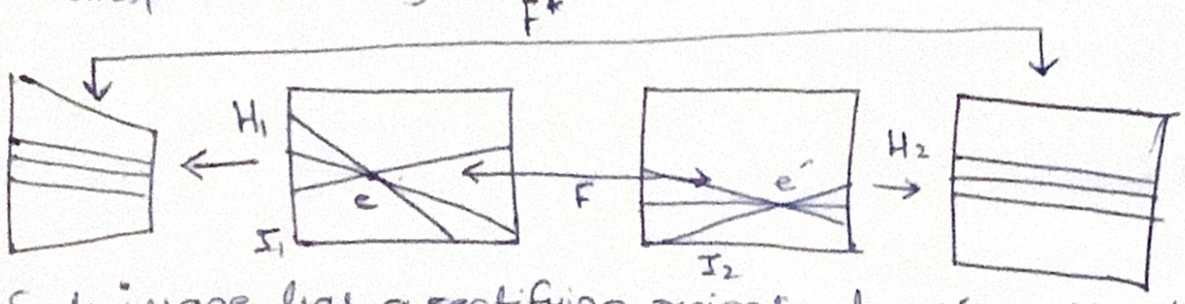
5) Let F be last column of V (9×1). Reshape into 3×3 matrix $\hat{F}$.

6) Compute SVD of $\hat{F} = \hat{U}\hat{D}\hat{V}^T$. Zero out lowest singular value of D. Then recompute $F = \hat{U}\hat{D}\hat{V}^T$. ↳ 3×3 diagonal matrix.

7) Renormalization F.

8-point algorithm

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*?

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

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

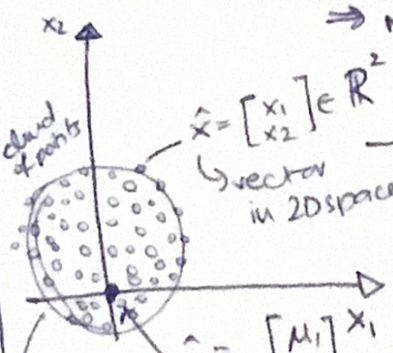
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ -y' \end{bmatrix} = 0$$

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

Gaussian Processes

⇒ Multivariate gaussian use → fit gaussians to the data.

→ process of fitting a Gaussian to the points and then the Gaussian will describe those points.



⇒ Mean and (correlation) covariance should describe the data.

At any point, $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \mathbb{E}(x_1 x_2) \\ \mathbb{E}(x_1 x_2) & 1 \end{bmatrix}\right)$ (expectation)

→ information about x_1 doesn't help get more information about x_2. Not correlated.

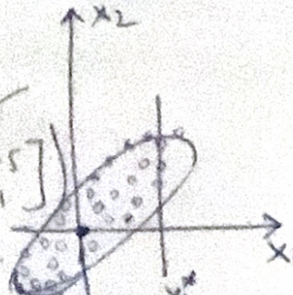
$$\rho_{x_1 x_2} = \frac{\text{covariance}(x_1, x_2)}{\sigma_1(x_1)\,\sigma_2(x_2)}$$

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

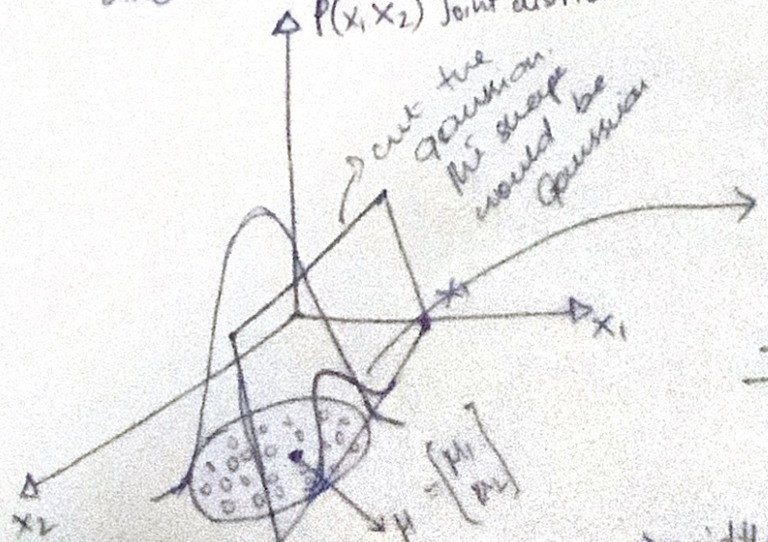
⇒ dot product measures similarity.
↳ if two points are similar, the dot product will be similar.

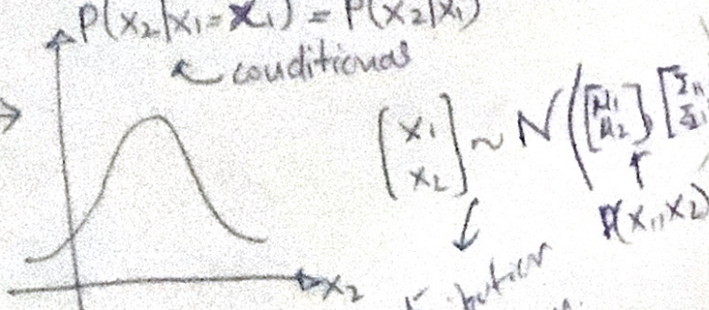
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right)$$

some +ve number.



→ If non-zero, then it means dependence is there.





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

↑ $P(x_1, x_2)$

↓ assume joint distribution is gaussian.

⇒ width of Gaussian should be +ve.

↳ the covariance matrix should be symmetric and +ve.

⇒ $x^T \Sigma x$ → has to be +ve.

↳ symmetric - positive definite.

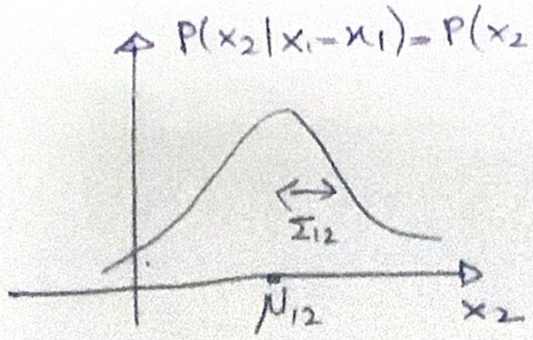
CH#4 → Kevin Murphy ML book

→ Schur complement or matrix inversion lemma.

↳ given a Gaussian, get the conditional.

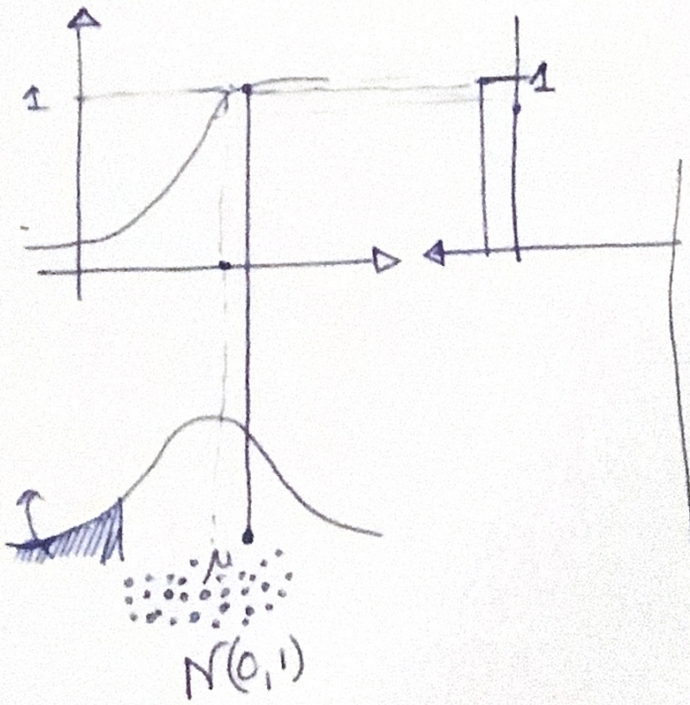
$\mu_{12} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$ → Multivariate Gaussian theorem.

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



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.



$x_i \sim N(0,1)$

$x_i \sim N(\mu, \sigma^2)$ → univariate.

$\sim \mu + \sigma N(0,1)$

↳ change the height

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

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

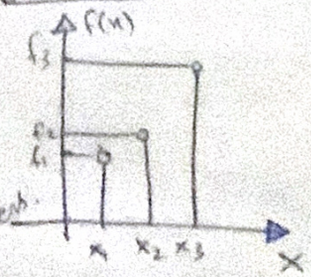
$x \quad \mu \quad \Sigma$

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

→ Cholesky decomposition

⇒ $\Sigma = LL^T$ → two matrices

⇒ x's given, want to model fc.



$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}\right)$

→ Learning the parameters.

↳ maximum likelihood ⇒ we get kernel ⇒ which fits the data.

$K_{ij} = e^{-\|x_i - x_j\|^2}$ → squared exponential kernel (similarity curve)

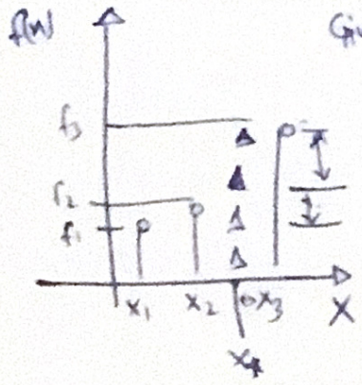
$= \begin{cases} 0 & \|x_i - x_j\| \to \infty \\ 1 & x_i = x_j \end{cases}$

$\sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.7 & 0.2 \\ 0.7 & 1 & 0.6 \\ 0.2 & 0.6 & 1 \end{bmatrix}\right)$

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

$f \sim N(0, K)$ ← self covariance

$f_* \sim N(0, K(x_*, x_*))$ ← assume the test data to come from same distr. as train data.

K_{**}

$K(x_*, x_*) = c^{-\|x_* - x_*\|^2}$ $\quad K_{**} = 1$

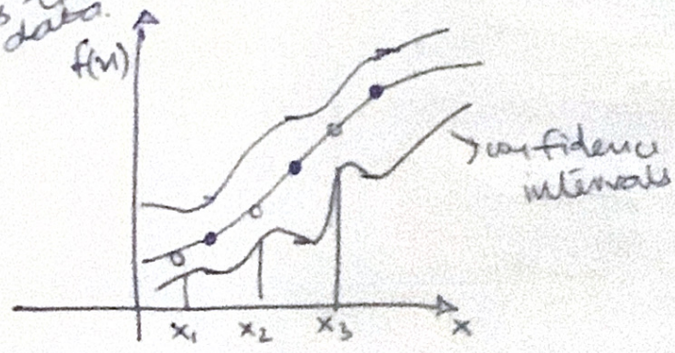
$\begin{bmatrix} f \\ f_* \end{bmatrix}_{3D} \sim N\left(0, \begin{bmatrix} K & K_{*} \\ K_{*} & K_{**} \end{bmatrix}_{3D}\right)$ $\quad K_* = K(x_1, x_*)$

$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$ $\quad K_* = \begin{bmatrix} k_{1*} \\ k_{2*} \\ k_{3*} \end{bmatrix}$ $\quad K_{**} = [k_{**}]$

$\Rightarrow f^* = K_*^T K^{-1} f$

$\mu_* = \mathbb{E}(f^*) = K_*^T K^{-1} f$

$\sigma^* = K_* K^{-1} K_*^T + K_{**}$



GP

↳ Gaussian distribution over functions

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

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

$K(x,x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T]$ $\quad k(x,x') = e^{-\frac{1}{2}(x-x')^2}$

GP Posterior ← 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 $f_i = f(x_i)$

$x_* \rightarrow N_* \times D$

$\begin{pmatrix} f \\ f_* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu \\ \mu_* \end{pmatrix}, \begin{pmatrix} K & K_* \\ K_* & K_{**} \end{pmatrix}\right)$

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

$\Rightarrow K_* = K(X, X_*)$

$\Rightarrow K_{**} = K(X_*, X_*)$

$K(x,x') = \sigma_f^2 e^{-\frac{1}{2\ell^2}(x-x')^2}$

↳ kernel

$p(f_* | X_*, X, f) = N(f_* | \mu_*, \Sigma_*)$

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

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

Algorithm GP regression (non linear reg.)

$\bar{f}_* = k_*^T K_y^{-1} y$ $\quad K_y = LL^T$

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

$\alpha = K_y^{-1} y = L^{-T} L^{-1} y$ (m)

$\alpha = \hat{L}^T \backslash (\hat{L} \backslash y)$ → Linear operator (\) ↳ solve it

$\Rightarrow m = L^{-1} y$

$\Rightarrow Lm = y$ $\quad L$, m

$\mathbb{E}[f_*] = k_*^T \alpha$ $\quad L^T \alpha = m$

$v = L \backslash k_*$

$\text{var}[f_*] = K(\hat{x}_*, \hat{x}_*) - \hat{v}^T \hat{v}$

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

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

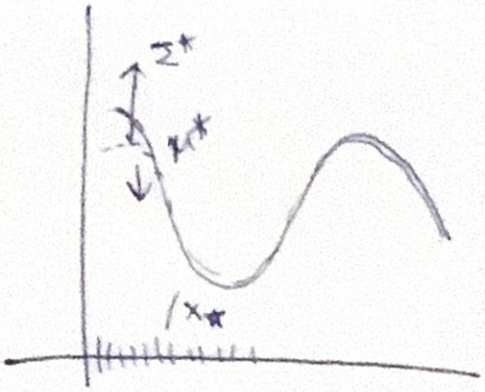
Noiseless GP Regression

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

$$p(f_* | X_*, X, f) = \mathcal{N}(f_* | \mu_*, \Sigma_*) \quad \leftarrow \text{gaussian dist.}$$

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

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



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

→ width parameter

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

✓ wiggly function.

✓ If ℓ is increased, it becomes smooth.

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

⇒ $y = f(x) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_y^2)$ → how to deal with noise?

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

→ marginalize the f.

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

→ basic operation of probability

To get rid of the variable, you need to integrate over it.

$$p(f|X) = \mathcal{N}(f|0, K)$$

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

→ convolution theorem. Same mean but covariance is increased.

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

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

$$\begin{pmatrix} y \\ f_* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K_y & k_* \\ k_* & k_{**} \end{pmatrix}\right)$$

$$p(f_* | X_*, X, y) = \mathcal{N}(f_* | \mu_*, \Sigma_*)$$

$$\mu_* = k_*^T K_y^{-1} y$$

$$\Sigma_* = k_{**} - k_*^T K_y^{-1} k_*$$

⇒ 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.

→ To 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.

$p(y|x,\theta^2) = \int p(y|f,x)\,p(f|x)\,df$ ← parameters.

marginal likelihood

$\log p(y|x) = \log \mathcal{N}(y|0,K_y) = -\frac{1}{2} y K_y^{-1} y - \frac{1}{2}\log|K_y| - \frac{N}{2}\log(2\pi)$

$\frac{\partial}{\partial \theta_j}\log p(y|x) = \frac{1}{2} y^T K_y^{-1} \frac{\partial K_y}{\partial \theta_j} K_y^{-1} y - \frac{1}{2} tr\left(K_y^{-1}\frac{\partial K_y}{\partial \theta_j}\right)$ ← optimal θ max. likelihood.

$p(f_*|X_*, X, y) = \mathcal{N}(f_*|k_*^T K_y^{-1} y,\ k_{**} - k_*^T K_y^{-1} k_*)$

↳ mean ↳ covariance → similarity kernel

$k_* = [k(x_*, x_1), \ldots, k(x_*, x_N)]_{1\times N}$ & $k_{**} = K(x_*, x_*)$

$\bar{f}_* = \underbrace{k_*^T}_{1\times N}\underbrace{K_y^{-1}}_{N\times N}\underbrace{y}_{N\times 1} = \sum_{i=1}^{N}\alpha_i K(x_i, x_*)$ $\quad \alpha = \underbrace{K_y^{-1} y}_{\text{training}}$ → answers

⇒ mean $= k_*^T \alpha$ → RBF

$= \sum_i \alpha_i \underbrace{e^{-\frac{1}{\lambda}\|x_i - x_*\|^2}}_{\text{basis function.}}$ → linear combination of basis function.

⇒ subset of vector space V

Ridge Regression

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

$\Rightarrow (\underbrace{X^TX}_{d\times d} + \delta^2 I_d)\theta = X^T y$ $\quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$ $\quad x_i \in \mathbb{R}^{1\times d}$

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

↳ solution can be written as.

#1 $X^TX\theta + \delta^2\theta = X^Ty$

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

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

$\theta = X^T\alpha$

Solution #1 ⇒

#2 $\delta^2\alpha = y - X\theta$ $\quad \theta = X^T\alpha$, $\alpha = \delta^{-2}(y - X\theta)$ ($n\times 1$)

$\delta^2\alpha = y - XX^T\alpha$

$XX^T\alpha + \delta^2 I_n\alpha = y$ (I_n: identity)

$\Rightarrow \alpha = \underbrace{(XX^T}_{n\times n} + \delta^2 I_n)^{-1} y$ ⇒ Solution #2

⇒ Depends on d and n you can penalize the one. If $n >>> d$ use solution #2 else solution #1.

→ e.g. 20 patients, 20000 features ($d\times d$). So, penalize α ($n\times n$) rather than θ ($X^*X^T\alpha$) or ($X^*\theta$).

↳ for α parametrization ↳ for θ parametrization

$\Rightarrow \boxed{Y^* = X^*\theta = X^*X^T\alpha}$

If $Y^*_{x1} = X^*_{xd}\theta_{dx1}$ (prediction for a new point)

$= X^*X^T\alpha$

$Y^* = X^*X^T\underbrace{(XX^T + \delta^2 I_n)^{-1} y}_{\alpha}$ ← by

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

→ kernel matrix ← measure of similarity

$K = (\text{dot product})$ $\quad XX^T = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\begin{bmatrix} x_1^T & \cdots & x_n^T \end{bmatrix}$ ($n\times d$ $d\times n$ → $n\times n$)

$XX^T = \begin{bmatrix} x_1x_1^T & - \\ x_nx_1^T & - \end{bmatrix}$

$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n\times d}$ $\quad x_i \in \mathbb{R}^{1\times d}$

⇒ 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.

$k_*^T = [x^* x_1^T \;\; x^* x_2^T \;\; \dots \;\; x^* x_n^T]$

⇒ cross validation, max-likelihood ⇒ works only when we have alots of data.

→ Ridge regression ⇒ Tikhonov regularization
↳ adding noise to our data.

Murphy's Book (CH# 04)