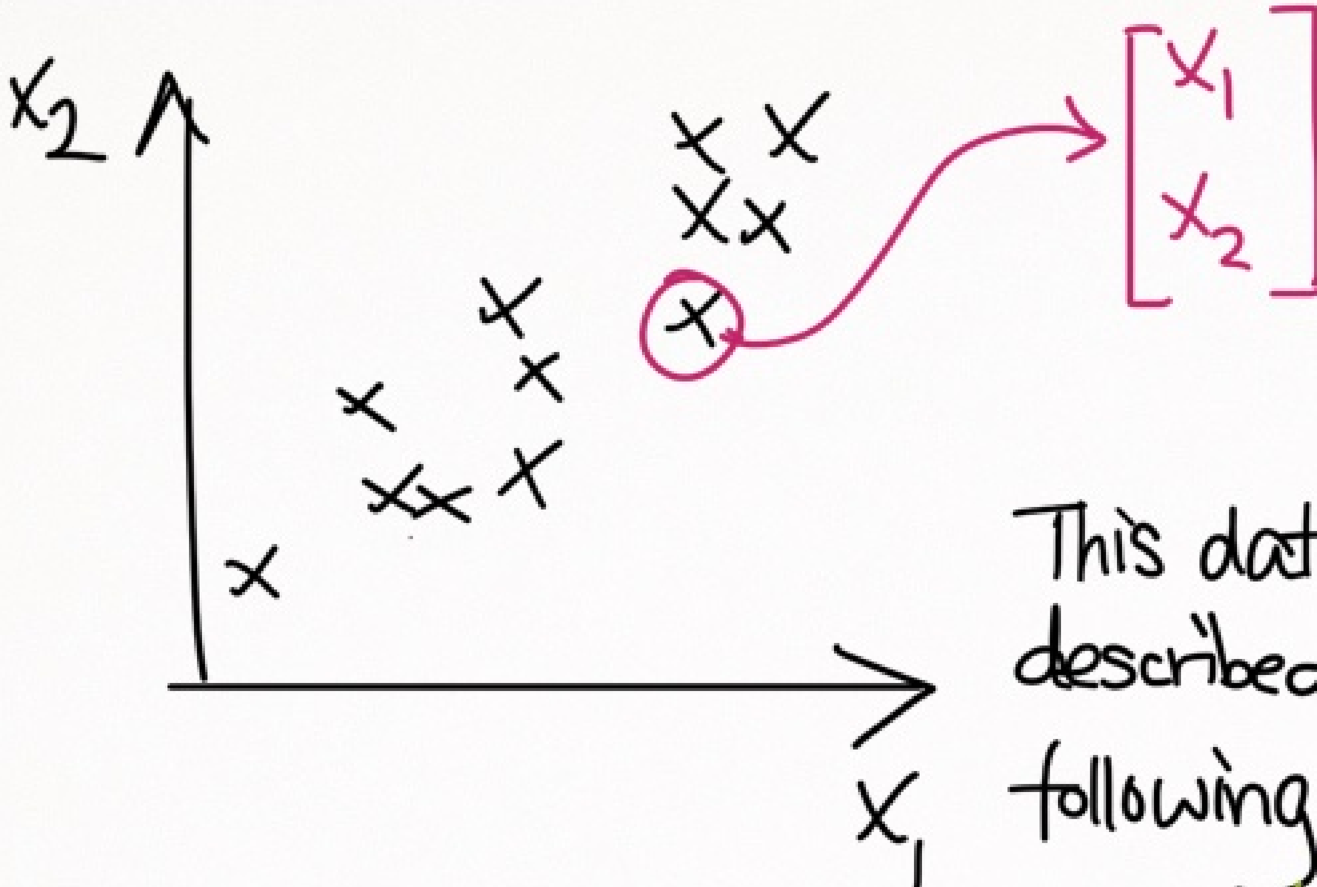
Gaussian Processes: The Math

Gaussian Basics



This data can be described using the following notation:

variance of x_1

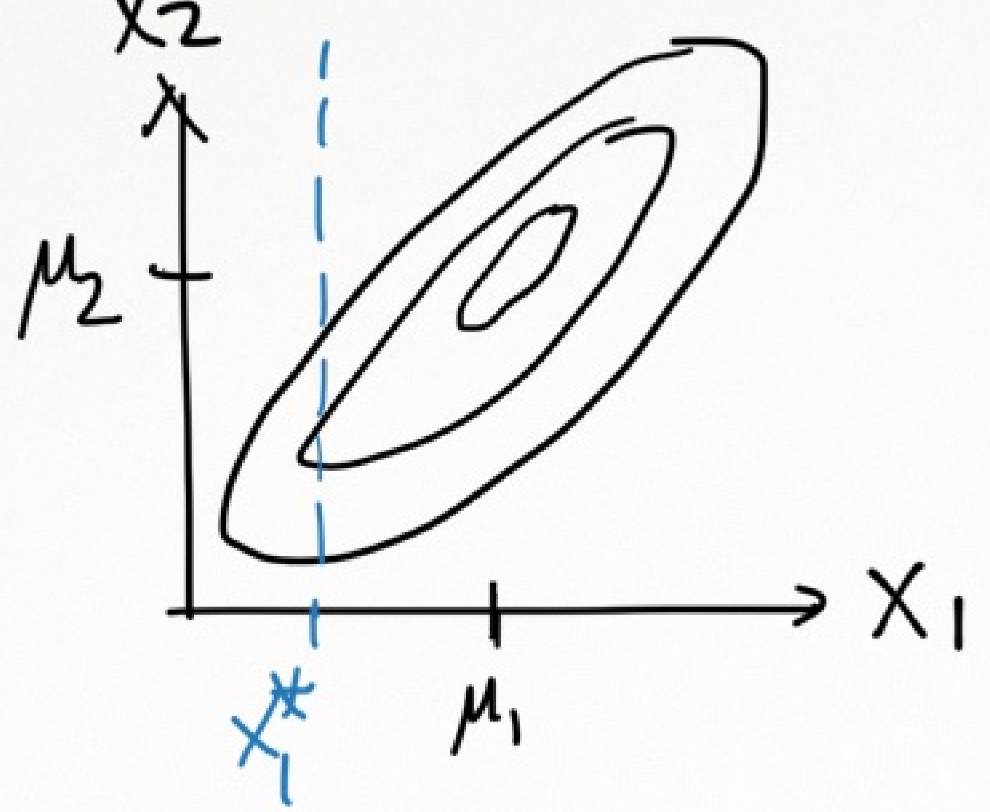
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) \quad \boxed{\mathbb{E}(x_1 x_2)}$$

$\boxed{\mathbb{E}(x_2 x_1)}$

variance of x_2

Measures whether knowing x_1 tells us something about x_2, and vice versa.

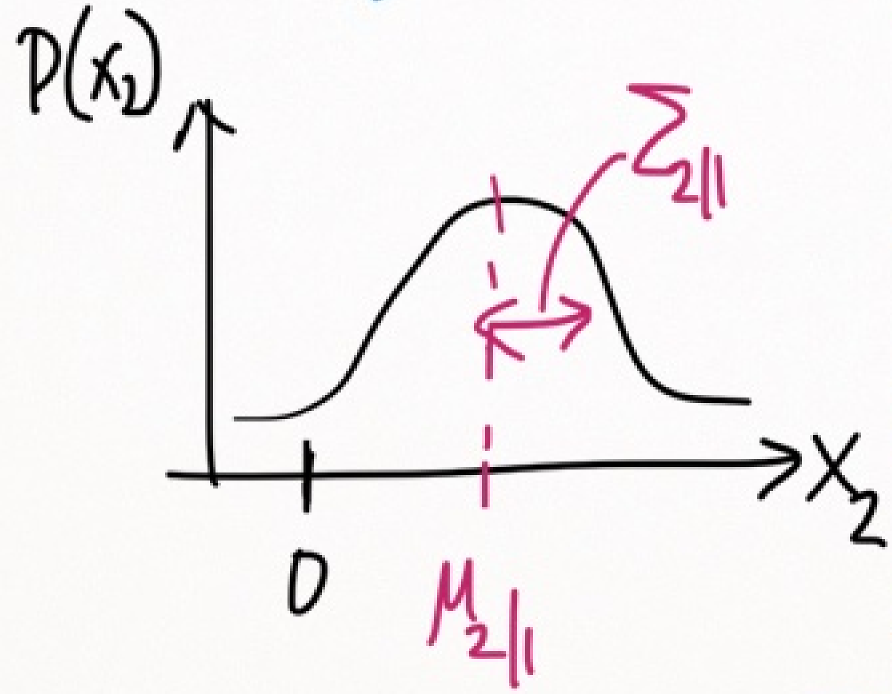
If we have data that are distributed jointly:



Joint distribution

$$N\left(\begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22}\end{bmatrix}\right)$$

$\downarrow$ conditional



$$\mu_{2|1} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1^* - \mu_1)$$

$$\Sigma_{2|1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$$

These equations are very important!!!

We also need to know the <u>reparam trick</u>

A univariate distribution

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

Can be rewritten as: no squares!

$$X \sim \mu + \sigma \cdot \mathcal{N}(0,1)$$

For multivariate distributions:

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{X} \sim \mathcal{N}\left(\underbrace{\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}}_{\mu} \cdot \underbrace{\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{bmatrix}}_{\Sigma} \right)$$

$\Sigma = L \cdot L^T$

find L by Cholesky decomp.

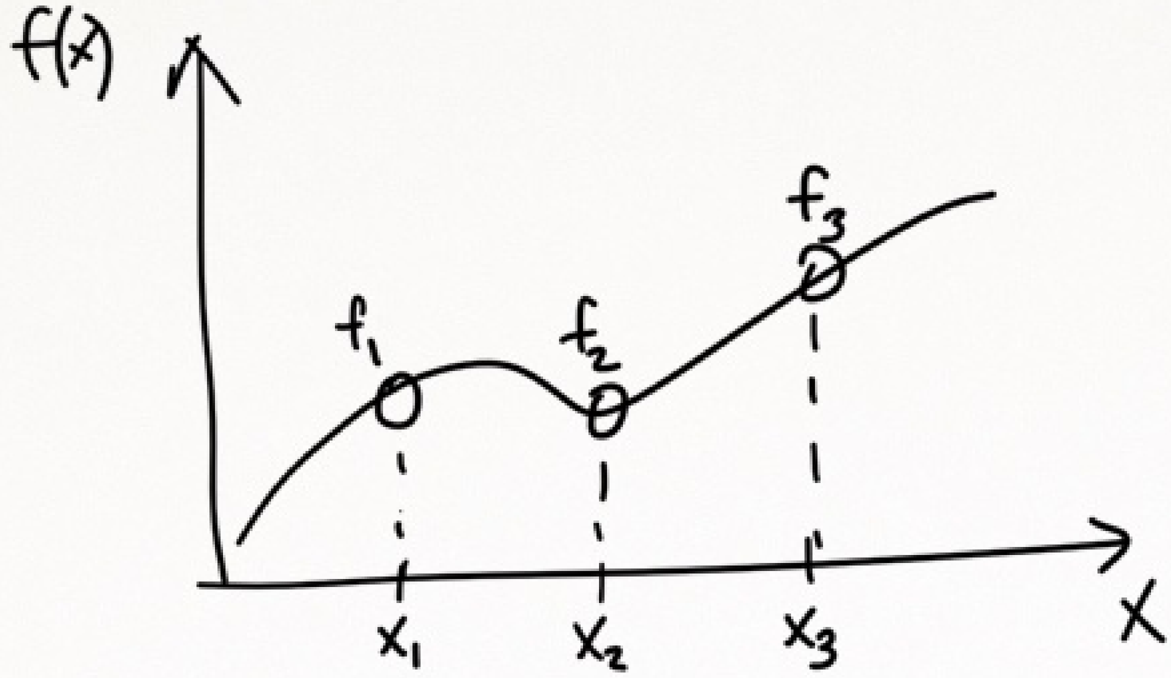
$\Updownarrow$

sq.rt. of matrix.

Rewritten:

$$X \sim \mu + L(\mathcal{N}(0,1))$$

Assume we have a function f.



We can model f(x) using multivariate Gaussians.

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

These Ks are very special! We can use them to express a prior belief that x_1 and x_2 that are close to one another should have similar f_1 and f_2

These Ks are our fabled covariance functions.

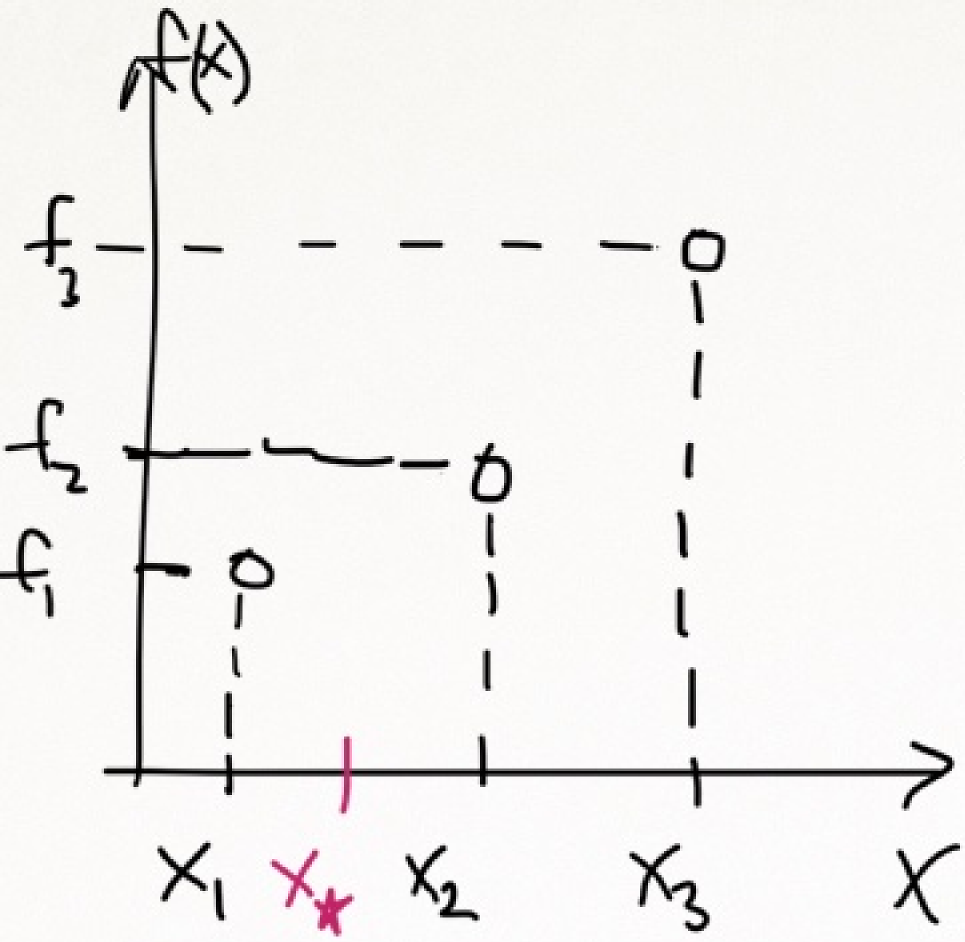
Here is one example:

$$K_{ij} = e^{-\lambda \| x_i - x_j \|^2} \begin{cases} 0 & \text{as } x_i - x_j \to \infty \\ 1 & \text{when } x_i = x_j \end{cases}$$

(exponentiated square kernel)

With this kernel, we can fill in the $\underline{\underline{K}}$ matrix.

$$K = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}$$



Q: Given $x_\star$ what is $f_\star$?

Let's model the covariances of $f_\star$ and the fs.

Assume $f_\star \sim N(0, \underbrace{K(x_\star, x_\star)}_{\substack{\text{self covariance} \\ = \text{variance of self}}}) = 1$

Then:

$$
\begin{bmatrix} f \\ f_\star \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_\star \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{array}{c} \begin{array}{cccc} & K & & K_\star \end{array} \\ \left[\begin{array}{ccc|c} K_{11} & K_{12} & K_{13} & K_{1\star} \\ K_{21} & K_{22} & K_{23} & K_{2\star} \\ K_{31} & K_{32} & K_{33} & K_{3\star} \\ \hline K_{\star 1} & K_{\star 2} & K_{\star 3} & K_{\star\star} \end{array}\right] \\ \begin{array}{cccc} & K_\star^T & & K_{\star\star} \end{array} \end{array} \right)
$$

We can write the previous covariance matrix more succinctly:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}\right)$$

Now, we can ask: under this particular modelling assumption (Mutivariate Gaussians), what is $P(f^*|f)$? To answer this question, we just have to follow the formula in Page 2!

$$\mu_{f_*|f} = \overset{0}{\overset{\shortparallel}{\cancel{\mu_{f_*}}}} + \Sigma_{f_*f} \cdot \Sigma_{ff}^{-1}(f - \overset{0}{\overset{\shortparallel}{\cancel{\mu_f}}})$$

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

$$\Sigma_{f_*|f} = \Sigma_{f_*f_*} - \Sigma_{f_*f} \Sigma_{ff}^{-1} \Sigma_{ff_*}$$

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

With this, we can write a numpy implementation!

How do we generalize this beyond 1 dimensional inputs?

The key lies in the covariance kernel function!

$$K_{ij} = e^{-\lambda \|x_i - x_j\|^2}$$

The $\|v\|$ notation refers to the norm of a vector/matrix. The norm is defined as the

$$\|v\|^p = \sum_i v_i^p$$

In this case, $p=1$, and $v = x_i - x_j$, hence x_i and x_j can be arbitrary-sized vectors/matrices!

1-D example:

i	X
1	1
2	2
3	3

$$\xrightarrow{x_i - x_j} \quad \begin{array}{c|ccc} i \backslash j & 1 & 2 & 3 \\ \hline 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 \\ 3 & 2 & 1 & 0 \end{array} \quad \xrightarrow{\|v\|^2} \quad \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 4 \\ 2 & 1 & 0 & 1 \\ 3 & 4 & 1 & 0 \end{array}$$

2-D example

i	x_1	x_2
1	1	1
2	1	0
3	0	0

$$\xrightarrow{x_i - x_j} \quad \begin{array}{c|ccc} i \backslash j & 1 & 2 & 3 \\ \hline 1 & (0,0) & (0,1) & (1,1) \\ 2 & (0,-1) & (0,0) & (1,0) \\ 3 & (-1,-1) & (-1,0) & (0,0) \end{array} \quad \xrightarrow{\|v\|^2} \quad \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{array}$$

As you can see, the covariance function, when defined properly, gives us a way to map high(-er) dimensional distance to a covariance scalar between our output values!