Gaussian Processes (GP) & Locally Weighted Project Regression(LWPR)

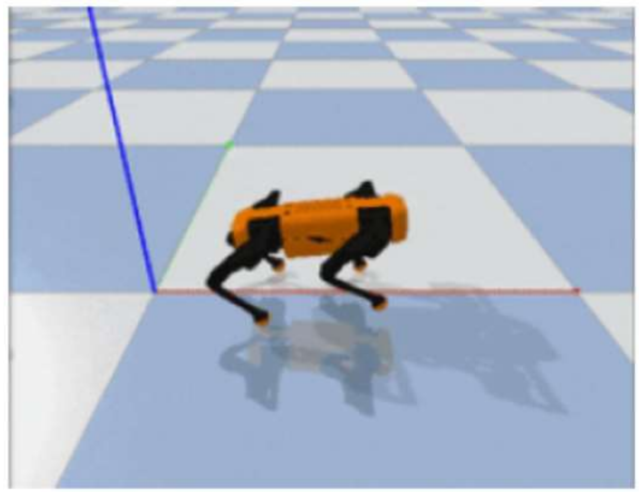
KUO-SHIH TSENG (曾國師)
DEPARTMENT OF MATHEMATICS
NATIONAL CENTRAL UNIVERSITY, TAIWAN

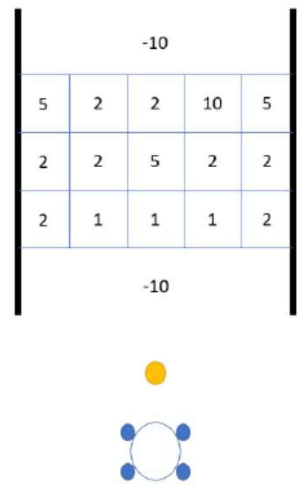
2021/05/26

Course Announcement

Course Announcement

- Final project proposal:
  - Check your proposal is
    - feasible?
    - Clear?





Course Announcement

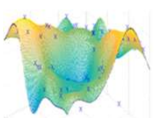
M e s

to me

Sun, May 2, 5:33 PM

Chinese (Traditional) > English Translate message Turn off for: Chinese (Traditional)

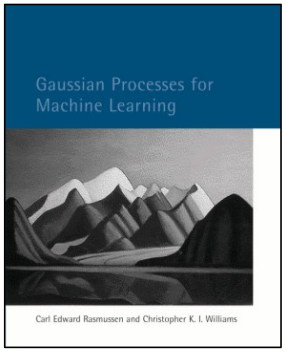
老師我聽朋友說做機器學習會一直用到高微和泛函，這是真的嗎XD 感覺上到現在都沒有看過。



Gaussian Processes: 我這不就來了嗎~

Outline

- Need
- Gaussian Distribution
- MAP of Bayesian Linear Regression
- Kernel Functions
- Gaussian Processes (GP)
  - EX: Coverage approximation
- Locally Weighted Project Regression(LWPR)
  - EX: Coverage approximation



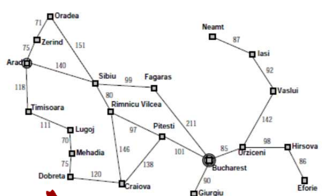
[6] Carl Edward Rasmussen and Christopher K. I. Williams, " Gaussian Processes for Machine Learning," MIT Press, 2006.

Outline

[Problem solving]

Search problems

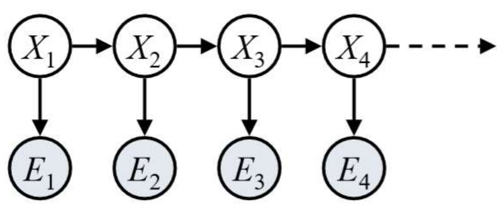
Adversarial Search



[Perception and Uncertainty]

Bayes Theorem

Bayes Filter and Smoothing

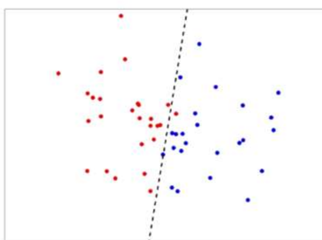


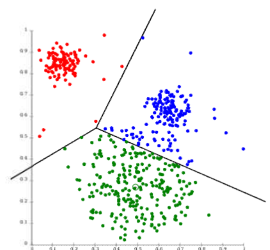
[Learning and Decision-making]

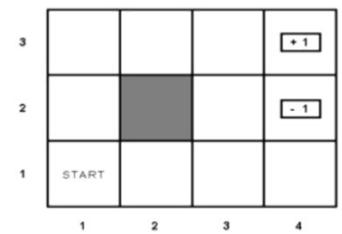
Supervised learning

Unsupervised learning

Reinforcement learning







GP

Need

- Parametric approximation needs good features. However, theses features are based on domain know-how. If we can learn the model <u>without features</u>, it could be applied to different domains.

$\{(\mathbf{x}_i, y_i)\}, i = 1 \sim N$

$y : measuremnt$

$\mathbf{x} : training\ sample$

$\boxed{\begin{array}{l} f_i, i = 1 \sim m \\ f : features \end{array}}$

Algorithm

$Q(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})$

Parametric approximation

$\boxed{\begin{array}{l} \{(\mathbf{x}_i, y_i)\}, i = 1 \sim N \\ y : measuremnt \\ \mathbf{x} : training\ sample \end{array}}$

Algorithm

$Q(\mathbf{x}) = ?$

Nonparametric approximation

Need

- Function approximation for these cases:

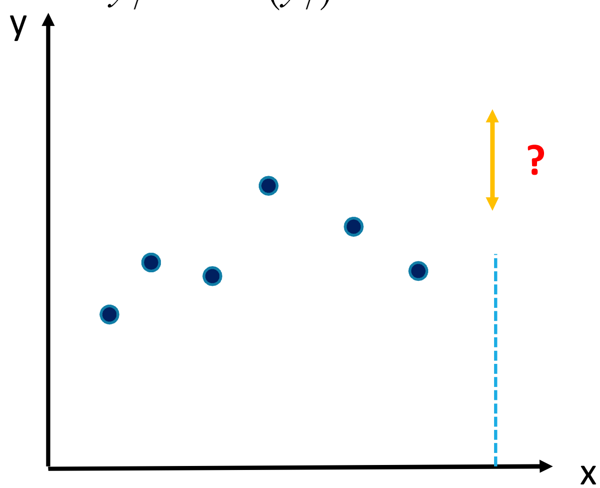
$Given:$

$\{(\mathbf{x}_i, y_i)\}, i = 1 \sim 6$

$\mathbf{x}_7$

$Find:$

$y_7 = ?\ \mathrm{cov}(y_7) = ?$



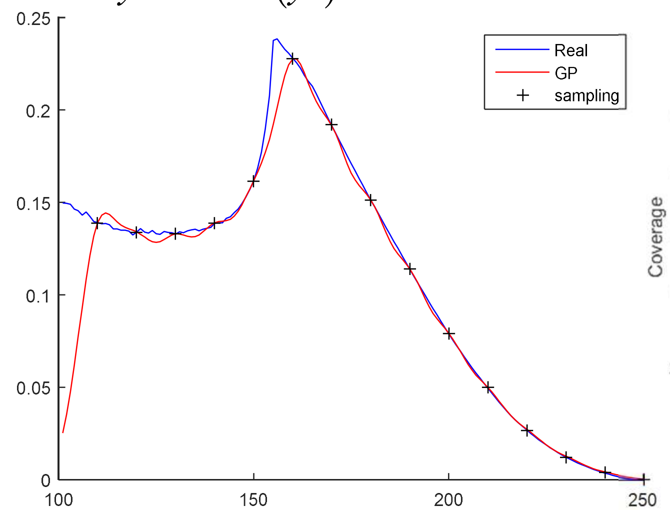
$Given:$

$\{(\mathbf{x}_i, y_i)\}, i = 1 \sim N$

$\mathbf{x}_*$

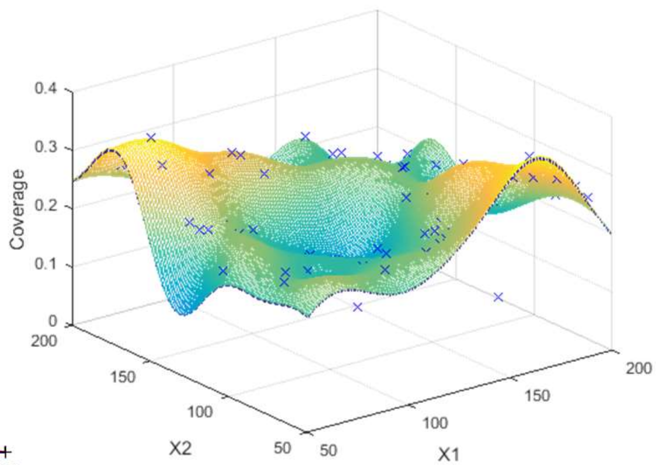
$Find:$

$y_* = ?\ \mathrm{cov}(y_*) = ?$



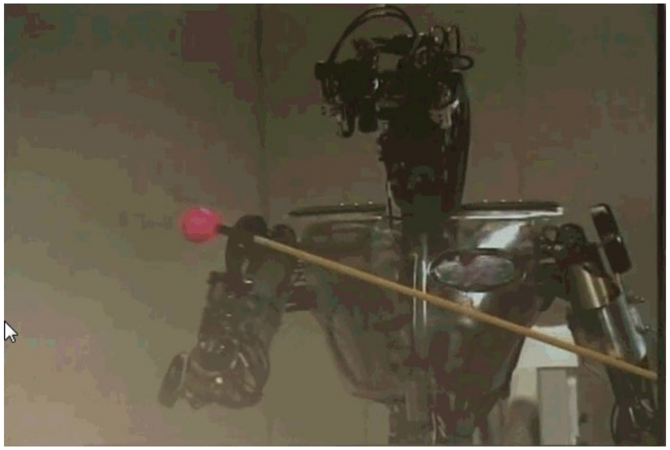
No more feature information!

$P(\mathbf{y}_* \mid \mathbf{x}, \mathbf{y}, \mathbf{x}_*) = ?$



Need

- Nonparametric approximation approaches include Gaussian processes (GP)[1], Locally Weighted Project Regression (LWPR) and neural network (NN).
- These approaches enable robots to learn some complex tasks[2].

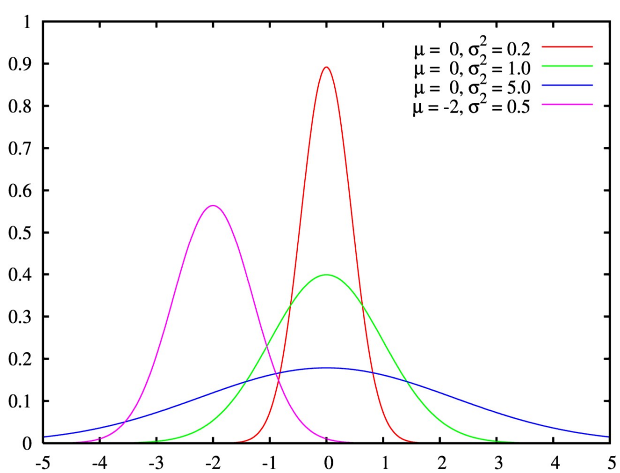


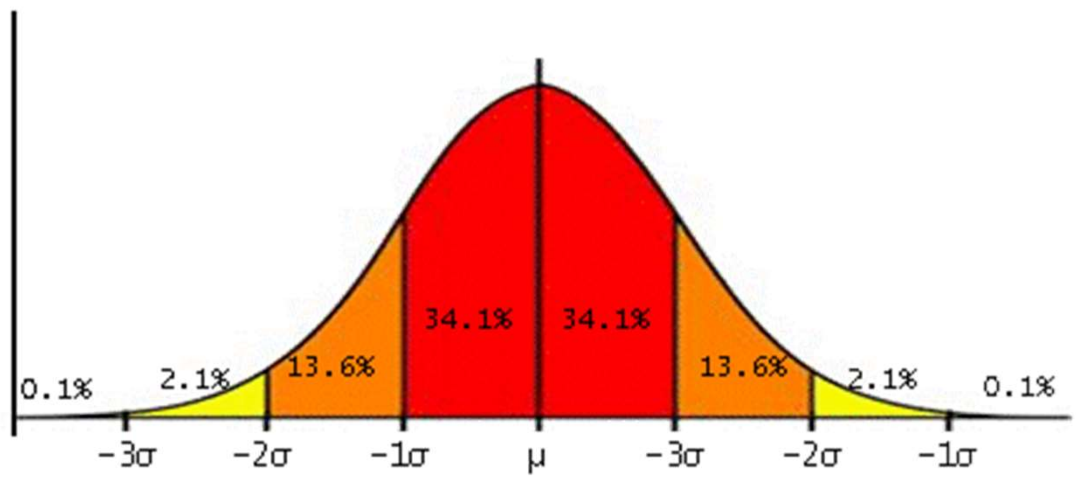
[1] http://www.gaussianprocess.org/gpml/
[2] http://www-clmc.usc.edu/~sschaal/

Gaussian Distribution

- Mean & variance → p(x)
- Probability density function (PDF)

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right)$$





https://en.wikipedia.org/wiki/Normal_distribution

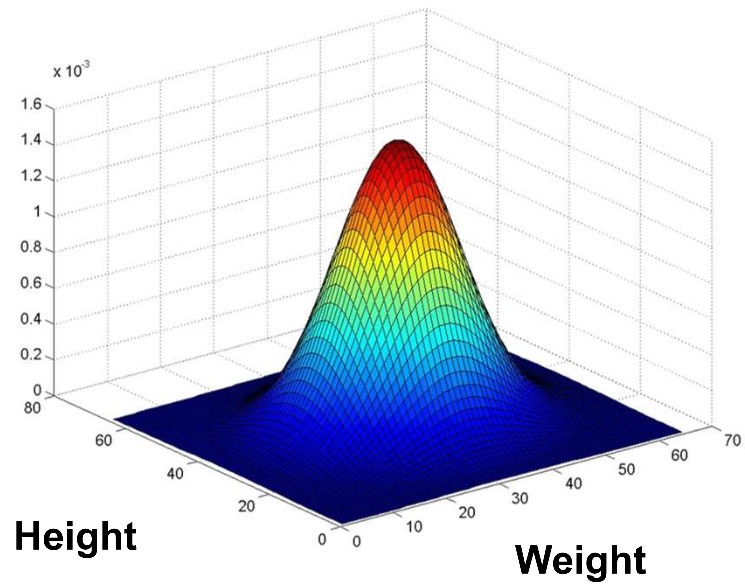
Gaussian Distribution

- Bivariate

$$\mathbf{x} = \begin{pmatrix} X \\ Y \end{pmatrix} \qquad X \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

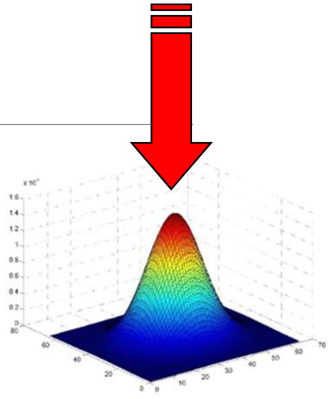
$$p(\mathbf{x}) = \frac{1}{2\pi \,||\, \boldsymbol{\Sigma} \,||^{1/2}} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

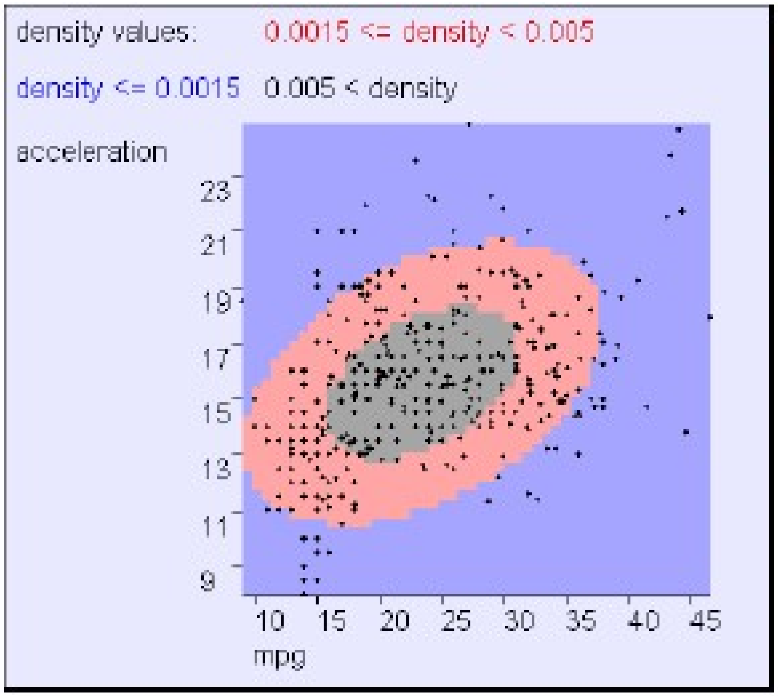
$$\boldsymbol{\mu} = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma^2{}_x & \sigma_{xy} \\ \sigma_{xy} & \sigma^2{}_y \end{pmatrix}$$

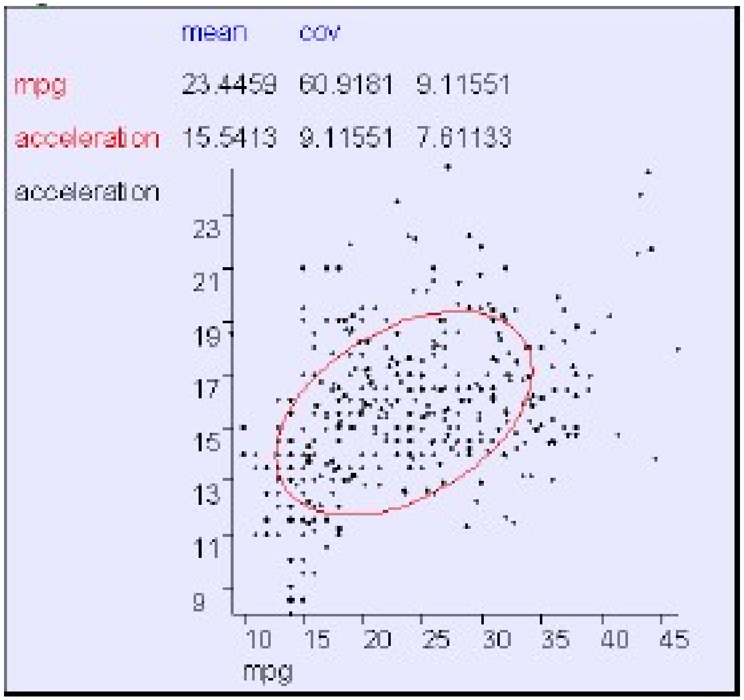


Gaussian Distribution

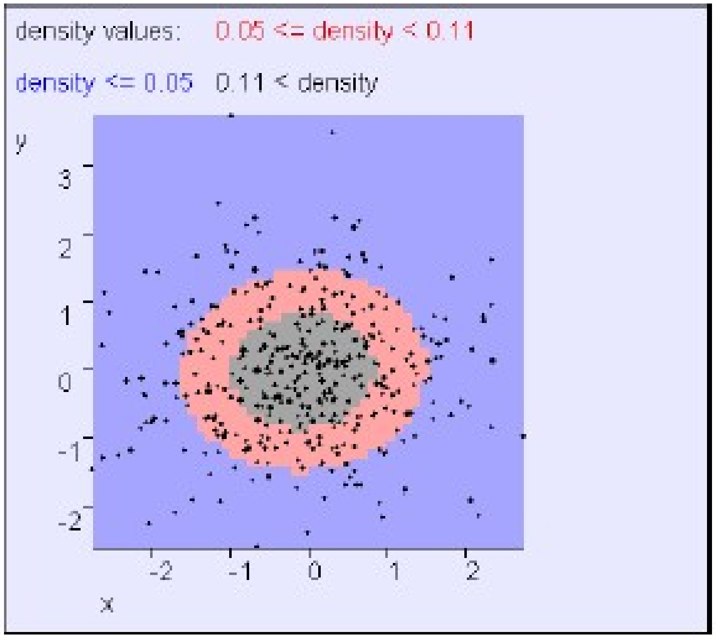
Statistical Data Mining Tutorials by Andrew Moore

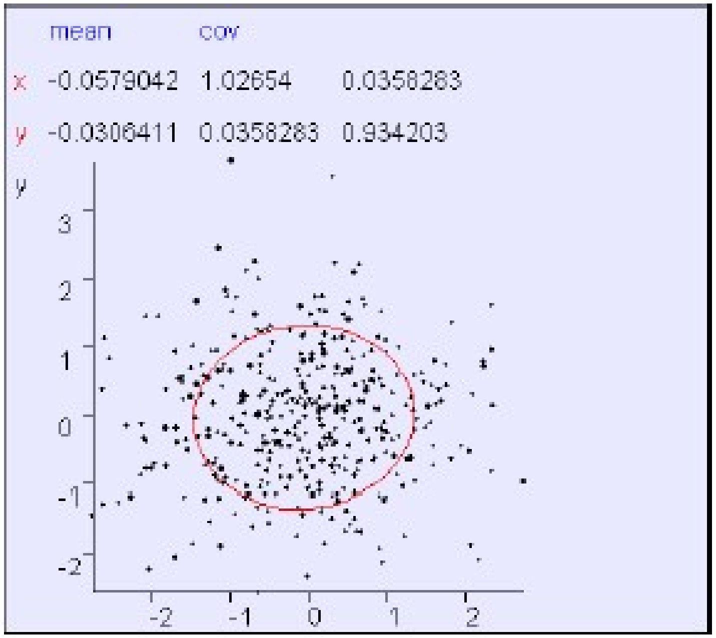






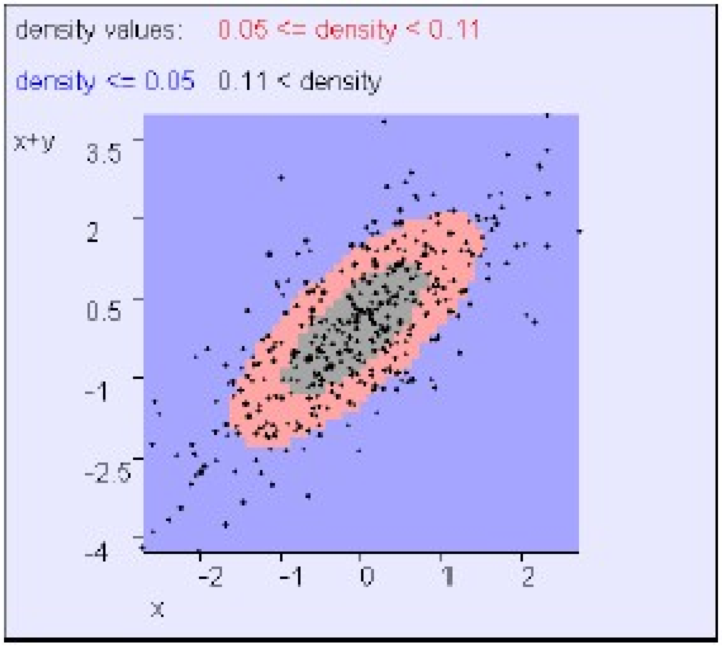
Gaussian Distribution

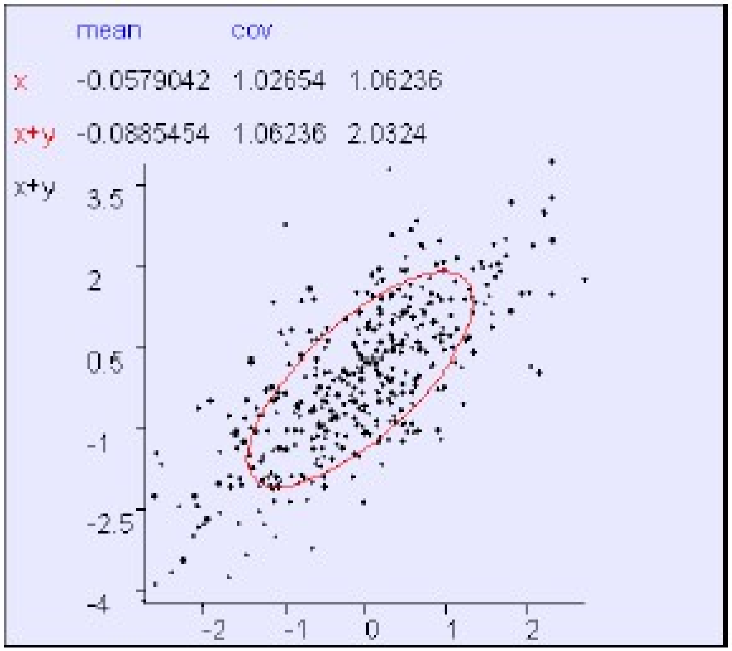




In this example, x and y are almost independent

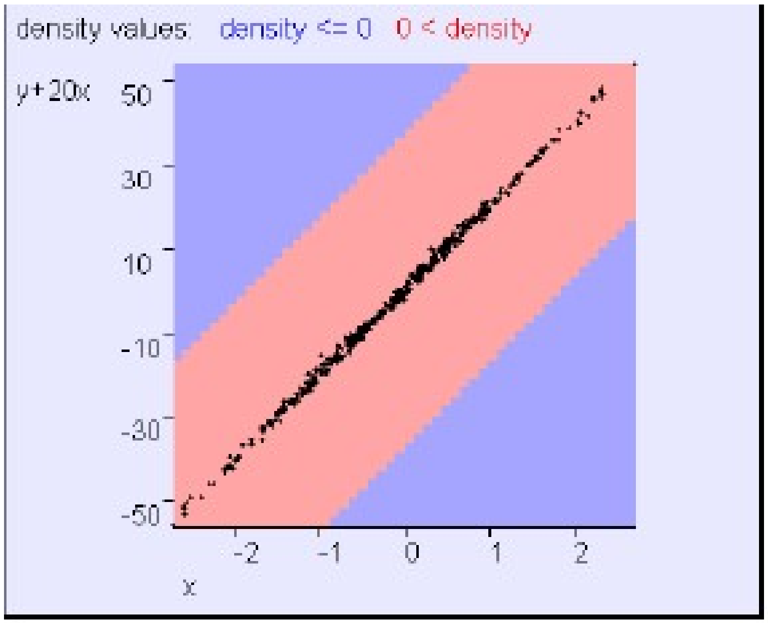
Gaussian Distribution

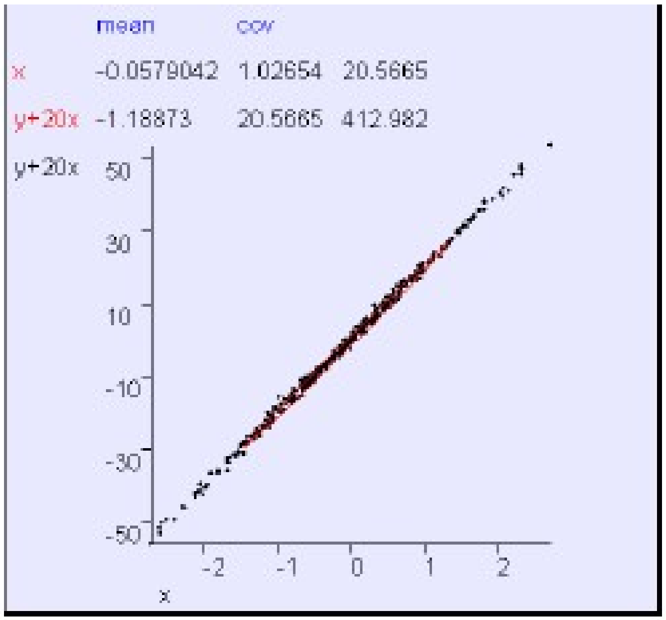




In this example, x and "x+y" are clearly not independent

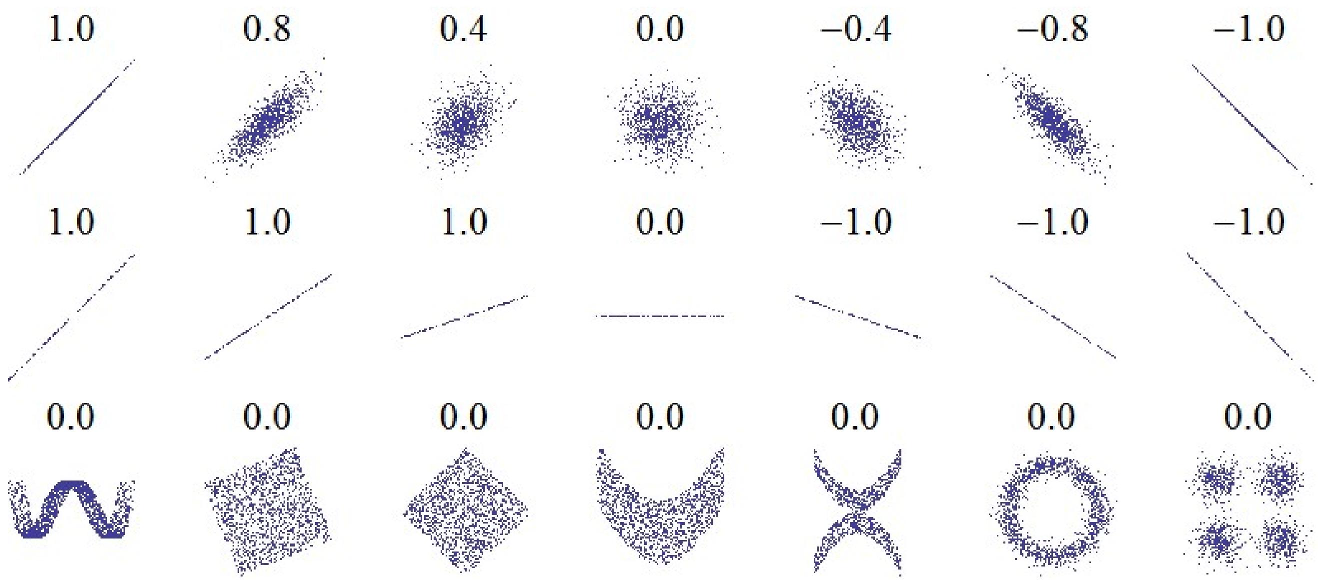
Gaussian Distribution





In this example, x and "20x+y" are clearly not independent

Gaussian Distribution



The correlation coefficient $\rho_{X,Y}$ between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y},$$

Source: WIKI

Correlation	Negative	Positive
Small	−0.3 to −0.1	0.1 to 0.3
Medium	−0.5 to −0.3	0.3 to 0.5
Large	−1.0 to −0.5	0.5 to 1.0

Gaussian Distribution

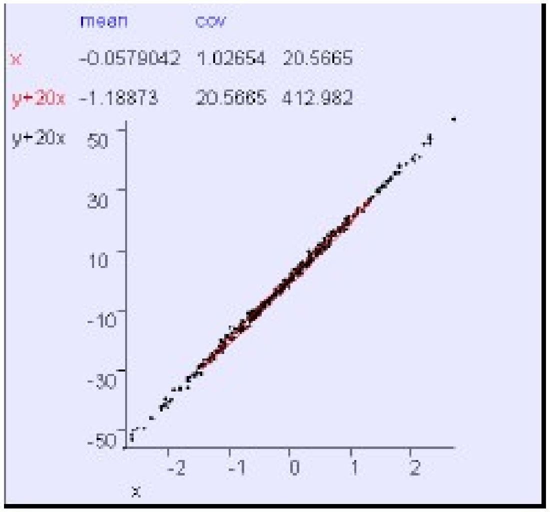
$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y},$$

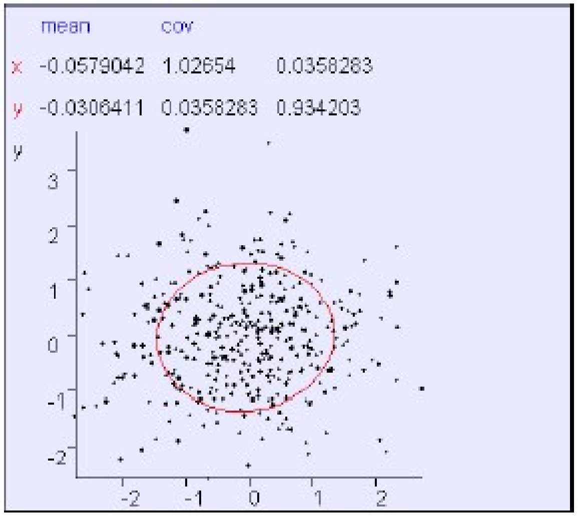
$$\rho_{X,Y} = \frac{\Sigma_{XY}}{\sigma_x \sigma_y}$$

$$= \frac{\Sigma_{XY}}{\Sigma_x^{0.5} \Sigma_y^{0.5}}$$

$$= \frac{20.5665}{\sqrt{1.02654}\sqrt{412.982}}$$

$$= 0.998$$





Gaussian Distribution

Error Ellipses in Action

Kai Arras
Social Robotics Lab, University of Freiburg

April 2010 Social Robotics Laboratory

Gaussian Distribution

- Multivariate

$$\mathbf{x} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix} \qquad X \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{m/2} \, \|\boldsymbol{\Sigma}\|^{1/2}} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \, \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma^2{}_1 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{12} & \sigma^2{}_2 & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1m} & \sigma_{2m} & \cdots & \sigma^2{}_m \end{pmatrix}$$

Gaussian Distribution

- **Calculations of Gaussian**
- Summation

$x \sim N(\mu_x, \Sigma_x)$
$y \sim N(\mu_y, \Sigma_y)$ $\Rightarrow$ $x + y \sim N(\mu_x + \mu_y, \Sigma_x + \Sigma_y)$

- Multiplication

$x \sim N(\mu_x, \Sigma_x)$ $\Rightarrow$ $bx \sim N(b\mu_x, b^T \Sigma_x b)$

- Conditional Gaussian

$x \sim N(\mu_x, \Sigma_{xc})$
$z \sim N(\mu_z, \Sigma_{zz})$ $\Rightarrow$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$

Gaussian Distribution

- Conditional Gaussian

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}, P(\mathbf{x}) \sim N(\mu_x, \Sigma_{xx}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{xx}|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_x)^T \Sigma_{xx}^{-1} (\mathbf{x}-\mu_x)}$$

$$\mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \mathbf{x} \text{ and } \mathbf{z} \text{ are jointly Gaussian, } P(\mathbf{z}) \sim N(\mu_z, \Sigma_{zz})$$

Reminder: We want to find $P(\mathbf{y}_* \mid \mathbf{x}, \mathbf{y}, \mathbf{x}_*) = ?$

$$Find\ P(\mathbf{x}|\mathbf{z}) = ?$$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$

See the appendix for the proof.

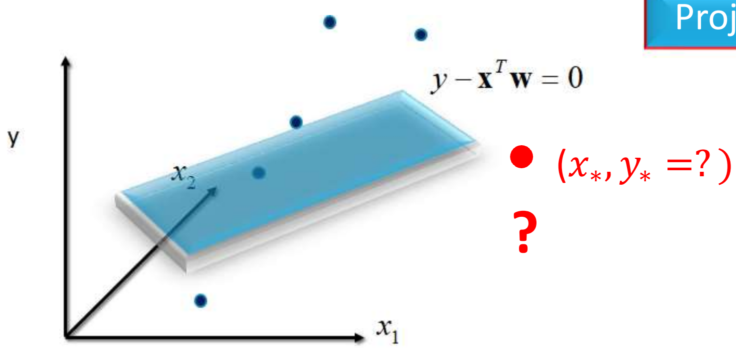
Go to Appendix

MAP of Bayesian Linear Regression

- Gaussian processes proof flow:

1

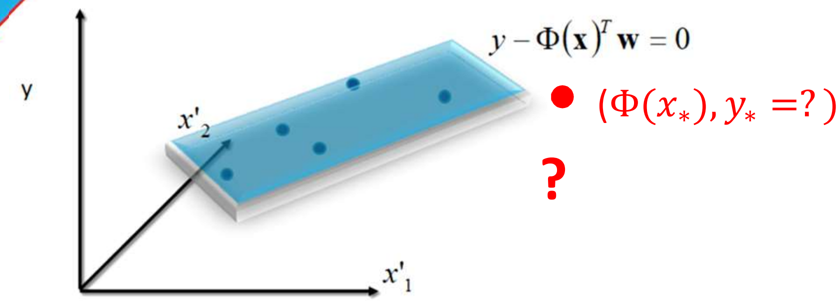
$$y = \mathbf{x}^T\mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$



Projection

3

$$y = \Phi(\mathbf{x})^T\mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$



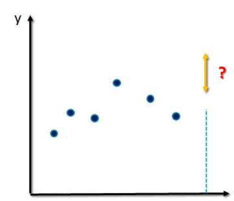
2

MAP :

$$P(\mathbf{w} \mid \mathbf{y}, X) \propto P(\mathbf{y} \mid X, \mathbf{w}) P(\mathbf{w})$$

Prediction :

$$P(y_* \mid x_*, y, X)$$



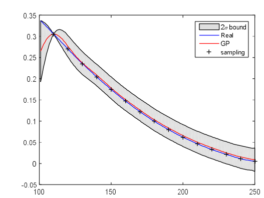
4

MAP :

$$P(\mathbf{w} \mid \mathbf{y}, X) \propto P(\mathbf{y} \mid X, \mathbf{w}) P(\mathbf{w})$$

Prediction :

$$P(y_* \mid x_*, y, X)$$



MAP of Bayesian Linear Regression

$\{(\mathbf{x}_i, y_i)\}, i = 1 \sim N$

$y : measuremnt$

$\mathbf{x} : traing\ sample$

$\boxed{\begin{array}{l} f_i, i = 1 \sim m \\ f : features \end{array}}$

$\longrightarrow$ Algorithm $\longrightarrow$ $Q(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})$

Parametric approximation

Training data D

$D = \{(\boldsymbol{x_i}, y_i) | i = 1, \ldots . N\} = \{X, \boldsymbol{y}\}$

$$\mathrm{y} = \begin{bmatrix} x_{\{1,1\}} & \cdots & x_{\{1,m\}} \\ \vdots & \ddots & \vdots \\ x_{\{N,1\}} & \cdots & x_{\{N,m\}} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix} + \epsilon$$

$$y = \sum_i w_i f_i(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$\Rightarrow y = \mathbf{x}^T \mathbf{w} + \varepsilon$$

Let's <u>take x vector</u> as **features**.
If we can find w vector, we got a linear approximation model!
We can use online/offline least square methods to find w.

MAP of Bayesian Linear Regression

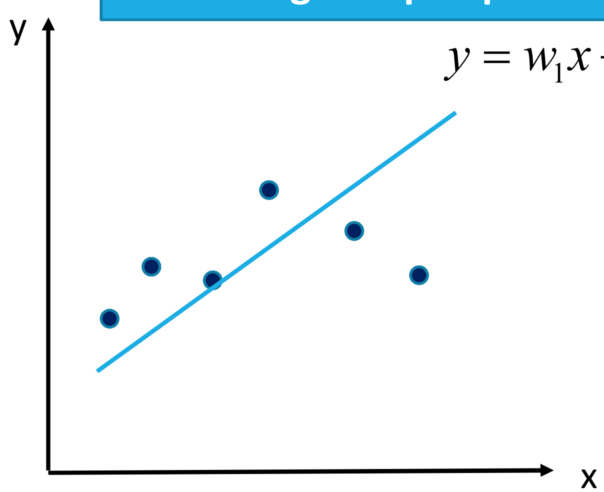
- Let's look at this problem from probabilistic perspective.

$$y = \mathbf{x}^T \mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

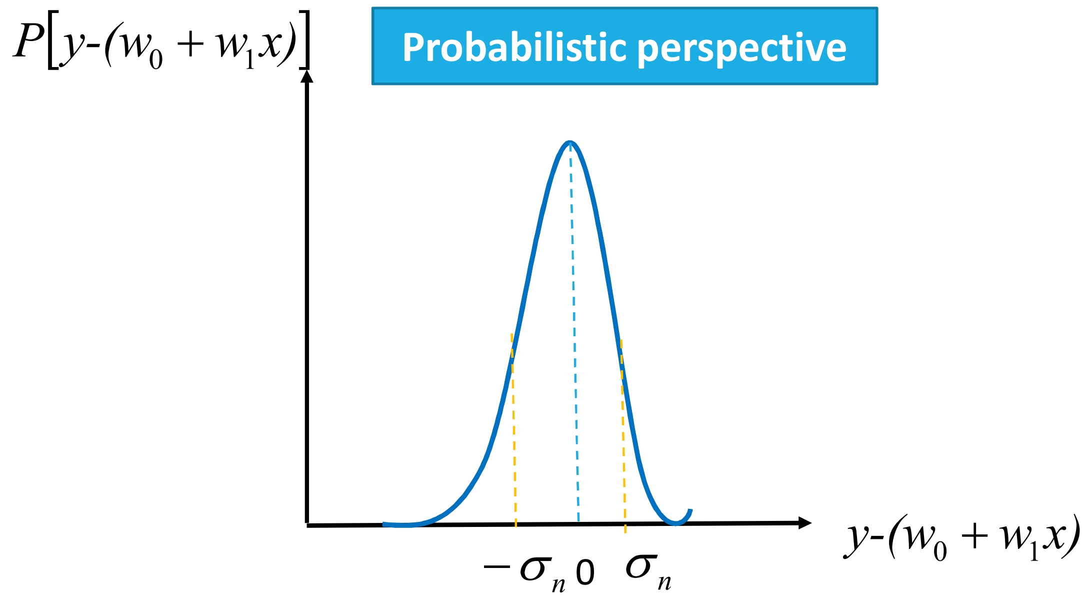
$$y = w_0 + w_1 x + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$y\text{-}(w_0 + w_1 x) = \varepsilon$$

Linear algebra perspective



Probabilistic perspective



MAP of Bayesian Linear Regression

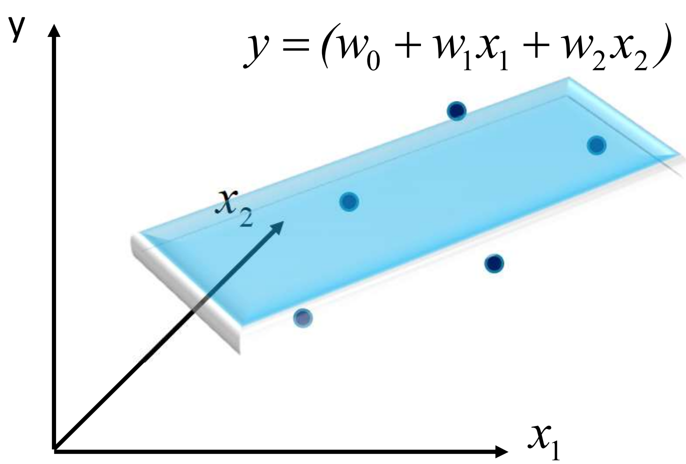
- Let's look at this problem from probabilistic perspective.

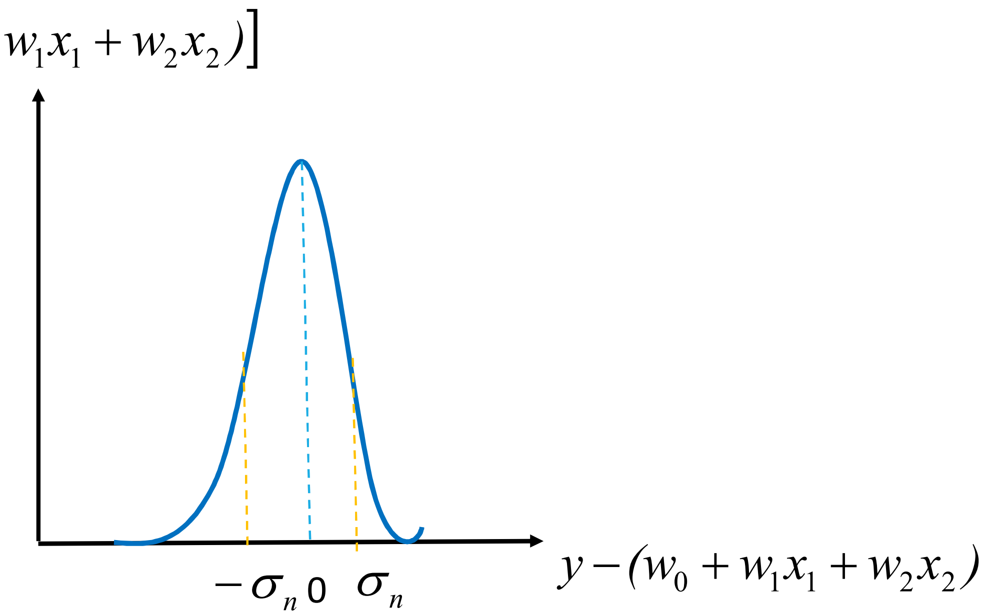
$$y = \mathbf{x}^T \mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$y = w_1 x + w_0$$

$$y = w_0 + w_1 x_1 + w_2 x_2 + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$y\text{-}(w_0 + w_1 x_1 + w_2 x_2) = \varepsilon$$





MAP of Bayesian Linear Regression

- Let's look at this problem in probabilistic domain and try to find **maximum a posterior** (MAP) estimation of w.

$$y = \mathbf{x}^T \mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}$$

$$P(\mathbf{w} \mid \mathbf{y}, X) = \frac{P(\mathbf{y} \mid X, \mathbf{w}) P(\mathbf{w})}{P(\mathbf{y} \mid X)}$$

$$P(\mathbf{w} \mid \mathbf{y}, X) \propto P(\mathbf{y} \mid X, \mathbf{w}) P(\mathbf{w})$$

Let's find w vector with **MAP** estimation.

$$P(\mathbf{w} \mid \mathbf{y}, X) = \frac{P(\mathbf{y}, X \mid \mathbf{w}) P(\mathbf{w})}{P(\mathbf{y}, X)}$$

$$= \frac{P(\mathbf{y} \mid X, \mathbf{w}) P(X \mid \mathbf{w}) P(\mathbf{w})}{P(\mathbf{y} \mid X) P(X)} \ldots\ldots \because P(X \mid \mathbf{w}) = P(X)$$

$$= \frac{P(\mathbf{y} \mid X, \mathbf{w}) P(\mathbf{w})}{P(\mathbf{y} \mid X)}$$

$\mathbf{x}^T$: x vector/matrix

X : input data

$\boldsymbol{y}$: $output\ data$

MAP of Bayesian Linear Regression

$$P(\mathbf{w} \mid y, X) \propto \underline{P(y \mid X, \mathbf{w})}\underline{P(\mathbf{w})}$$

$$\mathbf{y} = \mathbf{x}^T \mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$\underline{P(\mathbf{y} \mid X, \mathbf{w})}$$

$$= \prod_{i=1}^{n} p(y_i \mid \mathbf{x}_i, \mathbf{w}) \quad \because \text{iid}$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{\left(y_i - \mathbf{x}_i^T \mathbf{w}\right)^2}{2\sigma_n^2} \right)$$

$$= \frac{1}{\left(2\pi\sigma_n\right)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2} \left| \mathbf{y} - X^T \mathbf{w} \right|^2 \right)$$

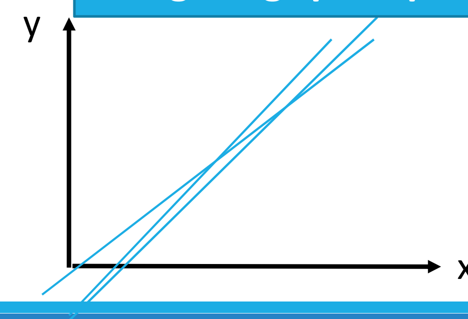
$$P(\mathbf{y} \mid X, \mathbf{w}) \sim N(X^T \mathbf{w}, \sigma_n^2)$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{m/2} \| \mathbf{\Sigma} \|^{1/2}} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)$$

Assume $P(\mathbf{w}) \sim N(0, \underline{\Sigma_p})$

$$\underline{P(\mathbf{w})} = \frac{1}{(2\pi)^{n/2} \left\| \Sigma_p \right\|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{w}^T \Sigma_p^{-1} \mathbf{w} \right)$$

Weighting space perspective



MAP of Bayesian Linear Regression

$$P(\mathbf{y} \mid X, \mathbf{w}) \sim N(\quad X^T \mathbf{w}, \sigma_n^2)$$

$$\underline{P(\mathbf{y} \mid X, \mathbf{w})}$$

$$= \frac{1}{(2\pi\sigma)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2} \left| \mathbf{y} \quad - X^T \mathbf{w} \right|^2 \right)$$

$$P(\mathbf{w}) \sim N(0, \Sigma_p)$$

$$\underline{P(\mathbf{w})} = \frac{1}{(2\pi)^{n/2} \left\| \Sigma_p \right\|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{w}^T \Sigma_p^{-1} \mathbf{w} \right)$$

$$P(\mathbf{w} \mid \mathbf{y}, X) \propto \underline{P(\mathbf{y} \mid X, \mathbf{w})}\,\underline{P(\mathbf{w})}$$

$$\propto \frac{1}{(2\pi\sigma)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2} \left| \mathbf{y} \quad - X^T \mathbf{w} \right|^2 \right) \frac{1}{(2\pi)^{n/2} \left\| \Sigma_p \right\|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{w}^T \Sigma_p^{-1} \mathbf{w} \right)$$

$$\propto \exp\left(-\frac{1}{2\sigma_n^2} \left| \mathbf{y} \quad - X^T \mathbf{w} \right|^2 \right) \exp\left(-\frac{1}{2} \mathbf{w}^T \Sigma_p^{-1} \mathbf{w} \right)$$

MAP of Bayesian Linear Regression

$$P(\mathbf{w} \mid \mathbf{y}, X) \propto \underline{P(\mathbf{y} \mid X, \mathbf{w})}\,\underline{P(\mathbf{w})}$$

$$\propto \exp\left(-\frac{1}{2\sigma_n^2}\left|\mathbf{y} - X^T\mathbf{w}\right|^2\right)\exp\left(-\frac{1}{2}\mathbf{w}^T \Sigma_p^{-1}\mathbf{w}\right)$$

$$\propto \exp\left(-\frac{1}{2}(\mathbf{w}-\overline{\mathbf{w}})^T\left(\frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1}\right)(\mathbf{w}-\overline{\mathbf{w}})\right)$$

$$\text{where } \overline{\mathbf{w}} = \sigma_n^{-2}\left(\sigma_n^{-2}XX^T + \Sigma_p^{-1}\right)^{-1}X\mathbf{y}$$

$$P(\mathbf{w} \mid \mathbf{y}, X) \sim N\left(\overline{\mathbf{w}} = \frac{1}{\sigma_n^2}A^{-1}X\mathbf{y}, A^{-1}\right)$$

$$\text{where } A = \sigma_n^{-2}XX^T + \Sigma_p^{-1}$$

We got a new distribution and can use it to predict yi.

$$\boxed{p(\mathbf{x}) = \frac{1}{(2\pi)^{m/2}\,\|\boldsymbol{\Sigma}\|^{1/2}}\exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}$$

Expand the summation term to Gaussian form.

$$\underline{P(\mathbf{y} \mid X, \mathbf{w})} \sim N(X^T\mathbf{w}, \sigma_n^2)$$

$$\underline{P(\mathbf{w})} \sim N(0, \Sigma_p)$$

MAP of Bayesian Linear Regression

- Prediction based on a Gaussian distribution

$$P(\mathbf{w} \mid y, X) \sim N\left(\overline{\mathbf{w}} = \frac{1}{\sigma_n^2} A^{-1} X\mathbf{y}, A^{-1}\right)$$

Prediction

(Total probability & independence) (Scaling a Gaussian to another Gaussian)

$$P(y_* \mid x_*, y, X) = \int \underline{P(y_* \mid x_*, \mathbf{w})} P(\mathbf{w} \mid y, X) d\mathbf{w} = \int \underline{x_*^T} \cdot P(\mathbf{w} \mid y, X) d\mathbf{w}$$

$$= N\left(\frac{1}{\sigma_n^2} \underline{x_*^T} A^{-1} X\mathbf{y}, \underline{x_*^T} A^{-1} \underline{x_*}\right)$$

$$x \sim N(\mu_x, \Sigma_x)$$
$$bx \sim N(b\mu_x, b^T \Sigma_x b)$$

where $A = \sigma_n^{-2} XX^T + \Sigma_p^{-1}$

$$\begin{cases} \overline{y}_* = \dfrac{1}{\sigma_n^2} x_*^T A^{-1} X\mathbf{y} \\ \Sigma_{y_*} = x_*^T A^{-1} x_* \end{cases}$$

Where is **W**?

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz} \Sigma_{zz}^{-1} (\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz} \Sigma_{zz}^{-1} \Sigma_{zx} \end{cases}$$

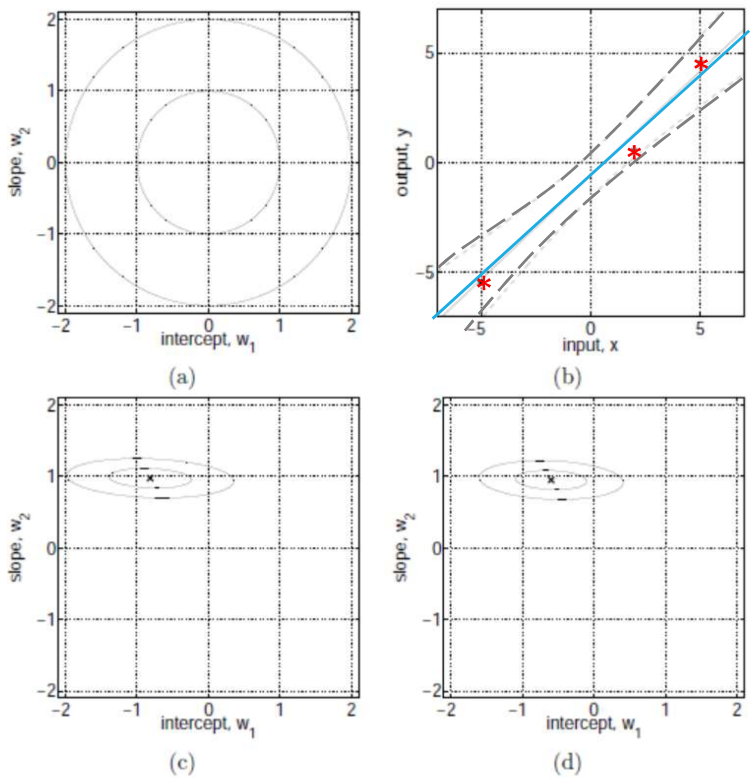
MAP of Bayesian Linear Regression

- Prediction based on a Gaussian distribution $P(\mathbf{w} \mid \mathbf{y}, X) \propto P(\mathbf{y} \mid X, \mathbf{w}) P(\mathbf{w})$

$\mathrm{P}(w) \sim N(0, I)$

$y = w_1 + w_2 x + \epsilon$

$P(y_* | x_* . X, y)$



$\mathrm{P}(y|X, w)$

Likelihood

$\mathrm{P}(w|y, X)$

Posterior

MAP of Bayesian Linear Regression

- Prediction based on a Gaussian distribution

Given :

$\{(\mathbf{x}_i, y_i)\}, i = 1 \sim 6$

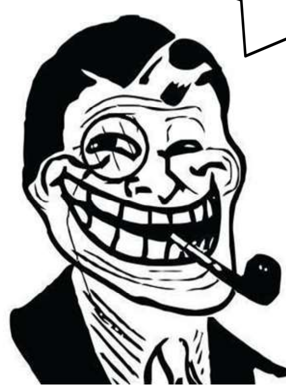
$\mathbf{x}_7$

Find :

$y_7 = ?\ \mathrm{cov}(y_7) = ?$

$$\begin{cases} \bar{y}_* = \dfrac{1}{\sigma_n^2} x_*^T A^{-1} X\mathbf{y} \\ \Sigma_{y_*} = x_*^T A^{-1} x_* \end{cases}$$

We can use it to predict yi based on the training data!



Kuo-Shih

Math students

Kernel functions

- The prediction of linear model could not work very well.

$$y = \mathbf{x}^T \mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

- In this model, it's a special case of feature base approximation. We adopt "X" as features.

Kuo-Shih

Math students

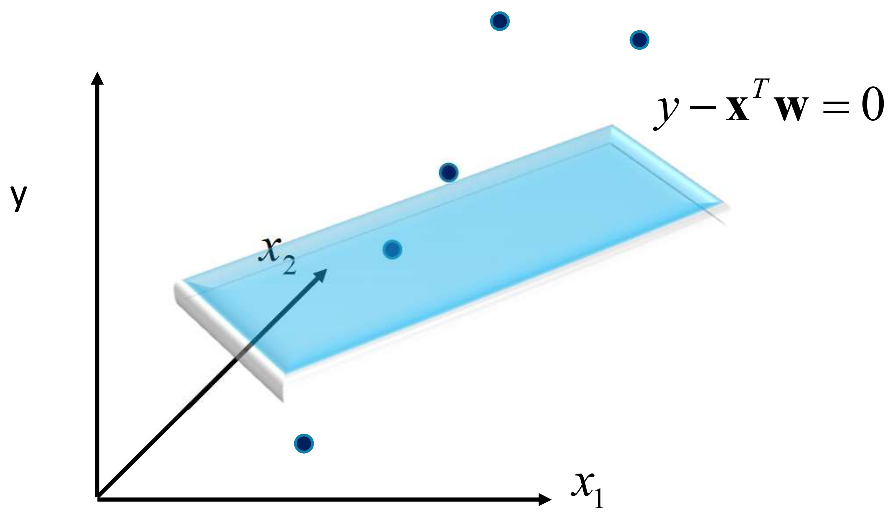
Kernel functions

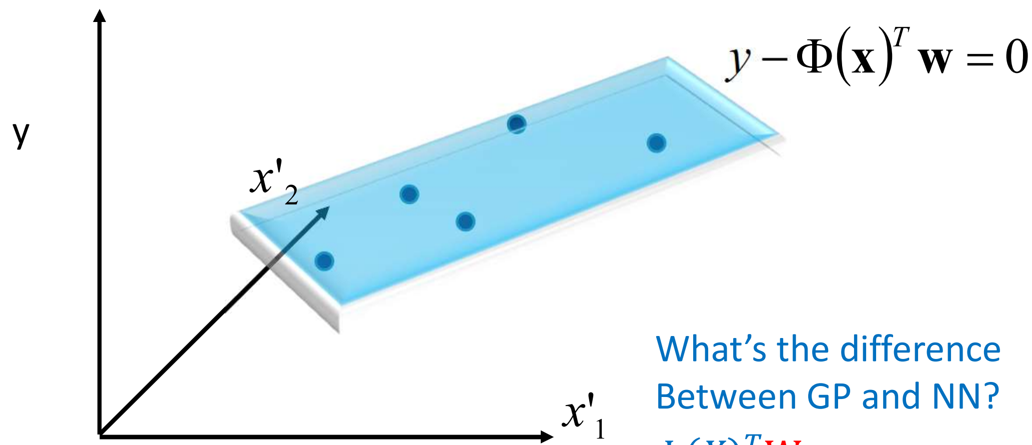
- Let's project data (X) to feature space.

$$y = \mathbf{x}^T \mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$

$$y\text{-}(w_0 + w_1 x_1 + w_2 x_2) = \varepsilon$$

$$y = \Phi(\mathbf{x})^T \mathbf{w} + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2)$$





What's the difference
Between GP and NN?

$\Phi(X)^T W$

$\Phi(X^T W)$

Kernel functions

- Let’s project data (X) to feature space.

$$\underset{D\times 1\ \text{vector}}{\mathbf{x}} \longrightarrow \boxed{\Phi(.)} \longrightarrow \underset{N\times 1\ \text{vector}}{\Phi(\mathbf{x})}$$

$$y = \mathbf{x}^T\mathbf{w} + \varepsilon \qquad\qquad y = \Phi(\mathbf{x})^T\mathbf{w} + \varepsilon$$

If $\Phi(x) = (1, x, x^2, x^3, \ldots.)^T$,

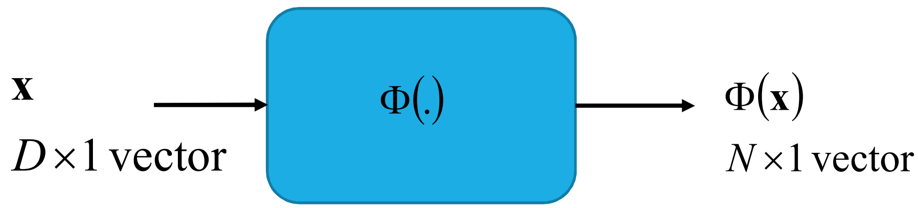
it's a polynomial regression!

This is still a feature based regression!!

Math students

Kernel functions

- Let's project data (X) to feature space.



$$y = \mathbf{x}^T \mathbf{w} + \varepsilon$$

$$P(y_* \mid x_*, y, X) = N\left(\frac{1}{\sigma_n^2} x_*^T A^{-1} X\mathbf{y}, x_*^T A^{-1} x_*\right)$$

$$\text{where } A = \sigma_n^{-2} X X^T + \Sigma_p^{-1}$$

$$\begin{cases} \bar{y}_* = \dfrac{1}{\sigma_n^2} x_*^T A^{-1} X\mathbf{y} \\ \Sigma_{y_*} = x_*^T A^{-1} x_* \end{cases}$$

$$y = \Phi(\mathbf{x})^T \mathbf{w} + \varepsilon$$

$$P(y_* \mid x_*, y, X) = N\left(\frac{1}{\sigma_n^2} \underline{\Phi(x_*)}^T A^{-1} X\mathbf{y}, \underline{\Phi(x_*)}^T A^{-1} \underline{\Phi(x_*)}\right)$$

$$\text{where } A = \sigma_n^{-2} \underline{\Phi \Phi^T} + \Sigma_p^{-1}$$

$$\begin{cases} \bar{y}_* = \dfrac{1}{\sigma_n^2} \underline{\Phi(x_*)}^T A^{-1} \Phi\mathbf{y} \\ \Sigma_{y_*} = \underline{\Phi(x_*)}^T A^{-1} \underline{\Phi(x_*)} \end{cases}$$

Kernel functions

$$P(y_* \mid x_*, y, X) = N\left(\frac{1}{\sigma_n^2}\Phi(x_*)^T A^{-1} X\mathbf{y}, \Phi(x_*)^T A^{-1}\Phi(x_*)\right)$$

where $A = \sigma_n^{-2}\Phi\Phi^T + \Sigma_p^{-1} = \sigma_n^{-2}K + \Sigma_p^{-1}$

$$\begin{cases} \bar{y}_* = \dfrac{1}{\sigma_n^2}\Phi(x_*)^T A^{-1}\Phi\mathbf{y} \\ \Sigma_{y_*} = \Phi(x_*)^T A^{-1}\Phi(x_*) \end{cases}$$

Let $Z^{-1} = \Sigma_p, W^{-1} = \sigma_n^2 I, U = V = \Phi$

$$A^{-1} = \left(\sigma_n^{-2}\Phi\Phi^T + \Sigma_p^{-1}\right)^{-1}$$

$$= \Sigma_p - \Sigma_p \Phi\left(\sigma_n^2 I + \Phi \Sigma_p \Phi^T\right)\Phi^T \Sigma_p$$

$$\Sigma_{y_*} = \Phi(x_*)^T A^{-1}\Phi(x_*) = \underline{\Phi_*^T \Sigma_p \Phi_*} - \underline{\Phi_*^T \Sigma_p \Phi}\ \underline{\left(K + \sigma_n^2 I\right)^{-1}}\ \underline{\Phi^T \Sigma_p \Phi_*}$$

Matrix inversion lemma :

$$\left(\mathrm{Z} + \mathrm{UWV}^{\mathrm{T}}\right)^{-1}$$

$$= \mathrm{Z}^{-1} - \mathrm{Z}^{-1}U\left(W^{-1} + V^T \mathrm{Z}^{-1} U\right)^{-1} V^T \mathrm{Z}^{-1}$$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}\left(\mathbf{z} - \mu_z\right) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$

Kernel functions

$$\Sigma_{y_*} = \Phi_*^T \Sigma_p \Phi_* - \Phi_*^T \Sigma_p \Phi \left(K + \sigma_n^2 I\right)^{-1} \Phi^T \Sigma_p \Phi_*$$

$$\mu_x = \mathbf{0}, \mu_z = \mathbf{0}$$

$$\Sigma_{xz} = \Phi_*^T \Sigma_p \Phi$$

$$\Sigma_{zz} = K + \sigma_n^2 I$$

$$\therefore \bar{y}_* = \Phi_*^T \Sigma_p \Phi\left(K + \sigma_n^2 I\right)^{-1} \mathbf{y}$$

$$\boxed{\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}\left(\mathbf{z} - \mu_z\right) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}}$$

The Gaussian distribution is as follows:

$$\begin{cases} \bar{y}_* = \Phi_*^T \Sigma_p \Phi\left(K + \sigma_n^2 I\right)^{-1} \mathbf{y} \\ \Sigma_{y_*} = \Phi_*^T \Sigma_p \Phi_* - \Phi_*^T \Sigma_p \Phi \left(K + \sigma_n^2 I\right)^{-1} \Phi^T \Sigma_p \Phi_* \end{cases}$$

Kernel functions

- Let's rewrite it.

$$y = \Phi(\mathbf{x})^T \mathbf{w} + \varepsilon$$

$$P(y_* \mid x_*, y, X) = N\left(\Phi_*^T \Sigma_p \Phi\left(K + \sigma_n^2 I\right)^{-1} \mathbf{y}, \Phi_*^T \Sigma_p \Phi_* - \Phi_*^T \Sigma_p \Phi \left(K + \sigma_n^2 I\right)^{-1} \Phi^T \Sigma_p \Phi_*\right)$$

where $K = \Phi^T \Sigma_p \Phi$

let's define $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \Sigma_p \Phi(\mathbf{x}')$

$$k_* = k(\mathbf{x}, \mathbf{x}_*)$$

$$k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$$

$$P(y_* \mid x_*, y, X) = N\left(k_*^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y}, k_{**} - k_*^T \left(K + \sigma_n^2 I\right)^{-1} k_*\right)$$

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz} \Sigma_{zz}^{-1} (\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz} \Sigma_{zz}^{-1} \Sigma_{zx} \end{cases}$$

We call "K" kernel function. It can project data to feature space.
How to choose a good kernel?

Kernel functions

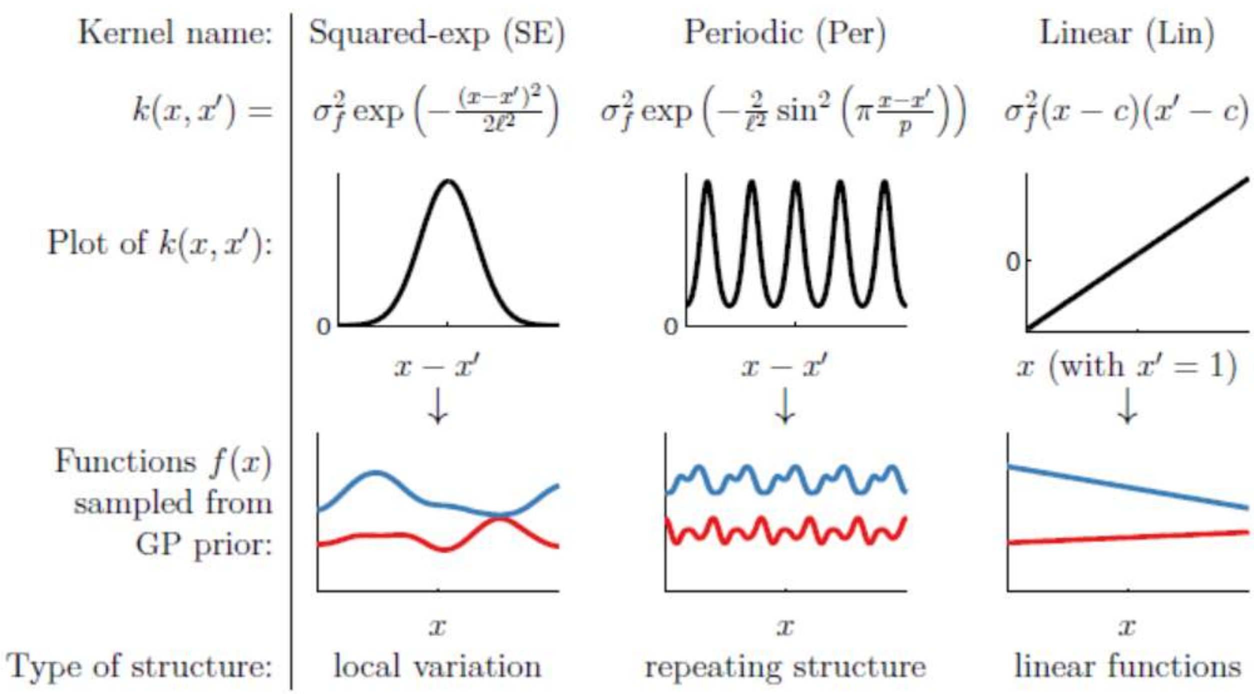


Figure 1.1: Examples of structures expressible by some basic kernels.

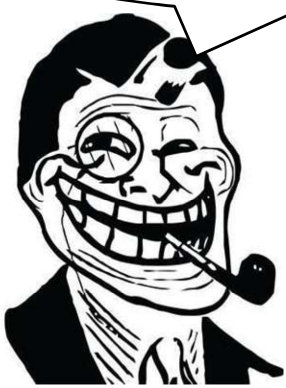
https://github.com/duvenaud/phd-thesis

Kernel functions

- The squared exponential kernel

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

We call them
"Hyperparameters"



Kuo-Shih

You use two
parameters!!

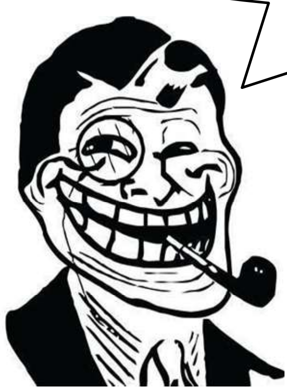
Math students

Kernel functions

- The squared exponential kernel

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

Hyperparameters can be learned from data (X)!



Kuo-Shih

What’s the difference?

Math students

Gaussian Processes (GP)

Definition : A Gaussian process is a collection of random variables with a joint Gaussian distribution.

(Function space view)

$$y = \Phi(\mathbf{x})^T \mathbf{w} + \varepsilon, \;\; \varepsilon \sim N(0, \sigma_n^2)$$

let's define $f(\mathbf{x}) = \Phi(\mathbf{x})^T \mathbf{w}, \mathbf{w} \sim N(0, \Sigma_P)$

Mean function : $m(\mathbf{x}) = E[f(\mathbf{x})]$

Covariance function : $k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x})')]$

$\Rightarrow f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$

$\boxed{\text{Assume } P(\mathbf{w}) \sim N(0, \Sigma_p)}$

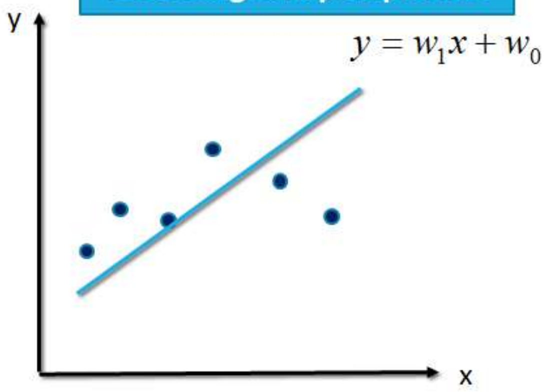
$$m(\mathbf{x}) = E[f(\mathbf{x})] = \Phi(\mathbf{x})^T E[\mathbf{w}] = 0$$

$$E[f(\mathbf{x}) f(\mathbf{x}')] = \Phi(\mathbf{x})^T E[\mathbf{w}\mathbf{w}^T] \Phi(\mathbf{x}') = \Phi(\mathbf{x})^T \Sigma_p \Phi(\mathbf{x}')$$

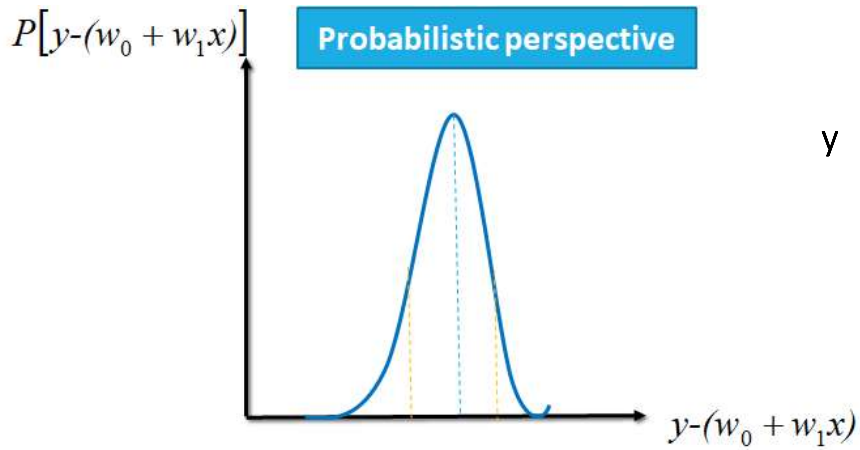
$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

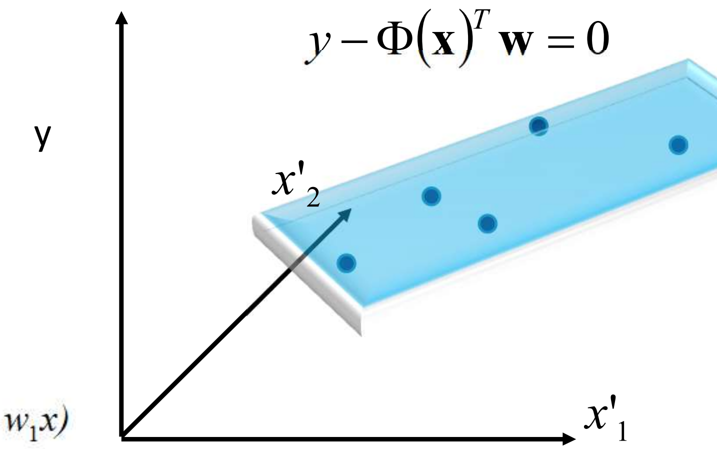
Gaussian Processes (GP)

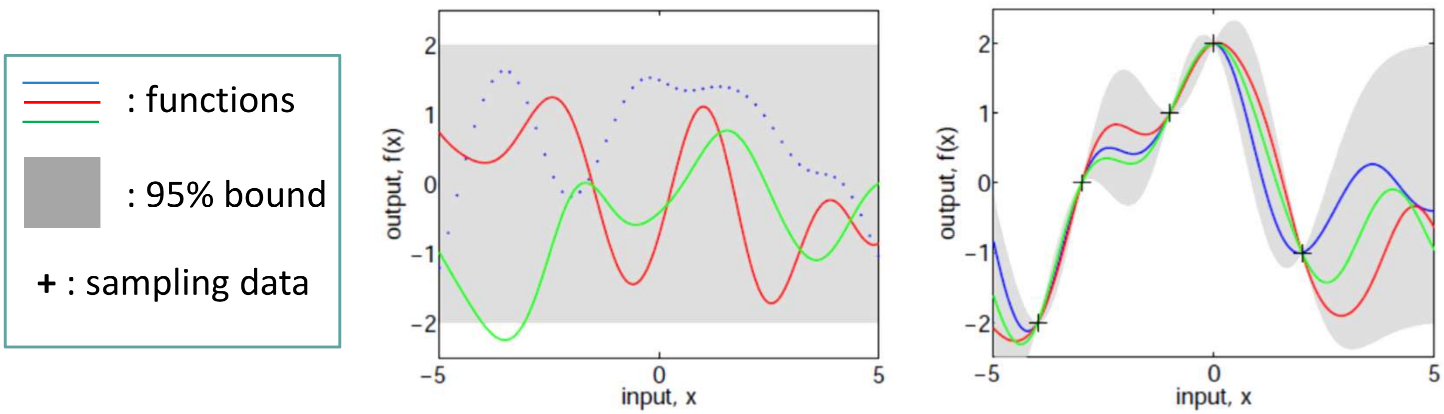
Linear algebra perspective



Probabilistic perspective







(a), prior (b), posterior

Function space perspective

$$P(\mathbf{w} \mid \mathbf{y}, X) \propto P(\mathbf{y} \mid X, \mathbf{w}) P(\mathbf{w})$$

$$y = \Psi(\boldsymbol{x})^T \boldsymbol{w} + \epsilon$$

$$\Rightarrow y = f(\boldsymbol{x}) + \epsilon$$

Gaussian Processes (GP)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}, P(\mathbf{x}) \sim N(\mu_x, \Sigma_{xx}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{xx}|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_x)^T \Sigma_{xx}^{-1} (\mathbf{x}-\mu_x)}$$

$$\mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \mathbf{x} \text{ and } \mathbf{z} \text{ are jointly Gaussian, } P(\mathbf{z}) \sim N(\mu_z, \Sigma_{zz})$$

$$Find\ P(\mathbf{x}|\mathbf{z}) = ?$$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$

Gaussian Processes (GP)

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

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z}-\mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$

$$\begin{cases} \mu_{y_*|y} = \mu_{y_*} + \Sigma_{y_* y}\Sigma_{yy}^{-1}(\mathbf{y}-\mathbf{0}) = K(X_*,X)\left(K(X,X)+\sigma_n^2 I\right)^{-1}\mathbf{y} \\ \Sigma_{y_* y_*|y} = \Sigma_{y_* y_*} - \Sigma_{y_* y}\Sigma_{yy}^{-1}\Sigma_{yy_*} = K(X_*,X_*) - K(X_*,X)\left(K(X,X)+\sigma_n^2 I\right)^{-1}K(X,X_*) \end{cases}$$

$$\begin{cases} \bar{y}_* = K_*^T\left(K+\sigma_n^2 I\right)^{-1}\mathbf{y} \\ \Sigma_{y_*} = K_{**} - K_*^T\left(K+\sigma_n^2 I\right)^{-1}K_* \end{cases}$$

Gaussian Processes (GP)

- Prediction algorithm

[*Input*]

X : data input, σ_n : *noise*, $\mathbf{y}$: data label

k : covariance function

X_* : test input

$1. \bar{y}_* = K_*^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y}$

$2. \Sigma_{y_*} = K_{**} - K_*^T \left(K + \sigma_n^2 I\right)^{-1} K_*$

Return

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

[Training]

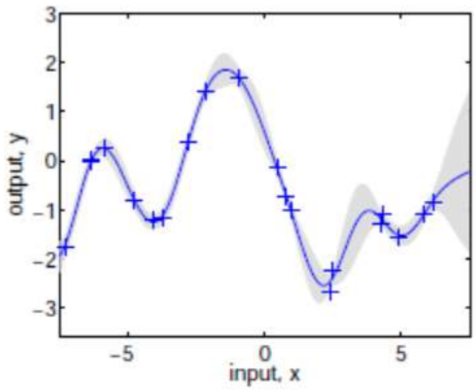
$O\left(N^3\right)$ Matrix invresion

[Prediction]

$O(N^2)$

Gaussian Processes (GP)

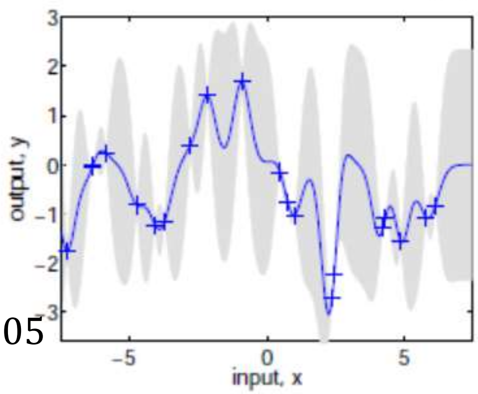
$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x - x')^2\right)$$



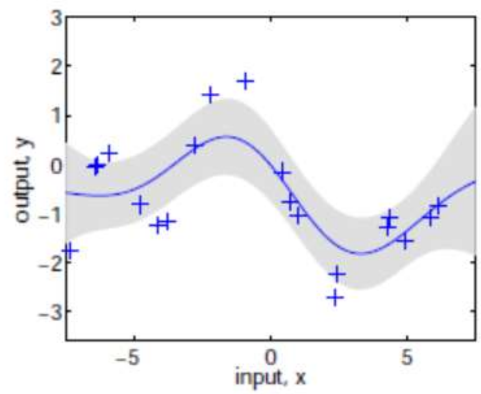
(a), $\ell = 1$

$l = 1$
$\sigma_f = 1$
$\sigma_n = 0.1$

$l = 0.3$
$\sigma_f = 1.08$
$\sigma_n = 0.00005$



(b), $\ell = 0.3$



(c), $\ell = 3$

$l = 3$
$\sigma_f = 1.16$
$\sigma_n = 0.89$

How to tune **hyperparameters**?

Gaussian Processes (GP)

- Learning Hyperparameters

$$\log P(\mathbf{y} \mid X, \mathbf{w})$$

$$= \frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi$$

$$\frac{\partial \log P(\mathbf{y} \mid X, \mathbf{w})}{\partial w_j}$$

$$= \frac{1}{2}\mathbf{y}^T K^{-1}\frac{\partial K}{\partial w_j}\mathbf{y} - \frac{1}{2}tr\left(K^{-1}\frac{\partial K}{\partial w_j}\right)$$

$$= \frac{1}{2}tr\left((K^{-1}\mathbf{y}\mathbf{y}^T K^{-1^T} - K^{-1})\frac{\partial K}{\partial w_j}\right)$$

$$w_j = w_j + \alpha\frac{\partial \log P(\mathbf{y} \mid X, \mathbf{w})}{\partial w_j}$$

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

$$P(\boldsymbol{y}|\mathrm{X}, \boldsymbol{W}) = \frac{1}{2\pi\sigma^{n/2}}\exp(-\frac{1}{2\sigma_n^2}|y - \mathrm{X}^{\mathrm{T}}\boldsymbol{w}|^2)$$

$$P(\boldsymbol{y}|\mathrm{X}, \boldsymbol{W}) = \frac{1}{2\pi\sigma^{n/2}}\exp(-\frac{1}{2\sigma_n^2}|y - \Phi(\mathrm{X})^{\mathrm{T}}\boldsymbol{w}|^2)$$

Gaussian Processes (GP)

- GP Library:
- scikit-learn (Python)
  - https://scikit-learn.org/stable/modules/gaussian_process.html
- SheffieldML (C++)
  - https://github.com/SheffieldML/GPc
- GP ML (MATLAB)
  - http://www.gaussianprocess.org/gpml/code/matlab/doc/
- Gpstuff (MATLAB)
  - https://github.com/gpstuff-dev/gpstuff

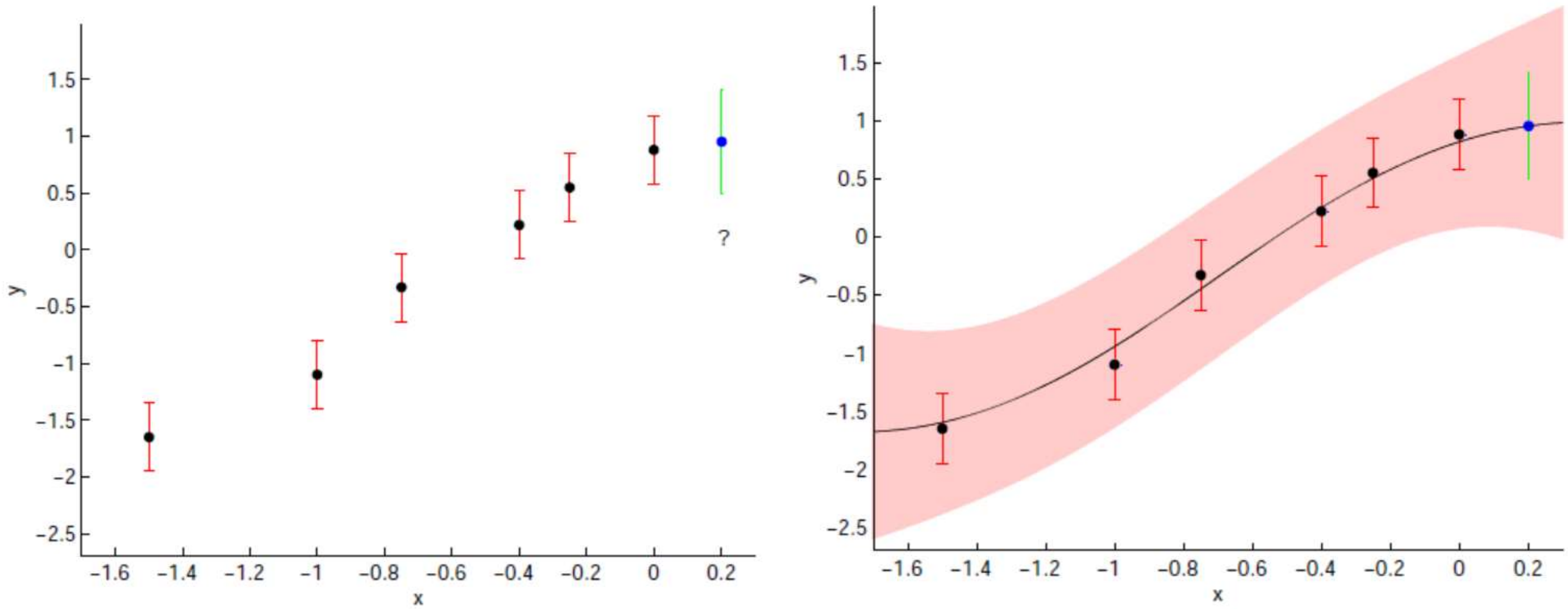
GP — EX:

Given :

$\{(\mathbf{x}_i, y_i)\}, i = 1 \sim 6$

Find :

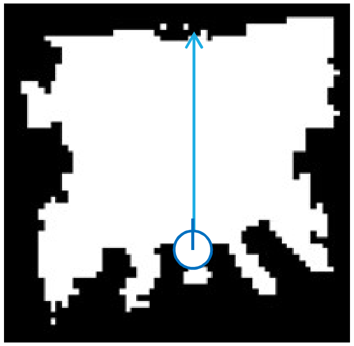
$y_7 = ?\, \mathrm{cov}(y_7) = ?$

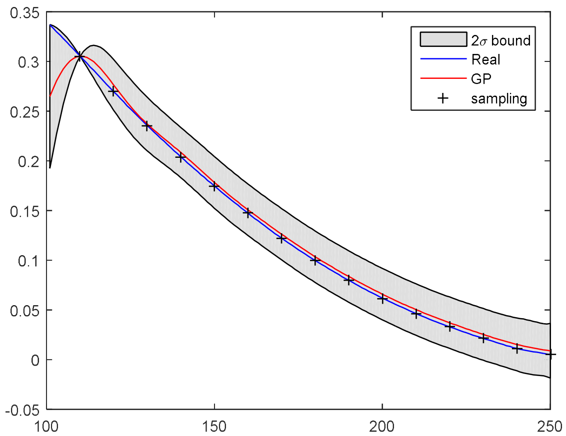


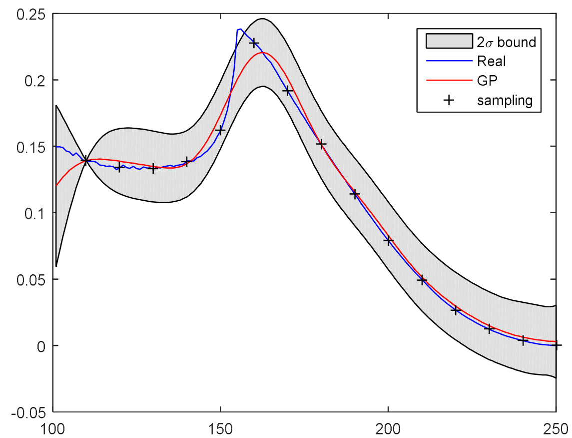
Example source: Mark Ebden, "Gaussian Processes: A Quick Introduction," arXiv, 2015

GP — EX: Coverage Approximation

- 1D case (X: robot's Y position)

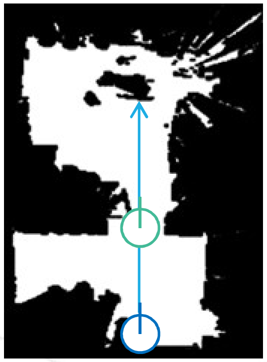


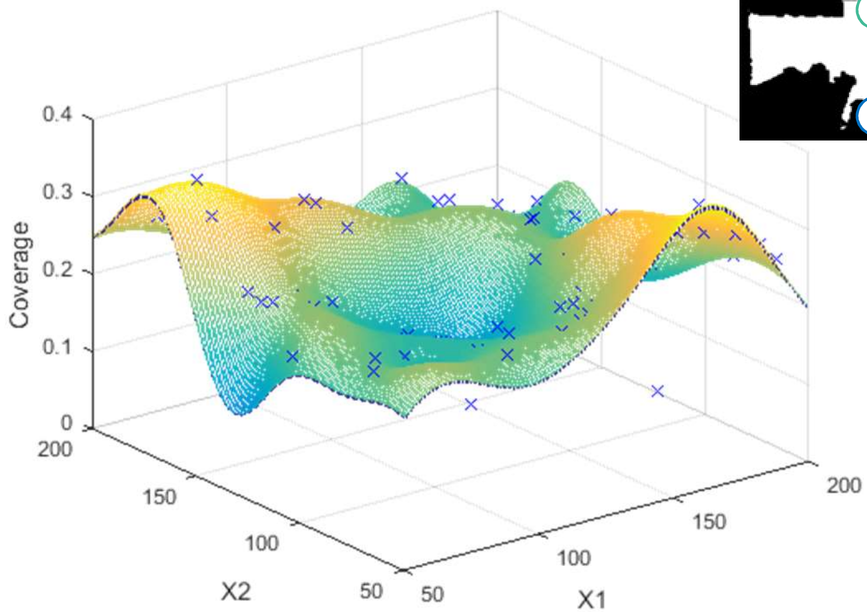


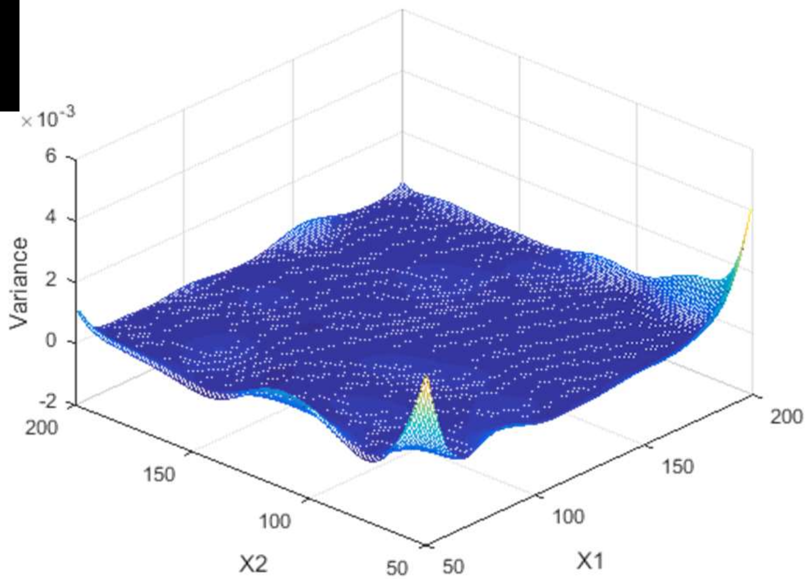


GP — EX: Coverage Approximation

- 2D case (X1: robot1's Y position, X2: robot2's Y position,)







Gaussian Processes (GP)

- ☺
  - Don’t need hand crafting (features)
  - Provide uncertainty information (mean and covariance)
  - Easy to predict (O(N^2))
- ☹
  - Need to select kernels
  - Computational complexity of training: O(N^3), <15,000 samples

Locally Weighted Project Regression(LWPR)

- Fastest & Scalable
  - O(N^2) for training
- Input space is high-dimension, data lies on low-dimension manifold.

Schaal, S.;Atkeson, C. G.;Vijayakumar, S.,“Scalable techniques from nonparameteric statistics for real-time robot learning,” Applied Intelligence, 2002

Locally Weighted Project Regression(LWPR)

Least square(LS):

$$y = \beta X$$

$$\beta = \left(X^T X\right)^{-1} X^T y$$

Weighted LS(WLS):

$$y = \beta X$$

$$w_{ii} = \frac{1}{\sigma^2}$$

$$\beta = \left(X^T W X\right)^{-1} X^T W y$$

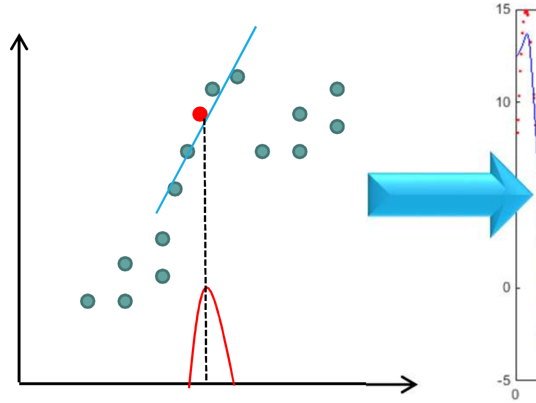
Locally Weighted Regression(LWR):

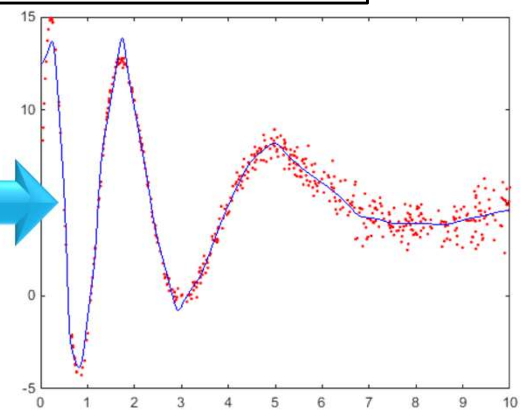
$$y = \beta X$$

$$w_{ii} = \exp\left(-\frac{1}{2}\left(x_i - x_q\right)^T D\left(x_i - x_q\right)\right)$$

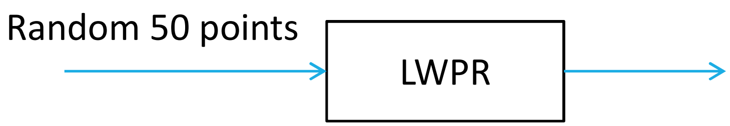
$$\beta = \left(X^T W X\right)^{-1} X^T W y$$

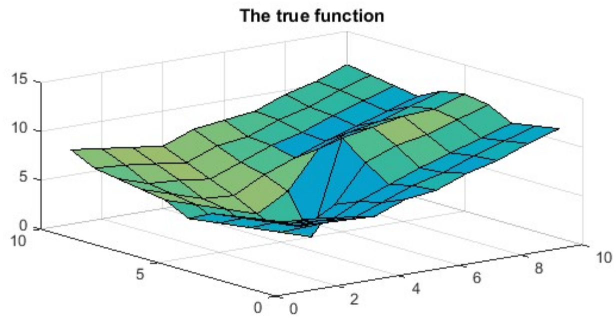
LWPR=LWR + PLS (partial least square)

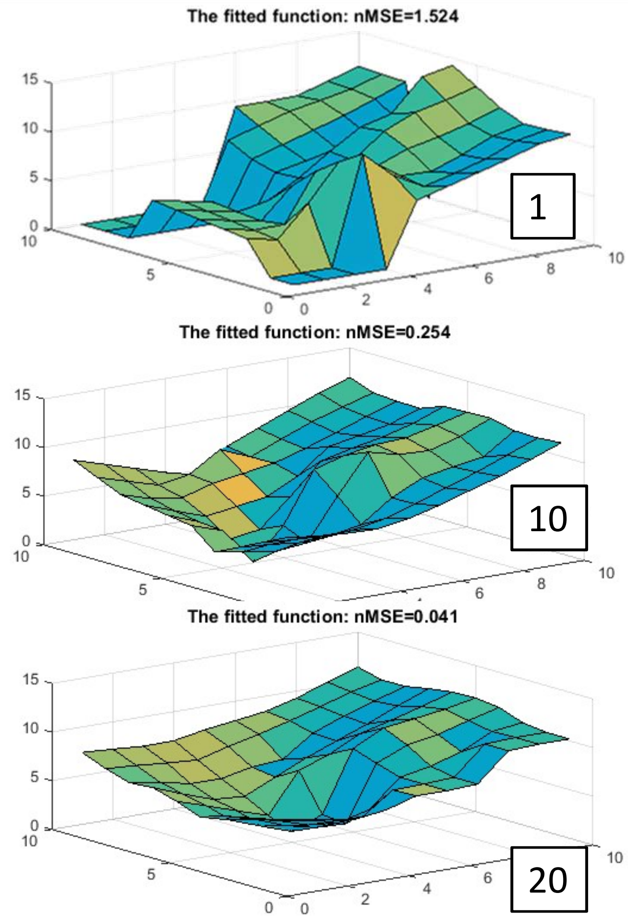




LWPR – EX: CTG







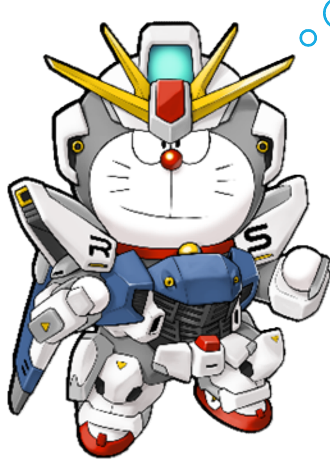
http://wcms.inf.ed.ac.uk/ipab/slmc/research/software-lwpr

Conclusions

- Nonparametric approximation :
- GP generates the most accurate approximation but its complexity is O(N^3).
- LWPR is the fastest method.
- If you have a few data (N<15,000), use GP.
- If you want to be fast, use LWPR.
- If you want to make a trade-off, use LGP, a mixture of GP and LWPR.
- After 2012, researchers start to adapt deep learning.
- During 2018~2020, researchers indicate that GP outperforms DL in many cases.

Nguyen-Tuong, D.; Seeger, M.; Peters, "*Model Learning with Local Gaussian Process Regression*, Advanced Robotics," 2009.

Q&A



Conditional Gaussian

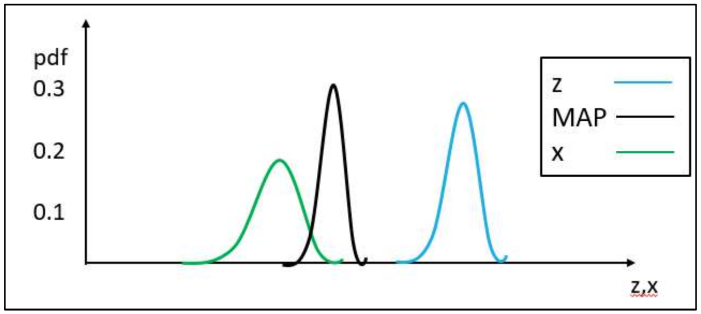
$\mathbf{x}^{MMSE}$ in Gaussian distribution.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}, P(\mathbf{x}) \sim N(\mu_x, \Sigma_{xx}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{xx}|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_x)^T \Sigma_{xx}^{-1} (\mathbf{x}-\mu_x)}$$

$$\mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \mathbf{x} \text{ and } \mathbf{z} \text{ are jointly Gaussian, } P(\mathbf{z}) \sim N(\mu_z, \Sigma_{zz})$$

(multivariate normal distribution)

$$Find\ P(\mathbf{x}|\mathbf{z}) = ?$$



Conditional Gaussian

$$P(\mathbf{x}) \sim N(\mu_x, \Sigma_{xx}) \text{ and } P(\mathbf{z}) \sim N(\mu_z, \Sigma_{zz}), P(\mathbf{x}, \mathbf{z}) = P(\mathbf{y}) = N(\mu_y, \Sigma_{yy})$$

$$\text{Mean}: \mathbf{y} = \begin{bmatrix} \mathbf{x}_{(m,1)} \\ \mathbf{z}_{(n,1)} \end{bmatrix}$$

$$\mu_y = \mathbf{E}[\mathbf{y}] = \mathbf{E}\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{E}[\mathbf{x}] \\ \mathbf{E}[\mathbf{z}] \end{bmatrix} = \begin{bmatrix} \mu_x \\ \mu_z \end{bmatrix}$$

$$\text{Covariance}: \Sigma_{yy} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}$$

$$\Sigma_{xx} = \mathbf{E}\left[(\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^T\right], \Sigma_{zz} = \mathbf{E}\left[(\mathbf{z} - \mu_z)(\mathbf{z} - \mu_z)^T\right]$$

$$\Sigma_{xz} = \mathbf{E}\left[(\mathbf{x} - \mu_x)^T (\mathbf{z} - \mu_z)^T\right], \Sigma_{zx} = \Sigma_{xz}^T$$

Conditional Gaussian

$$P(\mathbf{x}|\mathbf{z}) = \frac{P(\mathbf{x},\mathbf{z})}{P(\mathbf{z})} = \frac{\dfrac{1}{\sqrt{(2\pi)^{m+n}\left|\Sigma_{yy}\right|}}e^{-\frac{1}{2}(\mathbf{y}-\mu_y)^T\Sigma_{yy}^{-1}(\mathbf{y}-\mu_y)}}{\dfrac{1}{\sqrt{(2\pi)^{m}\left|\Sigma_{zz}\right|}}e^{-\frac{1}{2}(\mathbf{z}-\mu_z)^T\Sigma_{zz}^{-1}(\mathbf{z}-\mu_z)}}$$

$$= \frac{1}{\sqrt{(2\pi)^{n}\underline{\left|\Sigma_{yy}\right|/\left|\Sigma_{zz}\right|}}}e^{-\frac{1}{2}\left[\underline{(\mathbf{y}-\mu_y)^T\Sigma_{yy}^{-1}(\mathbf{y}-\mu_y)-(\mathbf{z}-\mu_z)^T\Sigma_{zz}^{-1}(\mathbf{z}-\mu_z)}\right]}$$

$$= \frac{1}{\sqrt{(2\pi)^{n}\beta}}e^{-\frac{1}{2}[\alpha]}$$

If we can find α and β, we know the mean and covariance.

Conditional Gaussian

$$\alpha = (\mathbf{y} - \mu_y)^T \Sigma_{yy}^{-1} (\mathbf{y} - \mu_y) - (\mathbf{z} - \mu_z)^T \Sigma_{zz}^{-1} (\mathbf{z} - \mu_z)$$

$$\begin{cases} \tilde{\mathbf{x}} = \mathbf{x} - \mu_x \\ \tilde{\mathbf{z}} = \mathbf{z} - \mu_z \end{cases}$$

$$= \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{z}} \end{bmatrix}^T \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{z}} \end{bmatrix} - \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \tilde{\mathbf{z}}$$

$$= \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{z}} \end{bmatrix}^T \begin{bmatrix} I_{xx} & I_{xz} \\ I_{zx} & I_{zz} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{z}} \end{bmatrix} - \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \tilde{\mathbf{z}}$$

$$= \tilde{\mathbf{x}}^T I_{xx} \tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T I_{xz} \tilde{\mathbf{z}} + \tilde{\mathbf{z}}^T I_{zx} \tilde{\mathbf{x}} + \tilde{\mathbf{z}}^T I_{zz} \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \tilde{\mathbf{z}}$$

Conditional Gaussian

$$\alpha = \tilde{\mathbf{x}}^T I_{xx} \tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T I_{xz} \tilde{\mathbf{z}} + \tilde{\mathbf{z}}^T I_{zx} \tilde{\mathbf{x}} + \tilde{\mathbf{z}}^T I_{zz} \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \tilde{\mathbf{z}}$$

$$= \tilde{\mathbf{x}}^T I_{xx} \tilde{\mathbf{x}} - \tilde{\mathbf{x}}^T I_{xx} \Sigma_{xz} \Sigma_{zz}^{-1} \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \Sigma_{zx} I_{xx} \tilde{\mathbf{x}}$$

$$+ \tilde{\mathbf{z}}^T \left(\Sigma_{zz}^{-1} + \Sigma_{zz}^{-1} \Sigma_{zx} I_{xx} \Sigma_{xz} \Sigma_{zz}^{-1} \right) \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \tilde{\mathbf{z}}$$

$$= \tilde{\mathbf{x}}^T I_{xx} \tilde{\mathbf{x}} - \tilde{\mathbf{x}}^T I_{xx} \Sigma_{xz} \Sigma_{zz}^{-1} \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \Sigma_{zx} I_{xx} \tilde{\mathbf{x}} + \tilde{\mathbf{z}}^T \Sigma_{zz}^{-1} \Sigma_{zx} I_{xx} \Sigma_{xz} \Sigma_{zz}^{-1} \tilde{\mathbf{z}}$$

$$= \left(\tilde{\mathbf{x}} - \Sigma_{xz} \Sigma_{zz}^{-1} \tilde{\mathbf{z}} \right)^T I_{xx} \left(\tilde{\mathbf{x}} - \Sigma_{xz} \Sigma_{zz}^{-1} \tilde{\mathbf{z}} \right)$$

See Appendix 2

$$\begin{cases} I_{xx} = \left(\Sigma_{xx} - \Sigma_{xz} \Sigma_{zz}^{-1} \Sigma_{zx} \right)^{-1} \\ I_{xz} = -I_{xx} \Sigma_{xz} \Sigma_{zz}^{-1} \\ I_{zx} = -\Sigma_{zz}^{-1} \Sigma_{zx} I_{xx} \\ I_{zz} = \Sigma_{zz}^{-1} + \Sigma_{zz}^{-1} \Sigma_{zx} I_{xx} \Sigma_{xz} \Sigma_{zz}^{-1} \end{cases}$$

Conditional Gaussian

$$\alpha = \left(\tilde{\mathbf{x}} - \Sigma_{xz}\Sigma_{zz}^{-1}\tilde{\mathbf{z}}\right)^T I_{xx}\left(\tilde{\mathbf{x}} - \Sigma_{xz}\Sigma_{zz}^{-1}\tilde{\mathbf{z}}\right)$$

$$I_{xx} = \left(\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right)^{-1}$$

$$= \left(\tilde{\mathbf{x}} - \Sigma_{xz}\Sigma_{zz}^{-1}\tilde{\mathbf{z}}\right)^T \left(\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right)^{-1}\left(\tilde{\mathbf{x}} - \Sigma_{xz}\Sigma_{zz}^{-1}\tilde{\mathbf{z}}\right)$$

$$\begin{cases} \tilde{\mathbf{x}} = \mathbf{x} - \mu_x \\ \tilde{\mathbf{z}} = \mathbf{z} - \mu_z \end{cases}$$

$$= \left[\mathbf{x} - \left(\mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z)\right)\right]^T \left(\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right)^{-1}\left[\mathbf{x} - \left(\mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z)\right)\right]$$

$$P(\mathbf{x}|\mathbf{z}) = \frac{P(\mathbf{x},\mathbf{z})}{P(\mathbf{z})} = \frac{1}{\sqrt{(2\pi)^n \left|\Sigma_{yy}\right| / \left|\Sigma_{zz}\right|}} e^{-\frac{1}{2}\left[(\mathbf{y}-\mu_y)^T \Sigma_{yy}^{-1}(\mathbf{y}-\mu_y) - (\mathbf{z}-\mu_z)^T \Sigma_{zz}^{-1}(\mathbf{z}-\mu_z)\right]}$$

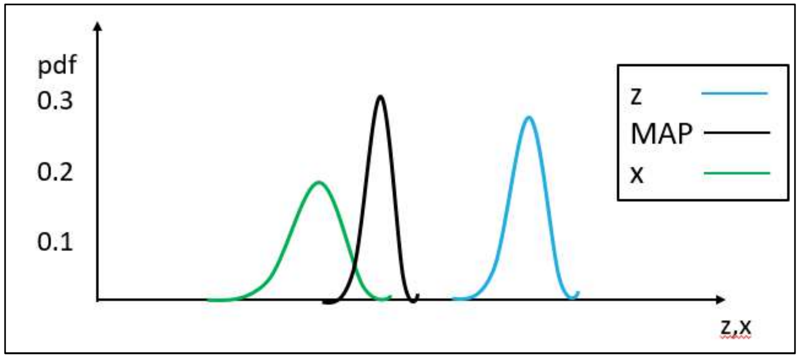
$$= \frac{1}{\sqrt{(2\pi)^n \left|\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right|}} e^{-\frac{1}{2}\left\{\mathbf{x} - \left[\mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z}-\mu_z)\right]\right\}^T \left(\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right)^{-1}\left\{\mathbf{x} - \left[\mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z}-\mu_z)\right]\right\}}$$

$$\frac{\left|\Sigma_{yy}\right|}{\left|\Sigma_{zz}\right|} = \left|\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right|$$

Conditional Gaussian

$$P(\mathbf{x}|\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^n \left|\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right|}} e^{-\frac{1}{2}\left\{\mathbf{x}-\left[\mu_x+\Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z}-\mu_z)\right]\right\}^T\left(\Sigma_{xx}-\Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}\right)^{-1}\left\{\mathbf{x}-\left[\mu_x+\Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z}-\mu_z)\right]\right\}}$$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$



- Now, we know how to fuse two data based on jointly Gaussian distribution.

Sensor Fusion

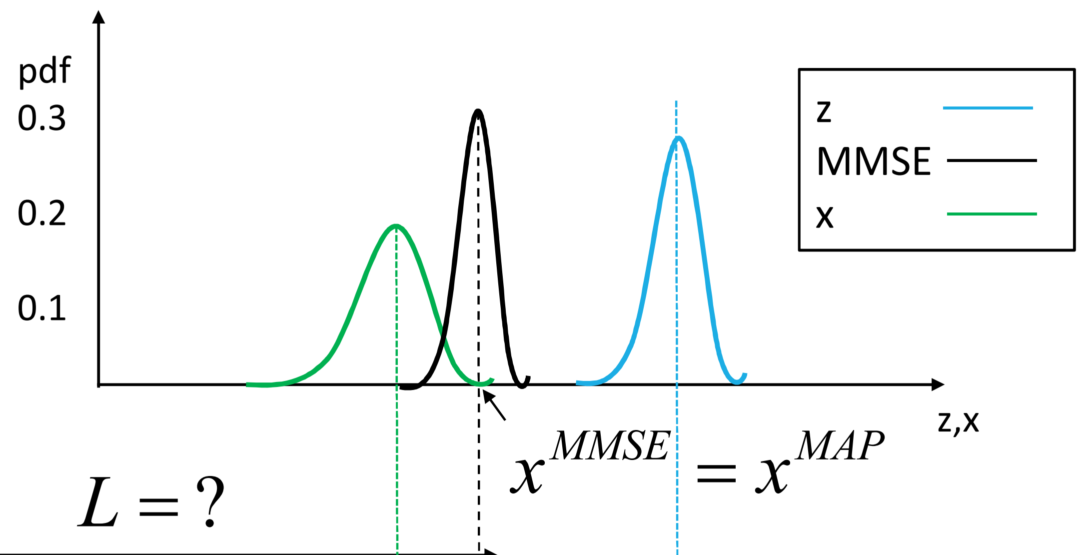
- Example:
- Location estimation

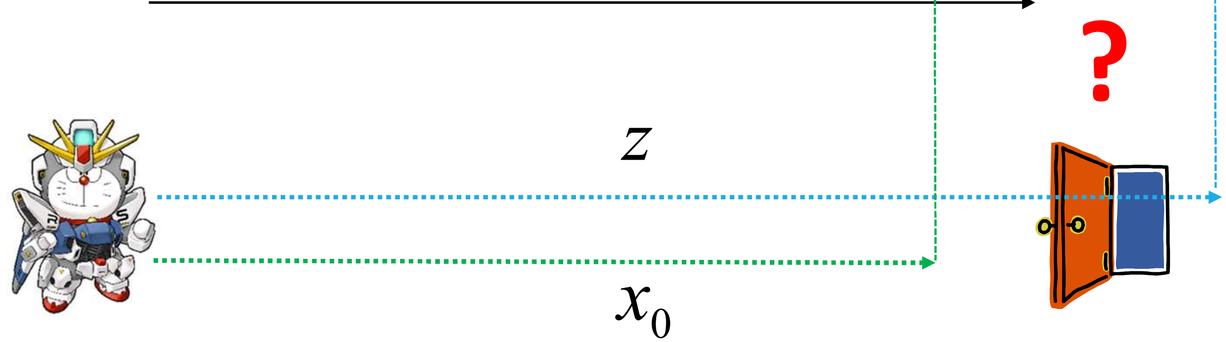
$x \sim N(10,2)$

$z \sim N(12,1), z = 12.5$

$\text{case}(i)\Sigma_{xz} = 0.1$

$\text{case}(ii)\Sigma_{xz} = 0.5$





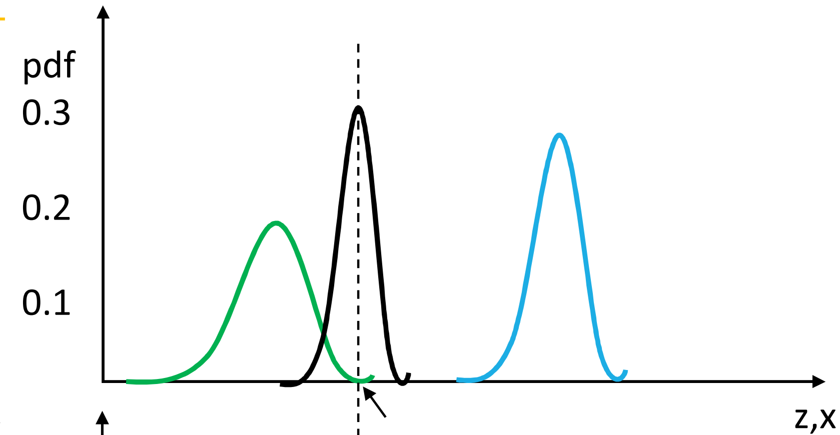
Sensor Fusion

$\text{case}(i)\, z = 12.5,\, x \sim N(10,2),\, z \sim N(12,1),\, \Sigma_{xz} = 0.1$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$

$$\mu_{x|z} = 10 + (0.1)1^{-1}(12.5 - 12) = 10.05$$

$$\Sigma_{xx|z} = 2 - (0.1)1^{-1}(0.1) = 1.99$$

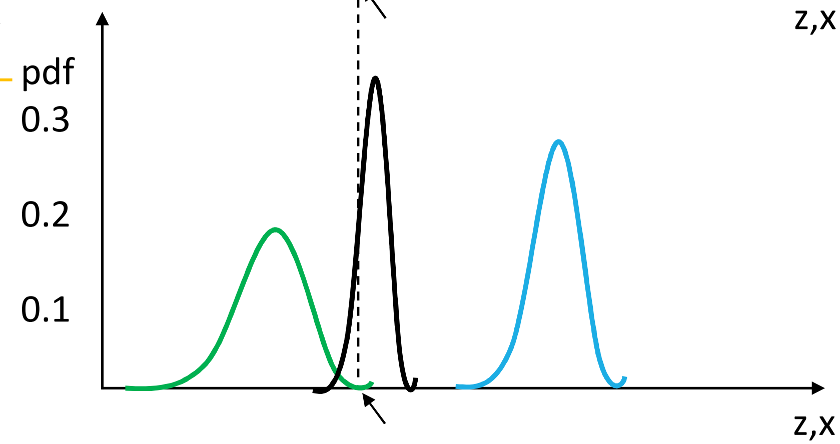


$\text{case}(ii)\, z = 12.5,\, x \sim N(10,2),\, z \sim N(12,1),\, \Sigma_{xz} = 0.5$

$$\begin{cases} \mu_{x|z} = \mu_x + \Sigma_{xz}\Sigma_{zz}^{-1}(\mathbf{z} - \mu_z) \\ \Sigma_{xx|z} = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx} \end{cases}$$

$$\mu_{x|z} = 10 + (0.5)1^{-1}(12.5 - 12) = 10.25$$

$$\Sigma_{xx|z} = 2 - (0.5)1^{-1}(0.5) = 1.75$$



Appendix 1:

Determinant of conditional covariance of jointly Gaussian variables

$$\frac{|\Sigma_{yy}|}{|\Sigma_{zz}|} = |\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}|$$

$$\Sigma_{yy} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ \Sigma_{zx} & \Sigma_{zz} \end{bmatrix} = \begin{bmatrix} C & \Sigma_{xz} \\ 0 & \Sigma_{zz} \end{bmatrix} \begin{bmatrix} I_{n\times n} & 0 \\ D & I_{m\times m} \end{bmatrix}$$

$$\Sigma_{xx} = C + \Sigma_{xz} D$$

$$\Sigma_{zx} = \Sigma_{zz} D$$

$$D = \Sigma_{zz}^{-1}\Sigma_{zx}$$

$$C = \Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}$$

$$\because |AB| = |A||B|$$

$$|\Sigma_{yy}| = (|C||\Sigma_{zz}|)(|I_{n\times n}||I_{m\times m}|) = |\Sigma_{xx} - \Sigma_{xz}\Sigma_{zz}^{-1}\Sigma_{zx}||\Sigma_{zz}|$$

Appendix 2:
Inversion of a Partitioned Matrix

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

$$Where$$

$$E = A^{-1} + A^{-1}BHCA^{-1} = \left(A - BD^{-1}C\right)^{-1}$$

$$F = -A^{-1}BH = -EBD^{-1}$$

$$G = -HCA^{-1} = -D^{-1}CE$$

$$H = \left(D - CA^{-1}B\right)^{-1} = D^{-1} + D^{-1}CEBD^{-1}$$