

Department of Teleinformatics Engineering , Federal University of Ceará

---

# Introduction to Gaussian Processes

Filipe P. de Farias, IC  
filipepfarias@fisica.ufc.br

September 14, 2019

- 1 Linear Regression
  - 1.1 Defining models
  - 1.2 Optimizing the parameters
  
- 2 Bayesian Linear Regression
  - 2.1 The D-dimensional Gaussian Distribution
  - 2.2 Bayes' rule for Gaussian variables
  
- 3 Gaussian Processes
  - 3.1 Recap
  - 3.2 Change of Space
  - 3.3 Gaussian processes

# Linear Regression

- If we have a set of points in a space that comes from observations of an experiment and we want to predict other points, this could be done with **curve fitting**.
- So we could define some strategy to find our model.

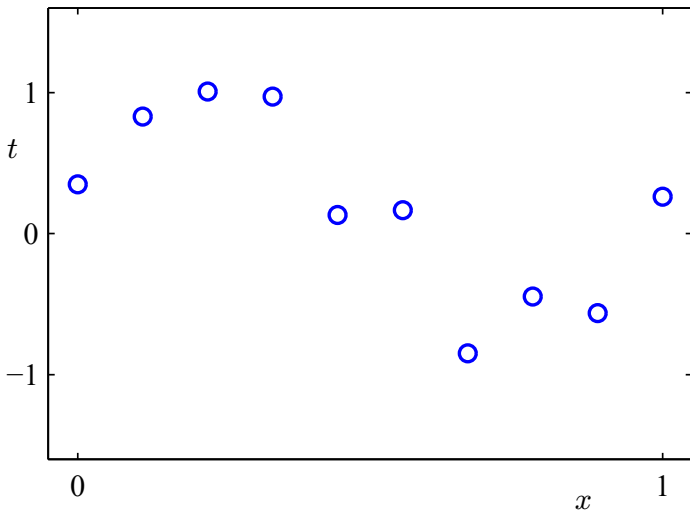
## Strategy

- 1 Purpose a **model**, e.g. functions like exponential, polynomial and others.
- 2 Train our model with the training data set, finding the **unknown parameters** or **weights**.

# Defining models

## An initial curve fitting problem

- Let's take the points below generated from the function  $y(x) = \sin(2\pi x)$ .



- We can express the curve with a polynomial, being the **model**

$$y(x, \mathbf{w}) = w_0x^0 + w_1x^1 + w_2x^2 + \dots + w_{M-1}x^{M-1} = \sum_{j=1}^{M-1} w_jx^j$$

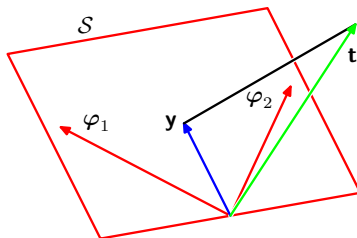
- In general, we could write this **weighted sum** with any other function. In other words, we can put this in terms of  $\phi_n(x) = x^n$ , where  $\phi$  could be other **basis function**.
- e.g. we could have different  $y(x)$  for different basis functions, or **features**.

$$\begin{aligned} y(x, \mathbf{w}) &= w_0\phi_0(x) + w_1\phi_1(x) + w_2\phi_2(x) + \dots + w_{M-1}\phi_{M-1}(x) \\ &= w_0 \exp \left\{ -\frac{(x - \mu_0)^2}{2\sigma^2} \right\} + w_1 \exp \left\{ -\frac{(x - \mu_1)^2}{2\sigma^2} \right\} + \\ &\dots + w_{M-1} \exp \left\{ -\frac{(x - \mu_{M-1})^2}{2\sigma^2} \right\} \\ &= w_0 \sin(0 \cdot x) + w_1 \cos(1 \cdot x) + \\ &\dots + w_{M_2} \sin((M - 2) \cdot x) + w_{M-1} \cos((M - 1) \cdot x) \end{aligned}$$

- For simplicity, we'll carry this notation along.

$$\begin{aligned} y(x, \mathbf{w}) &= w_0\phi_0(x) + w_1\phi_1(x) + \dots \\ &\quad + w_{M-1}\phi_{M-1}(x) \\ &= \sum_{j=1}^{M-1} w_j\phi_j(x) \end{aligned}$$

- We'll evaluate  $\phi$  for all  $x$ , and then project it in the  $w$  vector space, the **feature-space**, then our model could be formed by **non-linear** functions. But, remaining **linear on parameters**.

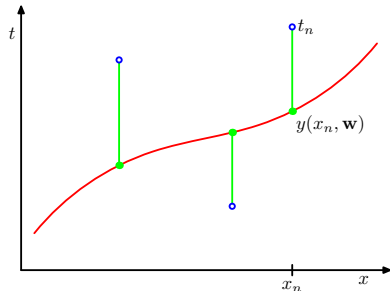


# Optimizing the parameters

## The model parameters

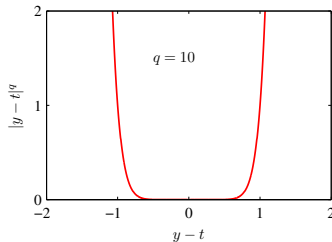
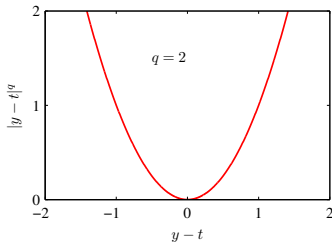
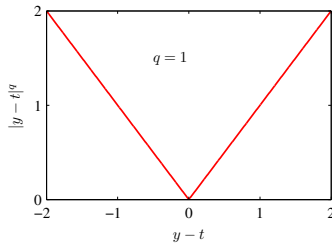
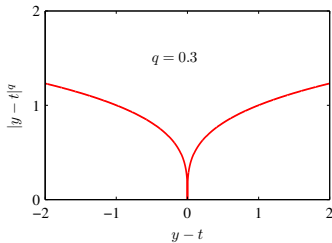
- The chosen model will give us some curve that is needed to adjust such that we'll **minimize its distance** to the **targets**  $t$ .
- This approach lead us to use the **least squares** to estimate the weights and minimize the **error**  $E$ .

$$E(\mathbf{w}) \triangleq \frac{1}{2} \sum_{n=1}^N \{y_n - t_n\}^2$$





### Why choose a quadratic norm distance?



### Why choose a quadratic norm distance?<sup>1</sup>

- The first row figures could be used for the derivations, taking care with some **non-continuous derivatives**.
- We'll use the **quadratic norm** because its the minor integer  $q$  differentiable, and then the error measures  $E$  between the model  $y(x, \mathbf{w})$  and the targets  $t$  will be euclidean.
- More, increasing the value of  $q$ , the smallests than 1 and bigger than 0 errors between the model and the targets that become irrelevant for  $E$ .

---

<sup>1</sup>See Appendix ?

- Remembering that

$$y(x, \mathbf{w}) = w_0\phi_0(x) + w_1\phi_1(x) + w_2\phi_2(x) + \dots + w_{M-1}\phi_{M-1}(x)$$

- We'll evaluate for all  $x_i$  values, and then put  $y_n(x_i, \mathbf{w})$  in the matrix form and get

$$y_n = [\phi_0(x_n) \quad \phi_1(x_n) \quad \dots \quad \phi_{M-1}(x_n)] [w_0 \quad w_1 \quad \dots \quad w_{M-1}]^\top$$

- And then

$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \dots & \phi_{M-1}(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \dots & \phi_{M-1}(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_{N-1}) & \phi_1(x_{N-1}) & \dots & \phi_{M-1}(x_{N-1}) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{N-1} \end{bmatrix}}_{\mathbf{w}}$$

where  $\Phi$  is the **design matrix**.

- This represents the system  $\mathbf{y} = \Phi\mathbf{w}$ .

- If  $E(\mathbf{w}) = \frac{1}{2} (\mathbf{y} - \mathbf{t})^\top (\mathbf{y} - \mathbf{t})$  where  $\mathbf{t} = [t_1 \quad t_2 \quad \dots \quad t_n]^\top$
- Then we'll have

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2} (\mathbf{y}^\top \mathbf{y} - \mathbf{t}^\top \mathbf{y} - \mathbf{y}^\top \mathbf{t} + \mathbf{t}^\top \mathbf{t}) \\ &= \frac{1}{2} ((\Phi \mathbf{w})^\top (\Phi \mathbf{w}) - \mathbf{t}^\top (\Phi \mathbf{w}) - (\Phi \mathbf{w})^\top \mathbf{t} + \mathbf{t}^\top \mathbf{t}) \\ &= \frac{1}{2} (\mathbf{w}^\top \Phi^\top \Phi \mathbf{w} - 2\mathbf{t}^\top \Phi \mathbf{w} + \mathbf{t}^\top \mathbf{t}) \end{aligned}$$

- In sequence, we'll try to minimize it in terms of the weights ( $\mathbf{w}$ ) by

$$\begin{aligned} 0 &= \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{2} (2\mathbf{w}^\top \Phi^\top \Phi - 2\mathbf{t}^\top \Phi + 0) \\ \mathbf{w}^\top &= \mathbf{t}^\top \Phi (\Phi^\top \Phi)^{-1} \\ \mathbf{w}^* &= (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{t} \end{aligned}$$

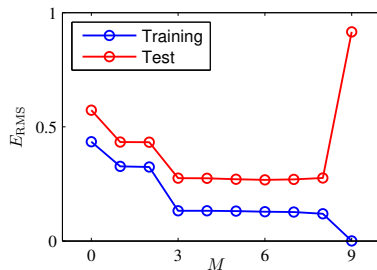
- Here, we've obtained the weights  $\mathbf{w}^*$  with the **best fit** of the curve.
- We could say that the model **learned** the parameters.

### Why the prediction is so distant from the deterministic curve?

- A visible effect of the **increase of the complexity** of the model, is the increase of the **number of features**  $M$ .
- It's easy to see that our model start's to differ from the  $y$  and starts to interpolate the noise. We call this of **over-fitting**.
- This phenomenon illustrate a method of always search for the **best estimation of the parameters**.

### Could be over-fitting a problem?

- We could **train** our model, it means evaluate  $\mathbf{w}^*$ , for only a part of our dataset.
- If the model be a good one, the error must be small when its **testing**, i.e. the error must be small when we evaluate all dataset with the  $\mathbf{w}^*$  of the trained part.
- But this in general does not occur and the **error increases**.

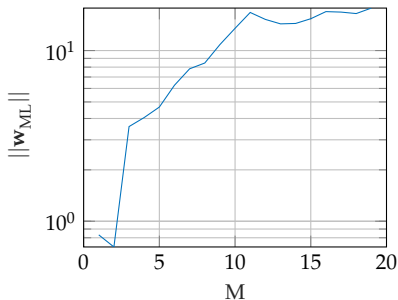


### How to control the over-fitting?

- With the increase of the model complexity, the value of  $\mathbf{w}^*$  increases too.
- A solution could be add a **penalty term** as the norm of the weights increases.
- To control the over-fitting, we try to **regularize** the weights by adding a penalty term  $\lambda$  to error function, by this we force the coefficients to not reach high values.

$$\tilde{E}(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{t})^\top (\mathbf{y} - \mathbf{t}) + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w}$$

$$\Rightarrow \mathbf{w}_{\text{reg}}^* = (\Phi^\top \Phi + \lambda \mathbf{I})^{-1} \Phi^\top \mathbf{t}$$



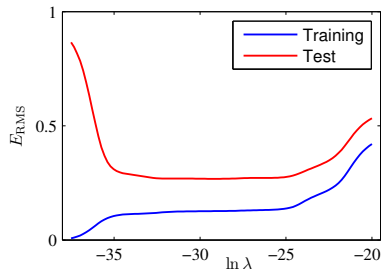
- To control the over-fitting, we try to **regularize** the weights by adding a penalty term  $\lambda$  to error function, by this we force the coefficients to not reach high values.

$$\begin{aligned}
 \tilde{E}(\mathbf{w}) &= \frac{1}{2}(\mathbf{y} - \mathbf{t})^\top (\mathbf{y} - \mathbf{t}) + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \\
 &= \frac{1}{2} \left( \mathbf{w}^\top \Phi^\top \Phi \mathbf{w} - 2\mathbf{t}^\top \Phi \mathbf{w} + \mathbf{t}^\top \mathbf{t} + \lambda \mathbf{w}^\top \mathbf{I} \mathbf{w} \right) \\
 \Rightarrow \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{2} \left( 2\mathbf{w}^\top \Phi^\top \Phi - 2\mathbf{t}^\top \Phi + 0 + 2\lambda \mathbf{w}^\top \mathbf{I} \right) \\
 0 &= \mathbf{w}^\top \Phi^\top \Phi - \mathbf{t}^\top \Phi + \lambda \mathbf{w}^\top \mathbf{I} \\
 \mathbf{w}_{\text{reg}}^* &= \left( \Phi^\top \Phi + \lambda \mathbf{I} \right)^{-1} \Phi^\top \mathbf{t}
 \end{aligned}$$

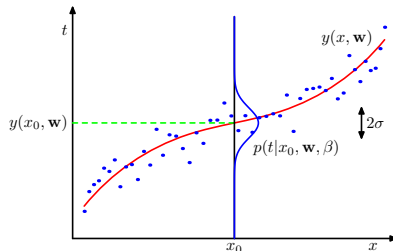


### A more sophisticated approach?

- We could **train and test** and evaluate the error for several values of  $\lambda$ .
- Even yet, it's too wasteful partitionate and optimize data to find good model parameters or even a **flexible** one.
- We need a more sophisticated approach.



- Having an **uncertainty** in the measured value, we could represent it with a **probability distribution**.
- Now, each **target** could be expressed as a **random variable**.
- Its **mean** is given by  $y(x, \mathbf{w})$ , and the **variance** by  $1/\sigma^2 = \beta$ .
- $\beta$  is known as **precision parameter** too.



- Being the random variables independent and identically distributed, we can say that our **joint probability** is given by

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^N p(t_n|x_n, \mathbf{w}, \beta)$$

- Our goal is, given the **parameters**  $\mathbf{w}$ , maximize the **probability** of the **targets**.
- Before, consider a property of the probability distributions

$$\int_{-\infty}^{\infty} p(x)dx = 1 \text{ and } p(x) \geq 0$$

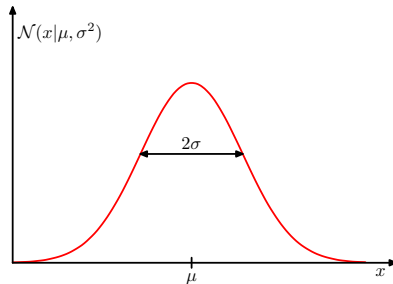
- Then, to avoid computational singularity and obtain a monotonically increasing function, we apply

$$\ln(p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)) = \sum_{n=1}^N \ln(p(t_n|x_n, \mathbf{w}, \beta))$$

- We'll choose the **Gaussian distribution** just for convenience.
- Then we can state the distribution for each target and then

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}\right)$$

- And we'll try to **maximize** the targets probability.



## One-dimensional Gaussian distribution

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x - \mu)^2\right\} > 0$$

where  $\mu$  is the mean and  $\sigma^2$  the variance.

- From the **joint probability** of the Gaussians distributions we have

$$\begin{aligned}\ln(p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)) &= \mathcal{N}\left(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}\right) \\ &= \sum_{n=1}^N -\frac{1}{2} \ln(2\pi) + \sum_{n=1}^N \frac{1}{2} \ln \beta - \sum_{n=1}^N \frac{\beta}{2} (x_n - y(x_n, \mathbf{w}))^2\end{aligned}$$

- If we make

$$\frac{\partial}{\partial \mathbf{w}} \ln(p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)) = 0$$

we'll obtain the **cost function** obtained before in the linear regression.

- In other words, we'll obtain the weights  $\mathbf{w}_{ML}$  that **maximize** the log probability of the targets, as  $\beta_{ML}$ .
- With this approach, we expect to obtain too the same **over-fitting** problem.<sup>2</sup>
- Next we'll take a step towards to a **Bayesian** approach.

---

<sup>2</sup>See Appendix ?

# Bayesian Linear Regression

At this point, we have a probabilistic model and we may want to predict values for  $x$ . Then, we need a *predictive distribution*.

Let's say we have the probabilities of some idea we desire to update it in the light of some new evidence. This could be done with **Bayes' Rule**, to convert a *prior* probability in a *posterior* probability and put some uncertainty in the parameters too.

Mathematically, by Bayes' Rule, we could infer



Mathematically, by Bayes' Rule, we could infer

$$\underbrace{p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta)}_{\text{posterior}} \propto \underbrace{p(\mathbf{t}|\mathbf{w}, \mathbf{x}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w}|\alpha)}_{\text{prior}}$$

Mathematically, by Bayes' Rule, we could infer

$$\underbrace{p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta)}_{\text{posterior}} \propto \underbrace{p(\mathbf{t}|\mathbf{w}, \mathbf{x}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w}|\alpha)}_{\text{prior}}$$

and for simplicity, consider the follow prior for  $\mathbf{w}$

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^\top \mathbf{w}\right\}$$

where  $\alpha$  the precision of the distribution and  $M + 1$  is the dimension of  $\mathbf{w}$ , for a polynomial of  $M^{th}$  order. Variables such  $\alpha$  are called *hyperparameters* and control the distribution of model parameters.

By this, we can find a distribution and its maximum, or most probable value of  $\mathbf{w}$  given the data taking the minimum of the negative logarithm of the inferred expression, that will lead us to a term

$$\sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2} \mathbf{w}^\top \mathbf{w} + \text{const.}$$

Note that if we consider  $\lambda = \alpha/\beta$ , this will back to the regularized form of *least squares*. This technique is called *maximum posterior* (MAP).

So, observe that even making some probabilistic assumptions, we don't have yet a fully bayesian model, given that finding the *maximum likelihood*, we're finding only the parameters given one model such that maximize our targets probabilities. Furthermore, even with some probabilistic assumptions, our model still have a **over-fitting** problem, given that we obtained the same expressions for the simple regression, adding some constants.

The next step is put some **uncertainty in predictive model**, and makes adjustments in the light of our new evidences. By that we could obtain a "more Bayesian" model, in other words, a **Bayesian Linear Regression**.

Seeking a Bayesian approach, the next steps consists to apply the **sum** and **product** rules of probability to evaluate the predictive distribution. By now we assume that the hyperparameters are fixed, but they could assume a distribution too.

We saw that the posterior distribution for  $\mathbf{w}$  could be given by

$$\underbrace{p(\mathbf{w}|\mathbf{x}, \mathbf{t})}_{\text{posterior}} \propto \underbrace{p(\mathbf{t}|\mathbf{w}, \mathbf{x})}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}}$$

Remember the One-dimensional Gaussian distribution

## One-dimensional Gaussian distribution

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\} > 0$$

where  $\mu$  is the mean and  $\sigma^2$  the variance.

Remember the One-dimensional Gaussian distribution

## One-dimensional Gaussian distribution

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\} > 0$$

where  $\mu$  is the mean and  $\sigma^2$  the variance.

First we'll consider a geometrical approach by the quadratic distance  $(x - \mu)^2$  normalized by the variance  $\sigma^2$ . This comprehension will help us with the D-dimensional case.

To more than one dimensions, we'll consider the points ( $\mathbf{x}$ ) distance for the mean of the distribution, as we done in the one dimensional case, by adding a term to prioritize some dimension distribution in particular. Then

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

called *Mahalanobis distance*. And it's becomes the *Euclidean distance*, when  $\boldsymbol{\Sigma}$  is the identity matrix. This means that the all the distances are equally normalized. The matrix  $\boldsymbol{\Sigma}$  is the covariance matrix of the distributions, by definition.



And then

## D-dimensional Gaussian distribution

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right\}$$

where  $\boldsymbol{\mu}$  is the D-dimensional mean vector,  $\boldsymbol{\Sigma}$  the  $D \times D$ -dimensional variance matrix and  $|\boldsymbol{\Sigma}|$  its determinant.

## Partitioned Gaussians

Given a joint Gaussian distribution  $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with  $\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1}$  and

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}, \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}.$$

Will give us

- Conditional distribution:

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Lambda}_{aa}^{-1}), \quad \boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)$$

- Marginal distribution:

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$$

To proceed we'd like to prove that the Gaussians are **closed under linear transformations**. This will allow us to transform the Gaussians under the likelihood distribution given a prior. For example, given a distribution

$$p(\mathbf{z}) = p(\mathbf{x}, \mathbf{y})$$

In other words, we're trying to find the marginal distribution  $p(\mathbf{y})$  and the conditional distribution  $p(\mathbf{x}|\mathbf{y})$ , given

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{Ax} + \mathbf{b}, \mathbf{L}^{-1})$$

In other words, we're trying to find the parginal distribution  $p(\mathbf{y})$  and the conditional distribution  $p(\mathbf{x}|\mathbf{y})$ , given

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{Ax} + \mathbf{b}, \mathbf{L}^{-1})$$

So, applying the joint distribution and the its ln after

$$p(\mathbf{z}) = p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x}) p(\mathbf{x})$$

$$\ln p(\mathbf{z}) = \ln p(\mathbf{y}|\mathbf{x}) + \ln p(\mathbf{x})$$

$$= -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Lambda} (\mathbf{x} - \boldsymbol{\mu})$$

$$- \frac{1}{2} (\mathbf{y} - \mathbf{Ax} - \mathbf{b})^\top \mathbf{L} (\mathbf{y} - \mathbf{Ax} - \mathbf{b}) + \text{const}$$

The "const" is the term independent of  $\mathbf{x}$  and  $\mathbf{y}$ . Then, expanding the quadratic form

$$\begin{aligned}\ln p(\mathbf{z}) &= -\frac{1}{2}\mathbf{x}^\top \left( \mathbf{\Lambda} + \mathbf{A}^\top \mathbf{L} \mathbf{A} \right) \mathbf{x} - \frac{1}{2}\mathbf{y}^\top \mathbf{L} \mathbf{y} + \frac{1}{2}\mathbf{y}^\top \mathbf{L} \mathbf{A} \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{A}^\top \mathbf{L} \mathbf{y} \\ &= -\frac{1}{2} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^\top \begin{pmatrix} \mathbf{\Lambda} + \mathbf{A}^\top \mathbf{L} \mathbf{A} & -\mathbf{A}^\top \mathbf{L} \\ -\mathbf{L} \mathbf{A} & \mathbf{L} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = -\frac{1}{2} \mathbf{z}^\top \mathbf{R} \mathbf{z}\end{aligned}$$

The "const" is the term independent of  $\mathbf{x}$  and  $\mathbf{y}$ . Then, expanding the quadratic form

$$\begin{aligned}\ln p(\mathbf{z}) &= -\frac{1}{2}\mathbf{x}^\top \left( \mathbf{\Lambda} + \mathbf{A}^\top \mathbf{L} \mathbf{A} \right) \mathbf{x} - \frac{1}{2}\mathbf{y}^\top \mathbf{L} \mathbf{y} + \frac{1}{2}\mathbf{y}^\top \mathbf{L} \mathbf{A} \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{A}^\top \mathbf{L} \mathbf{y} \\ &= -\frac{1}{2} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^\top \begin{pmatrix} \mathbf{\Lambda} + \mathbf{A}^\top \mathbf{L} \mathbf{A} & -\mathbf{A}^\top \mathbf{L} \\ -\mathbf{L} \mathbf{A} & \mathbf{L} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = -\frac{1}{2} \mathbf{z}^\top \mathbf{R} \mathbf{z}\end{aligned}$$

We'll apply the partitioned matrices inversion to obtain  $\mathbf{R}^{-1}$

$$\mathbf{R}^{-1} = \begin{pmatrix} \mathbf{\Lambda}^{-1} & \mathbf{\Lambda}^{-1} \mathbf{A}^\top \\ \mathbf{A} \mathbf{\Lambda}^{-1} & \mathbf{L}^{-1} + \mathbf{A} \mathbf{\Lambda}^{-1} \mathbf{A}^\top \end{pmatrix}$$

The expanded form of  $\ln p(\mathbf{z})$  give us the mean too by the linear terms, then

$$\mathbf{x}^\top \Lambda \boldsymbol{\mu} - \mathbf{x}^\top \mathbf{A}^\top \mathbf{L} \mathbf{b} + \mathbf{y}^\top \mathbf{L} \mathbf{b} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^\top \begin{pmatrix} \Lambda \boldsymbol{\mu} - \mathbf{A}^\top \mathbf{L} \mathbf{b} \\ \mathbf{L} \mathbf{b} \end{pmatrix}$$



The expanded form of  $\ln p(\mathbf{z})$  give us the mean too by the linear terms, then

$$\mathbf{x}^\top \mathbf{\Lambda} \boldsymbol{\mu} - \mathbf{x}^\top \mathbf{A}^\top \mathbf{L} \mathbf{b} + \mathbf{y}^\top \mathbf{L} \mathbf{b} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^\top \begin{pmatrix} \mathbf{\Lambda} \boldsymbol{\mu} - \mathbf{A}^\top \mathbf{L} \mathbf{b} \\ \mathbf{L} \mathbf{b} \end{pmatrix}$$

By inspection of the linear terms

$$\mathbb{E}[\mathbf{z}] = \mathbf{R}^{-1} \begin{pmatrix} \mathbf{\Lambda} \boldsymbol{\mu} - \mathbf{A}^\top \mathbf{L} \mathbf{b} \\ \mathbf{L} \mathbf{b} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{A} \boldsymbol{\mu} + \mathbf{b} \end{pmatrix}$$

And then we we'll have that

$$\begin{aligned}\mathbb{E}[\mathbf{y}] &= \mathbf{A}\boldsymbol{\mu} + \mathbf{b} \\ \text{cov}[\mathbf{y}] &= \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top\end{aligned}$$

And then we we'll have that

$$\begin{aligned}\mathbb{E}[\mathbf{y}] &= \mathbf{A}\boldsymbol{\mu} + \mathbf{b} \\ \text{cov}[\mathbf{y}] &= \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top\end{aligned}$$

$$\begin{aligned}\mathbb{E}[\mathbf{x}|\mathbf{y}] &= \left(\boldsymbol{\Lambda} + \mathbf{A}^\top\mathbf{L}\mathbf{A}\right)^{-1} \left\{ \mathbf{A}^\top\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu} \right\} \\ \text{cov}[\mathbf{x}|\mathbf{y}] &= \left(\boldsymbol{\Lambda} + \mathbf{A}^\top\mathbf{L}\mathbf{A}\right)^{-1}\end{aligned}$$

In the next step, we'll assume a **prior distribution over parameters**,  $p(\mathbf{w})$ , and define it as a Gaussian distribution, then

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_0, \mathbf{S}_0)$$

with mean  $\mathbf{m}_0$  and variance  $\mathbf{S}_0$ .

## Marginal and Conditioned Gaussians

- For  $\mathbf{y}$  given  $\mathbf{x}$ :

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$$

$$p(\mathbf{y} | \mathbf{x}) = \mathcal{N}(\mathbf{y} | \mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1})$$

- For  $\mathbf{x}$  given  $\mathbf{y}$ :

$$p(\mathbf{x} | \mathbf{y}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\top}), \text{ where } \boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^{\top}\mathbf{L}\mathbf{A})^{-1}$$

By the derivations, we make the assumptions of given  $p(\mathbf{w})$  and for  $p(\mathbf{t}|\mathbf{w})$  such that

$$\begin{aligned} p(\mathbf{t}|\mathbf{w}) &= \mathcal{N}(\mathbf{t}|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \\ &= \mathcal{N}(\mathbf{t}|\Phi^T \mathbf{w}, \beta^{-1}) \end{aligned}$$

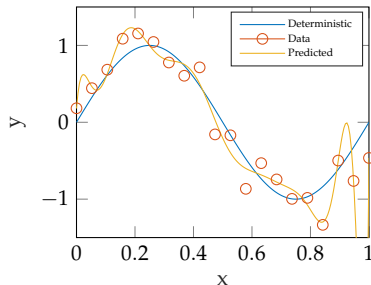
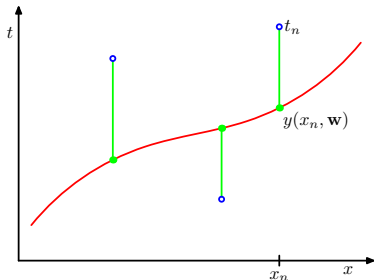
And then  $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$  where

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N (\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \Phi^T \mathbf{t}) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta \Phi^T \Phi \end{aligned}$$

# Gaussian Processes

### What was done until here?

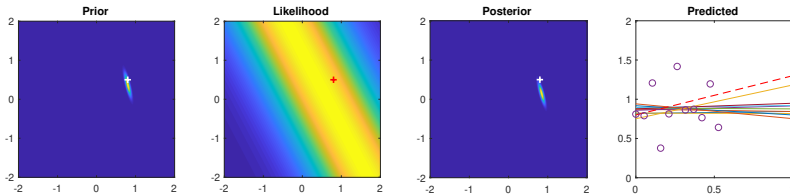
- We assumed that our targets  $t$  were **i.i.d.** and given by  $t = y(\mathbf{x}) + \varepsilon$ , where  $\varepsilon \sim \mathcal{N}(0, \beta)$ .
- Our model is given by  $y(\mathbf{x}) = \Phi^\top \mathbf{w}$ , where  $\Phi$  is the **design matrix**, and this characterizes our model as **linear in parameters**.
- The **design matrix** was defined as  $\phi_{i,j} = \phi_i(\mathbf{x}_j)$ .
- The **parameters** were given by  $\mathbf{w} = (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{t}$ .
- These **parameters** calculated at the minimum of the cost function are called **maximum likelihood**.





### What was done until here?

- We put an **uncertainty** over the targets  $t$  and the parameters  $\mathbf{w}$ .
- We assumed that targets being **distributed** as  $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$ .
- By **Bayes' Rule** we obtained that  $p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{w}, \mathbf{x}, \beta) p(\mathbf{w}|\alpha)$
- This allowed to make an **inference** to obtain a **prediction** of the parameters in the **weight-space**.



# Recap

A more clear way to see what is happening...

## From Bayesian inference

- We have

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

- We'll change from **weight-space**

$$p(\mathbf{t}_*|\mathbf{x}_*, \mathbf{x}, \mathbf{t}) = \int p(\mathbf{t}_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\mathbf{x}, \mathbf{t})d\mathbf{w}$$

- To **feature-space**

$$\begin{aligned} p(f_*|\mathbf{x}_*, \Phi, \mathbf{t}) &= \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|\Phi, \mathbf{t})d\mathbf{w} = \int \mathbf{x}_*^\top \mathbf{w} p(\mathbf{w}|\Phi, \mathbf{t})d\mathbf{w} \\ &= \mathcal{N}\left(\beta \phi(\mathbf{x}_*)^\top \mathbf{S}_N \Phi \mathbf{t}, \phi(\mathbf{x}_*)^\top \mathbf{S}_N^{-1} \phi(\mathbf{x}_*)\right) \end{aligned}$$

where  $f_* \triangleq f(\mathbf{x}_*)$  at  $\mathbf{x}_*$  and  $\Phi = \Phi(\mathbf{x})$

## Alternative formulation

$$f_* | \mathbf{x}_*, \Phi, \mathbf{t} \sim \mathcal{N} \left( \phi_*^\top \mathbf{S}_0 \Phi \left( K + \beta^{-2} I \right)^{-1} \mathbf{t}, \phi_*^\top \mathbf{S}_0 \phi_* - \phi_*^\top \mathbf{S}_0 \Phi \left( K + \beta^{-2} I \right)^{-1} \Phi^\top \mathbf{S}_0 \phi_* \right)$$

where  $K = \Phi^\top \mathbf{S}_0 \Phi$

### What is kernel?

$$f_* | \mathbf{x}_*, \Phi, \mathbf{t} \sim \mathcal{N} \left( \phi_*^\top \mathbf{S}_0 \Phi \left( K + \beta^{-2} I \right)^{-1} \mathbf{t}, \phi_*^\top \mathbf{S}_0 \phi_* - \phi_*^\top \mathbf{S}_0 \Phi \left( K + \beta^{-2} I \right)^{-1} \Phi^\top \mathbf{S}_0 \phi_* \right)$$

- We could observe the appearance of terms like  $\Phi^\top \mathbf{S}_0 \Phi$ ,  $\phi_*^\top \mathbf{S}_0 \Phi$ , or  $\phi_*^\top \mathbf{S}_0 \phi_*$ .
- The common term between these operations is  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \mathbf{S}_0 \phi(\mathbf{x}')$
- Then we define  $k(\cdot, \cdot)$  as **kernel function**
- This technique is particularly valuable in situations where it is more convenient to compute the kernel than the design matrix vectors themselves.

- Previously we make the inference in the **feature-space** and then we find the function distribution.
- Now we'll make the inference directly on **function-space**.
- Let's define

## Definition

*A **Gaussian process** is a collection of random variables which any finite number of them have a joint Gaussian distribution.*

## Mean and covariance function

- As the Gaussian distribution, the  $\mathcal{GP}$  is characterized by its **mean function**  $m(\mathbf{x})$  and its **covariance function**  $k(\mathbf{x}, \mathbf{x}')$  of a real process  $f(\mathbf{x})$ .
- For a Gaussian processes

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

- We have

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

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





- C.M. Bishop (2016). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York. ISBN: 9781493938438. URL: <https://books.google.com.br/books?id=kOXDtAEACAAJ>.
- P.J. Dhrymes (2013). *Mathematics for Econometrics*. SpringerLink : Bücher. Springer New York. ISBN: 9781461481454. URL: <https://books.google.com.br/books?id=HIK8BAAAQBAJ>.
- J. L. Doob (Sept. 1944). "The Elementary Gaussian Processes". In: *Ann. Math. Statist.* 15.3, pp. 229–282. DOI: 10.1214/aoms/1177731234. URL: <https://doi.org/10.1214/aoms/1177731234>.
- F.A. Graybill (2001). *Matrices with Applications in Statistics*. Duxbury Classic Series. Brooks/Cole. ISBN: 9780534401313. URL: <https://books.google.com.br/books?id=BV3CAAAACAAJ>.
- Philipp Hennig (Sept. 2013). *Animating Samples from Gaussian Distributions*. Technical Report 8. Spemannstraße, 72076 Tübingen, Germany: Max Planck Institute for Intelligent Systems.
- T.B. Schön (2011). *Manipulating the Multivariate Gaussian Density*. URL: <http://user.it.uu.se/~thosc112/pubpdf/schon12011.pdf>.