Department of Teleinformatics Engineering , Federal University of Ceará

Introduction to Gaussian Processes

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

September 14, 2019

Outline

Linear Regression

Defining models

An initial curve fitting problem

- 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.

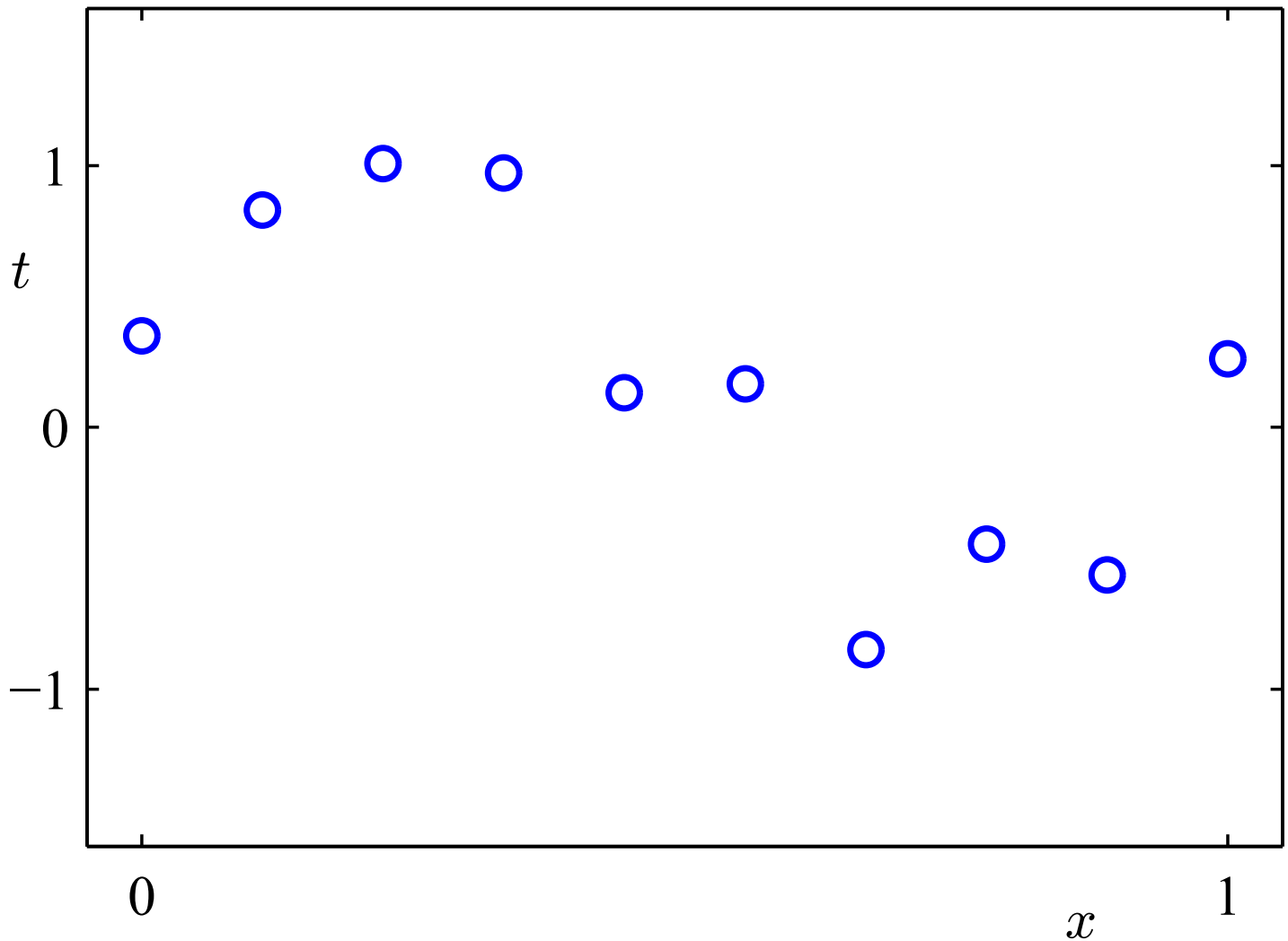
Strategy

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 $y(x) = \sin(2\pi x)$ with addition of Gaussian noise with zero mean and 0.2 of standard deviation.



Defining models
Chosing a model

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

$$y(x, \mathbf{w}) = w_0 x^0 + w_1 x^1 + w_2 x^2 + \ldots + w_{M-1} x^{M-1} = \sum_{j=1}^{M-1} w_j x^j$$

- 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 ϕ could be other **basis function**.
- e.g. we could have different $y(x)$ for different basis functions, or **features**.

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

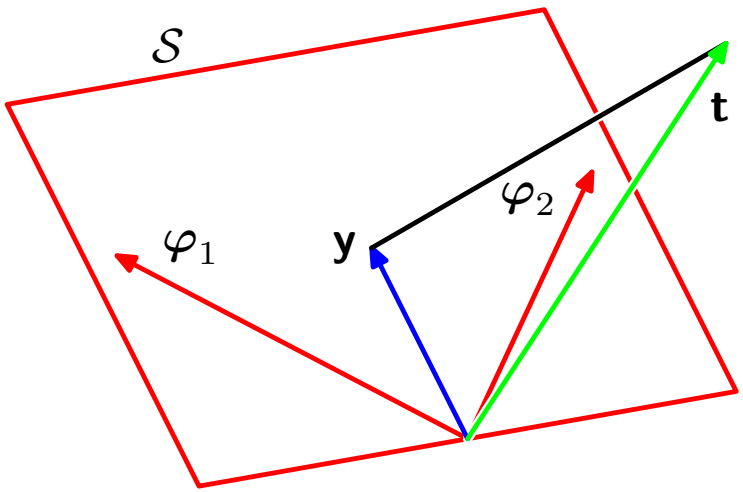
Defining models

A non-linear model linear in parameters

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

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

- We'll evaluate ϕ 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**.

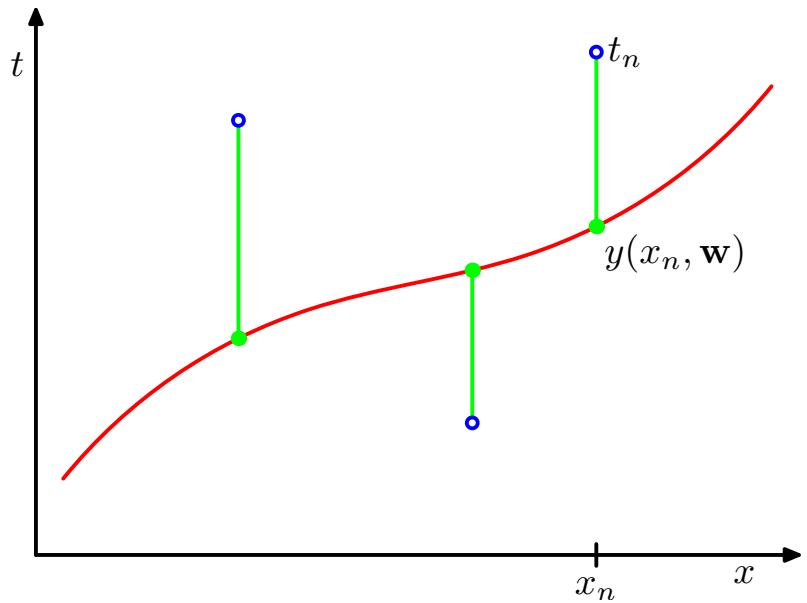


Optimizing the parameters

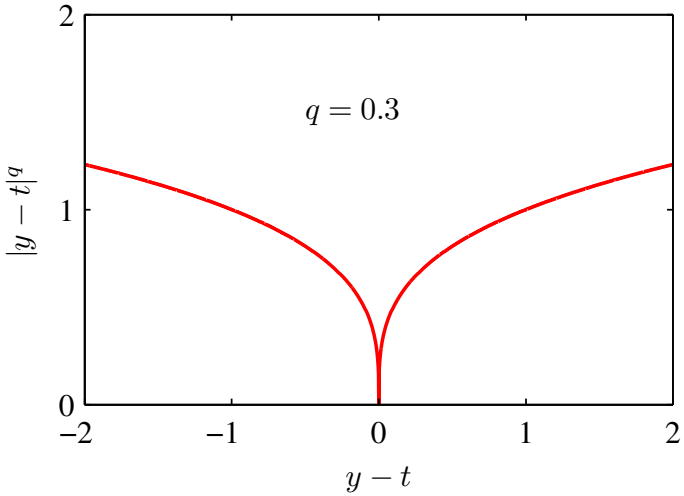
The model parameters

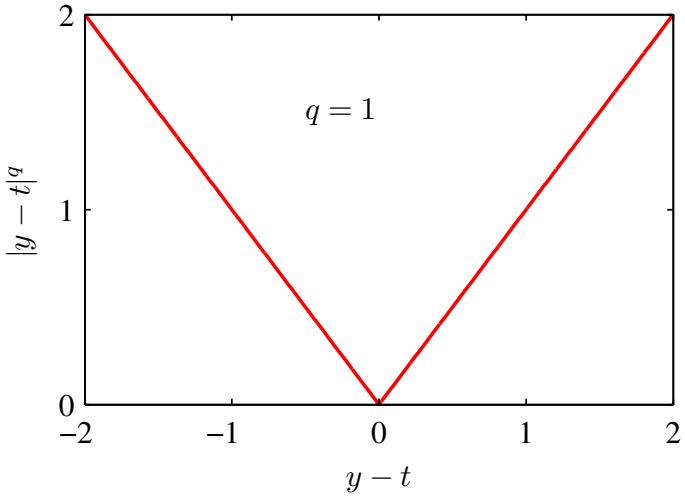
- The chosen model will give us some curve that is needed to adjust such that we'll **minimize its distance** to the **targets** t.
- This approach lead us to use the **least squares** to estimate the weights and minimize the **error** E.

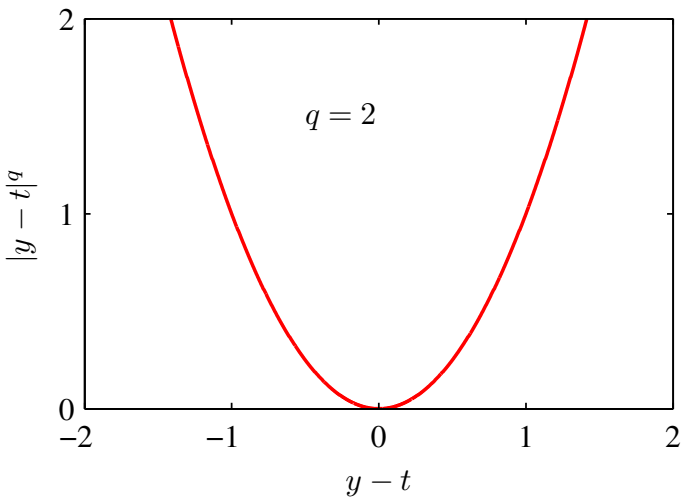
$$E(\mathbf{w}) \triangleq \frac{1}{2} \sum_{n=1}^{N} \{y_n - t_n\}^2$$

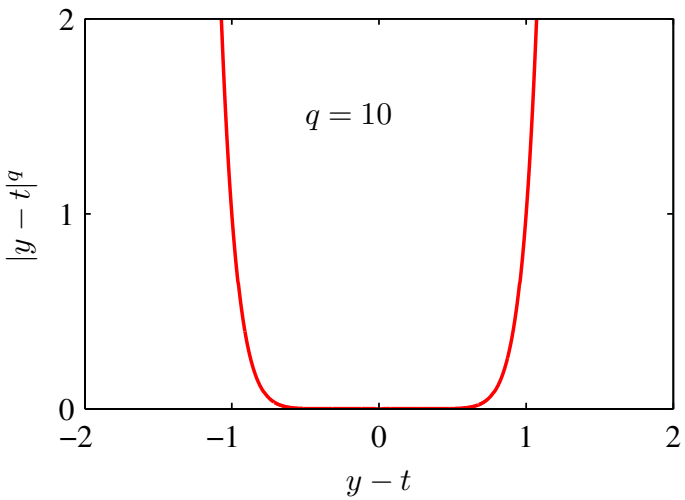


Why choose a quadratic norm distance?









Optimizing the parameters
The model parameters

Why choose a quadratic norm distance?[1]

- 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.

[1] See Appendix ?

Optimizing the parameters

Matrix form

- Remembering that

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

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

$$y_n = \begin{bmatrix}\phi_0(x_n) & \phi_1(x_n) & \ldots & \phi_{M-1}(x_n)\end{bmatrix}\begin{bmatrix}w_0 & w_1 & \cdots & w_{M-1}\end{bmatrix}^\top$$

- And then

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

  where Φ is the **design matrix**.

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

Optimizing the parameters

The cost function and minimization problem

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

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

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

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

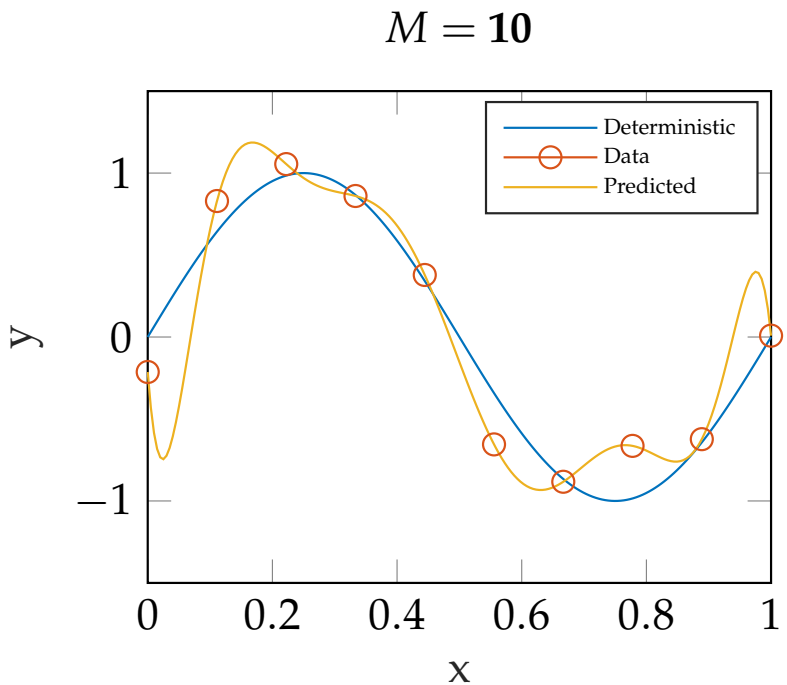
- Here, we've obtained the weights $\mathbf{w}^*$ with the **best fit** of the curve.
- We could say that the model **learned** the parameters.

Optimizing the parameters

The over-fitting phenomenon

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**.

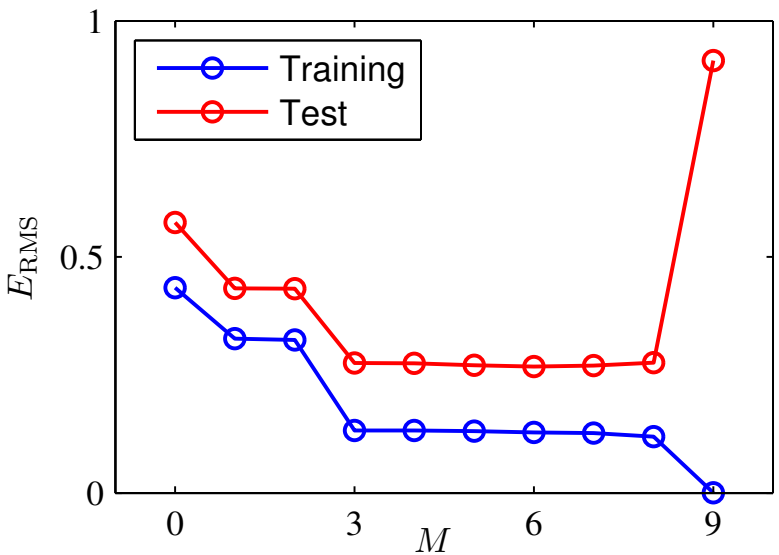


Optimizing the parameters

Training and testing

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**.



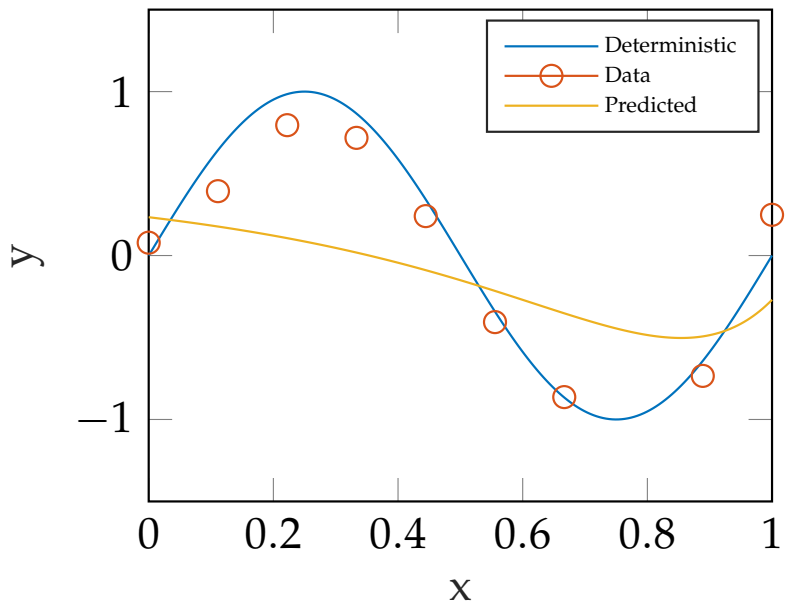
How to control the over-fitting?

- 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 λ to error function, by this we force the coefficients to not reach high values.

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

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

$\ln\lambda = \mathbf{0} \mid M = \mathbf{10}$

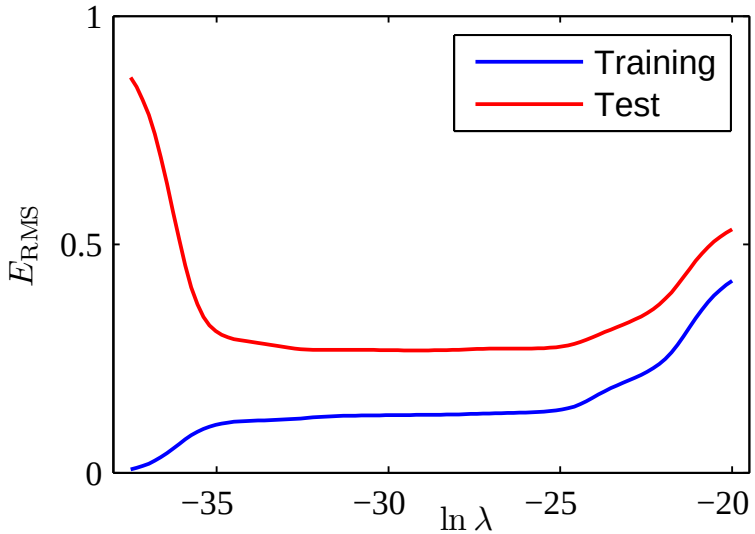


How to control the over-fitting?

- 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 λ to error function, by this we force the coefficients to not reach high values.

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

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

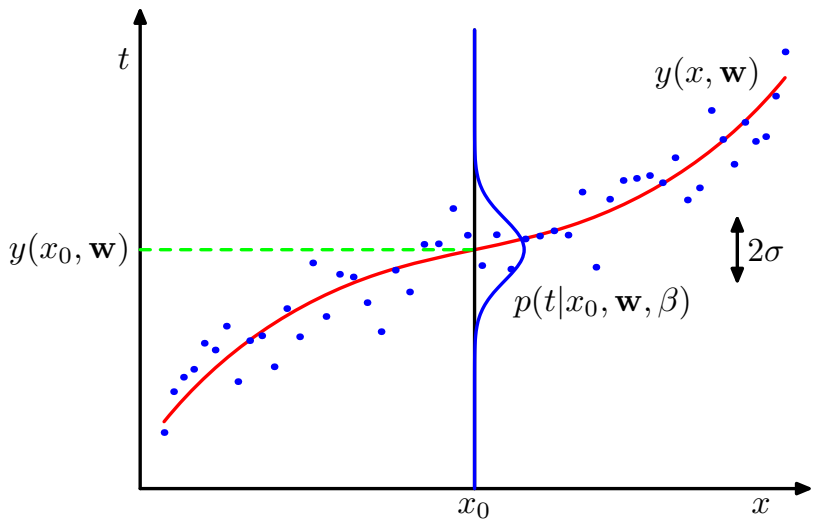


An uncertainty perspective

Looking back the regression

What if we assume not knowing the data exactly?

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



An uncertainty perspective

Targets as distributions

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

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

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

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

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

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

An uncertainty perspective

Back to the cost function

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

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

- If we make

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

  we'll obtain the **cost function** obtained before in the linear regression, then our assumptions are well grounded.
- With the maximization, we'll obtain the weights $\mathbf{w}$ that **maximize** the log probability of the targets, given the parameters.
- This is called **maximum likelihood**, since we are looking for the **parameters** distribution that are more probable to had been generated the data.
- This is a initial step towards to a **Bayesian** approach.

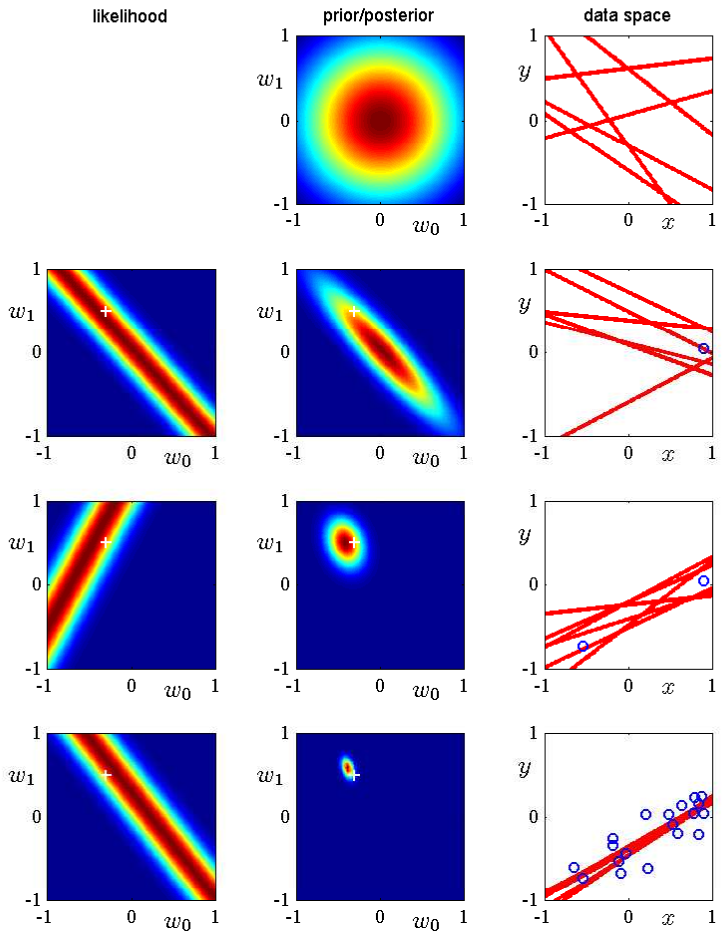
Bayesian Linear Regression

What if we assume not knowing the data exactly?

- The principle of the Bayesian statistics is express our **degree of belief** in an event.
- This belief could be based on some **prior** knowledge about the event or personal beliefs.
- This differs from frequentist statistics, where the probability is based on the number of trials.
- Suppose a die to be thrown once

Frequentist

There's empiric evidence that similar dice thrown in past produce similar outcomes with the same frequency.

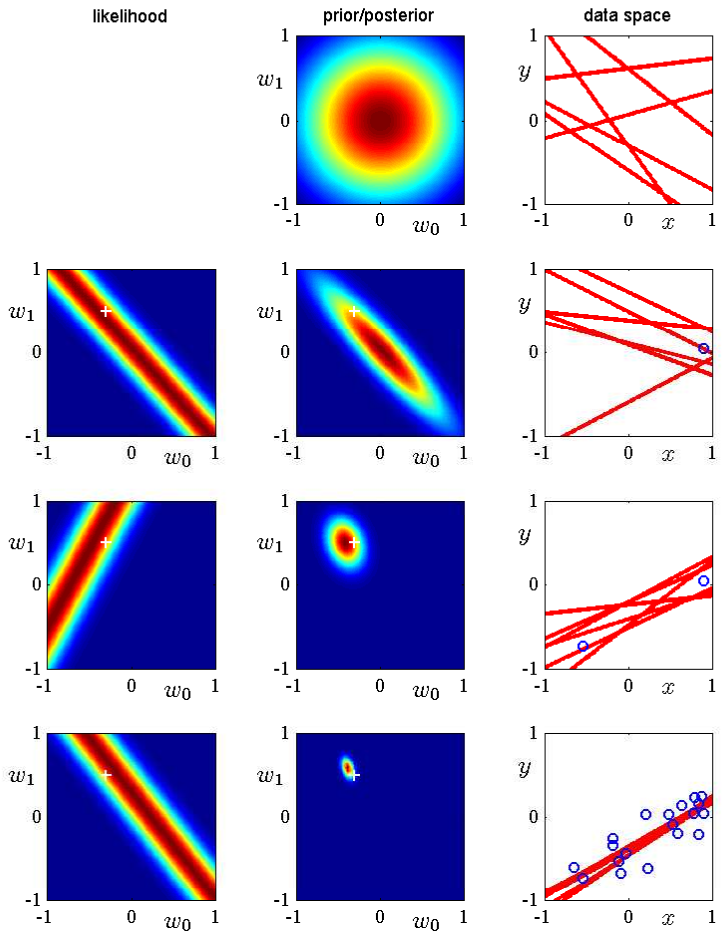


What if we assume not knowing the data exactly?

- The principle of the Bayesian statistics is express our **degree of belief** in an event.
- This belief could be based on some **prior** knowledge about the event or personal beliefs.
- This differs from frequentist statistics, where the probability is based on the number of trials.
- Suppose a die to be thrown once

Bayesian

The last argument is right, but the **belief of the observer** it's important to the statistics, as the number of the Lotto.



Mathematically, by Bayes' Rule, we could infer

Mathematically, by Bayes' Rule, we could infer

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

Mathematically, by Bayes' Rule, we could infer

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

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

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

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

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

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

The D-dimensional Gaussian Distribution

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 μ is the mean and σ^2 the variance.

The D-dimensional Gaussian Distribution

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 μ is the mean and σ^2 the variance.

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

The D-dimensional Gaussian Distribution

To more than one dimensions, we'll consider the points (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 indentity matrix. This means that the all the distances are equally normalized. The matrix $\boldsymbol{\Sigma}$ is the covariance matrix of the distributions, by definition.

The D-dimensional Gaussian Distribution

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.

The D-dimensional Gaussian Distribution

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:

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

- Marginal distribution:

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

Bayes' rule for Gaussian variables

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

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

Bayes' rule for Gaussian variables

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

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

Bayes' rule for Gaussian variables

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

$$
\begin{aligned}
p(\mathbf{x}) &= \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right) \\
p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}\right)
\end{aligned}
$$

So, applying the joint distribution and the its ln after

$$
\begin{aligned}
p(\mathbf{z}) &= p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x}) \\
\ln p(\mathbf{z}) &= \ln p(\mathbf{y}|\mathbf{x}) + \ln p(\mathbf{x}) \\
&= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top} \boldsymbol{\Lambda}(\mathbf{x}-\boldsymbol{\mu}) \\
&\quad -\frac{1}{2}(\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{b})^{\top} \mathbf{L}(\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{b}) + \text{const}
\end{aligned}
$$

Bayes' rule for Gaussian variables

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

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

Bayes' rule for Gaussian variables

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 $\mathbf{R}^{-1}$

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

Bayes' rule for Gaussian variables

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

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

Bayes' rule for Gaussian variables

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

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

By inspection of the linear terms

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

Bayes' rule for Gaussian variables

And then we we'll have that

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

Bayes' rule for Gaussian variables

And then we we'll have that

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

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

Bayesian Linear Regression

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

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

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

Bayes' rule for Gaussian variables

Marginal and Conditioned Gaussians

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

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

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

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

Bayes' rule for Gaussian variables

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

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

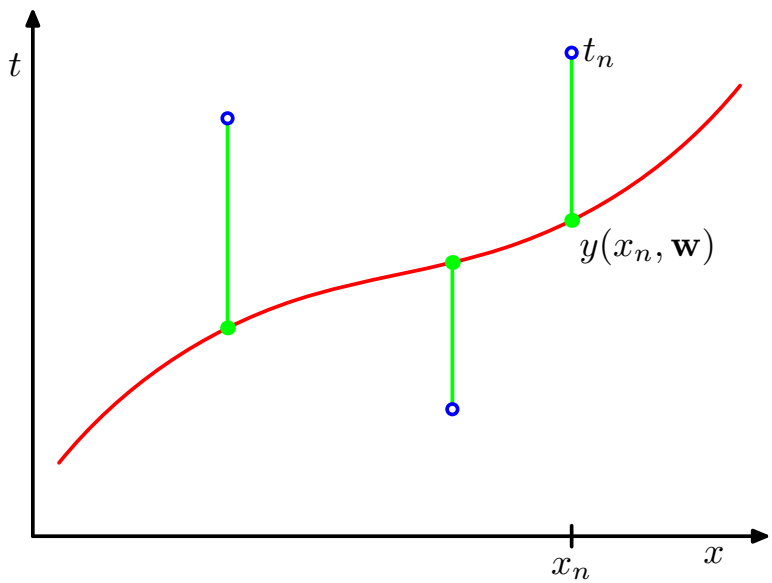
And then $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}\left(\mathbf{w}|\mathbf{m}_N,\mathbf{S}_N\right)$ where

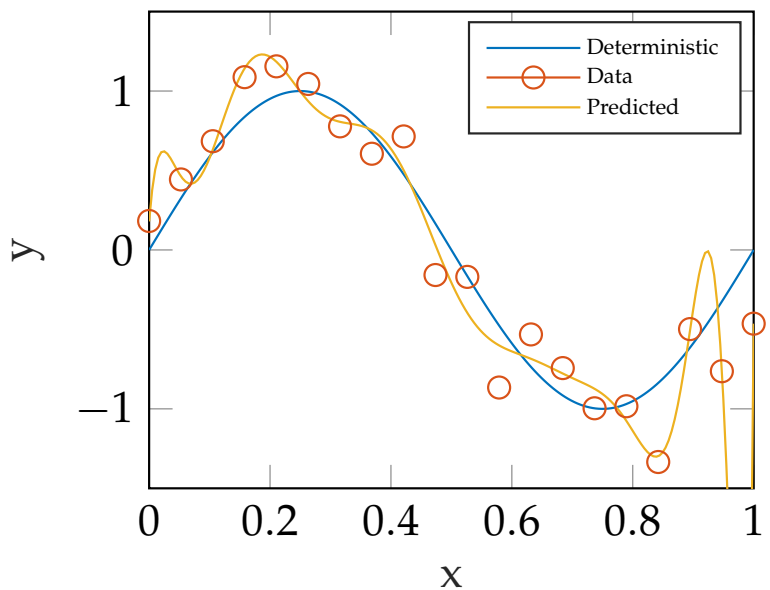
$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N\left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^\top\mathbf{t}\right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^\top\mathbf{\Phi} \end{aligned}$$

Gaussian Processes

What was done until here?

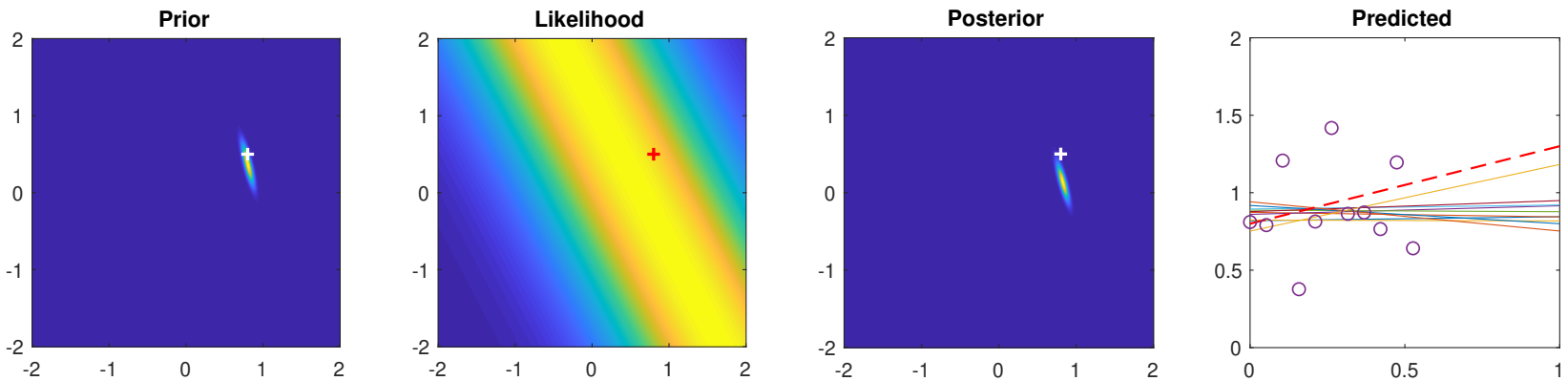
- We assumed that our targets t were **i.i.d.** and given by $t = y(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \beta)$.
- Our model is given by $\mathbf{y}(\mathbf{x}) = \Phi^\top \mathbf{w}$, where Φ is the **design matrix**, and this caracterize our model as **linear in parameters**.
- The **design matrix** was defined as $\phi_{i,j} = \phi_i(\mathbf{x}_j)$.
- The **parameters** were given by $\mathbf{w} = \left(\Phi^\top \Phi\right)^{-1} \Phi^\top \mathbf{t}$.
- 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 se 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**

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

- To **feature-space**

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

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

Alternative formulation

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

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

What is kernel?

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

- We could observe the appearance of terms like $\Phi^\top \mathbf{S}_0 \Phi$, $\boldsymbol{\phi}_*^\top \mathbf{S}_0 \Phi$, or $\boldsymbol{\phi}_*^\top \mathbf{S}_0 \boldsymbol{\phi}_*$.
- The common term between these operations is $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{S}_0 \boldsymbol{\phi}(\mathbf{x}')$
- 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.

Gaussian processes

In change of space

- 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

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

- We have

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k\left(\mathbf{x}, \mathbf{x}'\right) &= \mathbb{E}\left[\left(f(\mathbf{x}) - m(\mathbf{x})\right)\left(f\left(\mathbf{x}'\right) - m\left(\mathbf{x}'\right)\right)\right] \end{aligned}$$

Bibliography I

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/schonl2011.pdf.