Linear regression

Course of Machine Learning
Master Degree in Computer Science
University of Rome "Tor Vergata"
a.a. 2024-2025

Giorgio Gambosi

Linear models

Linear models are based on a linear combination of input features

$$h(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

More compactly,

$$h(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \bar{\mathbf{x}} = \begin{pmatrix} w_0 & w_1 & \cdots & w_d \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{pmatrix}$$

where $\bar{\mathbf{x}} = (1, x_1, \ldots, x_d)^T$

Observe that such models are linear both with respect to features and, more important, also to parameters. This is relevant since, during the learning phase of the models, parameters are treated as variables.

In general, the set of features can be modified (in particular, extended) by means of a set of predefined **base functions** $\phi_1, \ldots, \phi_m$ defined as $\phi_i : \mathbb{R}^d \mapsto \mathbb{R}$. That is, each vector $\mathbf{x}$ in $\mathbb{R}^d$ is mapped to a new vector in $\mathbb{R}^m$, $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x}))$. The prediction task is mapped from a d-dimensional to an m-dimensional space (usually with $m > d$). This is an action concerning **feature engineering**, which concerns the search of an effective representation of the data items from which predictions are to be made.

Clearly, applying base functions does not change the linearity of a linear model, which then has the structure

$$h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}) = \sum_{j=1}^{m} w_j \phi_j(\mathbf{x})$$

Most common types of base functions:

- Polynomial (global functions)

$$\phi_j(x) = x^j$$

- Gaussian (local)

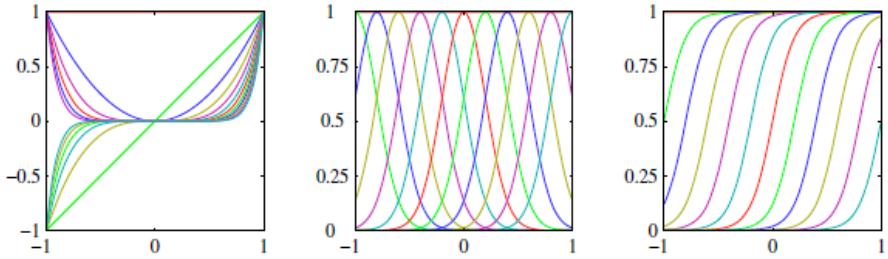
$$\phi_j(x) = \exp\left(-\frac{(x - \mu_j)^2}{2s^2}\right)$$

- Sigmoid (local)

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) = \frac{1}{1+e^{-\frac{x-\mu_j}{s}}}$$

- Hyperbolic tangent (local)

$$\phi_j(x) = \tanh(x) = 2\sigma(x) - 1 = \frac{1-e^{-\frac{x-\mu_j}{s}}}{1+e^{-\frac{x-\mu_j}{s}}}$$



Observe that a set of items

$$\mathrm{X} = \begin{pmatrix} - & \mathbf{x}_1 & - \\ & \vdots & \\ - & \mathbf{x}_n & - \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

is transformed by the set $\boldsymbol{\phi}$ of base functions into

$$\Phi = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_m(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_n) & \phi_2(\mathbf{x}_n) & \cdots & \phi_m(\mathbf{x}_n) \end{pmatrix}$$

The case when we extend $\mathbf{X}$ by 1 values to

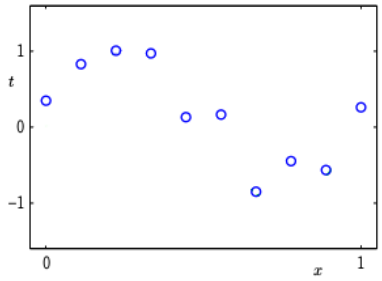
$$\overline{\mathrm{X}} = \begin{pmatrix} - & \overline{\mathbf{x}}_1 & - \\ & \vdots & \\ - & \overline{\mathbf{x}}_n & - \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{pmatrix}$$

is just a special case $\boldsymbol{\phi} = (1, \pi_1(\mathbf{x}), \ldots, \pi_d(\mathbf{x}))$, where $\pi_i(\mathbf{x}) = x_i$.

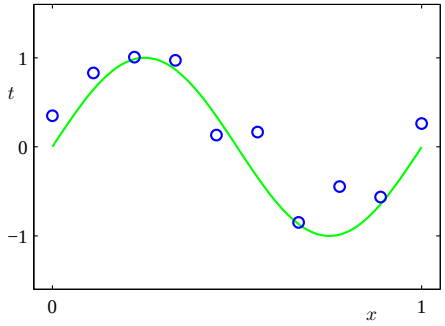
In the following, we will, for generality, usually refer to the training set (Φ, t) resulting from the application of a generic set of base function $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_m)$ to the items. We remark again that the original dataset $(\mathbf{X}, \mathbf{t})$ (or possibly $(\overline{\mathbf{X}}, \mathbf{t})$) is just a particular case.

- A set of n observations of two variables $x, t \in \mathbb{R}$: $(x_1, t_1), \ldots, (x_n, t_n))$ is available. We wish to exploit these observations to predict, for any value $\tilde{x}$ of x, the corresponding unknown value of the target variable t
- The training set is a pair of vectors $\mathbf{x} = (x_1, \ldots, x_n)^T$ and $\mathbf{t} = (t_1, \ldots, t_n)^T$, related through an unknown rule (function)

Example of a training set.



In this case, we assume that the (unknown) relation between x and t in the training set is provided by the function $t = \sin(2\pi x)$, with an additional gaussian noise with mean 0 and given variance σ^2. Hence, $t_i = \sin(2\pi x_i) + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.



Our purpose is guessing, or approximating as well as possible, the deterministic relation $t = \sin(2\pi x)$, on the basis of the analysis of data in the training set.

In polynomial regression we wish to approximate the unknown function through a suitable polynomial of given degree $m > 0$

$$h(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_m x_m = \sum_{j=0}^{m} w_j x^j$$

whose coefficients $\mathbf{w} = (w_0, w_1, \ldots, w_m)^T$ are to be computed.

This corresponds to applying a set $\boldsymbol{\phi}$ of $m + 1$ base functions $\phi_j(x) = x^j, j = 0, \ldots, m$ to the unique feature x

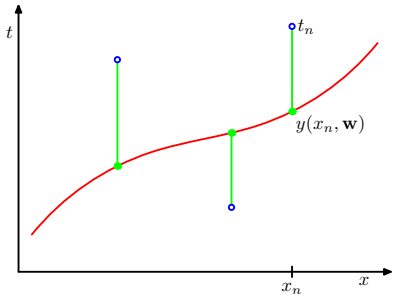
$$h(\boldsymbol{\phi}(x), \mathbf{w}) = \sum_{j=0}^{m} w_j \phi_j(x)$$

Regression loss

When base functions are applied, $h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w})$ is a nonlinear function of $\boldsymbol{\phi}(\mathbf{x})$, but it is still a linear function (model) of $\mathbf{w}$.

The values assigned to coefficients should minimize the empirical risk computed wrt some **error function** (a.k.a. **cost function**), when applied to data in the training set (then, to $\boldsymbol{\phi}(\mathbf{x})$, $\mathbf{t}$ and $\mathbf{w}$).

A most widely adopted error function is the **quadratic loss** $(h(\boldsymbol{\phi}(\mathbf{x}_i)) - t_i)^2$, which results into the **least quares** approach, i.e. minimizing the sum, for all items in the training set, of the (squared) difference between the value returned by the model and the target value.



$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} r_i(\mathbf{w})^2$$

where

$$r_i(\mathbf{w}) = h(\boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}) - t_i = \sum_{j=1}^{m} w_j \phi_j(\mathbf{x}_i) - t_i$$

is the **residue** for item $(\boldsymbol{\phi}(\mathbf{x}_i), t_i)$ if $\mathbf{w}$ is the vector of parameter values applied.

This is clearly equivalent to minimizing the empirical risk $\overline{\mathcal{R}}(\mathbf{w})$, since

$$E(\mathbf{w}) = \frac{|\mathcal{T}|}{2}\overline{\mathcal{R}}(\mathbf{w})(\mathbf{w})$$

To minimize $E(\mathbf{w})$, set its derivative w.r.t. $\mathbf{w}$ to $\mathbf{0}$. Since the quadratic loss is a convex function, only one (global) minimum is defined. The error $E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(h(\boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}) - t_i)^2$ is the sum of n convex functions $(h(\boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}) - t_i)^2$, which implies that only one (global) minimum is defined. In particular, $E(\mathbf{w})$ quadratic implies that its derivative is linear, hence that it is zero in one point $\mathbf{w}^*$: the resulting predictor is $h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}^*)$.

The gradient w.r.t. $\mathbf{w}$ is indeed a collection of derivatives. A linear system is obtained:

$$\frac{\partial E(\mathbf{w})}{\partial w_k} = 2\sum_{i=1}^{n} r_i(\mathbf{w})\frac{\partial}{\partial w_k} r_i(\mathbf{w}) = 2\sum_{i=1}^{n} r_i(\mathbf{w})\phi_k(\mathbf{x}_i) = 2\sum_{i=1}^{n}\left(\sum_{j=0}^{m} w_j \phi_j(\mathbf{x}_i) - t_i\right)\phi_k(\mathbf{x}_i)$$

since

$$\frac{\partial}{\partial w_k} r_i(\mathbf{w}) = \frac{\partial}{\partial w_k} h(\boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w})$$

Each of the $m+1$ equations is linear w.r.t. each coefficient in $\mathbf{w}$. A linear system results, with $m+1$ equations and $m+1$ unknowns $w_0, \ldots, w_m$, which, in general and with the exceptions of degenerate cases, has precisely one solution, that can be expressed in closed form by the **normal equations** for least squares.

$$\mathbf{w}^* = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$$

The minimum of $E(\mathbf{w})$ can be also computed numerically, by means of **gradient descent** methods with the following structure

1. Initial assignment $\mathbf{w}^{(0)} = (w_1^{(0)}, w_2^{(0)}, \ldots, w_m^{(0)})$, with a corresponding error value

$$E(\mathbf{w}^{(0)}) = \frac{1}{2}\sum_{i=1}^{n} r_i(\mathbf{w}^{(0)})^2$$

2. Iteratively, the current value $\mathbf{w}^{(s-1)}$ is modified in the direction of steepest descent of $E(\mathbf{w})$, that is the one corresponding to the negative of the gradient evaluated at $\mathbf{w}^{(s-1)}$

3. At step s, $w_k^{(s-1)}$ is updated as follows:

$$w_k^{(s)} := w_k^{(s-1)} - \eta \frac{\partial E(\mathbf{w})}{\partial w_k}\bigg|_{\mathbf{w}^{(s-1)}} = w_k^{(s-1)} - 2\eta \sum_{i=1}^{n} r_i(\mathbf{w}^{(s-1)})\phi_k(\mathbf{x}_i)$$

In matrix notation:

$$\mathbf{w}^{(s)} := \mathbf{w}^{(s-1)} - \eta \nabla E(\mathbf{w})\big|_{\mathbf{w}^{(s-1)}}$$

4. By definition of $E(\mathbf{w})$:

$$\mathbf{w}^{(s)} := \mathbf{w}^{(s-1)} - 2\eta \sum_{i=1}^{n} r_i(\mathbf{w}^{(s-1)})\boldsymbol{\phi}(\mathbf{x}_i)$$

$$\begin{pmatrix} w_1^{(s)} \\ w_2^{(s)} \\ \vdots \\ w_m^{(s)} \end{pmatrix} = \begin{pmatrix} w_1^{(s-1)} \\ w_2^{(s-1)} \\ \vdots \\ w_m^{(s-1)} \end{pmatrix} - 2\eta \sum_{j=1}^{n} r(\mathbf{w}^{(s-1)}) \begin{pmatrix} \phi_1(\mathbf{x}_i) \\ \phi_2(\mathbf{x}_i) \\ \vdots \\ \phi_m(\mathbf{x}_i) \end{pmatrix}$$

where, we remind,

$$r_i(\mathbf{w}^{(s-1)}) = \sum_{j=1}^{m} w_j^{(s-1)}\phi_j(\mathbf{x}_i) - t_i$$

As we may see, the update at each step is proportional to the linear combination of the items (possibly transformed by the application of base functions), each weighted by the corresponding residue, that is by the current error in its prediction.

Fitting of polynomials: polynomial degree. We apply here a set of base function $\boldsymbol{\phi} = (1, x, x^2, \ldots, x^M)$ to 1-dimemsional items.

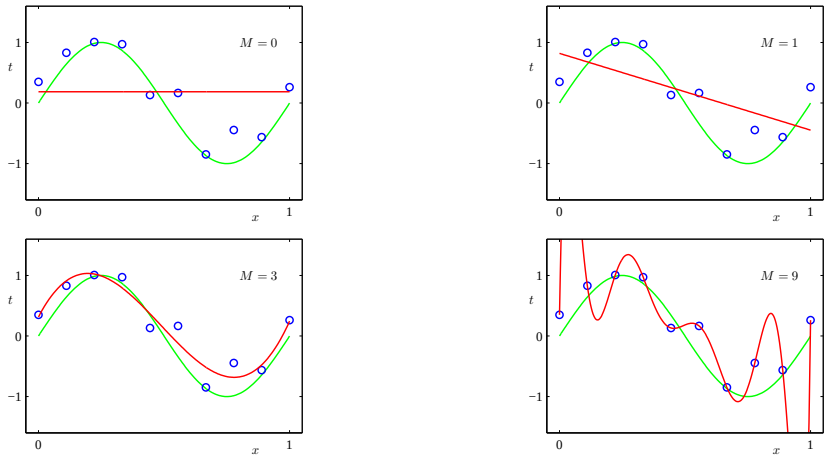
- Example of model selection: assigning a value to the degree M determines the representation of the items in the training set and as a consequence the specific model to be used, since the choice of M implies the number of coefficients in $\mathbf{w}$ to be estimated
- increasing M allows to better approximate the training set items, decreasing the error
- if $M + 1 = n$ the model allows to obtain a null error (overfitting)

Overfitting

- The function $h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w})$ is derived from items in the training set, but should provide good predictions for other items.
- It should provide a suitable generalization to all items in the whole domain.
- If $h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w})$ is derived as a too much accurate depiction of the training set, it results into an unsuitable generalization to items not in the training set

  Evaluation of the generalization

  – Training set $\mathcal{T}_{train}$ of 1-dimensional items, generated by uniformly sampling x in $[0, 1,]$ and ε from $\mathcal{N}(0, \sigma^2)$, and computing $t = \sin 2\pi x + \varepsilon$

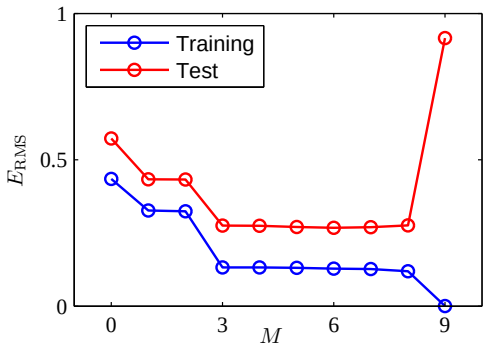


- Test set $\mathcal{T}_{test}$ of 1-dimensional items, generated in the same way as the training set
- For each M:
  - derives $\mathbf{w}^*$ by minimizing the empirical risk on the training set $\overline{\mathcal{R}}_{\mathcal{T}_{train}}(\mathbf{w})$
  - compute the empirical risk $\overline{\mathcal{R}}_{\mathcal{T}_{test}}(\mathbf{w}^*)$ on the test set: the square root of such value is considered here

$$E_{RMS}(\mathbf{w}^*, \mathcal{T}_{test}) = \sqrt{\overline{\mathcal{R}}_{\mathcal{T}_{test}}(\mathbf{w}^*)} = \sqrt{\frac{1}{|\mathcal{T}_{test}|} \sum_{(\mathbf{x},t)\in\mathcal{T}_{test}} (h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}^*) - t)^2}$$

- a lower value of $E_{RMS}(\mathbf{w}^*, \mathcal{T}_{test})$ denotes a good generalization

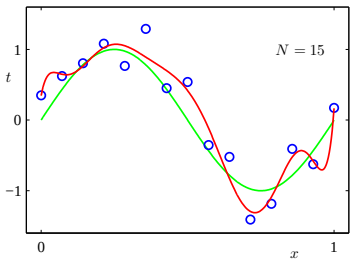
Plot of E_{RMS} w.r.t. M, on the training set and on the test set.

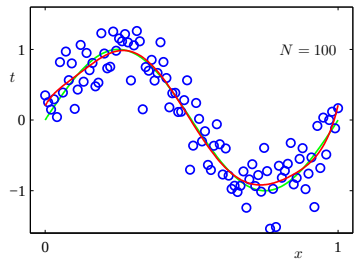


- As M increases, the error on the training set tends to 0.
- On the test set, the error initially decreases, since the higher complexity of the model allows to better deal with the characteristics of the data set. Next, the error increases, since the model becomes too dependent from the training set.

For a given model complexity (such as the degree in our example), overfitting decreases as the dimension of the training set increases.

The larger the training set, the higher the acceptable complexity of the model.





How to limit the complexity of the model?

A common approach is introducing a **regularization term** in the cost function

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Here, $E_D(\mathbf{w})$ is the empirical risk, which is dependent from the dataset (and the parameters): The regularization term $E_W(\mathbf{w})$ is instead dependent from the parameters alone.

The **regularization coefficient** λ controls the relative importance of the two terms.

If $E_D(\mathbf{w})$ is defined in terms of quadratic loss, we are dealing with regularized least squares learning. Different types of regularized least squares can be obtained in dependance of how the regularization term is defined.

The most common form is the sum of the squared values of the coefficients (times $1/2$, but this not relevant.

$$E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}\sum_{i=1}^{m} w_i^2$$

The resulting overall loss to be minimized is then

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} r_i(\mathbf{w})^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}\boldsymbol{r}(\mathbf{w})^T\boldsymbol{r}(\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

where $\boldsymbol{r}(\mathbf{w})$ is the vector of residues, which can be expressed in terms of Φ, $\mathbf{w}$ and $\mathbf{t}$ as

$$\boldsymbol{r}(\mathbf{w}) = \begin{pmatrix} r_1(\mathbf{w}) \\ r_2(\mathbf{w}) \\ \vdots \\ r_n(\mathbf{w}) \end{pmatrix} = \begin{pmatrix} h(\boldsymbol{\phi}(\mathbf{x}_1), \mathbf{w}) \\ h(\boldsymbol{\phi}(\mathbf{x}_2), \mathbf{w}) \\ \vdots \\ h(\boldsymbol{\phi}(\mathbf{x}_n), \mathbf{w}) \end{pmatrix} - \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_m(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \cdots & \phi_m(\mathbf{x}_2) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_n) & \cdots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} - \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix} = \Phi\mathbf{w} - \mathbf{t}$$

this is called **ridge regression**: its solution can be expressed in closed form as

$$\mathbf{w} = (\lambda\mathbf{I} + \Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$$

A more general form is obtained by considering the degree of the summed coefficients as a parameter

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} r_i(\mathbf{w})^2 + \frac{\lambda}{2}\sum_{j=1}^{m} |w_j|^q$$

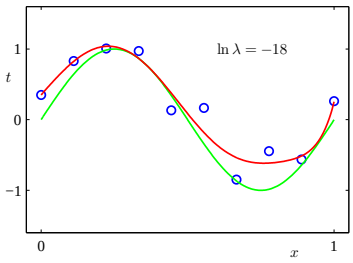
The case $q = 1$ is denoted as **lasso**. Lasso regression has the property of favor sparse models (that is returning parameter vectors with many null values).

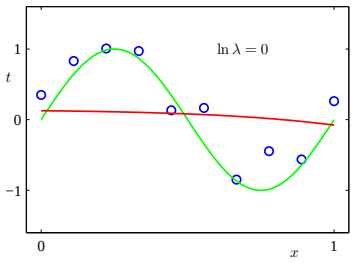
Example: polynomial regression
Use of **regularization** to limit complexity and overfitting.

- inclusion of a penalty term in the error function
- purpose: limiting the possible values of coefficients
- usually: limiting the absolute value of the coefficients

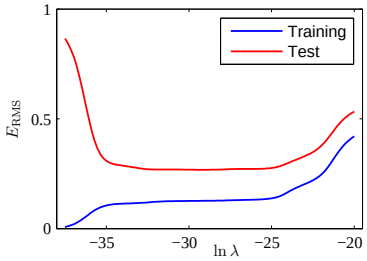
$$\tilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(h(\boldsymbol{\phi}(x_i),\mathbf{w}) - t_i)^2 + \frac{\lambda}{2}\sum_{k=0}^{M} w_k^2 = \frac{1}{2}\sum_{i=1}^{n}(h(\boldsymbol{\phi}(x_i),\mathbf{w}) - t_i)^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$

Dependance from the value of the hyperparameter λ.



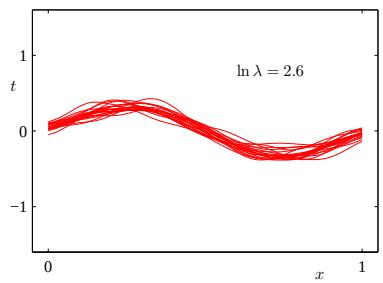


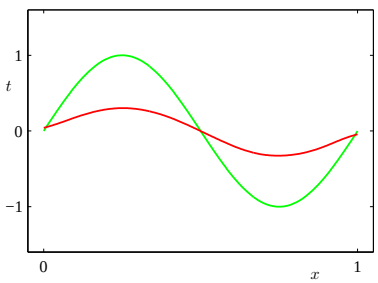
Plot of the error w.r.t λ, ridge regression.



- Small λ: overfitting. Small error on the training set, large error on the test set.
- Large λ: the effect of data values decreases. Large error on both test and training sets.
- Intermediate λ. Intermediate error on training set, small error on test set.
- Consider the case of function $y = \sin 2\pi x$ and assume $L = 100$ training sets $\mathcal{T}_1, \ldots, \mathcal{T}_L$ are available, each of size $n = 25$.
- Given $m = 24$ gaussian base functions $\boldsymbol{\phi} = (\phi_1(x), \ldots, \phi_m(x))$, from each training set $\mathcal{T}_i$ a prediction function $h_i(\boldsymbol{\phi}(x))$ is derived by minimizing the regularized cost function

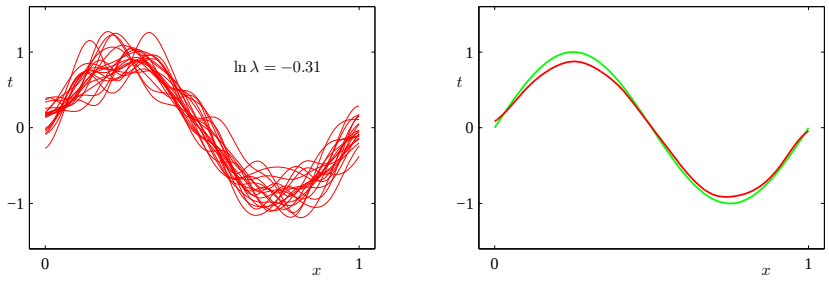
$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}\sum_{i=1}^{n}\left(\sum_{j=1}^{m} w_j\phi_j(\mathbf{x}_i) - t_i\right) + \frac{\lambda}{2}\sum_{k=1}^{m} w_k^2 \\ &= \frac{1}{2}(\Phi\mathbf{w} - \mathbf{t})^T(\Phi\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \end{aligned}$$



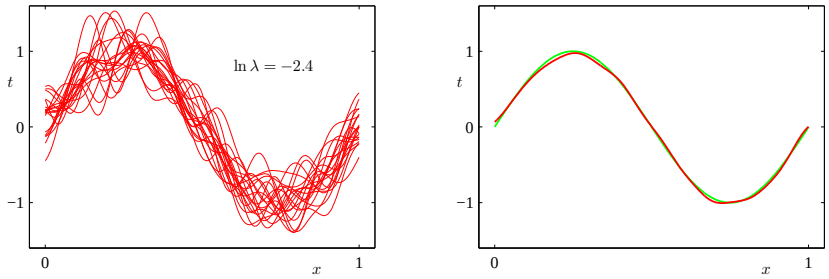


Left, a possible plot of prediction functions $h_i(\boldsymbol{\phi}(\mathbf{x}))$ $(i = 1, \ldots, 100)$, as derived, respectively, by training sets $\mathcal{T}_i, i = 1, \ldots, 100$ setting $\ln \lambda = 2.6$. Right, their expectation, with the unknown function $y = \sin 2\pi x$.

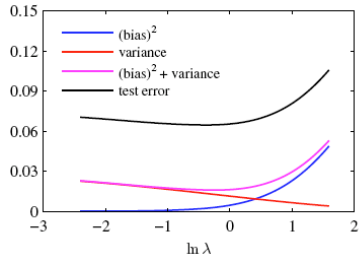
The prediction functions $h_i(\boldsymbol{\phi}(\mathbf{x}))$ do not differ much between them (small variance), but their expectation is a bad approximation of the unknown function (large bias).



Plot of the prediction functions obtained with $\ln \lambda = -0.31$.



Plot of the prediction functions obtained with $\ln \lambda = -2.4$. As λ decreases, the variance increases (prediction functions $h_i(\boldsymbol{\phi}(\mathbf{x}))$ are more different each other), while bias decreases (their expectation is a better approximation of $y = \sin 2\pi x$).

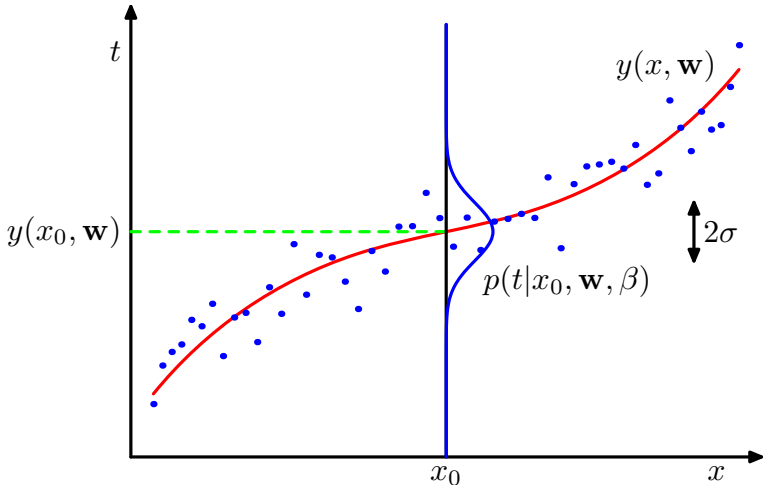


- Plot of $(\text{bias})^2$, variance and their sum as functions of λ: las λ increases, bias increases and varinace decreases. Their sum has a minimum in correspondance to the optimal value of λ.

Probabilistic model for regression

As said before, in this case we define a class of joint probability distributions in order to select the best one of them with respect to the training set. The class we define here is a class $p(\mathbf{x}, t) = p_M(\mathbf{x})p_C(t|\mathbf{x})$ where $p(\mathbf{x})$ is uniform (so we shall not take it into account) while $p_C(t|\mathbf{x})$ will be assumed to be a gaussian. In particular, we assume that, given an item $\mathbf{x}$, the corresponding unknown target t is normally distributed around the value returned by the model $h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w})$, with a given variance σ^2 or, equivalently precision β, where $\beta^{-1} = \sigma^2$.

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}), \beta^{-1})$$



An estimate β_{ML} of β and $\mathbf{w}_{ML}$ of the coefficients $\mathbf{w}$ can be performed on the basis of the likelihood of the training set with respect to the model:

$$L(\mathbf{w}, \beta|\Phi, \mathbf{t}) = p(\mathbf{t}|\Phi, \mathbf{w}, \beta) = \prod_{i=1}^{n} \mathcal{N}(t_i|y(\boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}), \beta^{-1}) = \prod_{i=1}^{n} \frac{\sqrt{\beta}}{\sqrt{2\pi}} e^{\frac{\beta}{2} r_i(\mathbf{w})^2}$$

Parameter values $\mathbf{w}_{ML}$ and β_{ML} can be estimated as the values which maximize the data likelihood $L(\mathbf{w}, \beta|\Phi, \mathbf{t})$ or equivalently its logarithm

$$l(\mathbf{w}, \beta|\Phi, \mathbf{t}) = \log p(\mathbf{t}|\Phi, \mathbf{w}, \beta) = \sum_{i=1}^{n} \log \mathcal{N}(t_i|y(\boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}), \beta^{-1})$$

which results into

$$p(\mathbf{t}|\Phi, \mathbf{w}, \beta) = -\frac{\beta}{2} \sum_{i=1}^{n} r_i(\mathbf{w})^2 + \frac{n}{2} \log \beta + c$$

where c is a constant, independent from $\mathbf{w}$ and β.

The maximization w.r.t. $\mathbf{w}$ is performed by determining a maximum w.r.t. $\mathbf{w}$ of the function

$$-\frac{1}{2} \sum_{i=1}^{n} r_i(\mathbf{w})^2$$

this is equivalent to minimizing the least squares sum.

The maximization w.r.t. the **precision** β is done by setting to 0 the corresponding derivative

$$\frac{\partial l(\mathbf{t}|\Phi, \mathbf{w}, \beta)}{\partial \beta} = -\frac{1}{2} \sum_{i=1}^{n} r_i(\mathbf{w})^2 + \frac{n}{2\beta}$$

which results into

$$\beta_{ML}^{-1} = \frac{1}{n} \sum_{i=1}^{n} r_i(\mathbf{w})^2$$

As a side result, the parameter estimate provides a **predictive distribution** of t given $\mathbf{x}$, that is the (gaussian) distribution of the target value for a given item $\mathbf{x}$.

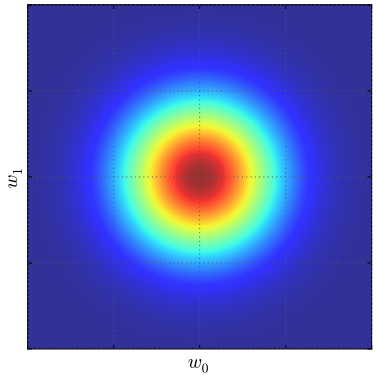
$$p(t|\mathbf{x};\mathbf{w},\beta) = \mathcal{N}(t|h(\boldsymbol{\phi}(\mathbf{x}),\mathbf{w}),\beta^{-1}) = \sqrt{\frac{\beta_{ML}}{2\pi}} e^{-\frac{\beta_{ML}}{2}(h(\boldsymbol{\phi}(\mathbf{x}),\mathbf{w}_{ML})-t)^2}$$

Remind that in the maximum likelihood framework parameters are considered as (unknown) values to determine with the best possible precision (**frequentist** approach). Moreover, being such maximization equivalent to minimizing the squares, it is prone to overfitting, thus a regularization term $\mathcal{E}(\mathbf{w})$ should be introduced. This can be done while staying in the probabilistic framework by applying a bayesian approach.

In this framework, parameters are considered as random variables, whose probability distributions has to be defined or estimated. In particular, we are interested to the probability distribution of parameters, given the observation of the training set, that is of the set of examples $(\mathbf{x}_i, t_i)$, on which the set $\boldsymbol{\phi}$ of base functions is applied. A prior distribution of the parameters will be assumed.

In the case we consider here, the prior distribution of parameters will be assumed to be a gaussian with mean $\mathbf{0}$ and diagonal covariance matrix, with variance equal to the inverse of **hyperparameter** α

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0},\alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{m+1}{2}} e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}}$$



Why a gaussian prior? Because the gaussian distribution is conjugated to itself. This means that the posterior distribution, being proportional to prior times likelihood, is gaussian if the likelihood is gaussian.

$$p(\mathbf{t}|\Phi,\mathbf{w},\beta) = \prod_{i=1}^{n} \mathcal{N}(t_i|h(\boldsymbol{\phi}(\mathbf{x}_i),\mathbf{w}),\beta^{-1}) = \prod_{i=1}^{n} e^{-\frac{\beta}{2}r_i(\mathbf{w})^2}$$

Given the prior $p(\mathbf{w}|\alpha)$, the posterior distribution for $\mathbf{w}$ derives from Bayes' rule

$$p(\mathbf{w}|\Phi,\mathbf{t},\alpha,\beta) = \frac{p(\mathbf{t}|\Phi,\mathbf{w},\beta)p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\Phi,\alpha,\beta)} \propto p(\mathbf{t}|\Phi,\mathbf{w},\beta)p(\mathbf{w}|\alpha)$$

Given the above likelihood, if the prior of $\mathbf{w}$ is a gaussian

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0,\Sigma_0)$$

than the posterior distribution is itself gaussian

$$p(\mathbf{w}|\Phi,\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_p,\Sigma_p)$$

with

$$\Sigma_p = (\Sigma_0^{-1} + \beta\Phi^T\Phi)^{-1}$$
$$\mathbf{m}_p = \Sigma_p(\Sigma_0^{-1}\mathbf{m}_0 + \beta\Phi^T\mathbf{t})$$

In the case we are considering here, we have

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \prod_{j=1}^{m} \frac{\sqrt{\alpha}}{\sqrt{2\pi}} e^{-\frac{\alpha}{2}w_i^2}$$

The posterior distribution is then a gaussian itself

$$p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \sigma) = \mathcal{N}(\mathbf{w}|\mathbf{m}_p, \Sigma_p)$$

with

$$\Sigma_p = (\alpha\mathbf{I} + \beta\Phi^T\Phi)^{-1}$$
$$\mathbf{m}_p = \beta\Sigma_p\Phi^T\mathbf{t}$$

Maximum a Posteriori

Given the posterior distribution $p(\mathbf{w}|\Phi, \mathbf{t}, \alpha, \beta)$, we may derive the value $\mathbf{w}_{MAP}$ of $\mathbf{w}$ which makes it maximum (the mode of the distribution). This is equivalent to maximizing its logarithm

$$\log p(\mathbf{w}|\Phi, \mathbf{t}, \alpha, \beta) = \log p(\mathbf{t}|\mathbf{w}, \Phi, \beta) + \log p(\mathbf{w}|\alpha) - \log p(\mathbf{t}|\Phi, \beta)$$

and, since $p(\mathbf{t}|\Phi, \beta)$ is a constant wrt $\mathbf{w}$

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\text{argmax}}\ \log p(\mathbf{w}|\Phi, \mathbf{t}, \alpha, \beta) = \underset{\mathbf{w}}{\text{argmax}}\ (\log p(\mathbf{t}|\mathbf{w}, \Phi, \beta) + \log p(\mathbf{w}|\alpha))$$

that is,

$$\mathbf{w}_{MAP} = \underset{\mathbf{w}}{\text{argmin}}\ (-\log p(\mathbf{t}|\Phi, \mathbf{w}, \beta) - \log p(\mathbf{w}|\alpha)) = \underset{\mathbf{w}}{\text{argmax}}\ (\log p(\mathbf{t}|\Phi, \mathbf{w}, \beta) + \log p(\mathbf{w}|\alpha))$$

In this case

$$\log p(\mathbf{t}|\Phi, \mathbf{w}, \beta) = \log \prod_{i=1}^{n} \frac{\sqrt{\beta}}{\sqrt{2\pi}} e^{-\frac{\beta}{2}r_i(\mathbf{w})^2} = \sum_{i=1}^{n} \left(\frac{1}{2}\log\beta - \frac{1}{2}\log(2\pi) - r_i(\mathbf{w})^2 \right) = \frac{n}{2}\log\beta - \frac{n}{2}\log(2\pi) - \frac{\beta}{2}\sum_{i=1}^{n} r_i(\mathbf{w})^2$$

and

$$\log p(\mathbf{w}|\alpha) = \log \prod_{j=1}^{m} \frac{\sqrt{\alpha}}{\sqrt{2\pi}} e^{-\frac{\alpha}{2}w_i^2} = \sum_{j=1}^{m} \left(\frac{1}{2}\log\alpha - \frac{1}{2}\log(2\pi) - w_j^2 \right) = \frac{m}{2}\log\alpha - \frac{n}{2}\log(2\pi) - \frac{\alpha}{2}\sum_{j=1}^{m} w_j^2$$

The value $\mathbf{w}_{MAP}$ which maximize the probability (mode of the distribution) minimizes

$$-\frac{\beta}{2}\sum_{i=1}^{n} r_i(\mathbf{w})^2 - \frac{\alpha}{2}\sum_{j=1}^{m} w_j^2 + \frac{n}{2}\log\beta + \frac{m}{2}\log\alpha - \frac{n+m}{2}\log(2\pi)$$

this is equivalent to maximizing

$$\frac{\beta}{2}\sum_{i=1}^{n} r_i(\mathbf{w})^2 + \frac{\alpha}{2}\sum_{j=1}^{m} w_j^2 \propto \frac{1}{2}\sum_{i=1}^{n} r_i(\mathbf{w})^2 + \frac{\alpha}{2\beta}\sum_{j=1}^{m} w_j^2$$

This corresponds to a ridge regression with regularization hyperparameter $\lambda = \frac{\alpha}{\beta}$.

The same considerations of ML appy here for what concerns deriving the predictive distribution of t given $\mathbf{x}$, which results now

$$p(t|\mathbf{x}; \mathbf{w}_{MAP}, \beta_{MAP}) = \mathcal{N}(t|h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_{MAP}), \beta_{MAP}^{-1}) = \sqrt{\frac{\beta_{MAP}}{2\pi}} e^{-\frac{\beta_{MAP}}{2}(h(\boldsymbol{\phi}(\mathbf{x}), \mathbf{w}_{MAP}) - t)^2}$$

where, as it is easy to see, $\beta_{MAP} = \beta_{ML}$

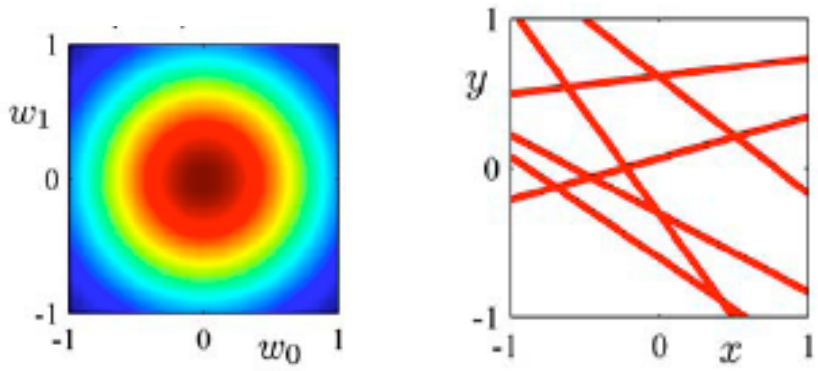
Sequential learning

Observe now that the posterior after observing $\mathcal{T}_1$ can be used as a prior for the next training set acquired.

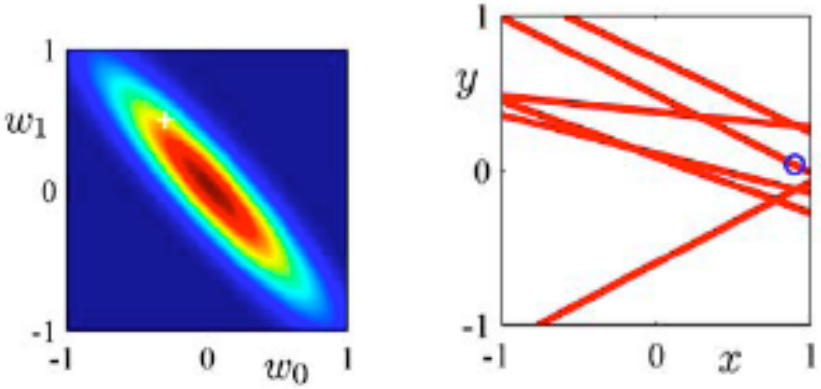
In general, for a sequence $\mathcal{T}_1, \ldots, \mathcal{T}_n$ of training sets,

$$\begin{aligned} p(\mathbf{w}|\mathcal{T}_1, \ldots \mathcal{T}_n) &\propto p(\mathcal{T}_n|\mathbf{w}) p(\mathbf{w}|\mathcal{T}_1, \ldots \mathcal{T}_{n-1}) \\ p(\mathbf{w}|\mathcal{T}_1, \ldots \mathcal{T}_{n-1}) &\propto p(\mathcal{T}_{n-1}|\mathbf{w}) p(\mathbf{w}|\mathcal{T}_1, \ldots \mathcal{T}_{n-2}) \\ &\ldots \\ p(\mathbf{w}|T_1) &\propto p(T_1|\mathbf{w}) p(\mathbf{w}) \end{aligned}$$

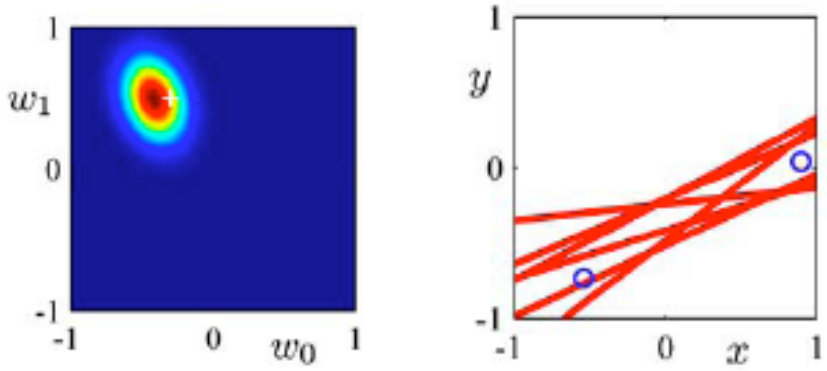
- Input variable x, target variable t, linear regression $y(x, w_0, w_1) = w_0 + w_1 x$.
- Dataset generated by applying function $y = a_0 + a_1 x$ (with $a_0 = -0.3$, $a_1 = 0.5$) to values uniformly sampled in $[-1, 1]$, with added gaussian noise ($\mu = 0$, $\sigma = 0.2$).
- Assume the prior distribution $p(w_0, w_1)$ is a bivariate gaussian with $\boldsymbol{\mu} = \mathbf{0}$ and $\Sigma = \sigma^2 \mathbf{I} = 0.04\mathbf{I}$



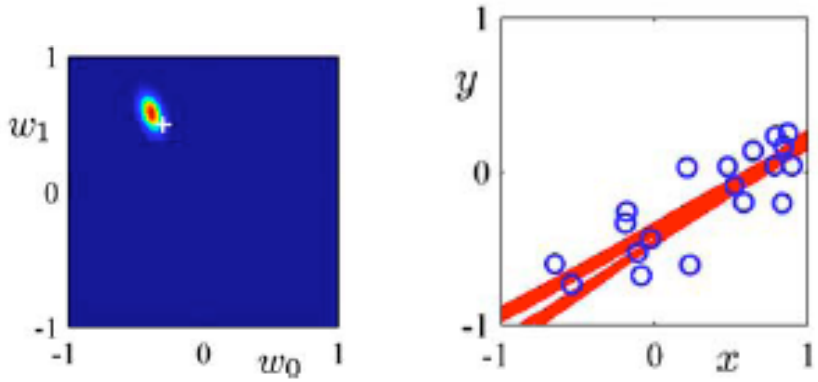
Left, prior distribution of w_0, w_1; right, 6 lines sampled from the distribution.
After observing item (x_1, y_1) (circle in right figure).



Left, posterior distribution $p(w_0, w_1|x_1, y_1)$; right, 6 lines sampled from the distribution.
After observing items $(x_1, y_1), (x_2, y_2)$ (circles in right figure).



Left, posterior distribution $p(w_0, w_1|x_1, y_1, x_2, y_2)$; right, 6 lines sampled from the distribution.
After observing a set of n items $(x_1, y_1), \ldots, (x_n, y_n)$ (circles in right figure).



Left, posterior distribution $p(w_0, w_1|x_i, y_i, i = 1, \ldots, n)$; right, 6 lines sampled from the distribution.

- As the number of observed items increases, the distribution of parameters w_0, w_1 tends to concentrate (variance decreases to 0) around a mean point a_0, a_1.
- As a consequence, sampled lines are concentrated around $y = a_0 + a_1 x$.

Approaches to prediction in linear regression

Classical

A value $\mathbf{w}_{LS}$ for $\mathbf{w}$ is learned through a point estimate, performed by minimizing a quadratic cost function, or equivalently by maximizing likelihood (ML) under the hypothesis of gaussian noise; regularization can be applied to modify the cost function to limit overfitting.

Given any $\mathbf{x}$, the obtained value $\mathbf{w}_{LS}$ is used to predict the corresponding t as $t = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}_{LS}$,.

Bayesian point estimation

The posterior distribution $p(\mathbf{w}|\Phi, \mathbf{t}, \alpha, \beta)$ is derived and a point estimate is performed from it, computing the mode $\mathbf{w}_{MAP}$ of the distribution (MAP).

This is equivalent to the classical approach, as $\mathbf{w}_{MAP}$ corresponds to $\mathbf{w}_{LS}$ if $\lambda = \frac{\alpha}{\beta}$. The prediction, for an element $\mathbf{x}$, is a gaussian distribution $\mathcal{N}(t|\boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}_{MAP}, \beta)$ for t, with mean $\boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}_{MAP}$ and variance β^{-1}.

The distribution is not derived directly from the posterior $p(\mathbf{w}|\Phi, \mathbf{t}, \alpha, \beta)$: it is built, instead, as a gaussian with mean depending from the expectation of the posterior, and variance given by the assumed noise.

Fully bayesian

The real interest is not in estimating $\mathbf{w}$ or its distribution $p(\mathbf{w}|\Phi, \mathbf{t}, \alpha, \beta)$, but in deriving the predictive distribution $p(t|\mathbf{x})$. This can be done through expectation of the probability $p(t|\mathbf{x}, \mathbf{w}, \beta)$ predicted by a model instance wrt model instance distribution $p(\mathbf{w}|\Phi, \mathbf{t}, \alpha, \beta)$, that is

$$p(t|\mathbf{x}, \mathbf{t}, \Phi, \alpha, \beta) = \int p(t|\mathbf{x}, \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta) d\mathbf{w}$$

The distribution $p(t|\mathbf{x}, \mathbf{w}, \beta)$ is assumed gaussian, and $p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta)$ is gaussian by the assumption that the likelihood $p(\mathbf{t}|\mathbf{w}, \Phi, \beta)$ and the prior $p(\mathbf{w}|\alpha)$ are gaussian themselves and by their being conjugate

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \beta)$$
$$p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\beta \mathbf{S}_N \Phi^T \mathbf{t}, \mathbf{S}_N)$$

where $\mathbf{S}_N = (\alpha \mathbf{I} + \beta \Phi^T \Phi)^{-1}$

Under such hypothesis, the predictive distribution is gaussian

$$p(t|\mathbf{x}, \mathbf{t}, \Phi, \alpha, \beta) = \mathcal{N}(t|m(\mathbf{x}), \sigma^2(\mathbf{x}))$$

with mean

$$m(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}_N \Phi^T \mathbf{t}$$

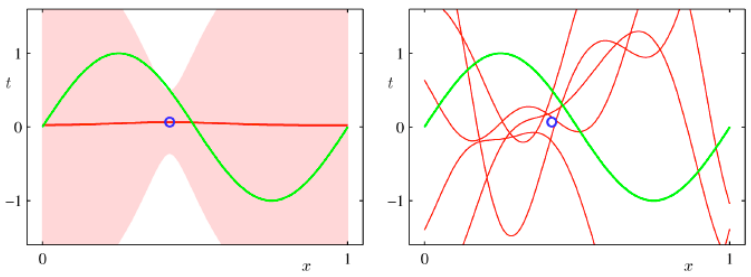
and variance

$$\sigma^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x})$$

- $\frac{1}{\beta}$ is a measure of the uncertainty intrinsic to observed data (noise)
- $\boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x})$ is the uncertainty wrt the values derived for the parameters $\mathbf{w}$
- as the noise distribution and the distribution of $\mathbf{w}$ are independent gaussians, their variances add

---

- predictive distribution for $y = \sin 2\pi x$, applying a model with 9 gaussian base functions and training sets of 1, 2, 4, 25 items, respectively
- left: items in training sets (sampled uniformly, with added gaussian noise); expectation of the predictive distribution (red), as function of x; variance of such distribution (pink shade within 1 standard deviation from mean), as a function of x
- right: items in training sets, 5 possible curves approximating $y = \sin 2\pi x$, derived through sampling from the posterior distribution $p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta)$

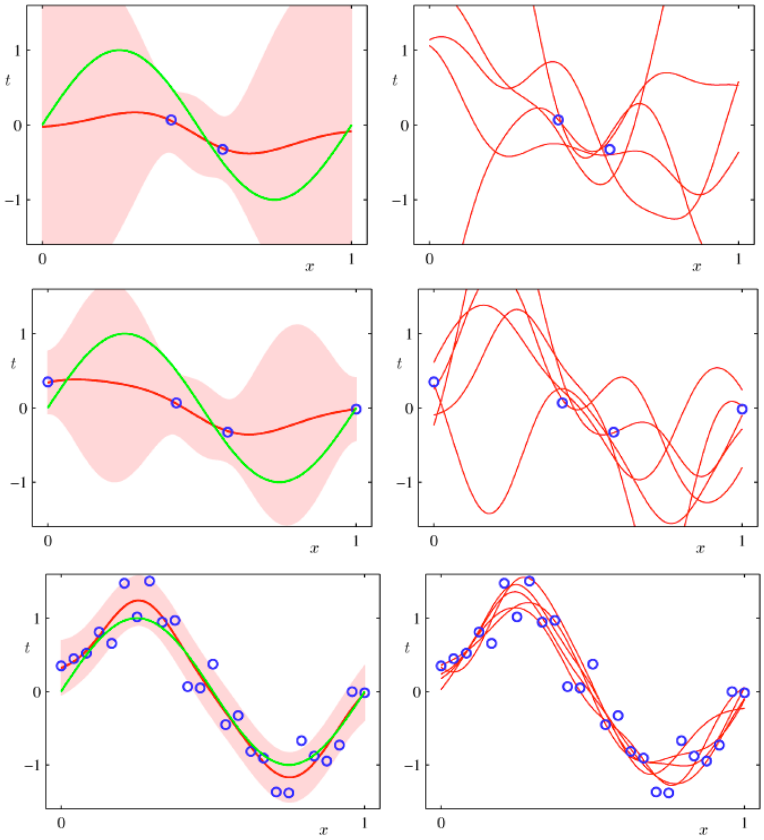


$n = 1$

$n = 2$

$n = 4$

$n = 25$



Fully bayesian regression and hyperparameter marginalization

In a fully bayesian approach, the hyper-parameters α, β are also marginalized

$$p(t|\mathbf{x}, \mathbf{t}, \Phi) = \int p(t|\mathbf{x}, \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta) p(\alpha, \beta|\mathbf{t}, \Phi) \; d\mathbf{w} \; d\alpha \; d\beta$$

where, as seen before,

- $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \beta)$
- $p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$, with $\mathbf{S}_N = (\alpha \mathbf{I} + \beta \Phi^T \Phi)^{-1}$ e $\mathbf{m}_N = \beta \mathbf{S}_N \Phi^T \mathbf{t}$

this marginalization wrt $\mathbf{w}, \alpha, \beta$ is analytically intractable we may consider approximation methods, that we do not introduce here.