# Linear regression

Giorgio Gambosi

## 1 Basic definitions

A linear model is a linear combination of the features $x_1, \ldots, x_d$ of the input element $\mathbf{x}$

$$y(\mathbf{x}, \mathbf{w}) \triangleq \sum_{j=1}^{d} w_j x_j + w_0$$

The values of a set of $d+1$ coefficients $w_0, w_1, \ldots, w_d$ completely defines the model. Clearly, it is a linear function of both the parameters $\mathbf{w}$ and the features $\mathbf{x}$.

In vector form, the model can be defined as

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \overline{\mathbf{x}}$$

with

$$\underset{1\times 1}{y(\mathbf{x}, \mathbf{w})} = \underset{1\times(d+1)}{\mathbf{w}^T} \times \underset{(d+1)\times 1}{\overline{\mathbf{x}}}$$

where $\overline{\mathbf{x}} = (1, x_1, \ldots, x_d)^T = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{pmatrix}$ and $\mathbf{w}^T = (w_0, w_1, \ldots, w_d)$

An extension of this definition can be obtained by introducing a set of basis functions $\phi_1, \ldots, \phi_m$ defined on $\mathbb{R}^d$ and considering a linear combination of the results $\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})$ obtained by applying the basis function to the item $\mathbf{x}$ considered.

$$y(\mathbf{x}, \mathbf{w}) \triangleq \sum_{j=1}^{m} w_j \phi_j(\mathbf{x})$$

More compactly, we may consider the vector of functions

$$\boldsymbol{\phi} \triangleq \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_m \end{pmatrix}$$

which maps any point in $\mathbf{x} \in \mathbb{R}^d$ to a point

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \vdots \\ \phi_m(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^m$$

That is, applying $\boldsymbol{\phi}$ maps the problem from a $d$-dimensional to an $m$-dimensional space (usually with $m > d$).

The introduction of $\boldsymbol{\phi}$ allows to define the prediction function in vector form as

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

Finding a number $m$ of basis functions and their definitions in order to improve the quality of the predictions performed by the model (modeled in terms of some quality measure) is the objective of the feature engineering process.

Examples of basis functions types on a single feature are:

- Polynomial

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

- Gaussian

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

- Sigmoid

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

- Hyperbolic tangent

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

This schemes can be easily extended to multiple features. Notice that gaussian, sigmoid and hyperbolic tangent functions are local, in the sense that they are not (almost) constant only in a limited interval of values.

In summary, given the feature matrix $\mathbf{X}$ in the training set $\mathcal{T}$,

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

applying a set of basis functions $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m$ to the elements $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbb{R}^d$ results into a new set of feature values, represented by the following $n \times m$ matrix

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

A special case here is when $m = d+1$ and $\phi_i(\mathbf{x}) = x_i$ for $i = 1, \ldots, d$ and $\phi_{d+1}(\mathbf{x}) = 1$, which clearly corresponds to the extension of $\mathbf{x}$ to $\overline{\mathbf{x}}$ showed above.

Given a set of values $\tilde{\mathbf{w}}$ of the coefficients, the set $\mathbf{Y}(\mathbf{X}, \tilde{\mathbf{w}})$ values predicted for all elements in the training set can be computed as

$$\mathbf{Y}(\mathbf{X}, \tilde{\mathbf{w}}) = \Phi\tilde{\mathbf{w}}$$

with

$$\underset{n\times 1}{\mathbf{Y}(\mathbf{X}, \tilde{\mathbf{w}})} = \underset{n\times m}{\Phi} \times \underset{m\times 1}{\tilde{\mathbf{w}}}$$

## 2 Parameter learning

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 $\mathbf{X}$, $\mathbf{t}$ and $\mathbf{w}$).

A most widely adopted error function is the quadratic loss $(y_i - t_i)^2$, which results into the least squares 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.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y(\mathbf{x}_i, \mathbf{w}) - t_i)^2 = \frac{1}{2}\sum_{i=1}^{n}\left(\sum_{j=1}^{m} w_j\phi_j(\mathbf{x}_i) - t_i\right)^2$$

or, in matrix form,

$$E(\mathbf{w}) = \frac{1}{2}(\Phi\mathbf{w} - \mathbf{t})^T(\Phi\mathbf{w} - \mathbf{t})$$

with

$$\underset{1\times 1}{E(\mathbf{w})} = \frac{1}{2} \times \underset{1\times n}{(\Phi\mathbf{w} - \mathbf{t})^T} \times \underset{n\times 1}{(\Phi\mathbf{w} - \mathbf{t})}$$

In order to minimize $E(\mathbf{w})$, set its gradient to $\mathbf{0}$. That is, set

$$\frac{\partial E(\mathbf{w})}{\partial w_k} = 0 \qquad\qquad k = 1, \ldots, m$$

We observe that

- the quadratic loss $(y - t)^2$ is a convex function, which implies that only one (global) minimum is defined
- $E(\mathbf{w})$ is convex itself, being the sum of $n$ convex functions $(y(x_k, \mathbf{w}) - t_k)^2$
- in particular, $E(\mathbf{w})$ quadratic implies that its derivative is linear, hence that it is zero for a unique value $\mathbf{w}^*$
- the resulting prediction function is $y(x, \mathbf{w}^*)$

The set of equations obtained by setting $\nabla_{\mathbf{w}}E(\mathbf{w}) = \mathbf{0}$ results into a linear system :

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

In matrix-vector form,

$$\nabla_{\mathbf{w}}E(\mathbf{w}) = \Phi^T(\Phi\mathbf{w} - \mathbf{t}) = \mathbf{0}$$

Each of the $m$ equations is linear w.r.t. each coefficient in $\mathbf{w}$. A linear system results, with $m$ equations and $m$ unknowns $w_1, \ldots, w_m$, which, in general and with the exceptions of degenerate cases, has precisely one solution.

In this case, the solution is defined in closed form by the normal equations for least squares

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

where

$$\underset{m\times 1}{\mathbf{w}^*} = \left(\underset{m\times n}{\Phi^T} \times \underset{n\times m}{\Phi}\right)^{-1} \times \underset{m\times n}{\Phi^T} \times \underset{n\times 1}{\mathbf{t}} = \underset{m\times m}{(\Phi^T\Phi)^{-1}} \times \underset{m\times n}{\Phi^T} \times \underset{n\times 1}{\mathbf{t}}$$

### 2.1 Gradient descent

The minimum of $E(\mathbf{w})$ can be computed numerically, by means of gradient descent methods

- Start from an initial assignment $\mathbf{w}^{(0)} = (w_1^{(0)}, w_2^{(0)}, \dots, w_m^{(0)})$, with a corresponding error

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

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

$$\mathbf{w}^{(i)} = \mathbf{w}^{(i-1)} - \eta\nabla_{\mathbf{w}}E(\mathbf{w})\Big|_{\mathbf{w}^{(i-1)}}$$

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

$$\begin{aligned} w_k^{(i)} = w_k^{(i-1)} - \eta\frac{\partial E(\mathbf{w})}{\partial w_k}\Bigg|_{\mathbf{w}^{(i-1)}} &= w_k^{(i-1)} - \eta\sum_{i=1}^{n}(y(\mathbf{x}_i, \mathbf{w}^{(i-1)}) - t_i)\phi_k(\mathbf{x}_i) \\ &= w_k^{(i-1)} - \eta\sum_{i=1}^{n}\left(\sum_{j=1}^{m} w_j^{(i-1)}\phi_j(\mathbf{x}_i) - t_i\right)\phi_k(\mathbf{x}_i) \end{aligned}$$

In matrix notation:

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

## 3 Example

Assume a set of $n$ observations of two variables $x, t \in \mathbb{R}$: $(x_1, t_1), \dots, (x_n, t_n))$ 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 $\mathcal{T}$ is a pair of vectors $\mathbf{x} = (x_1, \dots, x_n)^T$ and $\mathbf{t} = (t_1, \dots, t_n)^T$, and we assume that pairs $x_i, t_i$ are related through an unknown rule (function)

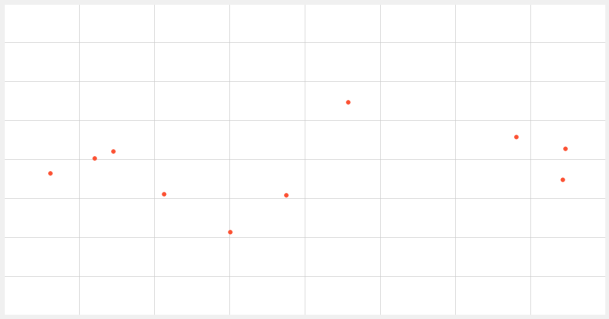


Figure 1: Observed dataset

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 having mean 0 and variance $\sigma^2$. Hence, $t_i = \sin(2\pi x_i) + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

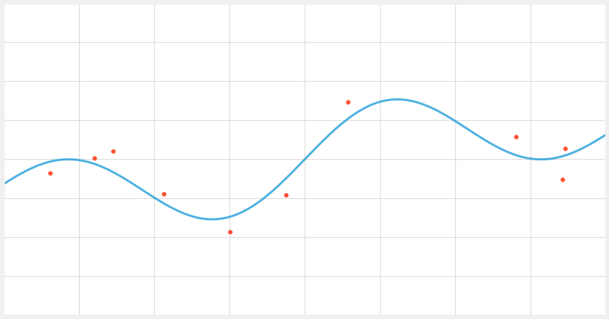


Figure 2: Observed dataset with underlying function

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

The approach we consider here is approximating the unknown function through a polynomial of suitable degree $m > 0$

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

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

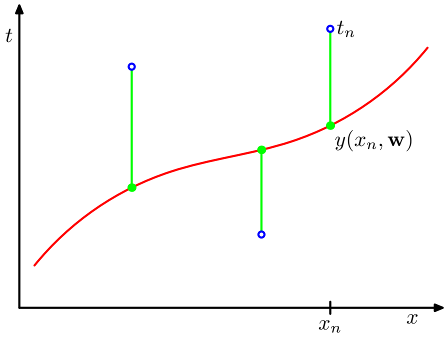
This corresponds to applying $m + 1$ basis functions $\phi_j(x) = x^j$, for $j = 0, \ldots, m$, to the unique feature $x$

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

observe that when basis functions are applied, $y(x, \mathbf{w})$ is a nonlinear function of $x$, but it is still a linear function (model) of $\mathbf{w}$.

In this case, the error function is defined as

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} (y(x_i, \mathbf{w}) - t_i)^2 = \frac{1}{2}\sum_{i=1}^{n}\left(t_i - \sum_{j=0}^{m} w_j x_i^j\right)^2$$



The solution in closed form is given by

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

where

$$\overline{\mathbf{X}} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{pmatrix}$$

### 3.1 Polynomial degree

Selecting the degree of the polynomial is a case of model selection: assigning a value to $m$ determines the precise model to be used, since the choice of $m$ implies the number of coefficients to be estimated.

Clearly, increasing $m$ allows to better approximate the items in the training set, decreasing the overall error. As a limit, if $m+1=n$ the model allows to obtain a null error on the training set itself.

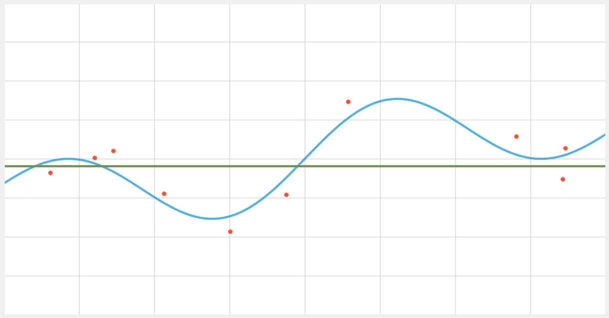


Figure 3: Approximation with $m=0$

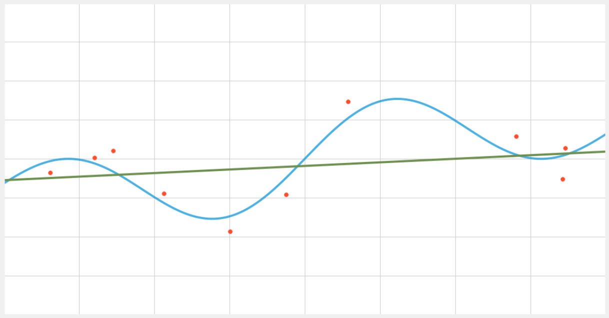


Figure 4: Approximation with $m=1$

This may result in overfitting if the function $y(x,\mathbf{w})$ derived from items in the training set, fails to provide good predictions for other items. That is, if fails to provide a suitable generalization to all items in the whole domain.

In general, if $y(x,\mathbf{w})$ is derived as a too much accurate depiction of the training set, it in fact results into an unsuitable generalization to items not in the training set.

### 3.2 Evaluation of the generalization

- Assume a test set $\mathcal{T}_{test}$ of 100 new items, generated by uniformly sampling $x$ in $[0,1]$ and $\varepsilon$ from $\mathcal{N}(0,\sigma^2)$, and computing $t=\sin 2\pi x+\varepsilon$

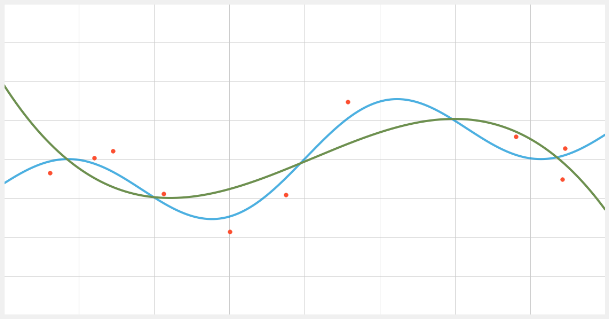


Figure 5: Approximation with $m = 3$

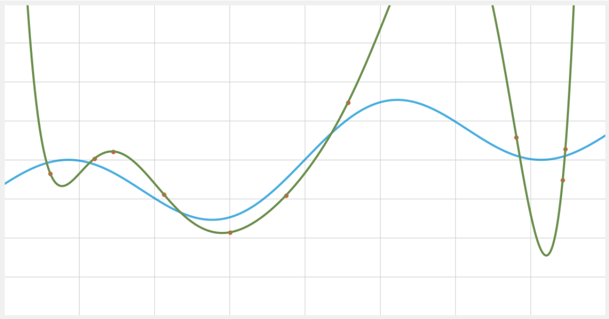


Figure 6: Approximation with $m = n - 1 = 9$

- For each $m \in \{0, \ldots, n\}$:
  - derive $\mathbf{w}^*$ from the training set $\mathcal{T}_{train}$
  - compute the root mean square error $E_{RMS}(\mathbf{w}^*, \mathcal{T}_{test})$ on the test set, defined as

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

- compare the RMS error for different values of $m$. A smaller value of $E_{RMS}(\mathbf{w}^*, \mathbf{X}_{test})$ denotes a better generalization

In the case considered here, $n = 9$. In figure 7, a typical 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 represent the characteristics of the data set. Next, the error increases, since the model becomes too dependent from the training set: the noise component in $t$ is also represented.

An immediate consequence of all this, is that, for a given model complexity (such as the degree in our example), overfitting decreases as the dimension of the dataset increases.

The larger the dataset, the higher the acceptable complexity of the model.

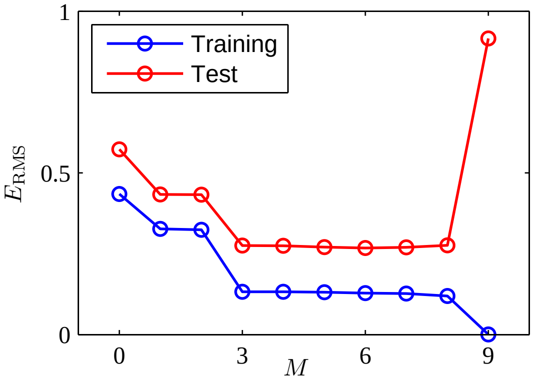
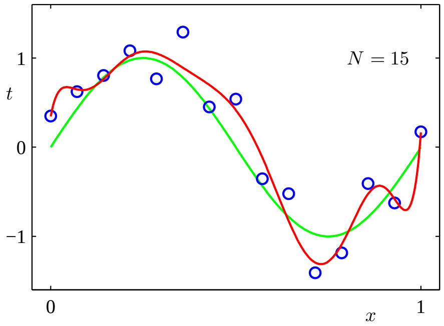


Figure 7:



### 3.3 Limiting the complexity of the model

Model complexity can be limited through regularization

- A regularization term is introduced in the cost function

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

  $E_D(\mathbf{w})$ dependent from the dataset (and the parameters), $E_W(\mathbf{w})$ dependent from the parameters alone.
- The regularization coefficient controls the relative importance of the two terms.

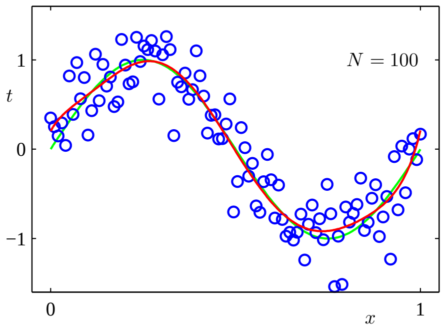
Regularized least squares

- Simple form

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

- Sum-of squares cost function: ridge regression

  $$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i))^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}(\Phi\mathbf{w}-\mathbf{y})^T(\Phi\mathbf{w}-\mathbf{y}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$



with solution

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

### Regularization

- A more general form

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} (t_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i))^2 + \frac{\lambda}{2} \sum_{j=1}^{m} |w_j|^q$$

- The case $q = 1$ is denoted as lasso: sparse models are favored

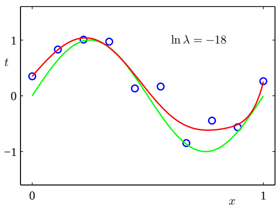
### Example: polynomial regression

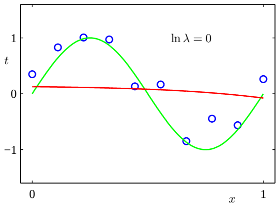
Use of regularization to limit complexity and overfitting.

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

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

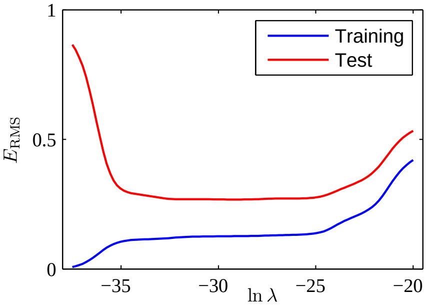
Dependance from the value of the hyperparameter $\lambda$.





### Example: polynomial regression

Plot of the error w.r.t $\lambda$, ridge regression.



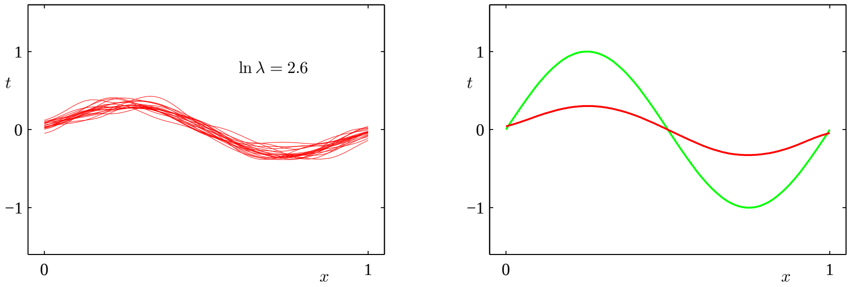
- Small $\lambda$: overfitting. Small error on the training set, large error on the test set.
- Large $\lambda$: the effect of data values decreases. Large error on both test and training sets.
- Intermediate $\lambda$. Intermediate error on training set, small error on test set.

### Example: polynomial regression

- 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 basis functions $\phi_1(x), \ldots, \phi_m(x)$, from each training set $\mathcal{T}_i$ a prediction function $y_i(x)$ is derived by minimizing the regularized cost function

$$E(\mathbf{w}) = \frac{1}{2}(\Phi\mathbf{w} - \mathbf{t})^T(\Phi\mathbf{w} - \mathbf{t}) + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

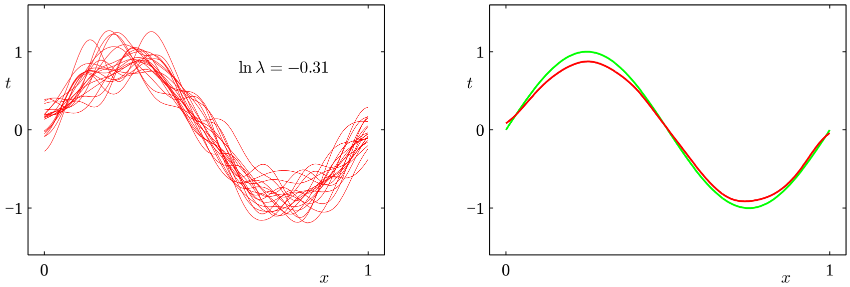
### Example: polynomial regression



Left, a possible plot of prediction functions $y_i(\mathbf{x})$ $(i = 1, \ldots, 100)$, 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 $y = \sin 2\pi x$.

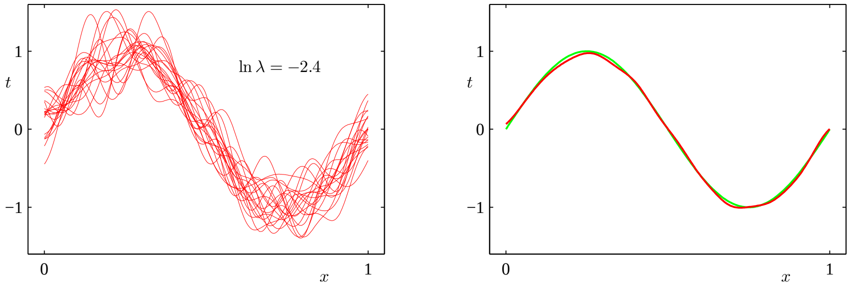
The prediction functions $y_i(\mathbf{x})$ do not differ much between them (small variance), but their expectation is a bad approximation of the unknown function (large bias).

### Example: polynomial regression



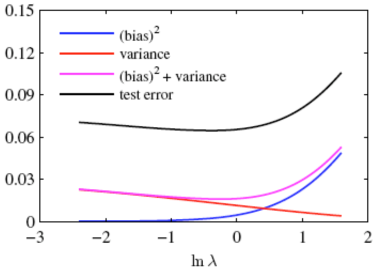
Plot of the prediction functions obtained with $\ln\lambda = -0.31$.

### Example: polynomial regression



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

### Example: polynomial regression

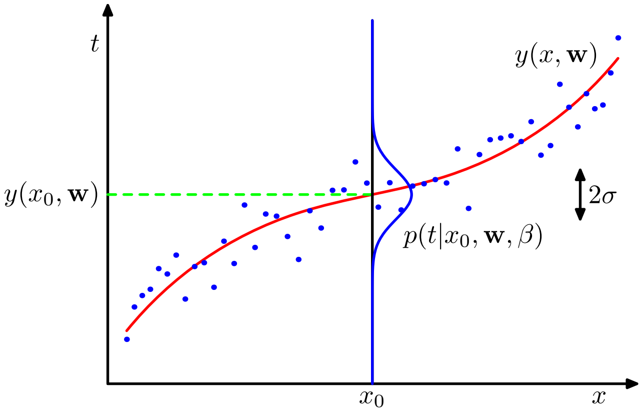


- Plot of $(\text{bias})^2$, variance and their sum as functions of $\lambda$: las $\lambda$ increases, bias increases and varinace decreases. Their sum has a minimum in correspondance to the optimal value of $\lambda$.
- The term $E_{\mathbf{x}}[\sigma^2_{y|\mathbf{x}}]$ shows an inherent limit to the approximability of $y = \sin 2\pi x$.

### Probabilistic model for regression

Assume that, given an item $\mathbf{x}$, the corresponding unknown target $t$ is normally distributed around the value returned by the model $\mathbf{w}^T\overline{\mathbf{x}}$, with a given variance $\sigma^2 = \beta^{-1}$:

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



Probabilistic model for regression

An estimate of both $\beta_{ML}$ and the coefficients $\mathbf{w}_{ML}$ can be performed on the basis of the likelihood w.r.t. the assumed normal distribution:

$$L(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{n} \mathcal{N}(t_i|y(\mathbf{x}_i, \mathbf{w}), \beta^{-1})$$

Parameters $\mathbf{w}$ and $\beta$ can be estimated as the values which maximize the data likelihood, or its logarithm

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

which results into

$$\begin{aligned} l(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) &= \sum_{i=1}^{n} \log \left( \frac{\sqrt{\beta}}{\sqrt{2\pi}} e^{-\frac{\beta}{2}(t_i - y(\mathbf{x}_i, \mathbf{w}))^2} \right) \\ &= -\sum_{i=1}^{n} \frac{\beta}{2} (t_i - y(\mathbf{x}_i, \mathbf{w}))^2 + \frac{n}{2} \log \beta - \frac{n}{2} \log(2\pi) \\ &= -\frac{\beta}{2} \sum_{i=1}^{n} (t_i - y(\mathbf{x}_i, \mathbf{w}))^2 + \frac{n}{2} \log \beta + \text{cost} \end{aligned}$$

Probabilistic model for regression

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

$$-\frac{1}{2} \sum_{i=1}^{n} (t_i - y(\mathbf{x}_i, \mathbf{w}))^2$$

this is equivalent to minimizing the least squares sum.

The maximization w.r.t. the precision $\beta$ is done by setting to 0 the corresponding derivative

$$\frac{\partial l(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta)}{\partial \beta} = -\frac{1}{2} \sum_{i=1}^{n} (t_i - y(\mathbf{x}_i, \mathbf{w}))^2 + \frac{n}{2\beta}$$

which results into

$$\beta_{ML}^{-1} = \frac{1}{n} \sum_{i=1}^{n} (t_i - y(\mathbf{x}_i, \mathbf{w}))^2$$

### Probabilistic model for regression

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 $\mathbf{x}$.

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

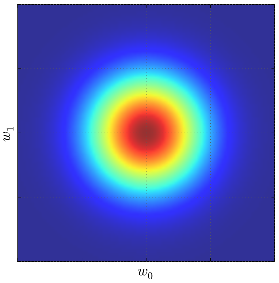
### Probabilistic model for regression

- In the maximum likelihood framework parameters are considered as (unknown) values to determine with the best possible precision (frequentist approach).
- Applying maximum likelihood to determine the values of model parameters is prone to overfitting: need of a regularization term $\mathcal{E}(\mathbf{w})$.
- In order control model complexity, a bayesian approach assumes a prior distribution of parameter values.
- The bayesian framework looks at parameters as random variables, whose probability distribution has to be derived.

### Probabilistic model for regression

Prior distribution of parameters: gaussian with mean $\mathbf{0}$ and diagonal covariance matrix with variance equal to the inverse of hyperparameter $\alpha$

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



### Why a gaussian prior?

Posterior proportional to prior times likelihood: likelihood is gaussian (gaussian noise).

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

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

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

### Why a gaussian prior?

In general, conjugate of gaussian is gaussian: choosing a gaussian prior distribution of $\mathbf{w}$

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

results into a gaussian posterior distribution

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

where

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

### Why a gaussian prior?

Here, we have

$$p(\mathbf{w})=\mathcal{N}(\mathbf{w}|\mathbf{0},\alpha^{-1}\mathbf{I}) \qquad\qquad p(\mathbf{t}|\mathbf{w},\Phi)=\mathcal{N}(\mathbf{t}|\mathbf{w}^T\Phi,\beta^{-1}\mathbf{I})$$

and the posterior distribution is gaussian

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

with

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

### Why a gaussian prior?

Note that as $\alpha\to 0$ the prior tends to have infinite variance, and we have minimum information on $\mathbf{w}$ before the training set is considered. In this case,

$$\mathbf{m}_p\to(\Phi^T\Phi)^{-1}(\Phi^T\mathbf{t})$$

that is $\mathbf{w}_{ML}$, the ML estimation of $\mathbf{w}$.

### Maximum a Posteriori

- Given the posterior distribution $p(\mathbf{w}|\Phi,\mathbf{t},\alpha,\beta)$, we may derive the value of $\mathbf{w}_{MAP}$ 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 $\mathbf{w}$

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

  that is,

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

### Maximum a Posteriori

In this case

$$\begin{aligned}p(\mathbf{w}|\mathbf{X},\mathbf{t};\alpha,\beta) &\propto p(\mathbf{t}|\mathbf{X},\mathbf{w};\beta)p(\mathbf{w}|\alpha)\\ &=\prod_{i=1}^n\left(\frac{\sqrt{\beta}}{\sqrt{2\pi}}e^{-\frac{\beta}{2}(t_i-y(\mathbf{x}_i,\mathbf{w}))^2}\right)\left(\frac{\alpha}{2\pi}\right)^{\frac{M+1}{2}}e^{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}}\end{aligned}$$

The maximization of the posterior distribution (MAP) is equivalent to the maximization of the corresponding logarithm

$$-\frac{\beta}{2}\sum_{i=1}^n(t_i-y(\mathbf{x}_i,\mathbf{w}))^2+\frac{n}{2}\log\beta-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}+\frac{m+1}{2}\log\frac{\alpha}{2\pi}+\text{cost}$$

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

$$\frac{\beta}{2}\sum_{i=1}^{n}(t_i - y(\mathbf{x}_i,\mathbf{w}))^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} = \beta\left(\frac{1}{2}\sum_{i=1}^{n}(t_i - y(\mathbf{x}_i,\mathbf{w}))^2 + \frac{\alpha}{2\beta}||\mathbf{w}||^2\right)$$

The ratio $\frac{\alpha}{\beta}$ corresponds to a regularization hyperparameter.

### Maximum a Posteriori

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

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

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

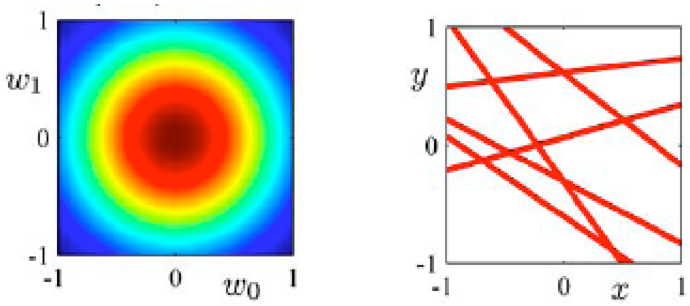
### Sequential learning

- The posterior after observing $T_1$ can be used as a prior for the next training set acquired.
- In general, for a sequence $T_1,\ldots,T_n$ of training sets,

$$\begin{aligned}
&p(\mathbf{w}|T_1,\ldots T_n) \propto p(T_n|\mathbf{w})p(\mathbf{w}|T_1,\ldots T_{n-1})\\
&p(\mathbf{w}|T_1,\ldots T_{n-1}) \propto p(T_{n-1}|\mathbf{w})p(\mathbf{w}|T_1,\ldots T_{n-2})\\
&\ldots\\
&p(\mathbf{w}|T_1) \propto p(T_1|\mathbf{w})p(\mathbf{w})
\end{aligned}$$

### Example

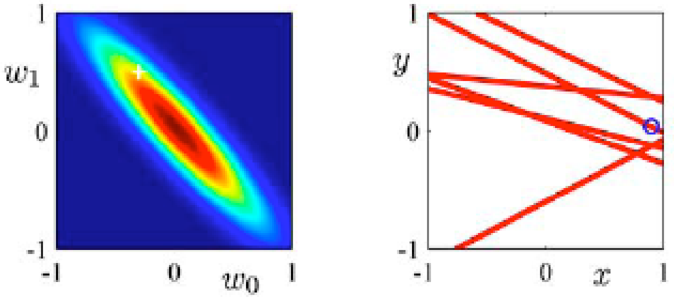
- Input variable $x$, target variable $t$, linear regression $y(x,w_0,w_1) = w_0 + w_1 x$.
- Dataset generated by applying function $y = a_0 + a_1 x$ (with $a_0 = -0.3$, $a_1 = 0.5$) to values uniformly sampled in $[-1,1]$, with added gaussian noise ($\mu = 0$, $\sigma = 0.2$).
- Assume the prior distribution $p(w_0,w_1)$ 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.

### Example

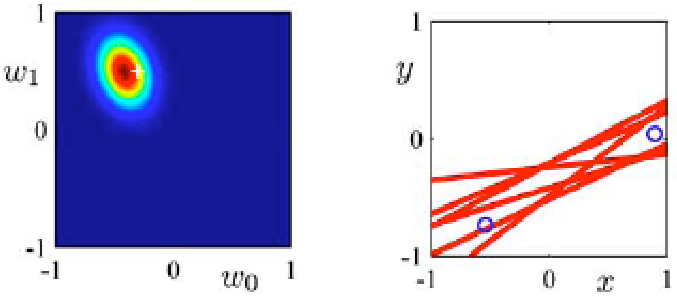
After observing item $(x_1, y_1)$ (circle in right figure).



Left, posterior distribution $p(w_0, w_1|x_1, y_1)$; right, 6 lines sampled from the distribution.

Example

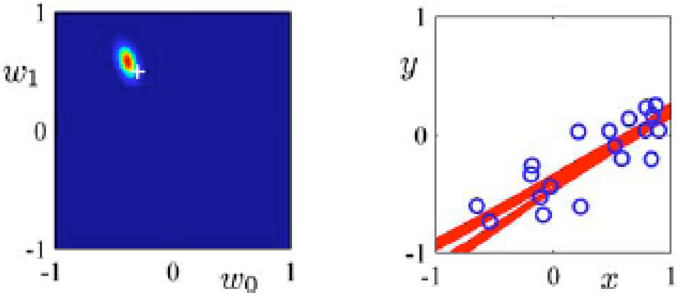
After observing items $(x_1, y_1), (x_2, y_2)$ (circles in right figure).



Left, posterior distribution $p(w_0, w_1|x_1, y_1, x_2, y_2)$; right, 6 lines sampled from the distribution.

Example

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.

Example

- 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 $y = \overline{\mathbf{x}}^T \mathbf{w}_{LS}$, where $\overline{\mathbf{x}}^T = (1, \mathbf{x})^T$, or, in general, as $y = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}_{LS}$

Approaches to prediction in linear regression

### Bayesian point estimation

- The posterior distribution $p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta)$ is derived and a point estimate is performed from it, computing the mode $\mathbf{w}_{MAP}$ of the distribution (MAP)
- Equivalent to the classical approach, as $\mathbf{w}_{MAP}$ corresponds to $\mathbf{w}_{LS}$ if $\lambda = \dfrac{\alpha}{\beta}$
- The prediction, for a value $\mathbf{x}$, is a gaussian distribution $p(y|\boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}_{MAP}, \beta)$ for $y$, with mean $\boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}_{MAP}$ and variance $\beta^{-1}$
- The distribution is not derived directly from the posterior $p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta)$: it is built, instead, as a gaussian with mean depending from the expectation of the posterior, and variance given by the assumed noise.

Approaches to prediction in linear regression

### Fully bayesian

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

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

- $p(y|\mathbf{x}, \mathbf{w}, \beta)$ 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

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

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

Approaches to prediction in linear regression

### Fully bayesian

Under such hypothesis, $p(y|\mathbf{x})$ is gaussian

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

with mean

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

and variance

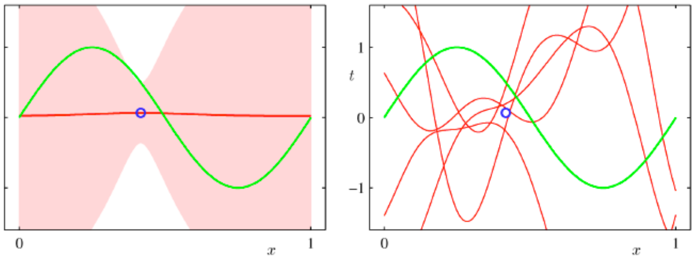
$$\sigma^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x})$$

- $\frac{1}{\beta}$ 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

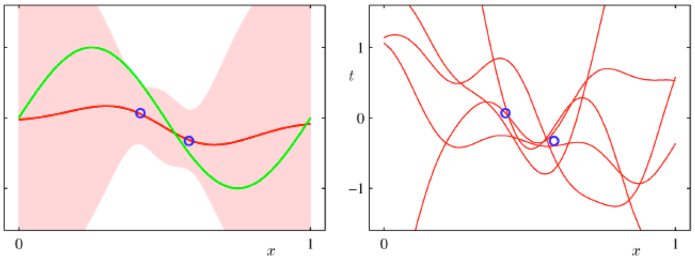
### Example

- predictive distribution for $y = \sin 2\pi x$, applying a model with 9 gaussian basis 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 $p(\mathbf{w}|\mathbf{t}, \Phi, \alpha, \beta)$

### Example

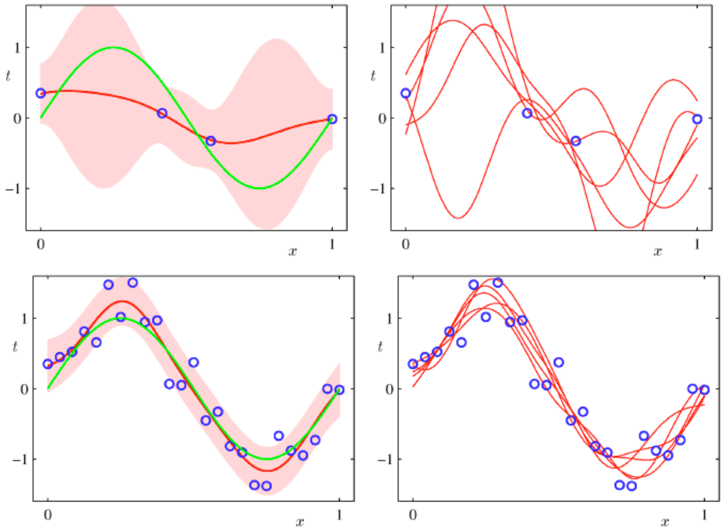


$n = 1$



$n = 2$

### Example



$n = 4$

$n = 25$

### Fully bayesian regression and hyperparameter marginalization

- In a fully bayesian approach, also the hyper-parameters $\alpha, \beta$ are 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