Linear regression

Course of Machine Learning
Master Degree in Computer Science
University of Rome “Tor Vergata”
a.a. 2021-2022

Giorgio Gambosi

Linear models

- Linear combination of input features

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_dx_d$$

  with $\mathbf{x} = (x_1, \ldots, x_d)$

- Linear function of parameters $\mathbf{w}$
- Linear function of features $\mathbf{x}$

More compactly,

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

where $\overline{\mathbf{x}} = (1, x_1, \ldots, x_d)$

Base functions

- Extension to linear combination of *base functions* $\phi_1, \ldots, \phi_m$ defined on $\mathbb{R}^d$

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

- 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 problem is mapped from a d-dimensional to an m-dimensional space (usually with $m > d$)

Base functions

- Many types:
  - Polynomial (global functions)

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

  - Gaussian (local)

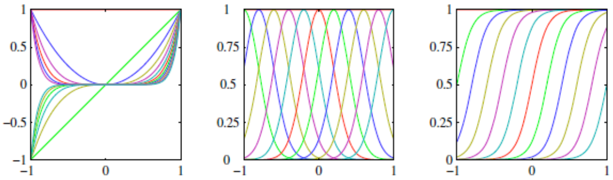
$$\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$$

- 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)

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



Base functions

Observe that a set of items (extended by 1 values)

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

is transformed into

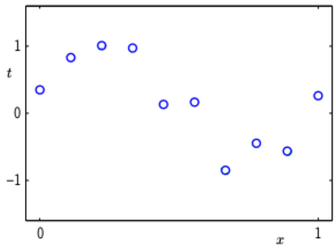
$$\Phi = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_m(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_n) & \phi_2(\mathbf{x}_n) & \cdots & \phi_m(\mathbf{x}_n) \end{pmatrix}$$

Example: polynomial regression

Problem

- A set of n observations of two variables $x, t \in \mathbf{R}$: $(x_1, t_1), \ldots, (x_n, t_n))$ 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)

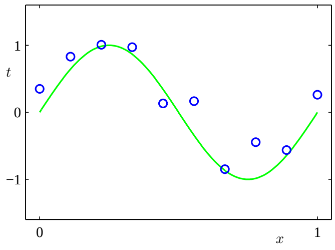
Example of a training set.



Example: polynomial regression

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 $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.



Purpose

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.

Example: polynomial regression

Approach

Let us approximate the unknown function through a suitable polynomial of given 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.

Linear models

$y(x, \mathbf{w})$ is a nonlinear function of x, but is a linear function (model) of $\mathbf{w}$.

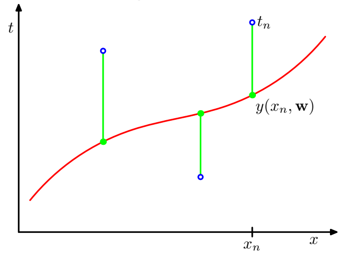
Parameter estimation

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

Regression loss

Least squares

A most widely adopted error function is *least quares*, i.e. the sum, for all items in the training set, of the (squared) difference between the value returned by the model and the target value.



$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} (y(x_i, \mathbf{w}) - t_i)^2 = \frac{1}{2}\sum_{i=1}^{n} \left(w_0 + w_1 x_i + w_2 x_i^2 + \ldots + w_m x_i^m - t_i\right)^2$$

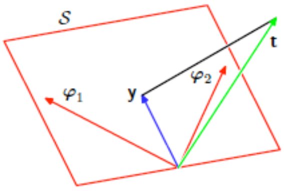
Regression loss

Error minimization

- To minimize $E(\mathbf{w})$, set its derivative w.r.t. $\mathbf{w}$ to $\mathbf{0}$
- $E(\mathbf{w})$ quadratic implies that its derivative is linear, hence that it is zero in one point w^*
- The resulting function is $y(x, \mathbf{w}^*)$

Least squares geometry

- $\mathbf{t} = (t_1, \ldots, t_n)^T$ is a vector in R^n
- Each basis function ϕ_j applied to $\mathbf{x}_1, \ldots, \mathbf{x}_n$ provides a vector $\varphi_j = (\phi_j(\mathbf{x}_1), \ldots, \phi_j(\mathbf{x}_n))^T \in \mathrm{R}^n$
- If $m < n$, vectors $\varphi_1, \ldots, \varphi_m$ define a subspace $\mathcal{S} \subset \mathrm{R}^n$ of dimension (at most) m
- $\mathbf{y} = (y(\mathbf{x}_1, \mathbf{w}), \ldots, y(\mathbf{x}_n, \mathbf{w}))^T$ is a vector in R^n: it can be represented as the linear combination $\mathbf{y} = \sum_{i=1}^m w_i \boldsymbol{\phi}(\mathbf{x}_i)$. Hence, it belongs to $\mathcal{S}$
- Given $\mathbf{t} \in \mathrm{R}^n$, $\mathbf{y} \in \mathrm{R}^n$ is the vector in subspace $\mathcal{S}$ at minimal squared distance from $\mathbf{t}$
- Given $\mathbf{t} \in \mathrm{R}^n$ and vectors $\phi_1, \ldots, \phi_m$, $\mathbf{w}_{ML}$ is such that $\mathbf{y}$ is the vector on $\mathcal{S}$ nearest to $\mathbf{t}$



Regression loss

Derivative with respect to w

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

$$\frac{\partial E(\mathbf{w})}{\partial w_0} = \sum_{i=1}^n (y(x_i, \mathbf{w}) - t_i) = 0$$

$$\frac{\partial E(\mathbf{w})}{\partial w_1} = \sum_{i=1}^n x_i \, (y(x_i, \mathbf{w}) - t_i) = 0$$

$$\ldots$$

$$\frac{\partial E(\mathbf{w})}{\partial w_m} = \sum_{i=1}^n x_i^m \, (y(x_i, \mathbf{w}) - t_i) = 0$$

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, which, in general and with the exceptions of degenerate cases, has precisely one solution.

Gradient descent

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

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

$$E(\mathbf{w}^{(0)}) = \frac{1}{2}\sum_{i=1}^{N}\Big(t_i - (\mathbf{w}^{(0)})^T\boldsymbol{\phi}(\mathbf{x}_i)\Big)^2$$

- Iteratively, 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 evaluated at $\mathbf{w}^{(i-1)}$
- At step i, $w_j^{(i-1)}$ is updated as follows:

$$w_j^{(i)} := w_j^{(i-1)} - \eta \frac{\partial E(\mathbf{w})}{\partial w_j}\bigg|_{\mathbf{w}^{(i-1)}}$$

Gradient descent

- In matrix notation:

$$\mathbf{w}^{(i)} := \mathbf{w}^{(i-1)} - \eta \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}\bigg|_{\mathbf{w}^{(i-1)}}$$

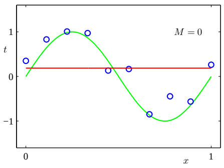
- By definition of $E(\mathbf{w})$:

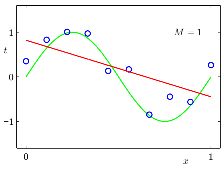
$$\mathbf{w}^{(i)} := \mathbf{w}^{(i-1)} - \eta(t_i - \mathbf{w}^{(i-1)}\boldsymbol{\phi}(\mathbf{x}_i))\boldsymbol{\phi}(\mathbf{x}_i)$$

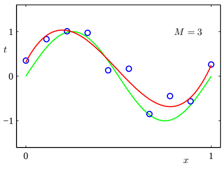
Example: fitting of polynomials

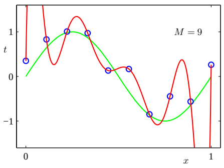
Polynomial degree

- Example of *model selection*: assigning a value to M determines the model to be used, the choice of M implies the number of coefficients 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*)









Example: fitting of polynomials

Overfitting

- The function $y(x, \mathbf{w})$ 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 $y(x, \mathbf{w})$ 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

Example: fitting of polynomials

Evaluation of the generalization

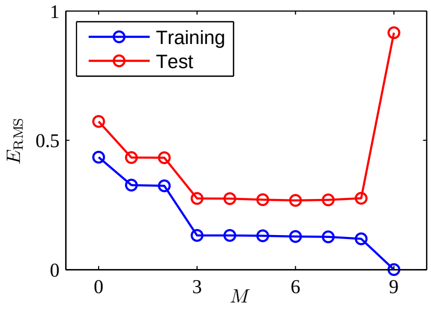
- Test set $\mathbf{X}_{test}$ of 100 new items, generated by uniformly sampling x in $[0, 1,]$ and ε from $\mathcal{N}(0, \sigma^2)$, and computing $t = \sin 2\pi x + \varepsilon$
- For each M:
  - derives $\mathbf{w}^*$ from the training set $\mathbf{X}_{train}$
  - compute the error $E(\mathbf{w}^*, \mathbf{X}_{test})$ on the test set, or the square root of its mean

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

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

Example: fitting of polynomials

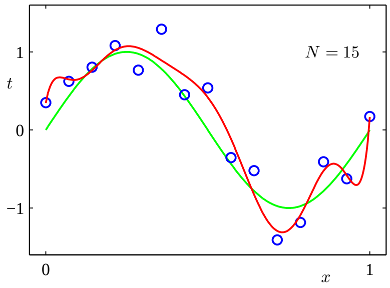
Plot of E_{RMS} w.r.t. M, on the training set and on the test set.

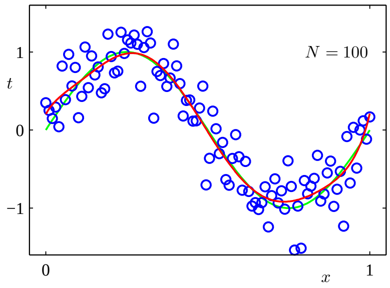


- As M increases, the error on the training set tends to 0.
- On the test set, the error initially decreases, since the higher complexity of the model allows to better represent the characteristics of the data set. Next, the error increases, since the model becomes too dependent from the training set: the noise component in t is too represented.

Example: fitting of polynomials

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.

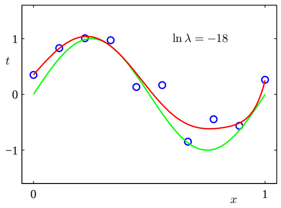
Example: fitting of polynomials

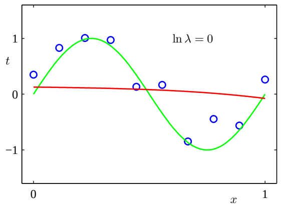
Use of *regularization* to limit complexity and overfitting.

- 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 λ.





Regularized least squares

- Regularization term 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*

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

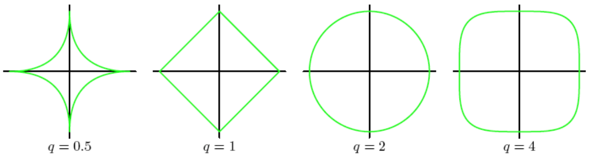
  with solution

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

Regularization

- A more general form

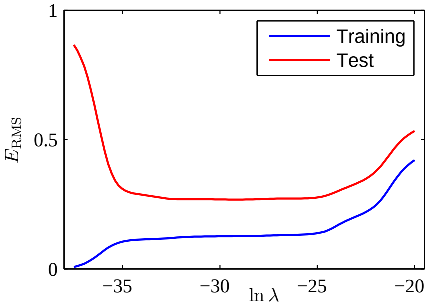
  $$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\{t_i - \mathbf{w}^T\phi(\mathbf{x}_i)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{m}|w_j|^q$$



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

Example: fitting of polynomials

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



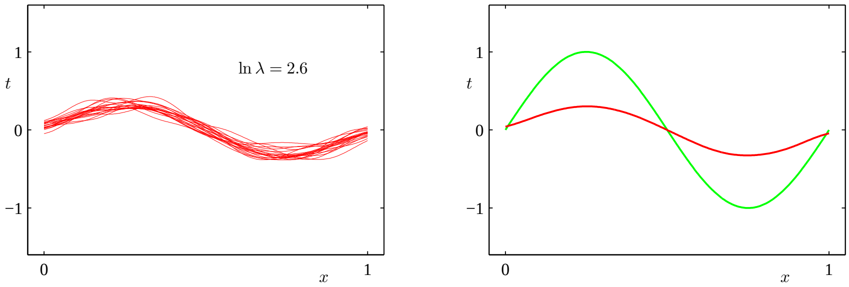
- Small λ: overfitting. Small error on the training set, large error on the test set.
- Large λ: the effect of data values decreases. Large error on both test and training sets.
- Intermediate λ. Intermediate error on training set, small error on test set.

Example: fitting of polynomials

- 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

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

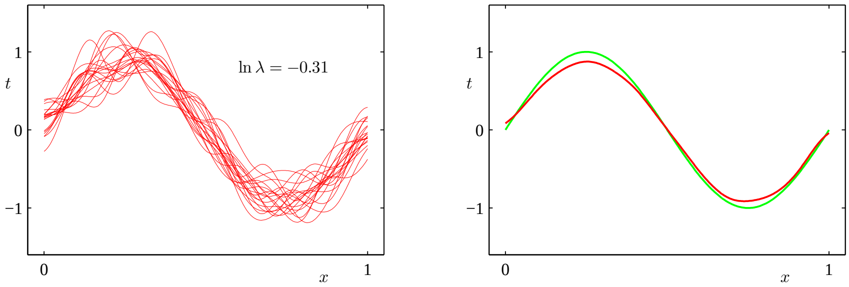
Example: fitting of polynomials



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 $y = \sin 2\pi x$.

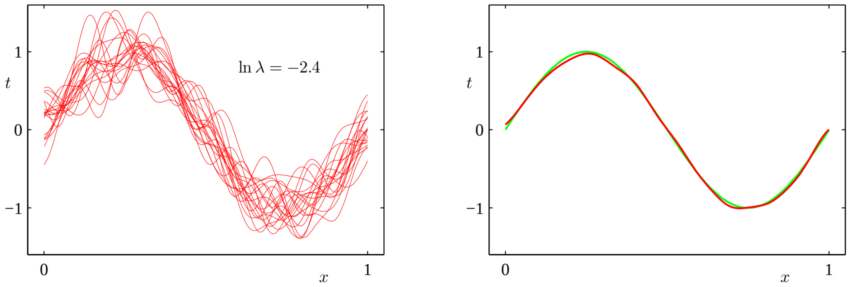
The prediction functions $y_i(\mathbf{x})$ do not differ much between them (small variance), but their expectation is a bad approximation of the unknown function (large bias).

Example: fitting of polynomials



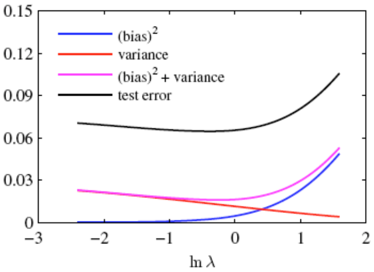
Plot of the prediction functions obtained with $\ln \lambda = -0.31$.

Example: fitting of polynomials



Plot of the prediction functions obtained with $\ln\lambda = -2.4$. As λ 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: fitting of polynomials

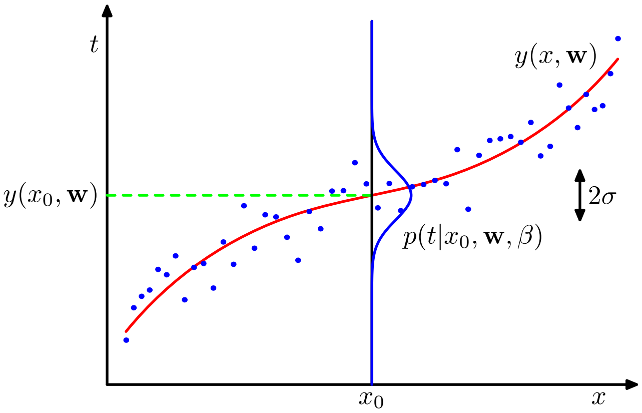


- Plot of (bias)^2, variance and their sum as unctions of λ: las λ increases, bias increases and varinace decreases. Their sum has a minimum in correspondance to the optimal value of λ.
- The term $E_{\mathbf{x}}[\sigma^2_{y|\mathbf{x}}]$ shows an inherent limit to the approximability of $y = \sin 2\pi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

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 β_{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 β 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. **w** is performed by determining a maximum w.r.t. **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* β 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 **x**, that is the (gaussian) distribution of the target value for a given item **x**.

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

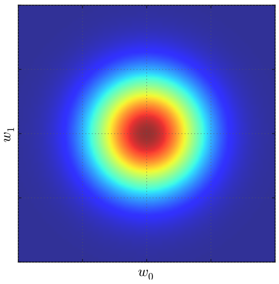
Probabilistic model for regression

- 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.
- An alternative framework (*bayesian*) looks at parameters as random variables, whose probability distribution has to be derived.

Probabilistic model for regression

Prior distribution of parameters: gaussian with mean **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?

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 $\mathbf{w}$

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

results into a gaussian posterior distribution

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

where

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

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

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

with

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

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,

$$\mathbf{m}_N \to (\Phi^T\beta\mathbf{I}\Phi)^{-1}(\Phi^T\beta\mathbf{I}\mathbf{t}) = (\Phi^T\Phi)^{-1}(\Phi^T\mathbf{t})$$

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 $\mathbf{w}$

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

  that is,

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

Fitting of polynomial in terms of probability

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.

Fitting of polynomial in terms of probability

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, $\beta_{MAP} = \beta_{ML}$

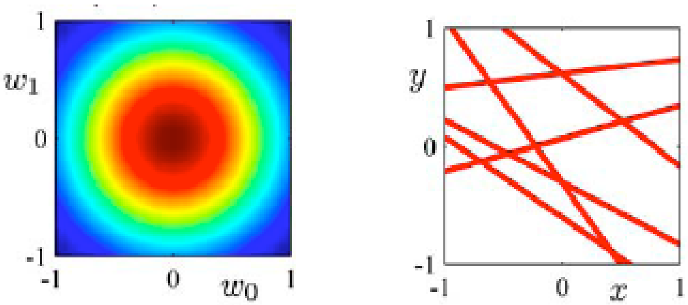
Sequential learning

- 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

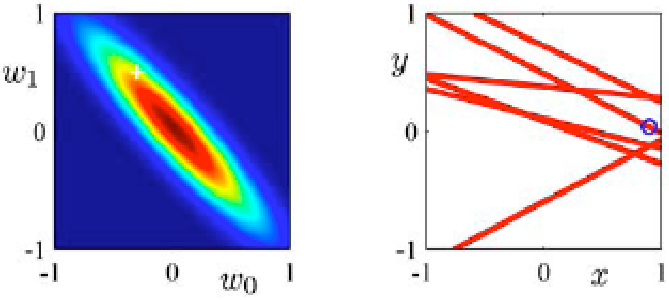
- Input variable x, target variable t, linear regression $y(x, w_0, w_1) = w_0 + w_1 x$.
- Dataset generated by applying function $y = a_0 + a_1 x$ (with $a_0 = -0.3$, $a_1 = 0.5$) to values uniformly sampled in $[-1, 1]$, with added gaussian noise ($\mu = 0$, $\sigma = 0.2$).
- Assume the prior distribution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

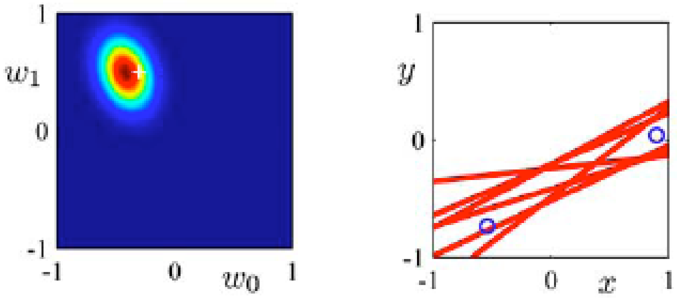
After observing item (x_1, y_1) (circle in right figure).



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

Esempio

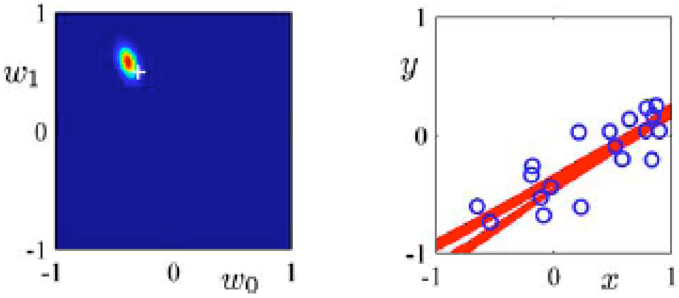
After observing items $(x_1, y_1), (x_2, y_2)$ (circles in right figure).



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 $y = \phi(\mathbf{x})^T \mathbf{w}_{LS}$

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 $p(y|\phi(\mathbf{x})^T \mathbf{w}_{MAP}, \beta)$ for y, with mean $\phi(\mathbf{x})^T \mathbf{w}_{MAP}$ and variance β^{-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

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

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

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

with mean

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

and variance

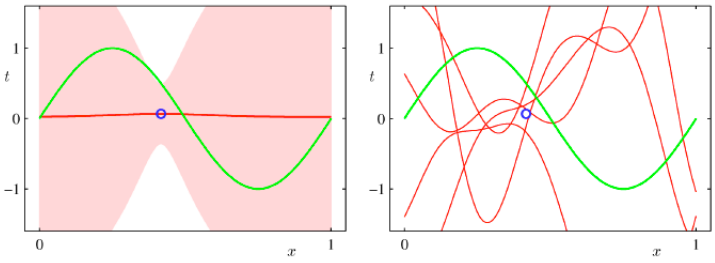
$$\sigma^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^T\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x})$$

- $\frac{1}{\beta}$ 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
- $\boldsymbol{\phi}(\mathbf{x})^T\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}) \to 0$ as $n \to \infty$, and the only uncertainty remaining is the one intrinsic into data observation

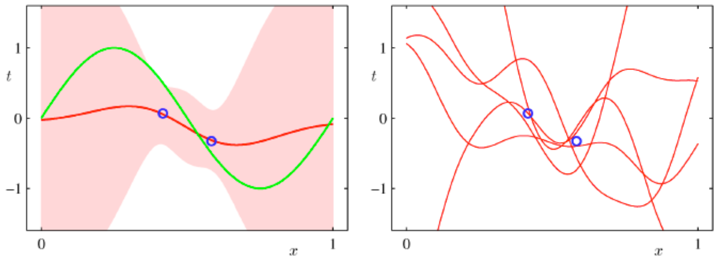
Example

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

Example

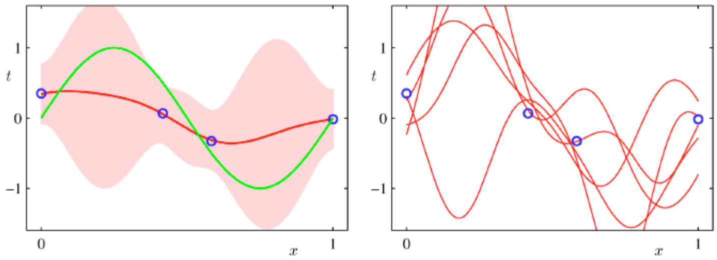


$n = 1$

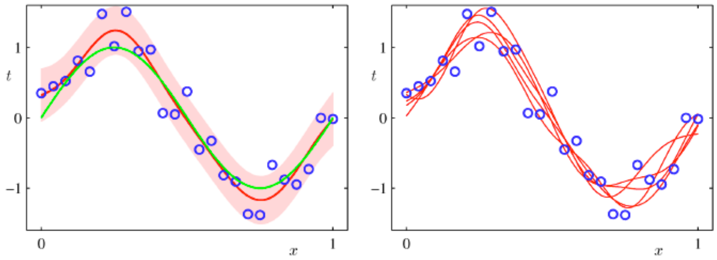


$n = 2$

Example



$n = 4$



$n = 25$

Fully bayesian regression and hyperparameter marginalization

- In a fully bayesian approach, also the hyper-parameters α, β 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 an approximation where point estimation is applied to derive hyper-parameter values by maximizing the posterior distribution $p(\alpha, \beta|\mathbf{t}, \Phi)$

Fully bayesian regression and hyperparameter marginalization

- since $p(\alpha, \beta|\mathbf{t}, \Phi) \propto p(\mathbf{t}|\Phi, \alpha, \beta) p(\alpha, \beta)$, if we assume that $p(\alpha, \beta)$ is relatively flat, then

  $$\underset{\alpha,\beta}{\text{argmax}}\, p(\alpha, \beta|\mathbf{t}, \Phi) \simeq \underset{\alpha,\beta}{\text{argmax}}\, p(\mathbf{t}|\Phi, \alpha, \beta)$$

  and we may consider the maximization of the *marginal likelihood* (marginal wrt to coefficients **w**)

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

- if we assume that $p(\Phi)$ is constant this is equivalent to maximize the evidence

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

Equivalent kernel

- The expectation of the predictive distribution can be written as

$$y(\mathbf{x}) = \beta\phi(\mathbf{x})^T \mathbf{S}_N \Phi^T \mathbf{t} = \sum_{i=1}^{n} \beta\phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}_i) t_i$$

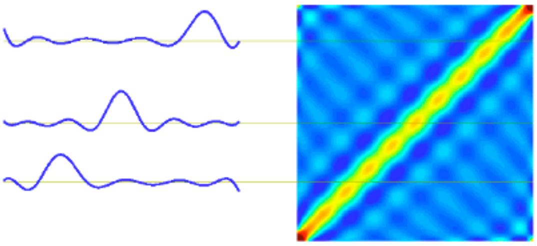
- The prediction can be seen as a linear combination of the target values t_i of items in the training set, with weights dependent from the item values $\mathbf{x}_i$ (and from $\mathbf{x}$)

$$y(\mathbf{x}) = \sum_{i=1}^{n} \kappa(\mathbf{x}, \mathbf{x}_i) t_i$$

The weight function $\kappa(\mathbf{x}, \mathbf{x}') = \beta\phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}')$ is said *equivalent kernel*

Equivalent kernel

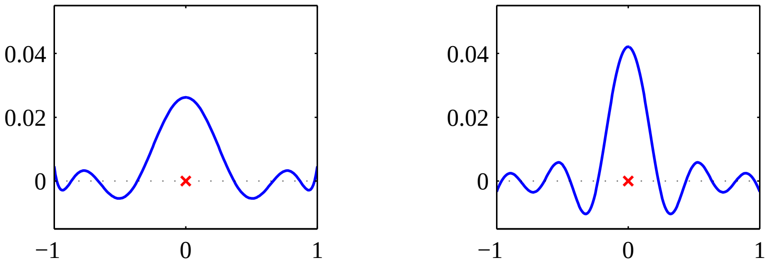
Right: plot on the plane (x, x_i) of a sample equivalent kernel, in the case of gaussian basis functions. Left: plot as a function of x_i for three different values of x



In deriving y, the equivalent kernel tends to assign greater relevance to the target values t_i corresponding to items x_i near to x.

Equivalent kernel

The same localization property holds also for different base functions.



Left, $\kappa(0, x')$ in the case of polynomial basis functions.
Right, $\kappa(0, x')$ in the case of gaussian basis functions.

Equivalent kernel

- The covariance between $y(\mathbf{x})$ and $y(\mathbf{x}')$ is given by

$$\text{cov}(\mathbf{x}, \mathbf{x}') = \text{cov}(\phi(\mathbf{x})^T \mathbf{w}, \mathbf{w}^T \phi(\mathbf{x}')) = \Phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}') = \frac{1}{\beta}\kappa(\mathbf{x}, \mathbf{x}')$$

predicted values are highly correlated at nearby points.

- Instead of introducing base functions which results into a kernel, we may define a localized kernel directly and use it to make predictions (this is the case of *gaussian processes*)
- The equivalent kernel can be expressed as inner product $\kappa(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^T \psi(\mathbf{x}')$ of a suitable set of functions

$$\psi(\mathbf{x}) = \beta^{1/2}\mathbf{S}_N^{1/2}\phi(\mathbf{x})$$

Alternative approach to linear regression

- First approach: define a set of base functions
  - used to derive **w**
  - or (by means of the resulting equivalent kernel) to directly computing $y(\mathbf{x})$ as a linear combination of training set items
- New approach: a suitable kernel is defined and used to compute $y(\mathbf{x})$