Linear basis function models

Linear models for regression

Linear models for regression

Linear models for regression

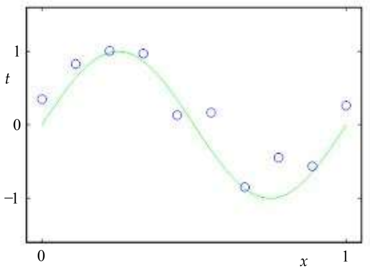
The focus so far on unsupervised learning, we turn now to supervised learning

- Regression

The goal of regression is to predict the value of one or morecontinuous target variables*t*, given the value of a*D*-dimensional vectorxofinput variables

- e.g., polynomial curvefitting

Linear models for regression (cont.)



Training data of $N = 10$ points, blue circles

- each comprising an observation of the input variable x along with the corresponding target variable t

The unknown function $\sin(2\pi x)$ is used to generate the data, green curve

- Goal: Predict the value of t for some new value of x
- without knowledge of the green curve

The input training data x was generated by choosing values of x_n, for $n = 1, \dots, N$, that are spaced uniformly in the range $[0, 1]$

The target training data t was obtained by computing values $\sin(2\pi x_n)$ of the function and adding a small level of Gaussian noise

Linear models for regression (cont.)

- We shall fit the data using a polynomial function of the form

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

- M is the polynomial order, x^j is x raised to the power of j
- Polynomial coefficients $w_0, \ldots, w_M$ are collected in vector $\mathbf{w}$

The coefficients values are obtained by fitting the polynomial to training data

- By minimising an error function, a measure of misfit between function $y(x, \mathbf{w})$, for any given value of $\mathbf{w}$, and the training set data points
- A choice of error function is the sum of the squares of the errors between predictions $y(x_n, \mathbf{w})$ for each point x_n and corresponding target values t_n

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \big(y(x_n, \mathbf{w}) - t_n \big)^2 \implies \mathbf{w}^{\star} \qquad (2)$$

Linear models for regression (cont.)

The goal in the curvefitting problem is to be able to make predictions for the target variable t, given some new value of the input variable x and

- a set of training data comprising N input values $\mathbf{x} = (x_1, \dots, x_N)^T$ and their corresponding target values $\mathbf{t} = (t_1, \dots, t_N)^T$

Uncertainty over the target value is expressed using a probability distribution

- Given the value of x, the corresponding value of t is assumed to have a Gaussian distribution with a mean the value $y(x, \mathbf{w})$ of the polynomial

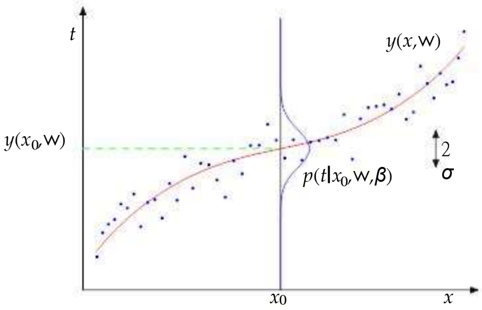
$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t \mid y(x, \mathbf{w}), \beta^{-1}\right) \quad (3)$$

and some precision β (the precision is the reciprocal of the variance σ^2)

Linear models for regression (cont.)

The conditional distribution over t given x is $p(t|x,\mathbf{w},\beta) = \mathcal{N}\left(t \mid y(x,\mathbf{w}), \beta^{-1}\right)$

- The mean is given by the polynomial function $y(x,\mathbf{w})$
- The precision is given by β, with $\beta^{-1} = \sigma^2$



We can use training data $\{\mathbf{x}, \mathbf{t}\}$ to determine the values of the parameters μ and β of this Gaussian distribution

- Likelihood maximisation

Linear models for regression (cont.)

Assuming that the data have been drawn independently from the conditional distribution $p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t \mid y(x, \mathbf{w}), \beta^{-1}\right)$, the likelihood function is

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid y(x_n, \mathbf{w}), \beta^{-1}\right) \tag{4}$$

It is again convenient to maximise its logarithm, the log likelihood function

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\left(y(x_n, \mathbf{w}) - t_n\right)^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) \tag{5}$$

Maximisation of log likelihood wrt $\mathbf{w}$ is minimisation of negative log likelihood

- This equals the minimisation of the sum-of-squares error function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n, \mathbf{w}) - t_n\right)^2 \implies \mathbf{w}_{ML} = \mathbf{w}^{\star}$$

Linear models for regression (cont.)

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

Determination of the maximum likelihood solution for β

- Maximising the log likelihood with respect to β gives

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\left(y(x_n,\mathbf{w}_{ML})-t_n\right)^2 \qquad (6)$$

- where again we decoupled the solution of $\mathbf{w}$ and β

Linear models for regression(cont.)

- Having an estimate of $\mathbf{w}$ and β we can make predictions for new values of x
  - We have a probabilistic model that gives the probability distribution over t

- We can make estimations that are much more than a plain point estimate of t
  - We can make predictions in terms of the predictive distribution

$$p(t|x,\mathbf{w}_{ML},\beta_{ML}) = \mathcal{N}\left(t \,|\, y(x,\mathbf{w}_{ML}),\beta_{ML}^{-1}\right) \qquad (7)$$

- The probability distribution over t, rather than a point estimate

Linear models for regression (cont.)

Polynomialfitting is only a specific example of a broad class of functions

- Linear regression models

They share the property of beinglinear functions of tunable parameters

- In the simplest form, also linear functions of the input variables

Linear models for regression (cont.)

A much more useful class of functions arises by taking linear combinations of afixed set of some nonlinear functions of the input variables

- Such functions are commonly know asbasis functions

Such models are linear functions of the parameters (read, simple analytical properties), and yet can benonlinear with respect to the input variables

Linear models for regression (cont.)

Given a training data set comprising N input observations $\{\mathbf{x}_n\}_{n=1}^{N}$ and corresponding responses target values $\{t_n\}_{n=1}^{N}$

- the goal is to predict the value of t for a new, unseen, value of $\mathbf{x}$

Simplest approach:

Directly construct an appropriate function $y(\mathbf{x})$

- The value for new inputs $\mathbf{x}$ constitute the predictions for the corresponding values of t

More generally:

From a probabilistic perspective, we aim to model the predictive distribution $p(t|\mathbf{x})$

- This expresses our uncertainty about the value of t for each value of $\mathbf{x}$
- This allows to make predictions of t, for any new value of $\mathbf{x}$, that minimise the expected value of a suitable loss function (squared loss)

Linear basis function models

Linear models for regression

Outline

Linear basis function models

- Maximum likelihood and least squares
- Geometry of least squares
- Regularised least squares
- Multiple outputs

Linear basis function models

The simplest linear model for regression is one that involves a linear combination of the input variables

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D, \text{ with } \mathbf{x} = (x_1, \ldots, x_D)^T \tag{8}$$

This is often simply known as linear regression

Key property: This model is a linear function of the parameters $w_0, \ldots, w_D$
Key limitation: It is also a linear function of the input variables $x_1, \ldots, x_D$

We immediately extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \tag{9}$$

Functions $\phi_j(\mathbf{x})$ of the input $\mathbf{x}$ are known as basis functions

Linear basis function models (cont.)

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

The total number of parameters in the linear basis function model is M

The parameter w_0 allow for any fixed offset in the data, it is called bias

- It is often convenient to define a dummy basis function $\phi_0(\mathbf{x}) = 1$

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \qquad (10)$$

- $\mathbf{w} = (w_0, \dots, w_{M-1})^T$
- $\boldsymbol{\phi} = (\phi_0, \dots, \phi_{M-1})^T$

Linear basis function models (cont.)

$$y(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

By using nonlinear basis functions, we allow the function $y(\mathbf{x},\mathbf{w})$ to be a nonlinear function of the input vector

- It is still a linear model, in $\mathbf{w}$

The linearity simplifies the analysis of this class of models

The example of polynomial regression is a particular example of this model on a single input variable x, the basis functions are powers of x, so that $\phi_j(x) = x^j$

$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Linear basis function models (cont.)

There are many possible choices for the basis functions, the classic

$$\varphi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right) \quad (11)$$

- the μ_j govern the location of the basis functions in input space
- the parameter s governs their spatial scale

This kind of functions are referred to as 'Gaussian' basis functions

- Though they are not required to have a probabilistic meaning
  - Normalisation coefficients are unimportant, we multiply by w_j

Linear basis function models (cont.)

Another used possibility is thesigmoidal basis functionof the form

$$\varphi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \quad (12)$$

where the function $\sigma(a)$ is the logistic sigmoid function, defined by

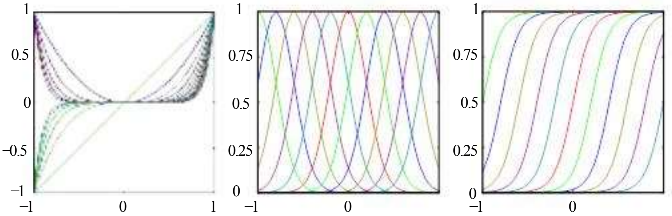
$$\sigma(a) = \frac{1}{1+\exp(-a)} \quad (13)$$

Or, We could also use the hyperbolic tangent function $\tanh(a)$

- It relates to the logistic sigmoid $\tanh(a) = 2\sigma(2a) - 1$

Linear basis function models (cont.)

Polynomials on the left, Gaussians in the centre, and sigmoids on the right



The analysis here is independent of the particular choice of basis function set

- We shall not specify the particular form of the basis functions

Applicable when the vector $\varphi(x)$ of basis functions is the identity $\varphi(x) = x$

Maximum likelihood and least squares

Linear basis function models

Maximum likelihood and least squares

We alreadyfitted polynomial functions to data

- by minimising sum-of-squares error function

We also showed that this error function could be motivated probabilistically

- Maximum likelihood solution under an assumed Gaussian noise model

We return to this problem and consider the least squares approach in detail

- Especially, its relation to maximum likelihood

Maximum likelihood and least squares (cont.)

We assume that the target variable t is given by

- a deterministic function $y(\mathbf{x}, \mathbf{w})$
- with additive Gaussian noise

$$t = y(\mathbf{x}, \mathbf{w}) + \varepsilon \quad (14)$$

Here ε is a zero-mean Gaussian random variable

- with inverse variance (precision) equal β
- $\varepsilon \sim \mathcal{N}(0, \beta^{-1})$

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \quad (15)$$

Given the value of $\mathbf{x}$, the corresponding value of t has a Gaussian distribution with a mean equal to the value $y(\mathbf{x}, \mathbf{w})$ of the deterministic function

Maximum likelihood and least squares (cont.)

If we assume a squared loss function[1], then the optimal prediction, for a new value $\mathbf{x}$, will be given by the conditional mean of the target variable

We have a Gaussian conditional distribution $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$

- The conditional average of t conditoned on $\mathbf{x}$ is

$$\mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x}) dt = y(\mathbf{x}, \mathbf{w}) \qquad (16)$$

- which is what we called the regression function

[1] With a squared loss function $L(t, y(\mathbf{x})) = \{y(\mathbf{x}) - t\}^2$, the expected loss is $\mathbb{E}[L] = \int\int \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt$ and we choose $y(\mathbf{x})$ that minimises $\mathbb{E}(L)$

Maximum likelihood and least squares (cont.)

Consider a set of inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with target values $\mathbf{t} = \{t_1, \ldots, t_N\}$

We group the target variables $\{t_n\}$ into a column vector that we denote by $\mathbf{t}$

Making the assumption that these points are drawn independently from the distribution $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$, we get the likelihood function

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_n), \beta) \quad (17)$$

- It is a function of the adjustable parameters $\mathbf{w}$ and β
- We used $y(\mathbf{x}_n, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \varphi_j(\mathbf{x}_n) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_n)$

In supervised learning, we are not trying to model the distribution of $\mathbf{x}$

- $\mathbf{x}$ is always in the set of conditioning variables
- We can drop it from expression like $p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)$

Maximum likelihood and least squares (cont.)

Taking the logarithm of the likelihood function and using the standard form for the univariate Gaussian, we have

$$\ln p(\mathbf{t}|\mathbf{w},\beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n),\beta^{-1})$$

$$= \frac{N}{2}\ln\beta - \frac{N}{2}\ln 2\pi - \beta E_D(\mathbf{w}) \qquad (18)$$

Where, as always, the sum-of-squares has been defined as

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \big(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\big)^2 \qquad (19)$$

Maximum likelihood and least squares (cont.)

Having the likelihood function, we use maximum likelihood to get $\mathbf{w}$ and β

Consider first the maximisation with respect to $\mathbf{w}$

- The maximisation of the likelihood under a conditional Gaussian is equivalent to the minimisation of the sum-of-squares error function $E_D(\mathbf{w})$
- The gradient of the log likelihood function takes the form

$$\nabla \ln p(\mathbf{t}|\mathbf{w},\beta) = \beta \sum_{n=1}^{N} \left(t_n - \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_n) \right) \boldsymbol{\varphi}(\mathbf{x}_n) \tag{20}$$

- Setting the gradient to zero gives

$$0 = \sum_{n=1}^{N} t_n \boldsymbol{\varphi}(\mathbf{x}_n)^T - \mathbf{w}^T \left(\sum_{n=1}^{N} \boldsymbol{\varphi}(\mathbf{x}_n)\boldsymbol{\varphi}(\mathbf{x}_n)^T \right) \tag{21}$$

- Solving for $\mathbf{w}$ gives

$$\mathbf{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t} \tag{22}$$

Maximum likelihood and least squares (cont.)

Normal equations for the LS problem: $\mathbf{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$

Matrix Φ is the $N \times M$ design matrix, with elements $\Phi_{nj} = \varphi_j(\mathbf{x}_n)$

$$\Phi = \begin{pmatrix} \varphi_0(\mathbf{x}_1) & \varphi_1(\mathbf{x}_1) & \cdots & \varphi_{M-1}(\mathbf{x}_1) \\ \varphi_0(\mathbf{x}_2) & \varphi_1(\mathbf{x}_2) & \cdots & \varphi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(\mathbf{x}_N) & \varphi_1(\mathbf{x}_N) & \cdots & \varphi_{M-1}(\mathbf{x}_N) \end{pmatrix} \quad (23)$$

The Moore-Penrose pseudo inverse[2] of matrix Φ is the quantity

$$\Phi^\dagger \equiv (\Phi^T\Phi)^{-1}\Phi^T \quad (24)$$

[2] Notion of matrix inverse for non-square matrices: If Φ square and invertible, $\Phi^\dagger = \Phi^{-1}$

Maximum likelihood and least squares (cont.)

Let us now get some insight on the bias parameter w_0, by making it explicit

- The error function $E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\big(t_n - \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_n)\big)^2$ becomes

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\Big(t_n - w_0 - \sum_{j=1}^{M-1} w_j\varphi_j(\mathbf{x}_n)\Big)^2 \qquad (25)$$

- We can set its derivative wrt w_0 to be equal to zero and get

$$\begin{aligned} w_0 &= \underbrace{\frac{1}{N}\sum_{n=1}^{N} t_n}_{\overline{t}} - \sum_{j=1}^{M-1} w_j \underbrace{\frac{1}{N}\sum_{n=1}^{N}\varphi_j(\mathbf{x}_n)}_{\overline{\varphi}_j} \\ &= \overline{t} - \sum_{j=1}^{M-1} w_j\overline{\varphi}_j \qquad (26) \end{aligned}$$

Maximum likelihood and least squares (cont.)

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\varphi}_j \quad \text{with} \quad \begin{cases} \bar{t} = \frac{1}{N} \sum_{n=1}^{N} t_n \\ \bar{\varphi}_j = \frac{1}{N} \sum_{n=1}^{N} \varphi_j(\mathbf{x}_n) \end{cases}$$

The bias w_0 compensates for the difference between the averages of the target values $\{t_n\}_{n=1}^N$ in the training set and the weighted sum of the averages of the basis functions $\{\varphi_j(\mathbf{x}_n)\}_{j=1}^{M-1}$ evaluated also over the whole training set $\{\mathbf{x}_n\}_{n=1}^N$

Maximum likelihood and least squares (cont.)

Maximising the likelihood $\ln p(\mathbf{t}|\mathbf{w},\beta) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln 2\pi - \beta E_D(\mathbf{w})$ wrt β

- We find that the inverse noise precision (the noise variance) is given by

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_n)\right)^2 \tag{27}$$

- It is the residual variance of the targets around the regression function

Geometry of least squares

Linear basis function models

Geometry of least squares

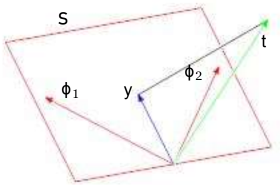
Let us now analyse the geometrical interpretation of the least-squares solution

Consider an N-dimensional space whose axes are given by the t_n

- $\mathbf{t} = (t_1, \ldots, t_N)^T$ is a vector in this space

Each basis function $\varphi_j(\mathbf{x}_n)$, evaluated at the N data points, can also be seen as a vector in the same space, denoted by $\boldsymbol{\phi}_j$

$\boldsymbol{\phi}_j$ corresponds to the j-th column of Φ, $\varphi(\mathbf{x}_n)$ corresponds to the n-th row of Φ

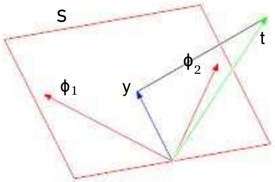


If the number M of basis functions is smaller than the number N of points, then the M vectors $\boldsymbol{\phi}_j$ will span a linear subspace $\mathcal{S}$ of dimensionality M

Geometry of least squares (cont.)

Define the N-dimensional vector $\mathbf{y}$ whose n-th element is $y(\mathbf{x}_n, \mathbf{w})$, $n = 1, \ldots, N$

- $\mathbf{y}$ is an arbitrary linear combination of the vectors ϕ_j
- it can live anywhere in this M-dimensional subspace



The sum-of-squares error $E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^T\varphi(\mathbf{x}_n))^2$ is equal (up to a factor 1/2) to the squared Euclidean distance between $\mathbf{y}$ and $\mathbf{t}$

The least-squares solution for $\mathbf{w}$ corresponds to that choice of $\mathbf{y}$ that lies in subspace $\mathcal{S}$ and that is closest to $\mathbf{t}$

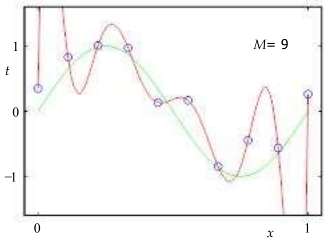
It can be shown that this solution is an orthogonal projection of $\mathbf{t}$ onto $\mathcal{S}$ ( )

Regularised least squares

Linear basis function models

Regularised least squares

We introduced the idea of adding a regularisation term to an error function in order to control over-fitting



The magnitude of the coefficients tends to explode trying to (over)fit the data

$$||w||^2 = w^T w = w_0^2 + w_1^2 + \cdots + w_M^2$$

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Regularised least squares (cont.)

- Add a penalty term to the error function $E(\mathbf{w})$, to discourage the coefficients from reaching large values
- The simplest such penalty term is the sum of squares of all of the coefficients, to get a new error function

$$\tilde{E}(\mathbf{w}) = \underbrace{\frac{1}{2}\sum_{n=1}^{N}\big(y(x_n,\mathbf{w}) - t_n\big)^2}_{E_D(\mathbf{w})} + \lambda \underbrace{\frac{1}{2}||\mathbf{w}||^2}_{E_W(\mathbf{w})} \qquad (28)$$

- where $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w} = w_0^2 + w_1^2 + \cdots + w_M^2$
- Coefficient λ trades off between the regularisation term and the standard sum-of-squares error

Regularised least squares (cont.)

- The total error function to be minimised became

$$E_D(\mathrm{w}) + \lambda E_W(\mathrm{w}) \tag{29}$$

- λ is the regularisation coefficient that controls the relative importance of the data-dependent error $E_D(\mathrm{w})$ and the regularisation term $E_W(\mathrm{w})$

One of the simplest forms of regulariser is the sum-of-squares of the weight vector elements

$$E_W(\mathrm{w}) = \frac{1}{2}\mathrm{w}^T\mathrm{w} \tag{30}$$

Regularised least squares (cont.)

Consider the sum-of-squares error function

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_n)\right)^2 \quad (31)$$

then, the total error function becomes

$$\frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_n)\right)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \quad (32)$$

This particular choice of regulariser is known in the machine learning literature asweight decaybecause in sequential learning algorithms, it encourages weight values to decay towards zero, unless supported by the data

In statistics, it provides an example of aparameter shrinkagemethod because it tends to shrinks the parameter values towards zero

Regularised least squares (cont.)

For the polynomial example, we developed a Bayesian treatment of the problem

- We introduced a prior distribution over the polynomial coefficients $\mathbf{w}$

A Gaussian prior $\mathcal{N}(\mathbf{x}|\mu,\sigma^2) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$

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

With $\mu = 0$ and $\Sigma = \alpha^{-1}\mathbf{I}$, α is the precision of the prior distribution

Regularised least squares (cont.)

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

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

Using Bayes' theorem, the posterior distribution for $\mathbf{w}$ is proportional to the product of the prior distribution and the likelihood function, thus

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

We determine $\mathbf{w}$ by finding its most probable value given the data

- by maximising the posterior distribution
- maximum posterior or MAP

Regularised least squares (cont.)

By taking the negative log of the posterior distribution over w and combining with the log likelihood function and the prior distribution over w, we found that

- the maximum of the posterior is given by the minimum of

$$\frac{\beta}{2}\sum_{n=1}^{N}\left(y(x_n,w)-t_n\right)^2+\frac{\alpha}{2}w^Tw$$

Thus, maximising the posterior is equivalent to minimising the regularised sum-of-squares error function with regularisation $\lambda=\alpha/\beta$

$$\tilde{E}(w)=\frac{1}{2}\sum_{n=1}^{N}(y(x_n,w)-t_n)^2+\frac{\lambda}{2}||w||^2$$

Though we included a prior $p(w|\alpha)$, we are still making point estimates of w

Regularised least squares (cont.)

$$\frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\right)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

It has the advantage that the error function remains a quadratic function of $\mathbf{w}$

- Its exact minimiser can be found in closed form

We first set the gradient of the total error function with respect to $\mathbf{w}$ to zero

- Then, we solve for $\mathbf{w}$ to get

$$\mathbf{w} = \left(\lambda \mathbf{I} + \Phi^T\Phi\right)^{-1}\Phi^T\mathbf{t} \quad (33)$$

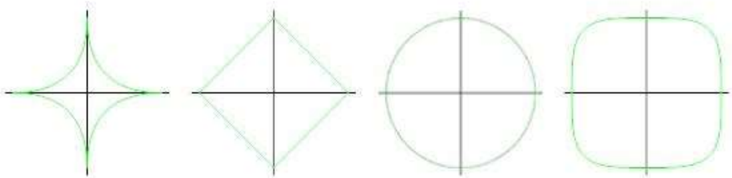
This result is an extension of the least-squares solution $\mathbf{w} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\mathbf{t}$

Regularised least squares (cont.)

A more general regulariser gives a regularised error in the form

$$\frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q \qquad (34)$$

where $q = 2$ corresponds to the classical quadratic regulariser



$q = 0.5$ $q = 1$ $q = 2$ $q = 4$

The case of $q = 1$ is know as the *lasso* in the statistics literature

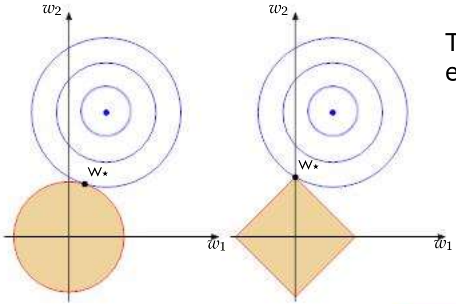
- If λ is sufficiently large, some of the coefficients w_j are driven to zero
- Sparse model in which the corresponding basis functions play no role

Regularised least squares (cont.)

To see this (⋆), we first note that minimising Eq. 34 is equivalent to minimising the un-regularised sum-of-squares error (Eq. 3.12) subject to the constraint

$$\sum_{j=1}^{M} |w_j|^q \leq \eta \qquad (35)$$

for an appropriate value of parameter η (related, using Lagrange multipliers)



The contours of the unregularised error function (blue)

- The constraint region for the quadratic regulariser ($q = 2$)
- The constraint region for the lasso regulariser ($q = 1$)

Regularised least squares (cont.)

Regularisation allows complex models to be trained on data sets of limited size without severe over-fitting, essentially by limiting the effective model complexity

The problem of determining the optimal model complexity is shifted

- fromfinding the appropriate number of basis functions
- to determining a suitable value of the coefficientλ

Multiple outputs

Linear basis function models

Multiple outputs

We have only discussed the case of a single target variable t

It happens that in some applications we wish to predict $K > 1$ target variables

- We denote collectively multivariate targets by the target vector **t**

This problem can be approached by introducing a different set of basis functions for each component of **t**, leading to multiple, independent regression problems

- A more interesting, and more common, approach is to use the same set of basis functions to model all of the components of the target vector so that

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \boldsymbol{\varphi}(\mathbf{x}) \qquad (36)$$

- **y** is a K-dimensional column vector
- **W** is a $M \times K$ matrix of parameters
- $\boldsymbol{\varphi}(\mathbf{x})$ is a M-dimensional column vector with elements $\varphi_j(\mathbf{x})$ ($\varphi_0(\mathbf{x}) = 1$)

Multiple outputs (cont.)

Suppose, the conditional distribution of the target to be an isotropic Gaussian

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\mathbf{I}) \tag{37}$$

If we have a set of observations $\mathbf{t}_1, \ldots, \mathbf{t}_N$, we can combine these into a matrix $\mathbf{T}$ of size $N \times K$ such that the n-th row is given by $\mathbf{t}_n^T$

Similarly, we can combine the input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ into matrix $\mathbf{X}$

The log likelihood function is then given by

$$\begin{aligned} \ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2}\sum_{n=1}^{N} ||\mathbf{t}_n - \mathbf{W}^T\boldsymbol{\phi}(\mathbf{x}_n)||^2 \end{aligned} \tag{38}$$

Multiple outputs (cont.)

The minimisation of this function with respect to $\mathbf{W}$ gives

$$\mathbf{W}_{ML} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{T} \qquad (39)$$

For each target variable t_k, we have

$$\mathbf{w}_k = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t}_k = \Phi^{\dagger} \mathbf{t}_k \qquad (40)$$

where $\mathbf{t}_k$ is a N-dimensional vector with component t_{nk}, for $n = 1, \ldots, N$

- The solution decouples between different target variables

We need compute a single pseudo-inverse matrix $\Phi^{\dagger}$, shared by all vectors $\mathbf{w}_k$

Multiple outputs (cont.)

The extension to general Gaussian noise distributions having arbitrary covariance matrices is straightforward (⋆)

Again, this leads to a decoupling into K independent regression problems

This result is unsurprising because:

- the parameters $\mathbf{W}$ define only the mean of the Gaussian noise distribution
- he maximum likelihood solution for the mean of a multivariate Gaussian is independent of the covariance