Linear basis function models

Linear models for regression

Francesco Corona
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 more *continuous target variables* $t$, given the value of some $D$-dimensional vector $x$ of *input variables*

• e.g., polynomial curve fitting
Linear basis function models

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

Linear models for regression (cont.)

\[ N = 10 \text{ training points, blue circles} \]
- each comprising an observation of the input variable \( x \) along with 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 \( x \)
- w/o knowledge of the green curve

The input training data \( x \) was generated by choosing values of \( x_n \), for \( n = 1, \ldots, N \), 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
• We shall fit the data using a polynomial function of the form

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

• \( M \) 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 \( t_n \)

\[ E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left( y(x_n, \mathbf{w}) - t_n \right)^2 \implies \mathbf{w}^* \quad (2) \]
The goal in the curve fitting 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 \( x = (x_1, \ldots, x_N)^T \) and their corresponding target values \( t = (t_1, \ldots, t_N)^T \)

**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 \( y(x, w) \) of the polynomial

\[
p(t|x, w, \beta) = \mathcal{N} \left( t \middle| y(x, w), \beta^{-1} \right)
\]

and some precision \( \beta \) (the precision is the reciprocal of the variance \( \sigma^2 \))
The conditional distribution over $t$ given $x$ is $p(t|x, w, \beta) = \mathcal{N}(t \mid y(x, w), \beta^{-1})$

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

We can use training data $\{x, t\}$ to determine the values of the parameters $\mu$ and $\beta$ of this Gaussian

- Likelihood maximisation
Linear models for regression (cont.)

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

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

(4)

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

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

(5)

Maximisation of log likelihood wrt \( w \) is minimisation of negative log likelihood

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

\[
E(w) = \frac{1}{2} \sum_{n=1}^{N} \left(y(x_n, w) - t_n\right)^2 \quad \Rightarrow \quad w_{ML} = w^*
\]
Linear models for regression (cont.)

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

Determination of the maximum likelihood solution for $\beta$

- Maximising the log likelihood with respect to $\beta$ gives

\[
\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \left(y(x_n, w_{ML}) - t_n\right)^2
\]

- where again we decoupled the solution of $w$ and $\beta$
Having an estimate of $w$ and $\beta$ 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**

$$p(t|x, w_{ML}, \beta_{ML}) = \mathcal{N}(t|\hat{y}(x, w_{ML}), \beta_{ML}^{-1})$$

- The probability distribution over $t$, rather than a point estimate
Polynomial fitting is only a specific example of a broad class of functions

- **Linear regression models**
  They share the property of being **linear functions of tuneable parameters**
  
  - In the simplest form, also linear functions of the input variables
A much more useful and richer class of functions arises by taking linear combinations of a fixed set of some nonlinear functions of input variables

- Such functions are commonly known as **basis functions**

Such models are linear functions of the parameters (read, simple analytical properties), and yet can be **nonlinear with respect to the input variables**
Linear models for regression (cont.)

Given training data comprising $N$ input observations $\{x_n\}_{n=1}^N$ and the corresponding responses target values $\{t_n\}_{n=1}^N$:

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

Simplest approach:

Directly construct an appropriate function $y(x)$

- The value for new inputs $x$ constitute the predictions for the corresponding values of $t$

More generally:

We aim to model the predictive distribution $p(t|x)$

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

Linear models for regression
Outline

1 Linear basis function models
   Maximum likelihood and least squares
   Geometry of least squares
   Regularised least squares
   Multiple outputs
The simplest linear model for regression is one that involves a linear combination of the input variables

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

(8)

This is often simply known as \textit{linear regression}

\textbf{Key property}: This model is a linear function of the parameters $w_0, \ldots, w_D$

\textbf{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

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

(9)

Functions \( \phi_j(x) \) of the input \( x \) are known as basis functions.
**Linear basis function models (cont.)**

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

The total number of parameters in the linear basis function model is \( M \).

The parameter \( w_0 \) allows for any fixed offset in the data, it is called **bias**.

- It is often convenient to define a **dummy basis function** \( \phi_0(x) = 1 \)

\[
y(x, w) = \sum_{j=0}^{M-1} w_j \phi_j(x) = w^T \phi(x)
\]  \hspace{1cm} (10)

- \( w = (w_0, \ldots, w_{M-1})^T \)
- \( \phi = (\phi_0, \ldots, \phi_{M-1})^T \)
Linear basis function models (cont.)

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

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

- It is still a linear model, in \( w \)

The linearity simplifies the analysis of this class of models

Example

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

\[ y(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j \]
There are many possible choices for the basis functions, the classic

$$\phi_j(x) = \exp \left( - \frac{(x - \mu_j)^2}{2s^2} \right)$$  \hspace{1cm} (11)

- the $\mu_j$ 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 $w_j$
Another used possibility is the **sigmoidal basis function** of the form

\[ \phi_j(x) = \sigma \left( \frac{x - \mu_j}{s} \right) \]  

(12)

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

\[ \sigma(a) = \frac{1}{1 + \exp(-a)} \]  

(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 $\phi(x)$ of basis functions is the identity $\phi(x) = x$
Maximum likelihood and least squares
Linear basis function models
Maximum likelihood and least squares

We already fitted 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
We assume that the target variable $t$ is given by

- a deterministic function $y(x, w)$
- with additive Gaussian noise

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

Here $\varepsilon$ is a zero-mean Gaussian random variable

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

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

$$ p(t|x, w, \beta) = \mathcal{N}(t|y(x, w), \beta^{-1}) \quad (15) $$
If we assume a squared loss function\(^1\), then the optimal prediction, for a new value \(x\), will be given by the conditional mean of the target variable

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

• The conditional average of \(t\) conditioned on \(x\) is

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

• which is what we called the regression function

\(^1\)With a squared loss function \(L(t, y(x)) = (y(x) - t)^2\), the expected loss is \(\mathbb{E}[L] = \int \int (y(x) - t)^2 p(x, t) dxdt\) and we choose \(y(x)\) that minimises \(\mathbb{E}(L)\)
Consider a set of inputs $X = \{x_1, \ldots, x_N\}$ with target values $t = \{t_1, \ldots, t_N\}$

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

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

\[
p(t|X, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|w^T \phi(x_n), \beta)
\]

(M17)

- It is a function of the adjustable parameters $w$ and $\beta$
- We used $y(x_n, w) = \sum_{j=0}^{M-1} w_j \phi_j(x_n) = w^T \phi(x_n)$
Remark

In supervised learning, we are not trying to model the distribution of $x$

- $x$ is always in the set of conditioning variables
- We can drop it from expression like $p(t|x, w, \beta)$
Taking the logarithm of the likelihood function and using the standard form for the univariate Gaussian, we have

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

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

(18)

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

$$
E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left( t_n - \mathbf{w}^T \phi(x_n) \right)^2
$$

(19)

Having the likelihood function, we use maximum likelihood to get $\mathbf{w}$ and $\beta$
Maximum likelihood and least squares (cont.)

Consider first the maximisation with respect to $w$

- The maximisation of the likelihood under a conditional Gaussian equals the minimisation of the sum-of-squares error function $E_D(w)$
- The gradient of the log likelihood function takes the form

\[
\nabla \ln p(t|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left( t_n - \mathbf{w}^T \phi(x_n) \right) \phi(x_n)
\]

(20)

- Setting the gradient to zero gives

\[
0 = \sum_{n=1}^{N} t_n \phi(x_n)^T - \mathbf{w}^T \left( \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T \right)
\]

(21)

- Solving for $w$ gives

\[
\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t
\]

(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 \( \Phi \) is the \( N \times M \) design matrix, with elements \( \Phi_{nj} = \phi_j(\mathbf{x}_n) \)

\[
\Phi = \begin{pmatrix}
\phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\
\phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\
\vdots & \vdots & \ddots & \vdots \\
\phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N)
\end{pmatrix}
\]  

(23)

The Moore-Penrose pseudo inverse\(^2\) of matrix \( \Phi \) is the quantity

\[
\Phi^\dagger \equiv (\Phi^T \Phi)^{-1} \Phi^T
\]

(24)

---

\(^2\)Notion of matrix inverse for non-square matrices: If \( \Phi \) square and invertible, \( \Phi^\dagger = \Phi^{-1} \)
Let us now get some insight on the bias parameter $w_0$, by making it explicit

- The error function $E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right)^2$ becomes

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \left( t_n - w_0 - \sum_{j=1}^{M-1} w_j \phi_j(x_n) \right)^2$$  \hspace{1cm} (25)

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

$$w_0 = \frac{1}{N} \sum_{n=1}^{N} t_n - \sum_{j=1}^{M-1} w_j \frac{1}{N} \sum_{n=1}^{N} \phi_j(x_n)$$

$$= \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j$$  \hspace{1cm} (26)
Maximum likelihood and least squares (cont.)

\[
\begin{aligned}
\hat{w}_0 &= \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j \\
\text{with} \\
\begin{cases}
\bar{t} = \frac{1}{N} \sum_{n=1}^{N} t_n \\
\bar{\phi}_j = \frac{1}{N} \sum_{n=1}^{N} \phi_j(x_n)
\end{cases}
\end{aligned}
\]

The bias \( \hat{w}_0 \) 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 \( \{\phi_j(x_n)\}_{j=1}^{M-1} \) evaluated over the whole training set \( \{x_n\}_{n=1}^{N} \).
Maximum likelihood and least squares (cont.)

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

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

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right)^2$$

(27)

- It is the residual variance of the targets around the regression function
Geometry of least squares
Linear basis function models
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 $\phi_j(x_n)$, evaluated at the $N$ data points, can also be seen as a vector in the same space, denoted by $\varphi_j$

$\varphi_j$ corresponds to the $j$-th column of $\Phi$, $\phi(x_n)$ corresponds to the $n$-th row of $\Phi$

If the number $M$ of basis functions is smaller than the number $N$ of points, then the $M$ vectors $\varphi_j$ will span a linear subspace $S$ of dimensionality $M$
Define the $N$-vector $y$ whose $n$-th element is $y(x_n, w)$, $n = 1, \ldots, N$

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

$$E_D(w) = \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi(x_n))^2,$$

the sum-of-squares error is equal (up to a factor 1/2) to the squared Euclidean distance between $y$ and $t$

A least-squares solution for $w$ corresponds to that choice of $y$ that lies in subspace $S$ and that is closest to $t$

It can be shown that this solution is an orthogonal projection of $t$ onto $S$ (⋆)
Regularised least squares
Linear basis function models
Linear basis function
models
UFC/DC
ATAI-I (CK0146)
2017.1

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.

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

<table>
<thead>
<tr>
<th></th>
<th>( M = 0 )</th>
<th>( M = 1 )</th>
<th>( M = 6 )</th>
<th>( M = 9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_0^* )</td>
<td>0.19</td>
<td>0.82</td>
<td>0.31</td>
<td>0.35</td>
</tr>
<tr>
<td>( w_1^* )</td>
<td>-1.27</td>
<td>7.99</td>
<td>232.37</td>
<td></td>
</tr>
<tr>
<td>( w_2^* )</td>
<td>-25.43</td>
<td>-5321.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_3^* )</td>
<td>17.37</td>
<td>48568.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_4^* )</td>
<td></td>
<td></td>
<td>-231639.30</td>
<td></td>
</tr>
<tr>
<td>( w_5^* )</td>
<td></td>
<td></td>
<td>640042.26</td>
<td></td>
</tr>
<tr>
<td>( w_6^* )</td>
<td></td>
<td></td>
<td>-1061800.52</td>
<td></td>
</tr>
<tr>
<td>( w_7^* )</td>
<td></td>
<td></td>
<td>1042400.18</td>
<td></td>
</tr>
<tr>
<td>( w_8^* )</td>
<td></td>
<td></td>
<td>-557682.99</td>
<td></td>
</tr>
<tr>
<td>( w_9^* )</td>
<td></td>
<td></td>
<td>125201.43</td>
<td></td>
</tr>
</tbody>
</table>
Add a penalty term to the error function $E(w)$, 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

$$\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2 + \lambda \frac{1}{2} ||w||^2$$  \hspace{1cm} (28)

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

Coefficient $\lambda$ trades off between the regularisation term and the standard sum-of-squares error.
The total error function to be minimised became

$$E_D(w) + \lambda E_W(w)$$  \hspace{1cm} (29)

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

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

$$E_W(w) = \frac{1}{2}w^T w$$  \hspace{1cm} (30)
Consider the sum-of-squares error function

\[
E_D(w) = \frac{1}{2} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right)^2
\]  
(31)

then, the total error function becomes

\[
\frac{1}{2} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right)^2 + \frac{\lambda}{2} w^T w
\]  
(32)
Remark

This choice of regulariser is known in machine learning as weight decay

- In sequential learning algorithms, it encourages weight values to decay towards zero, unless supported by the data

In statistics, it provides an example of a parameter shrinkage method

- It tends to shrink the parameter values towards zero
For the polynomial example, we developed a Bayesian treatment of the case

- We introduced a prior distribution over the polynomial coefficients $w$

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

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

With $\mu = 0$ and $\Sigma = \alpha^{-1} I$, $\alpha$ is the precision of the prior distribution
Regularised least squares (cont.)

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

\[ p(t|x, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, w), \beta^{-1}) \]

Using Bayes’ theorem, the posterior distribution for \( w \) is proportional to the product of the prior distribution and the likelihood function, thus

\[ p(w|x, t, \alpha, \beta) \propto p(t|x, w, \beta) p(w|\alpha) \]
We determined \( w \) by finding its most probable value given the data

- by \textit{maximising the posterior distribution}
- \textit{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 prior distribution over $w$, we found that

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

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

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.)

\[
\frac{1}{2} \sum_{n=1}^{N} \left( t_n - w^T \phi(x_n) \right)^2 + \frac{\lambda}{2} w^T w
\]

It has the advantage that the error function remains a quadratic function of $w$

- Its exact minimiser can be found in closed form

We first set the gradient of the total error function with respect to $w$ to zero

- Then, we solve for $w$ to get

\[
w = \left( \lambda I + \Phi^T \Phi \right)^{-1} \Phi^T t
\]

(33)

This result is an extension of the least-squares solution $w = \left( \Phi^T \Phi \right)^{-1} \Phi^T t$
A more general regulariser gives a regularised error in the form

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

(34)

where $q = 2$ corresponds to the classical quadratic regulariser.
Regularised least squares (cont.)

Remark

The case of $q = 1$ is known as the lasso in the statistics literature:

- If $\lambda$ is sufficiently large, some of the coefficients $w_j$ are driven to zero.
- Sparse model in which the corresponding basis functions play no role.
To see this (⋆), note that minimising Eq. 34 is equivalent to minimising the conventional, un-regularised, sum-of-squares error subject to the constraint

$$\sum_{j=1}^{M} |w_j|^q \leq \eta$$

for an appropriate value of parameter $\eta$ (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 effective model complexity.

The problem of determining the optimal model complexity is shifted:

- from finding the appropriate number of basis functions
- to determining a suitable value of the coefficient $\lambda$
Multiple outputs
Linear basis function models
Multiple outputs

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

In some applications we wish to predict $K > 1$ target variables

- We denote collectively multivariate targets by the target vector $\mathbf{t}$

The task can be approached by introducing a different set of basis functions for each component of $\mathbf{t}$, giving 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

$$y(x, w) = W^T \phi(x)$$

(36)

- $y$ is a $K$-dimensional column vector
- $W$ is a $M \times K$ matrix of parameters
- $\phi(x)$ is a $M$-dimensional column vector with elements $\phi_j(x) (\phi_0(x) = 1)$
Multiple outputs (cont.)

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

\[ p(t|x, W, \beta) = \mathcal{N}(t|W^T \phi(x), \beta^{-1}I) \]  

(37)

If we have a set of observations \( t_1, \ldots, t_N \), we can combine these into a matrix \( T \) of size \( N \times K \) such that the \( n \)-th row is given by \( t_n^T \).
Similarly, we can combine the input vectors $x_1, \ldots, x_N$ into matrix $X$.

The log likelihood function is then given by

$$
\ln p(T|X, W, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n|W^T \phi(x), \beta^{-1}I)
$$

$$
= \frac{NK}{2} \ln \left( \frac{\beta}{2\pi} \right) - \frac{\beta}{2} \sum_{n=1}^{N} ||t_n - W^T \phi(x_n)||^2 \quad (38)
$$
Multiple outputs (cont.)

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

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

(39)

For each target variable $t_k$, we have

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

(40)

$t_k$ is a $N$-D vector with components $t_{nk}$, $n = 1,\ldots,N$

- The solution decouples between different target variables

Need to compute a single pseudo-inverse $\mathbf{\Phi}^\dagger$, shared by all vectors $\mathbf{w}_k$
Multiple outputs (cont.)

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

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

This result is unsurprising

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