# Linear Models for Regression

## Notations and Overview

**Input:**

$N$ observations $\{\mathbf{x}_n\}$, where $n = 1, \ldots, N$

$N$ Target values $\{t_n\}$

**Aim**:

Model the predictive distribution $p(t|\mathbf{x})$ => this expresses the uncertainty about the value of t for each value of $\mathbf{x}$.

**Pros and Cons of linear models**:

Linear models have problems involving input spaces of high dimensionality, but they have nice analytical properties.

## Linear Basis Function Models

Linear means linear combination of the input variables.

$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D$

Where $\mathbf{x} = (x_1, \ldots, x_D)^T$

In the above equation, it is linear function of both $w_i$ and $x_i$, this imposes signifiant limitations on the model, so we extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables.

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

Where $\phi_j(x)$ are basis functions.

Define $\phi_0(\mathbf{x}) = 1$, so

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

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

**We call it is linear because the function is linear in $\mathbf{w}$**

Choices of basis functions:

- Polynomial: $\phi_j(x) = x^j$
- Exponential: $\phi_j(x) = \exp\{-\frac{(x-\mu_j)^2}{2s^2}\}$
- Sigmoid: $\phi_j(x) = \sigma(\frac{x-\mu_j}{s})$, where $\sigma(a) = \frac{1}{1+\exp(-a)}$
- tanh: $\tanh(a) = 2\sigma(a) - 1$
- Fourier basis

## Maximum Likelihood and Least Squares

Define target variable $t$ is given:

$t = y(\mathbf{x}, \mathbf{w}) + \epsilon$

Where $\epsilon$ is a zero mean Gaussian random variable with precision (inverse variance) $\beta$. (The Gaussian noise implies the conditional distribution of $t$ given $\mathbf{x}$ is unimodal.)

Now consider a data set of inputs $\mathbf{X} = \{\mathbf{x}_1 \dots, \mathbf{x}_N\}$ and we group target value $t_1, \dots t_N$ together into a column vector $\mathbf{t}$.

The likelihood function is:

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

To keep the notation simple, we will drop $\mathbf{x}$ since it will always appear in the set of conditioning variables.

Then we take logarithm of the likelihood function:

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{w},\beta) &= \sum_{n=1}^{N}\ln\mathcal{N}(t_n|\mathbf{w}^T\phi(\mathbf{x}_n),\beta^{-1}) \\ &= -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n-\mathbf{w}^T\phi(\mathbf{x}_n)\}^2+\frac{N}{2}\ln\beta-\frac{N}{2}\ln(2\pi) \\ &= \frac{N}{2}\ln\beta-\frac{N}{2}\ln(2\pi)-\beta E_D(\mathbf{w})\end{aligned}$$

Where $E_D(\mathbf{w})=\frac{1}{2}\sum_{n=1}^{N}\{t_n-\mathbf{w}^T\phi(\mathbf{x}_n)\}^2$ is the sum-of-squares error function.

We can use maximum likelihood to determine $\mathbf{w}$ and $\beta$.

- With respect to $\mathbf{w}$, Maximisation of the likelihood function under a conditional Gaussian noise distribution is equivalent to minimise the sum-of-squares error function given by $E_D(\mathbf{w})$

  $\frac{\partial E_D(\mathbf{w})}{\partial \mathbf{w}}=\sum_{n=1}^{N}\{t_n-\mathbf{w}^T\phi(\mathbf{x}_n)\}\phi(\mathbf{x}_n)^T$

  Setting the gradient to 0:

  $0=\sum_{n=1}^{N}t_n\phi(\mathbf{x}_n)^T-\mathbf{w}^T(\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T)$

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

  $\mathbf{\Phi}$ is the design matrix (features) with dimension NxM, and:

  $$\begin{pmatrix}\phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1)\\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_{M-1}(\mathbf{x}_2)\\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N)\end{pmatrix}$$

- With respect to $\beta$, maximise the log-likelihood

$$\frac{\partial \ln p(\mathbf{t}|\mathbf{w},\beta)}{\partial \beta} = \frac{N}{2}\frac{1}{\beta} - E_D(\mathbf{w})$$
$$= \frac{N}{2}\frac{1}{\beta} - \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2$$

$0 = \frac{N}{2}\frac{1}{\beta} - \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}^2$

$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{ML}^T\phi(\mathbf{x}_n)\}^2$

- We can gain some insight into the role of the bias $w_0$ by make $w_0$ explicit.

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

  Setting the derivative with respect $w_0$ to 0 and get:

  $w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j\bar{\phi}_j$

  Where $\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n$ and $\bar{\phi_j} = \frac{1}{N}\sum_{n=1}^{N}\phi_j(\mathbf{x}_n)$

  So, $w_0$ compensates for the difference between the averages (over the training set) of the target values and the weighted sum of the averages of the basis function values.

## Regularised least squares

Adding a regularisation term to the error function in order to control over-fitting.

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

Where $E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$

Then the error function becomes (linear ridged regression):

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

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

If we now setting the gradient of the error function with respect to $\mathbf{w}$ to 0:

$\frac{\partial(E_D(\mathbf{w})+\lambda E_W(\mathbf{w}))}{\partial \mathbf{w}} = \sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}\phi(\mathbf{x}_n)^T + \lambda w$

$0 = \sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\}\phi(\mathbf{x}_n)^T + \lambda w$

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

$\mathbf{w}_{ML} = \mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{t}$ [*]

*(Apply Woodbury push-through identity:
$(\alpha\mathbf{I} + cUV)^{-1}U = U(\alpha\mathbf{I} + cVU)^{-1}$)

**Prediction:**

$$\begin{aligned} t^* &= \mathbf{w}_{ML}^T\phi(\mathbf{x}^*) \\ &= \mathbf{t}^T\mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{I})^{-1} * \phi(\mathbf{x}^*) \\ &= \mathbf{t}^T(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{\Phi}\phi(\mathbf{x}^*) \end{aligned}$$

$$\begin{bmatrix} \cdots & \phi(\mathbf{x}_1)^T & \cdots \\ & \vdots & \\ \cdots & \phi(\mathbf{x}_n)^T & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \phi(\mathbf{x}^*) \\ \vdots \end{bmatrix} = \begin{bmatrix} \phi(\mathbf{x}_1)^T\phi(\mathbf{x}^*) \\ \vdots \\ \phi(\mathbf{x}_n)^T\phi(\mathbf{x}^*) \end{bmatrix} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}^*) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}^*) \end{bmatrix}$$

$\therefore t^* = \mathbf{t}^T(\mathbf{k} + \lambda\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}^*)$

Where $\mathbf{k}(\mathbf{x}^*) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}^*) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}^*) \end{bmatrix}$

## Overview of Linear Regression

| | Parametric | Non-parametric |
|---|---|---|
| Non-Bayesian | **Regularised linear regression** (also called linear ridged regression) | **Regularised kernel regression** (also called kernel ridged regression) |
| Bayesian | Bayesian linear regression | Bayesian kernel regression (also called Gaussian Process regression) |

3 Key components in machine learning:

- Model

  How data should behave
- Train

  How to find best model
- Predict

  How to predict data from model

## Practice

1. Explain why maximising the likelihood of logistic regression directly is a bad idea. What can be done to fix the problem?

2. Consider a linear model of the form

   $y(x, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i$

   And the sum-of-squares function:

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

   Now suppose the Gaussian noise $\epsilon_i \sim (0, \sigma^2)$ and is added independently to each of the input variables $x_i$. By making use of $E[\epsilon_i] = 0$ and $E[\epsilon_i \epsilon_j] = \delta_{ij}\sigma^2$.

   Show that minimising $E_D$ averaged over the noise distribution is equivalent to minimising the sum-of-squares error for noise-free input variables with the addition of a weight-decay regularisation term, in which the bias parameter $w_0$ is omitted from the regulariser.