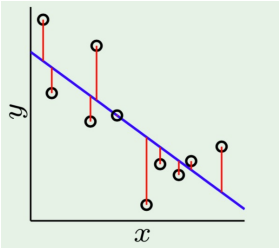
Lecture 6: Linear Regression

Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

1

Regression of a Line

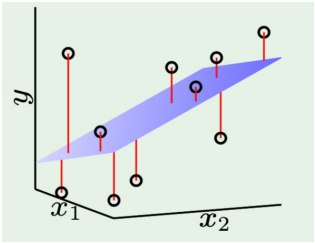


$$y = w_0 + w_1 \cdot x = a + b \cdot x$$

With w_0 being the intercept term and w_1 being the slop of the line. With $w_0 = 0$ (absent) the line goes through the origin.

2

Linear Regression



Simples linear model is the linear combination

$$y = y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{D} w_j \cdot x_j = w_0 + \langle \mathbf{w} | \mathbf{x} \rangle$$

Parameters w_j are values that control the behaviour of the system.

3

Bias

The intercept term w_0 is often called the bias parameter of the affine transformation. The output of the transformation y is biased toward being w_0 in the absence of any input. This term is different from the idea of a statistical bias!

In Neural Networks $net = y$

$$net = bias + \sum_{j=1}^{D} w_j \cdot x_j = w_0 + \sum_{j=1}^{D} w_j \cdot x_j$$

With $x_0 = 1$

$$y = y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{D} w_j \cdot x_j = \langle \mathbf{w} | \mathbf{x} \rangle = \mathbf{w}^T \cdot \mathbf{x}$$

4

Mean-squared-error (MSE)

Training set consists on N observations (sample)

$$X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\eta, \cdots, \mathbf{x}_N)^T$$

together with the values

$$\mathbf{t} = (t_1, t_2, \cdots, t_\eta, \cdots, t_N)^T$$

Mean-squared-error (MSE) over all N training points is defined as

$$E(\mathbf{w}) = \frac{1}{N} \cdot \sum_{\eta=1}^{N} (y(\mathbf{x}_\eta, \mathbf{w}) - t_\eta)^2 = \frac{1}{N} \cdot \|\mathbf{y} - \mathbf{t}\|^2$$

5

Sum-of-squares error

Sum-of-squares error function over all N training points is defined as

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (y(\mathbf{x}_\eta, \mathbf{w}) - t_\eta)^2 = \frac{1}{2} \cdot \|\mathbf{y} - \mathbf{t}\|^2 = \frac{1}{2} \cdot \|\mathbf{t} - \mathbf{y}\|^2$$

It is scaled by $1/2$ Euclidean distance between the predictions and the target values.

$$E(\mathbf{w}) = \frac{1}{2} \cdot \left\| \begin{pmatrix} t_1 \\ \vdots \\ t_\eta \\ \vdots \\ t_N \end{pmatrix} - \begin{pmatrix} y_1 \\ \vdots \\ y_\eta \\ \vdots \\ y_N \end{pmatrix} \right\|^2$$

6

Design Matrix

Data matrix with $x_{j,0} = 1$, also called design matrix is represented as

$$X = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_\eta^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} = \begin{pmatrix} x_{1,0} & x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,0} & x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots & \\ x_{N,0} & x_{N,1} & x_{N,2} & \cdots & x_{N,D} \end{pmatrix}$$

7

Linear Mapping

$$\begin{pmatrix} y_1 \\ \vdots \\ y_\eta \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & x_{N,1} & x_{N,2} & \cdots & x_{N,D} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_j \\ \vdots \\ w_D \end{pmatrix}$$

$$\mathbf{y} = X \cdot \mathbf{w} = \left(\mathbf{w}^T \cdot X^T \right)^T$$

8

Error Functions

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - y(\mathbf{x}_\eta, \mathbf{w}))^2 = \frac{1}{2} \cdot \|\mathbf{t} - \mathbf{y}\|^2$$

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} \left(t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta\right)^2 = \frac{1}{2} \cdot \sum_{\eta=1}^{N} \left(t_\eta - \mathbf{x}_\eta^T \cdot \mathbf{w}\right)^2$$

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

9

Least-Squares Estimation

We set the gradient of $E(\mathbf{w})$ to zero with the gradient operator

$$\nabla = \left[\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \cdots, \frac{\partial}{\partial w_D}\right]^T$$

$$\nabla E(\mathbf{w}) = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \cdots, \frac{\partial E}{\partial w_D}\right]^T$$

$$\nabla E(\mathbf{w}) = \nabla \left(\frac{1}{2} \cdot (\mathbf{t} - X \cdot \mathbf{w})^T \cdot (\mathbf{t} - X \cdot \mathbf{w})\right) = 0$$

10

The gradient rules

$$\nabla(a \cdot f(\mathbf{w}) + b \cdot g(\mathbf{w})) = a \cdot \nabla f(\mathbf{w}) + b \cdot \nabla g(\mathbf{w}), \quad a, b \in \mathbb{R}$$

and for $A = X^T \cdot X$ symmetric

$$\nabla\left((\mathbf{w}^T \cdot A \cdot \mathbf{w}\right) = 2 \cdot A \cdot \mathbf{w}$$

$$\nabla_w\left(\mathbf{t}^T \mathbf{w}\right) = \mathbf{t}$$

11

$$\nabla E(\mathbf{w}) = \nabla\left(\frac{1}{2} \cdot (\mathbf{t} - X \cdot \mathbf{w})^T \cdot (\mathbf{t} - X \cdot \mathbf{w})\right) = 0$$

$$\nabla\left(\mathbf{t}^T \cdot \mathbf{t} - 2 \cdot \mathbf{t}^T \cdot X \cdot \mathbf{w} + \mathbf{w}^T \cdot X^T \cdot X \cdot \mathbf{w}\right) = 0$$

$$\nabla\left(\mathbf{t}^T \cdot \mathbf{t}\right) - 2 \cdot \nabla\left(\mathbf{t}^T \cdot X \cdot \mathbf{w}\right) + \nabla\left(\mathbf{w}^T \cdot X^T \cdot X \cdot \mathbf{w}\right) = 0$$

$$-2 \cdot X^T \cdot \mathbf{t} + 2 \cdot X^T \cdot X \cdot \mathbf{w} = 0$$

$$X^T \cdot \mathbf{t} - X^T \cdot X \cdot \mathbf{w} = 0$$

$$X^T \cdot \mathbf{t} = X^T \cdot X \cdot \mathbf{w}$$

$$\left(X^T \cdot X\right)^{-1} \cdot X^T \cdot \mathbf{t} = \mathbf{w}$$

The matrix

$$X^\dagger = \left(X^T \cdot X\right)^{-1} \cdot X^T$$

$X^\dagger$ is Moore-Penrose or the pseudo-inverse of X.

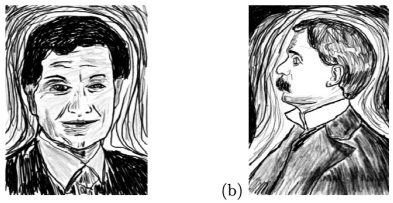
12

Moore-Penrose Matrix

$$X^{\dagger} = \left(X^T \cdot X\right)^{-1} \cdot X^T$$

With this new matrix defined, we can say that our **closed-form solution** is given by

$$X^{\dagger} \cdot \mathbf{t} = \mathbf{w}.$$



(a) Roger Penrose and (b) Eliakim Hastings Moore.

13

Let us go through an example where we employ our closed-form solution. In this example, we are given a training set that consists of 4 observations

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 3 \\ 3 \end{pmatrix},$$

with the corresponding targets

$$t_1 = 1.4, t_2 = 0.5, t_3 = 2, t_4 = 2.5.$$

We can represent our sample by a design matrix with $x_{j,0} = 1$ as

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 3 \\ 1 & 3 & 3 \end{pmatrix}$$

14

and a target vector

$$\mathbf{t} = \begin{pmatrix} 1.4 \\ 0.5 \\ 2 \\ 2.5 \end{pmatrix}.$$

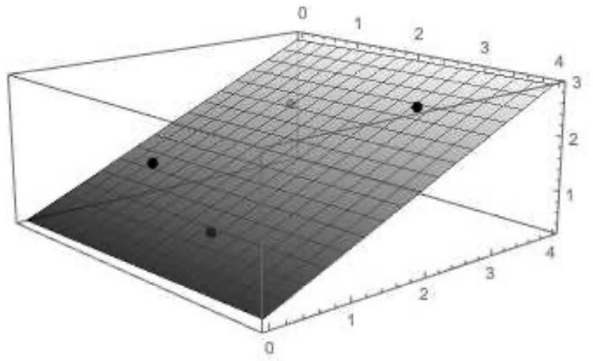
With that, we can employ our expression for the weights

$$\left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 3 \\ 1 & 1 & 3 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 3 \\ 1 & 3 & 3 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 3 \\ 1 & 1 & 3 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1.4 \\ 0.5 \\ 2 \\ 2.5 \end{pmatrix} = \mathbf{w}$$

to get the final solution

$$\begin{pmatrix} 0.275 \\ 0.02 \\ 0.645 \end{pmatrix} = \mathbf{w},$$

15



A plane y = y(w) = 0.275+0.02·x_1+0.645·x_2 is selected to find a best fit to a given data set.

16

- When some features are linear combinations of the others, or when $N < D$, the matrix $X^T X$ is not invertible, it is said to be *singular* or *degenerate*
  - D is the number of features
  - N number of examples
- However, the pseudo-inverse is always defined: it is based on the SVD decomposition of the matrix X

17

Moore-Penrose Pseudoinverse

- X is a $N \times D$ matrix, then SVD decomposition of the matrix X *is*

  $X=U \cdot D \cdot V^T$
  - U is an $N \times N$ orthogonal matrix
  - D is a diagonal $N \times D$ matrix with non-negative real numbers on the diagonal
    - the diagonal entries are the singular values, the square roots of eigenvalues V is an $D \times D$ orthogonal matrix

  and

  $X^{\dagger} = V \cdot D^{\dagger} \cdot U^T$
    - We get the pseudo-inverse of $D^{\dagger}$ by taking the reciprocal of each non-zero element on the diagonal, leaving the zeros in place, and then transposing the matrix.

18

Non-linear regression - Linear Basis Function Models

- Central idea of non-linear regression: same as linear regression, just with non-linear features
- Non-linear regression is the linear combination of fixed nonlinear functions

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

with $\phi_0(x) = 1$

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \cdot \phi_j(\mathbf{x}) = \langle \mathbf{w} | \Phi(\mathbf{x}) \rangle = \mathbf{w}^T \Phi(\mathbf{x})$$

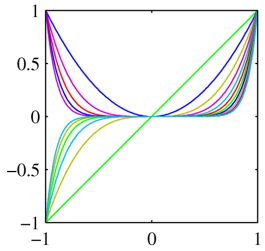
19

Linear Basis Function Models

- Polynomial basis functions:

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

- These are global; a small change in x affect all basis functions.



20

Non-linear regression - Linear Basis Function Models

- Non-linear regression is the linear combination of fixed nonlinear functions

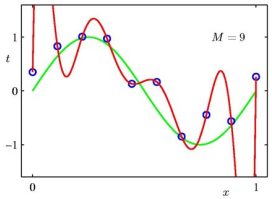
$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \cdot \phi_j(\mathbf{x})$$

with $\phi_0(x) = 1$

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \cdot \phi_j(\mathbf{x}) = \langle \mathbf{w} | \Phi(\mathbf{x}) \rangle = \mathbf{w}^T \Phi(\mathbf{x})$$

21

D=1



One should note that D and $M-1$ do not need to agree. For example with basis function power of x for $D = 1$

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

and $M - 1 = 9$

$$y(x, \mathbf{w}) = w_0 + \sum_{j=1}^{9} w_j \cdot x^j = \sum_{j=0}^{9} \phi_j(x)$$

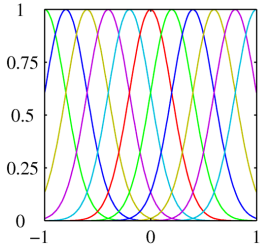
and we have to determine $M = 10$ parameters.

22

Linear Basis Function Models

- Gaussian basis functions:

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



•These are local; a small change in x only affect nearby basis functions. μ_j and *s* control location and scale (width).

23

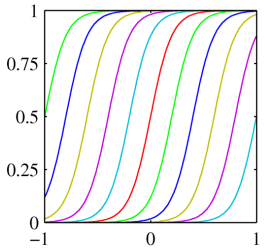
Linear Basis Function Models

- Sigmoidal basis functions:

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

- where

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



•Also these are local; a small change in *x* only affect nearby basis functions. μ_j and s control location and scale (slope).

24

With

$$\Phi_{\eta,j} = \phi_j(\mathbf{x}_\eta)$$

- Dimensions change since the dimension are not determined by the dimension of the vector **x** which is D
- The number of the is M-1

$$\begin{pmatrix} y_1 \\ \vdots \\ y_\eta \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,M-1} \\ 1 & \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,M-1} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & \phi_{N,1} & \phi_{N,2} & \cdots & \phi_{N,M-1} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_j \\ \vdots \\ w_{M-1} \end{pmatrix}$$

with $\Phi^\dagger$ is Moore-Penrose or the pseudo-inverse of Φ as before with

$$\Phi^\dagger = \left(\Phi^T \cdot \Phi\right)^{-1} \cdot \Phi^T$$

25

Example Polynomial Regression

$$\mathbf{x}_1 = \left(0.8\right), \mathbf{x}_2 = \left(1\right), \mathbf{x}_3 = \left(1.2\right), \mathbf{x}_4 = \left(1.4\right), \mathbf{x}_5 = \left(1.6\right),$$

with targets

$$t_1 = 24, t_2 = 20, t_3 = 10, t_4 = 13, t_5 = 12.$$

Using as basis functions the first $M = 4$ powers of x

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

we get a polynomial regression

$$y(x, \mathbf{w}) = \sum_{j=0}^{3} w_j \cdot \phi_j(x) = w_0 + w_1 \cdot x + w_2 \cdot x^2 + w_3 \cdot x^3.$$

26

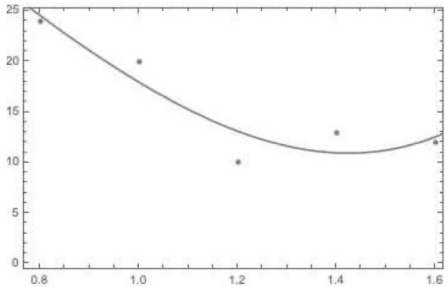
$$w_0 + w_1 \cdot x + w_2 \cdot x^2 + w_3 \cdot x^3$$

$$\Phi = \begin{pmatrix} 1 & 0.8 & 0.64 & 0.512 \\ 1 & 1 & 1 & 1 \\ 1 & 1.2 & 1.44 & 1.728 \\ 1 & 1.4 & 1.96 & 2.744 \\ 1 & 1.6 & 2.56 & 4.096 \end{pmatrix} \qquad \Phi^\dagger = \left(\Phi^T \cdot \Phi\right)^{-1} \cdot \Phi^T.$$

$$\Phi^\dagger = \begin{pmatrix} 22.5571 & -34.2286 & -4.65714 & 29.7714 & -12.4429 \\ -53.1548 & 90.9524 & 8.57143 & -82.381 & 36.0119 \\ 41.0714 & -76.7857 & -3.57143 & 73.2143 & -33.9286 \\ -10.4167 & 20.8333 & 0 & -20.8333 & 10.4167 \end{pmatrix}$$

27

- The closed-form solution is $\mathbf{w} = \Phi^\dagger \cdot \mathbf{t}$



The polynomial y(x, w) = 47.9429 − 9.7619 · x − 41.0714 · x_2 + 20.8333 · x_3 is selected to find a best fit to a given data set

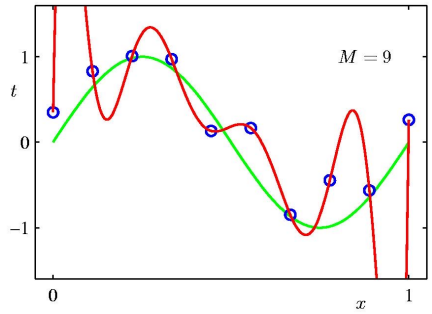
28

Non-linear regression - Linear Basis Function Models

- With many features, our prediction function becomes very expressive
- Can lead to overfitting
  - Low error on input data points, but high error nearby

29

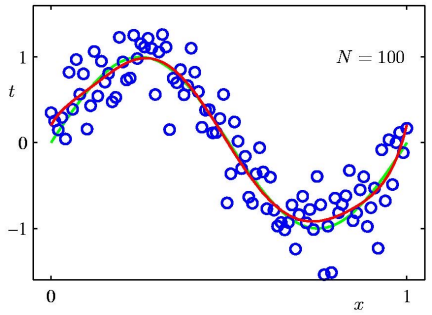
9th Order Polynomial



30

Data Set Size: $N = 100$

9th Order Polynomial



31

Bayesian Regression

$$p(\mathbf{w}|D) = \frac{P(D|\mathbf{w}) \cdot P(\mathbf{w})}{p(D)}$$

- *P(D|**w**)* is evaluated on the observed data set D and is called likelihood function.
  It indicates how probable the observed data set is for different settings of ***w***.
- Given likelihood we can state $posterior \propto likelihood \times prior$
  - *posterior* is related in a linear manner to *likelihood x prior*
- All parameters are viewed as a function of ***w***

32

Bayesian Regression

$$p(\mathbf{w}|D) = \frac{P(D|\mathbf{w}) \cdot P(\mathbf{w})}{p(D)}$$

- *p(D)* is a normalisation constant which ensures that *p(**w**|D)* is a valid probability density
- In frequentist paradigms ***w*** is considered as a **fixed parameter** determined by some estimator and errors are observed by considering the dataset *D*
- By Bayesian viewpoint there is only a dataset *D* and the **uncertainty** is represented by the distribution **w**

33

Maximising ML and MAP

- Maximising the likelihood (ML) is

$$\mathbf{w}_{ML} = \arg\max_{\mathbf{w}} p(D|\mathbf{w})$$

- Since **log** is monotically increasing function

$$\mathbf{w}_{ML} = \arg\max_{\mathbf{w}} \log(p(D|\mathbf{w}))$$

- Maximising a posteriori (MAP) is

$$\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} \log(p(\mathbf{w}|D))$$

34

Bayesian Learning

- We know that likelihood function is $p(t_\eta | \mathbf{w}, \mathbf{x}_\eta)$
- $\mathbf{w}$ in relation with $\mathbf{x}_\eta$ generates the data t_η
- What we liked is to have the posterior distribution $p(\mathbf{w} | t_\eta, \mathbf{x}_\eta)$
- what about $\mathbf{x}_\eta$?

$$p(\mathbf{w}, t_\eta) = p(\mathbf{w}|t_\eta) \cdot p(t_\eta) = p(t_\eta|\mathbf{w}) \cdot p(\mathbf{w})$$

and

$$p(\mathbf{w}, t_\eta|\mathbf{x}_\eta) = p(\mathbf{w}|t_\eta, \mathbf{x}_\eta) \cdot p(t_\eta) = p(t_\eta|\mathbf{w}, \mathbf{x}_\eta) \cdot p(\mathbf{w})$$

and we arrive at

$$p(\mathbf{w}|t_\eta, \mathbf{x}_\eta) = \frac{p(t_\eta|\mathbf{w}, \mathbf{x}_\eta) \cdot p(\mathbf{w})}{p(t_\eta)} \qquad p(\mathbf{w}|\mathbf{t}, X) \propto p(\mathbf{t}|\mathbf{w}, X) \cdot p(\mathbf{w})$$

35

Gaussian Environment

- The *N* examples $\mathbf{x}_\eta$ are drown independent from the same distribution. They are independent and identically distributed (iid).
- The environment environment for generating the training examples is Gaussian distributed. The error in the linear regression model is described by a Gaussian density function of zero mean and a common variance σ^2.
- The environment is stationary, the parameter vector w is fixed but unknown.

36

Likelihood

The Likelihood is

$$p(t_\eta|\mathbf{x}_\eta, \mathbf{w}, \sigma^2) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot \exp\left(-\frac{1}{2 \cdot \sigma^2} \cdot (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2\right)$$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2) = \prod_{\eta=1}^{N} p(t_\eta|\mathbf{x}_\eta, \mathbf{w}, \sigma^2)$$

results in total empirical knowledge about $\mathbf{w}$.

37

Precision

$$p(t_\eta|\mathbf{x}_\eta, \mathbf{w}, \sigma^2) = \mathcal{N}(t_\eta|\mathbf{w}^T\mathbf{x}_\eta, \sigma^2).$$

- Precision is often used in Bayesian software by convention.
- Some (Bishop) say that precision is more intuitive than variance because it says how concentrated are the values around the mean rather than how much spread they are.
- Precision is just an inverted variance

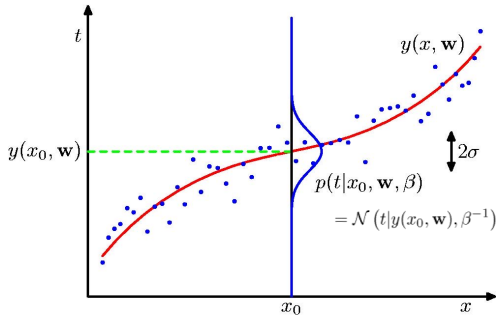
$$\beta = \frac{1}{\sigma^2}, \quad \beta^{-1} = \sigma^2$$

$$p(t_\eta|\mathbf{x}_\eta, \mathbf{w}, \beta) = \mathcal{N}(t_\eta|\mathbf{w}^T\mathbf{x}_\eta, \beta^{-1}).$$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{\eta=1}^{N} \mathcal{N}(t_\eta|\mathbf{w}^T\mathbf{x}_\eta, \beta^{-1})$$

38

Curve Fitting



Indicates how probable the observed data set is for different settings of ***w***

39

Likelihood

The Likelihood (without the precision notation) is

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2) = \frac{1}{\left(\sqrt{2 \cdot \pi} \cdot \sigma\right)^N} \prod_{\eta=1}^{N} \left(\exp\left(-\frac{1}{2 \cdot \sigma^2} \cdot (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2\right)\right)$$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2) = \frac{1}{\left(\sqrt{2 \cdot \pi} \cdot \sigma\right)^N} \exp\left(-\frac{1}{2 \cdot \sigma^2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2\right)$$

40

Prior

- *M* elements of the vector ***w*** are independent and identically distributed and described by a Gaussian density function of zero mean and a common variance

$$p(\mathbf{w}|\sigma_w^2) = \prod_{j=0}^{M-1} p(w_j|\sigma_w^2) = \prod_{j=0}^{M-1} \mathcal{N}(\mathbf{w}|0, \sigma_w^2)$$

$$p(\mathbf{w}|\sigma_w^2) = \frac{1}{\left(\sqrt{2 \cdot \pi} \cdot \sigma_w\right)^M} \prod_{j=0}^{M-1} \left(\exp\left(-\frac{w_j^2}{2 \cdot \sigma_w^2}\right)\right)$$

$$p(\mathbf{w}|\sigma_w^2) = \frac{1}{\left(\sqrt{2 \cdot \pi} \cdot \sigma_w\right)^M} \exp\left(-\frac{1}{2 \cdot \sigma_w^2} \sum_{j=0}^{M-1} w_j^2\right)$$

41

Prior

$$p(\mathbf{w}|\sigma_w^2) = \frac{1}{\left(\sqrt{2 \cdot \pi} \cdot \sigma_w\right)^M} \exp\left(-\frac{1}{2 \cdot \sigma_w^2} \sum_{j=0}^{M-1} w_j^2\right)$$

$$p(\mathbf{w}|\sigma_w^2) = \frac{1}{\left(\sqrt{2 \cdot \pi} \cdot \sigma_w\right)^M} \exp\left(-\frac{1}{2 \cdot \sigma_w^2} \mathbf{w}^T \cdot \mathbf{w}\right)$$

$$p(\mathbf{w}|\sigma_w^2) = \frac{1}{\left(\sqrt{2 \cdot \pi} \cdot \sigma_w\right)^M} \exp\left(-\frac{1}{2 \cdot \sigma_w^2} \|\mathbf{w}\|^2\right)$$

42

Posterior Density

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \sigma^2) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2) \cdot p(\mathbf{w}|\sigma_w^2)$$

Simplifying (no normalisation) we get

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \sigma^2) \propto \exp\left(-\frac{1}{2 \cdot \sigma^2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2 - \frac{1}{2 \cdot \sigma_w^2} \|\mathbf{w}\|^2\right)$$

43

Posterior Density

With

$$\lambda = \frac{\sigma^2}{\sigma_w^2}$$

we get

$$\mathbf{w}_{MAP}(N) = \max_{\mathbf{w}} \left(-\frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2 - \frac{\lambda}{2} \|\mathbf{w}\|^2\right)$$

because

$$\mathbf{w}_{MAP} = \arg\max_{\mathbf{w}} \log(p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \lambda))$$

44

Quadratic Function

- Now we can define the quadratic function, minimising it is equivalent to maximising $\boldsymbol{w}_{MAP}(N)$

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

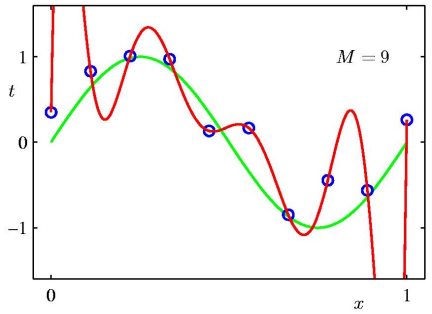
- We set the gradient of $E(\boldsymbol{w})$ to zero with the gradient operator

$$\nabla E(\mathbf{w}) = \nabla \left(\frac{1}{2} \cdot (\mathbf{t} - X \cdot \mathbf{w})^T \cdot (\mathbf{t} - X \cdot \mathbf{w}) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \right) = 0$$

$$-2 \cdot X^T \cdot \mathbf{t} + 2 \cdot X^T \cdot X \cdot \mathbf{w} + 2 \cdot \lambda \cdot \mathbf{w} = 0$$
$$-X^T \cdot \mathbf{t} + X^T \cdot X \cdot \mathbf{w} + \lambda \cdot \mathbf{w} = 0$$
$$X^T \cdot \mathbf{t} = \left(X^T \cdot X + \lambda \cdot I\right) \cdot \mathbf{w}$$
$$\left(X^T \cdot X + \lambda \cdot I\right)^{-1} \cdot X^T \cdot \mathbf{t} = \mathbf{w}$$

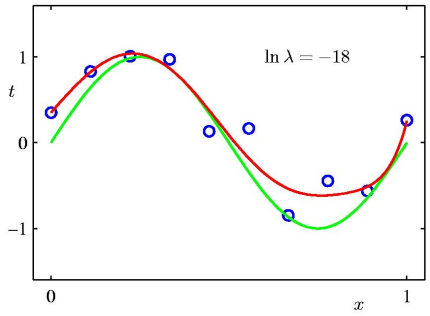
45

9th Order Polynomial



46

Regularization: $\ln \lambda = -18$



47

Relation between Regularised Least-Squares and MAP

Ordinary least-squares estimator

$$E_0(\mathbf{w}) = \frac{1}{2}\sum_{\eta=1}^{N}\left(t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta\right)^2$$

To overcome the problems one adds a new term in l_2 norm (We usually simplify $\|\mathbf{w}\|_2 = \|\mathbf{w}\|$)

$$E(\mathbf{w}) = \frac{1}{2}\cdot\sum_{\eta=1}^{N}(t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

which is identical to the MAP estimate.

48

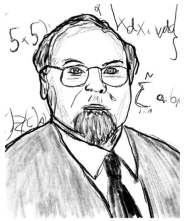
Tikhonov regularisation

The quadratic regulariser is called ridge regression or Tikhonov regularisation, named for Andrey Tikhonov

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2 + \|\Gamma \cdot \mathbf{w}\|_2^2$$

where is the Γ Tikhonov matrix with

$$\Gamma = I \cdot \frac{\lambda}{\sqrt{2}}$$



Andrey Nikolayevich Tikhonov a known Russian mathematician.

49

lasso

For the l_1 norm we have the lasso (least absolute shrinkage and selection operator)

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_1$$

For large λ some coefficient w_j are driven to zero leading to a sparse model. For the Bayesian interpretation it results from the MAP estimate where the prior distribution is Laplacian.

The prior is

$$p(\mathbf{w}|b) = \left(\frac{1}{2 \cdot b}\right)^M \cdot \prod_{j=0}^{M-1} \left(\exp\left(\frac{-|w_j|}{2 \cdot b}\right)\right)$$

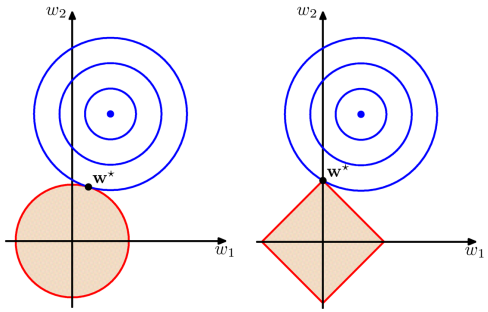
$$p(\mathbf{w}|b) = \left(\frac{1}{2 \cdot b}\right)^M \cdot \exp\left(-\frac{1}{2 \cdot b} \|\mathbf{w}\|_1\right)$$

$b > 0$ is referred to as the diversity, is a scale parameter.

50

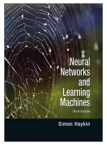
Regularized Least Squares

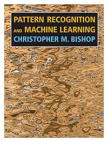
•Lasso tends to generate sparser solutions than a quadratic regularizer.



51

Literature

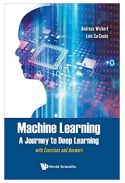




- Simon O. Haykin, Neural Networks and Learning Machine, (3rd Edition), Pearson 2008
  - Chapter 2
- Christopher M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer 2006
  - Section *1.1, 1.2.3, 1.2.4, 1.2.5, 1.2.6*
  - Chapter *3* till Section *3.3.3*
- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Aurélien Géron , O'Reilly Media; 1 edition, 2017
  - Chapter 4

52

Literature



- Machine Learning - A Journey to Deep Learning, A. Wichert, Luis Sa-Couto, World Scientific, 2021
  - Chapter 4

53