Aalto University
School of Electrical Engineering

CS-C3240 – Machine Learning D

Round 2: Regression

Dariush Salami

Department of Communications and Networking
Aalto University, School of Electrical Engineering
dariush.salami@aalto.fi

Version 1.4, September 9, 2022

Outline

Regression

Least squares estimation

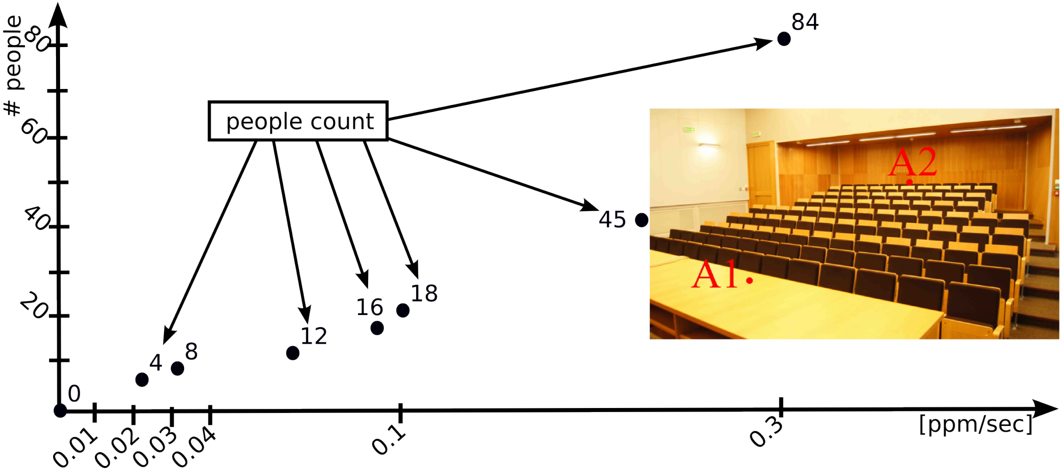
Polynomial regression

Multivariable linear regression

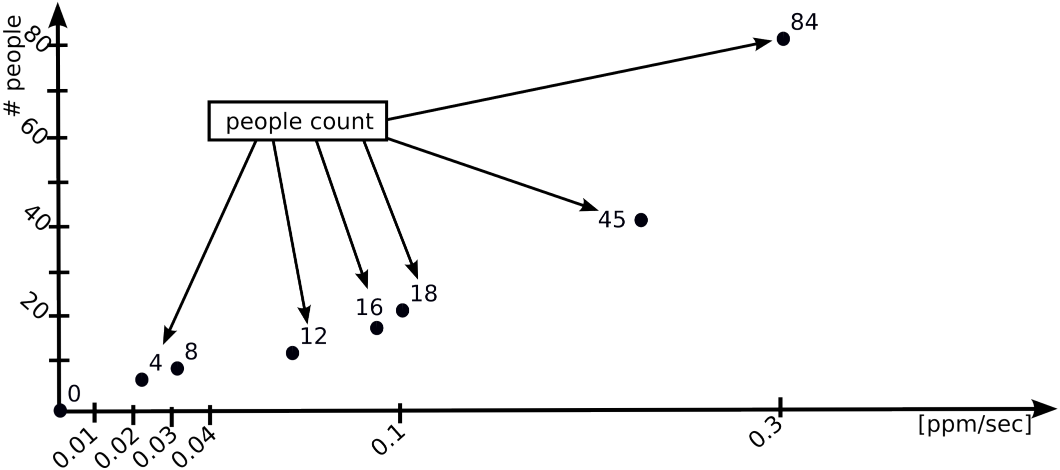
Multivariate linear regression

Logistic regression

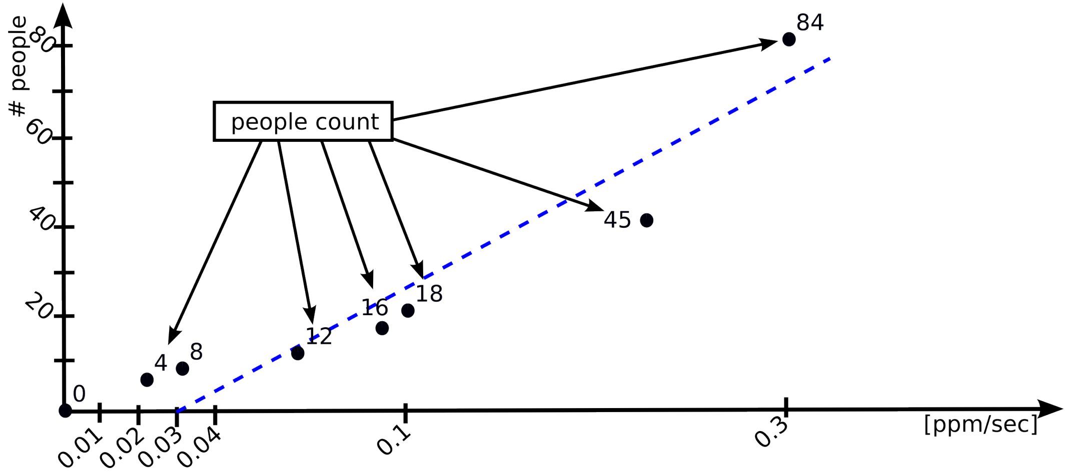
Linear regression



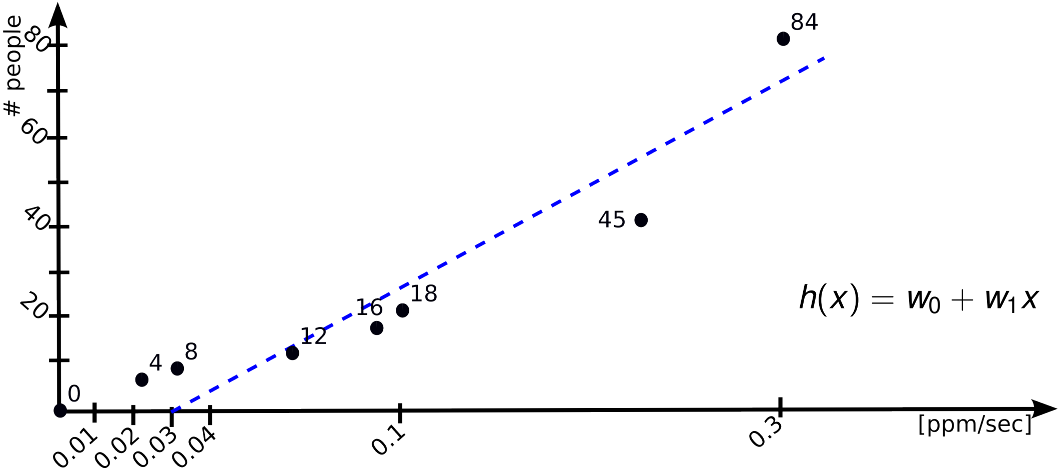
Linear regression



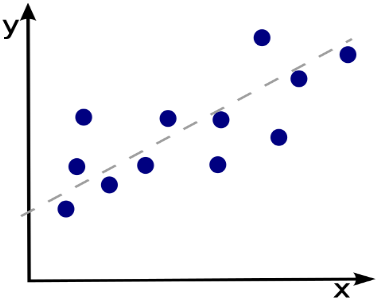
Linear regression



Linear regression



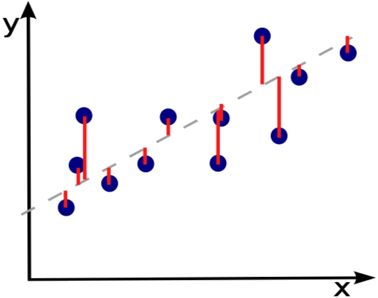
Linear regression



Objective function: $h(x) = w_0 + w_1 x$

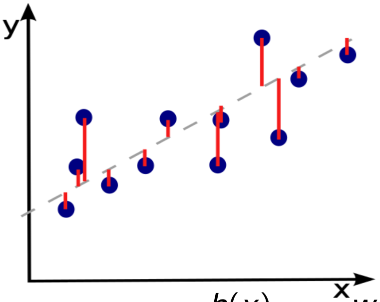
How to choose the parameter w_0 and w_1?

Linear regression



$$h(x) = w_0 + w_1 x$$

Linear regression

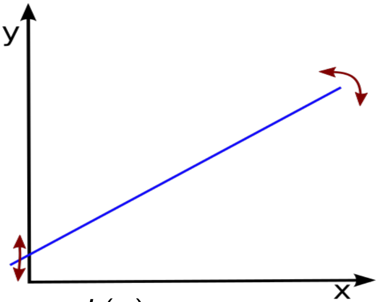


Some authors refer to the <u>Loss function</u> as *error function* or *cost function*. All these are synonyms.

$$
\begin{aligned}
h(x) &= w_0 + w_1 x \\
\text{minimize } E[w_0, w_1] = L[(X, Y), h(x)] &= \frac{1}{2n} \sum_{i=1}^{n} (h(x_i) - y_i)^2
\end{aligned}
$$

Loss function estimates quality of current solution (Gradient descent).

Gradient descent loss function – intuition

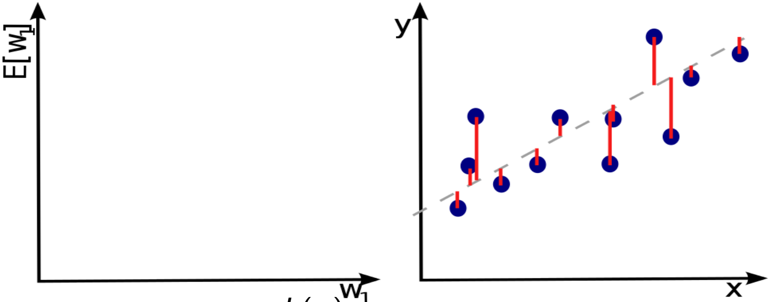


additive constant w_0 ignored in the following for simplicity

$$h(x) = w_0 + w_1 x$$

$$\text{minimize } E[w_0, w_1] = \frac{1}{2n} \sum_{i=1}^{n} (h(x_i) - y_i)^2$$

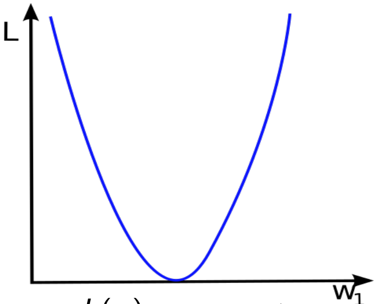
Gradient descent loss function – intuition



$$h(x) = w_0 + w_1 x$$

$$\text{minimize } E[w_0, w_1] = \frac{1}{2n}\sum_{i=1}^{n}(h(x_i) - y_i)^2$$

Gradient descent loss function – intuition



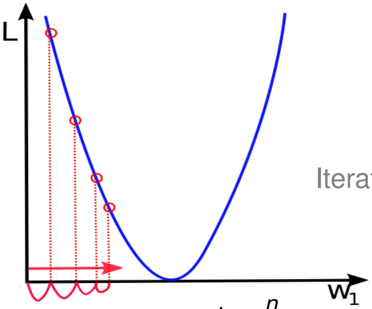
additive constant w_0
ignored in this figure

$$h(x) = w_0 + w_1 x$$

$$\text{minimize } E[w_0, w_1] = \frac{1}{2n}\sum_{i=1}^{n}(h(x_i) - y_i)^2$$

Gradient descent loss function – intuition

Gradient descent loss function

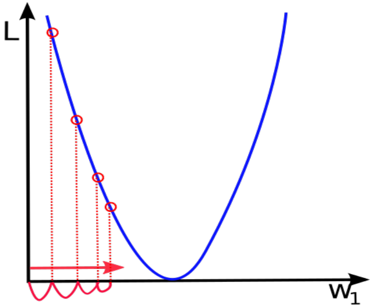


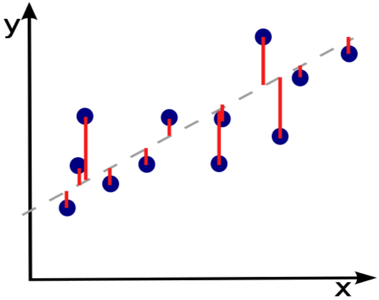
Iterative approximation of w_1

$$\text{minimize } E[w_0, w_1] = \frac{1}{2n}\sum_{i=1}^{n}(h(x_i) - y_i)^2$$

$$w_1 = w_1 - \delta \cdot \frac{\partial}{\partial w_1} E[w_0, w_1]$$

Gradient descent loss function





$$
\begin{aligned}
h(x) &= w_0 + w_1 x \\
\text{minimize } E[w_0, w_1] &= \frac{1}{2n}\sum_{i=1}^{n}\left(h(x_i) - y_i\right)^2
\end{aligned}
$$

Outline

Regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression

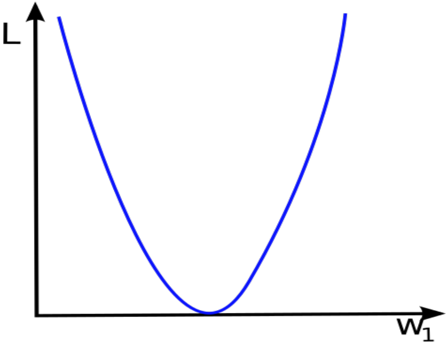
Least squares estimation

Given a loss function

$$E[w_0, w_1] = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i))^2$$

we can minimize the error by requiring

$$\frac{\partial E}{\partial w_0} = 0, \frac{\partial E}{\partial w_1} = 0$$



Least squares estimation

Given a loss function

$$E[w_0, w_1] = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_i))^2$$

we can minimize the error by requiring

$$\frac{\partial E}{\partial w_0} = 0, \frac{\partial E}{\partial w_1} = 0$$

Differentiation yields

$$\frac{\partial E}{\partial w_0} = \sum_{i=1}^{n} \frac{2}{2n} (y_i - (w_1 x_i + w_0)) \cdot 1$$

$$\frac{\partial E}{\partial w_1} = \sum_{i=1}^{n} \frac{2}{2n} (y_i - (w_1 x_i + w_0)) \cdot x_i$$

$$\frac{\partial E}{\partial w_0} = \sum_{i=1}^{n} \frac{2}{2n} (y_i - (w_1 x_i + w_0)) \cdot 1$$

$$\frac{\partial E}{\partial w_1} = \sum_{i=1}^{n} \frac{2}{2n} (y_i - (w_1 x_i + w_0)) \cdot x_i$$

Setting

$$\frac{\partial E}{\partial w_0} = \frac{\partial E}{\partial w_1} = 0$$

will lead to

$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) = 0$$

$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) \cdot x_i = 0$$

Least squares estimation

$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) = 0$$

$$\sum_{i=1}^{n} (y_i - (w_1 x_i + w_0)) \cdot x_i = 0$$

rewrite as

$$\left(\sum_{i=1}^{n} x_i\right) w_1 + \left(\sum_{i=1}^{n} 1\right) w_0 = \sum_{i=1}^{n} y_i$$

$$\left(\sum_{i=1}^{n} x_i^2\right) w_1 + \left(\sum_{i=1}^{n} x_i\right) w_0 = \sum_{i=1}^{n} x_i y_i$$

Least squares estimation

$$\left(\sum_{i=1}^{n} x_i\right) w_1 + \left(\sum_{i=1}^{n} 1\right) w_0 \quad = \sum_{i=1}^{n} y_i$$

$$\left(\sum_{i=1}^{n} x_i^2\right) w_1 + \left(\sum_{i=1}^{n} x_i\right) w_0 \quad = \sum_{i=1}^{n} x_i y_i$$

Consequently, values of w_0 and w_1 that minimize the error satisfy

$$\begin{pmatrix} \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \end{pmatrix} \begin{pmatrix} w_1 \\ w_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{pmatrix}$$

Least squares estimation

$$\begin{pmatrix} \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \end{pmatrix} \begin{pmatrix} w_1 \\ w_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{pmatrix}$$

for an invertible matrix this implies

$$\begin{pmatrix} w_1 \\ w_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{pmatrix}$$

Solve this linear equation system to find optimal values for w_0 and w_1.

Least squares estimation

$$\begin{pmatrix} \sum_{i=1}^n x_i & \sum_{i=1}^n 1 \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{pmatrix} \begin{pmatrix} w_1 \\ w_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}$$

for an invertible matrix this implies

$$\begin{pmatrix} w_1 \\ w_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i & \sum_{i=1}^n 1 \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}$$

Solve this linear equation system to find optimal values for w_0 and w_1.

Note: matrix must be invertible.

Outline

Regression

Least squares estimation

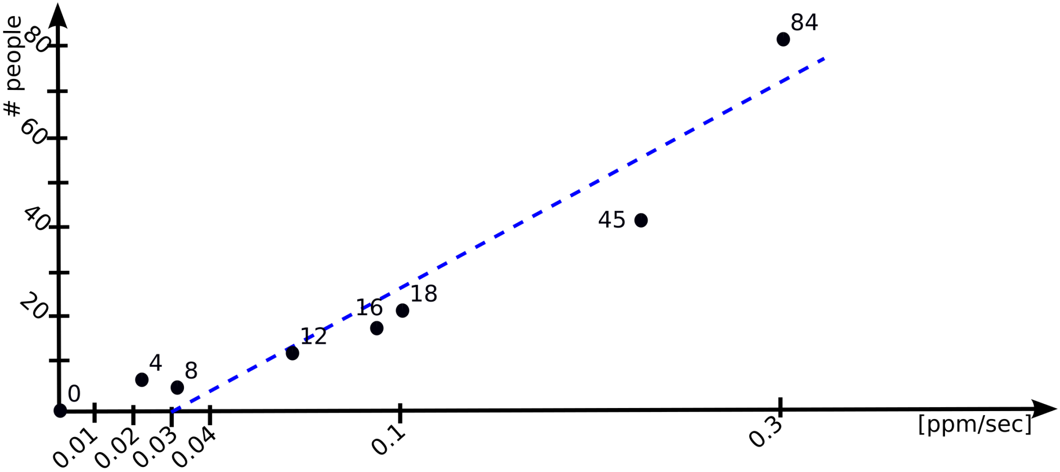
Polynomial regression

Multivariable linear regression

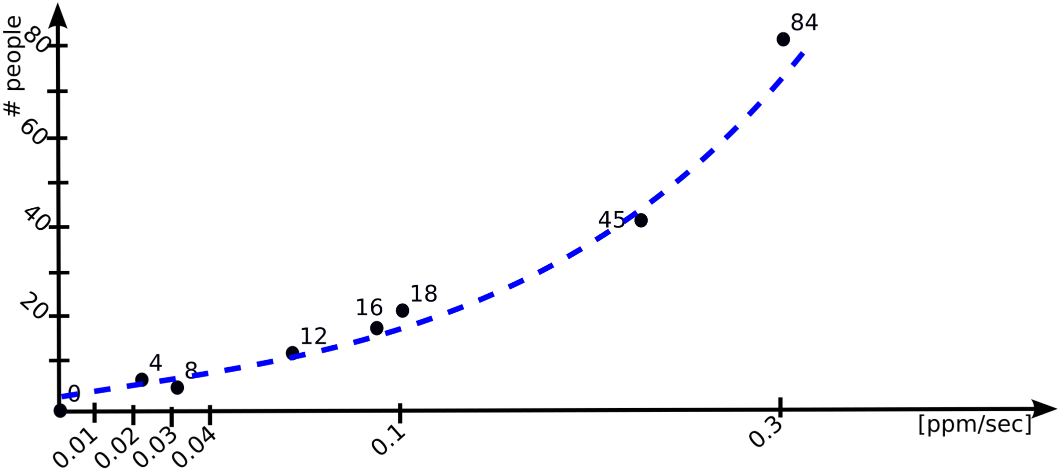
Multivariate linear regression

Logistic regression

Polynomial regression



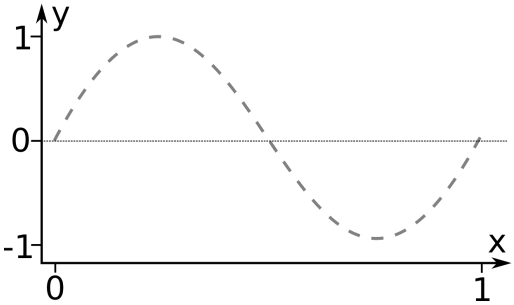
Polynomial regression



Polynomial regression (Polynomial curve fitting)

Example

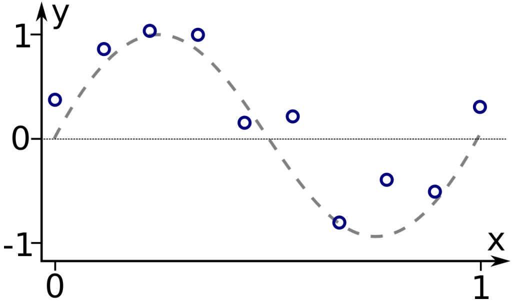
Sample points are created for the function $\sin(2\pi x) + \mathcal{N}$ where $\mathcal{N}$ is a random noise value



Polynomial regression (Polynomial curve fitting)

Example

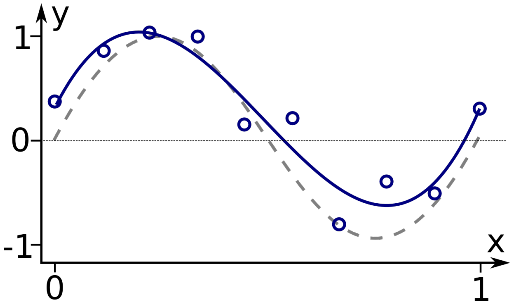
Sample points are created for the function $\sin(2\pi x) + \mathcal{N}$ where $\mathcal{N}$ is a random noise value



Polynomial regression (Polynomial curve fitting)

Example

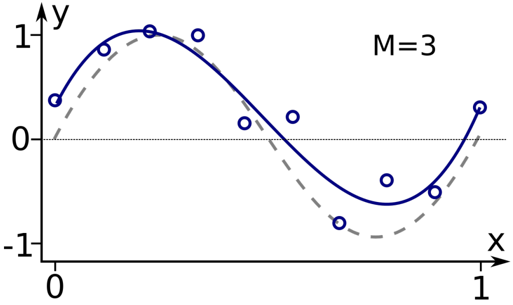
Sample points are created for the function $\sin(2\pi x) + \mathcal{N}$ where $\mathcal{N}$ is a random noise value



Polynomial curve fitting

We fit the data points into a polynomial function:

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



Polynomial curve fitting

We fit the data points into a polynomial function:

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

This can be obtained by minimising a loss function which measures the misfit between $h(x, \overrightarrow{w})$ and the training data set:

$$E(\overrightarrow{w}) = \frac{1}{2n} \sum_{i=1}^{n} \left[h(x_i, \overrightarrow{w}) - y_i \right]^2$$

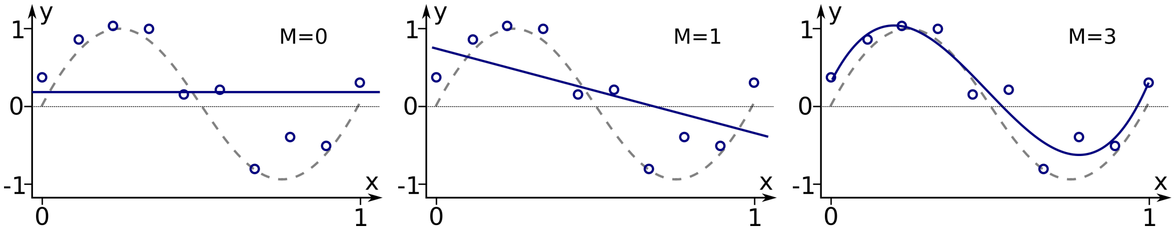
$E(\overrightarrow{w}) \geq 0$; $E(\overrightarrow{w}) = 0$ IFF all points are covered by the function

Polynomial curve fitting

One problem is the right choice of the dimension M

When M is too small, the approximation accuracy might be bad

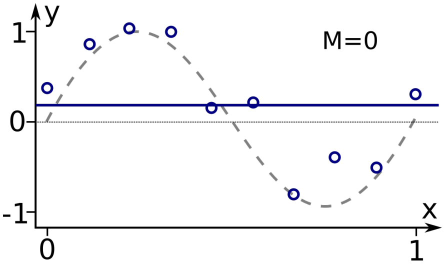
$$h(x, \overrightarrow{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

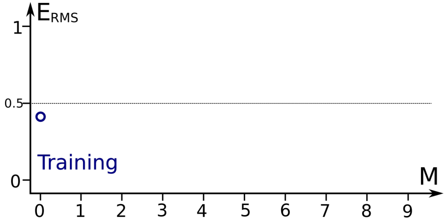


Polynomial curve fitting

Visualise error $E(\overrightarrow{w})$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{2E(\overrightarrow{w})}{n}}$$

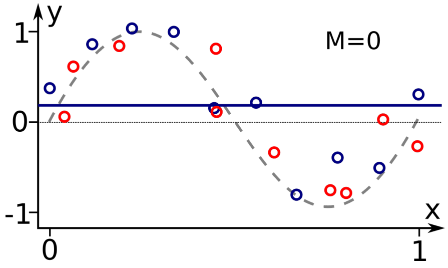


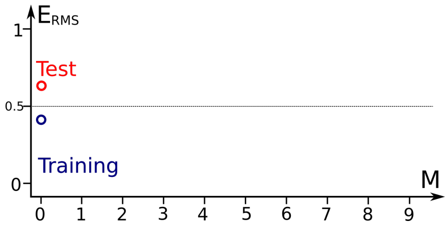


Polynomial curve fitting

Visualise error $E(\vec{w})$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{2E(\vec{w})}{n}}$$

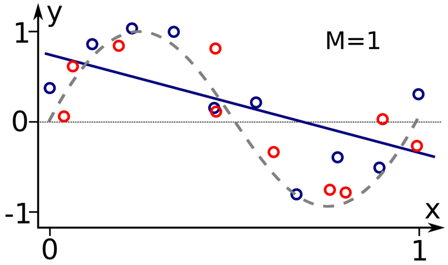


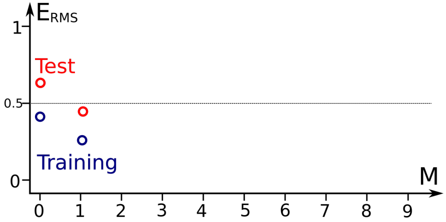


Polynomial curve fitting

Visualise error $E(\vec{w})$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{2E(\vec{w})}{n}}$$

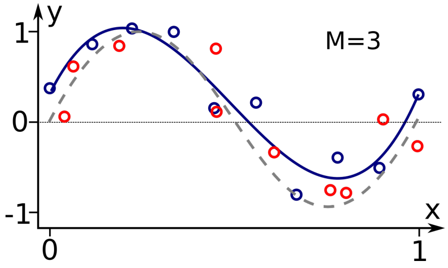


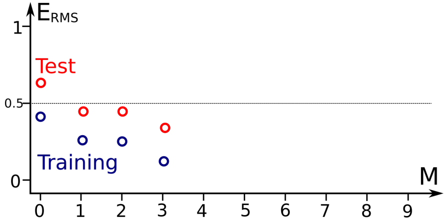


Polynomial curve fitting

Visualise error $E(\overrightarrow{w})$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{2E(\overrightarrow{w})}{n}}$$

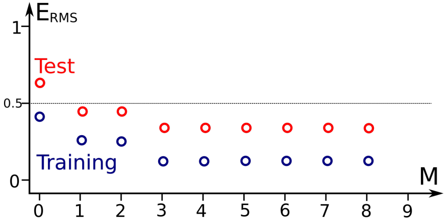




Polynomial curve fitting

Visualise error $E(\overrightarrow{w})$ wrt the data by Root of the Mean Squared (RMS)

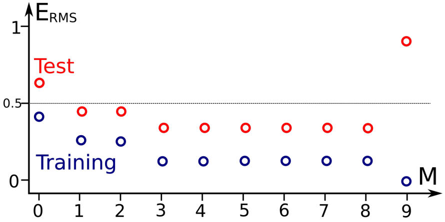
$$E_{RMS} = \sqrt{\frac{2E(\overrightarrow{w})}{n}}$$



Polynomial curve fitting

Visualise error $E(\overrightarrow{w})$ wrt the data by Root of the Mean Squared (RMS)

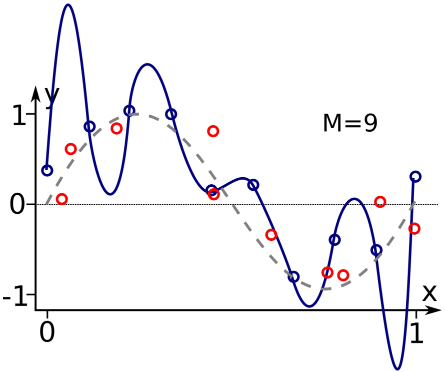
$$E_{RMS} = \sqrt{\frac{2E(\overrightarrow{w})}{n}}$$

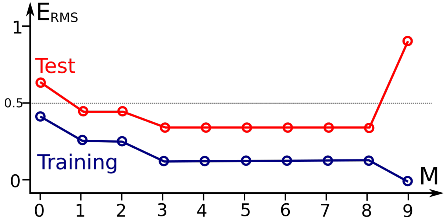


Polynomial curve fitting

Visualise error $E(\vec{w})$ wrt the data by Root of the Mean Squared (RMS)

$$E_{RMS} = \sqrt{\frac{2E(\vec{w})}{n}}$$



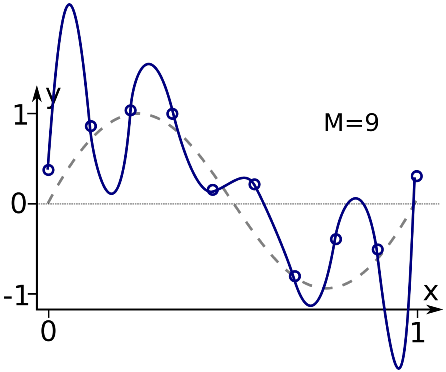


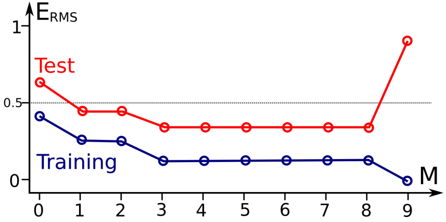
Polynomial curve fitting

This event is called overfitting

The polynomial is now trained too well to the training data

It performs badly on test data

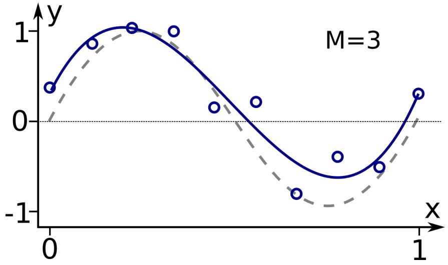


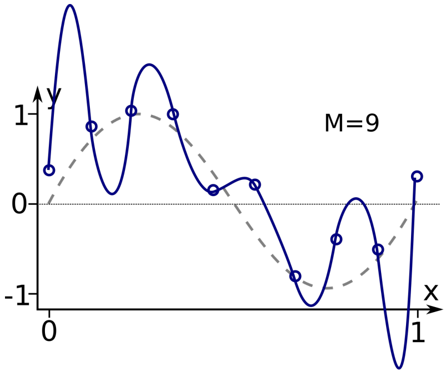


Polynomial curve fitting

When M becomes too big, the polynomial will cross all points exactly

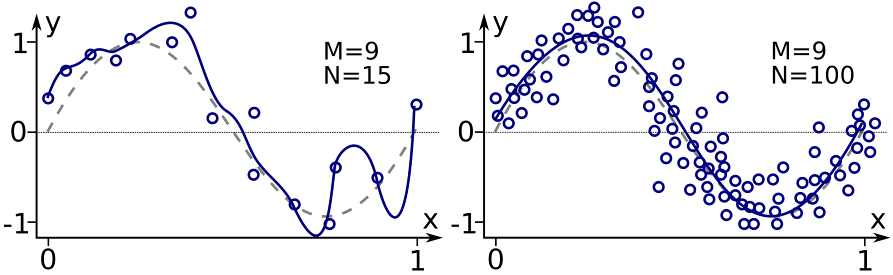
For $M = n$, it is always possible to create a polynomial of order M that contains all values in the data set.





Polynomial curve fitting

With increasing number of data points, the problem of overfitting becomes less severe for a given value of *M*



Polynomial curve fitting

One solution to cope with overfitting is regularisation

A penalty term is added to the loss function

This term discourages the coefficients $\overrightarrow{w}$ from reaching large values

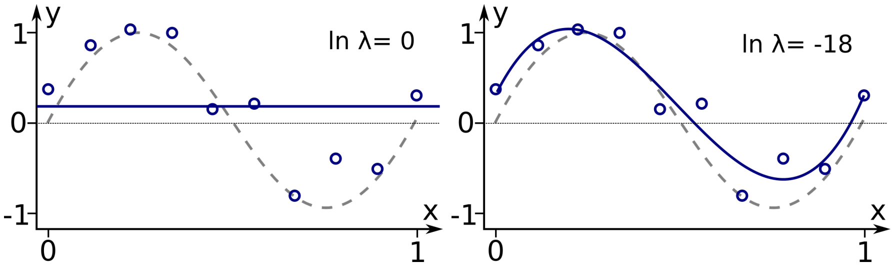
$$\overline{E}(\overrightarrow{w}) = \frac{1}{2n}\sum_{i=1}^{n}\left[h(x_i, \overrightarrow{w}) - y_i\right]^2 + \frac{\lambda}{2}||\overrightarrow{w}||^2$$

with

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

Polynomial curve fitting

Depending on the value of λ, overfitting is controlled



$$\overline{E}(\overrightarrow{w}) = \frac{1}{2n}\sum_{i=1}^{n}\left[h(x_i, \overrightarrow{w}) - y_i\right]^2 + \frac{\lambda}{2}||\overrightarrow{w}||^2$$


TIME FOR A
SHORT BREAK, RIGHT?
imgflip.com

Outline

Regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression

Multivariable linear regression

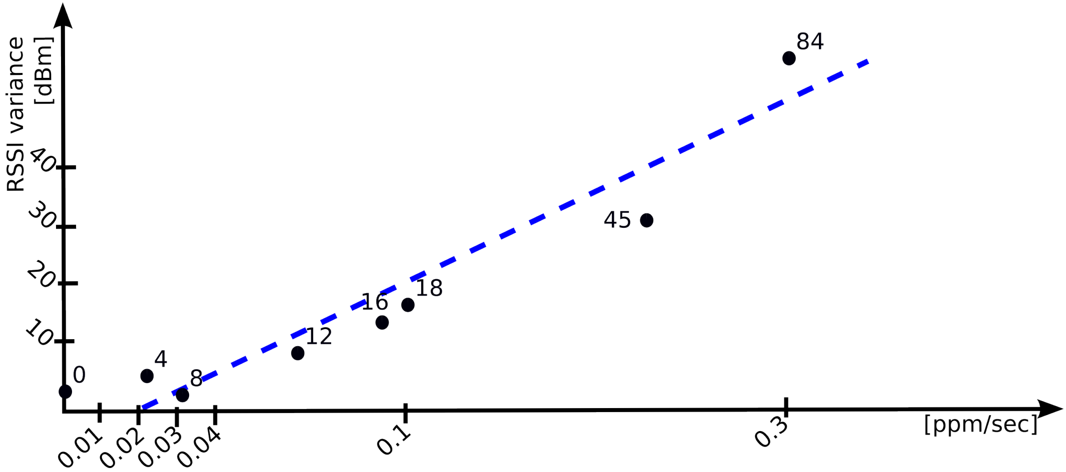
In multivariable linear regression problems we assume that multiple regression variables (features) apply.

Multivariable linear regression

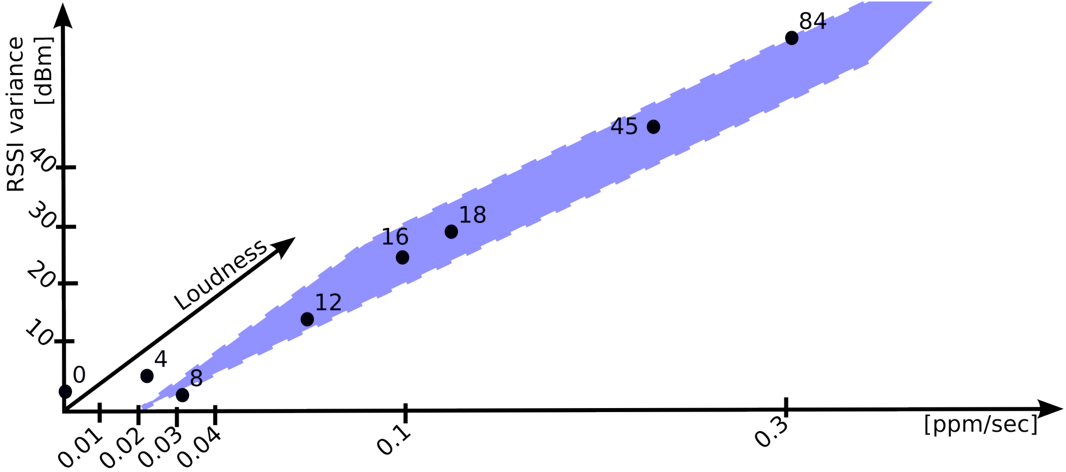
In multivariable linear regression problems we assume that multiple regression variables (features) apply.

$$
\begin{aligned}
h(x_{j1}, \ldots, x_{jm}) &= \sum_{i=0}^{m} w_i x_{ji} \\
\text{minimize } E[W] &= \frac{1}{2n} \sum_{j=1}^{n} \left(h(x_{j1}, \ldots, x_{jm}) - y_j \right)^2
\end{aligned}
$$

Multivariable linear regression



Multivariable linear regression



$$
\begin{aligned}
h(x_{j1}, \ldots, x_{jm}) &= \sum_{i=0}^{m} w_i x_{ji} \\
\text{minimize } E[W] &= \frac{1}{2n} \sum_{j=1}^{n} \left(h(x_{j1}, \ldots, x_{jm}) - y_j \right)^2
\end{aligned}
$$

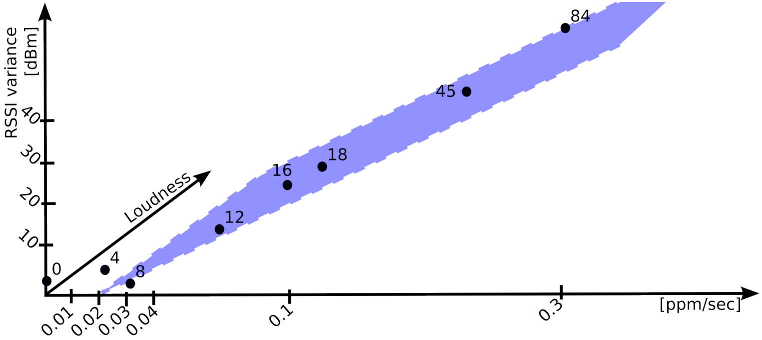
$$
\begin{aligned}
h(x_{j1}, \ldots, x_{jm}) &= \sum_{i=0}^{m} w_i x_{ji} \\
\text{minimize } E[W] &= \frac{1}{2n} \sum_{j=1}^{n} \left(h(x_{j1}, \ldots, x_{jm}) - y_j\right)^2 \\
w_i &= w_i - \delta \cdot \frac{\partial}{\partial w_i} E[W]
\end{aligned}
$$

w_i are optimised together over several iterations

Outline

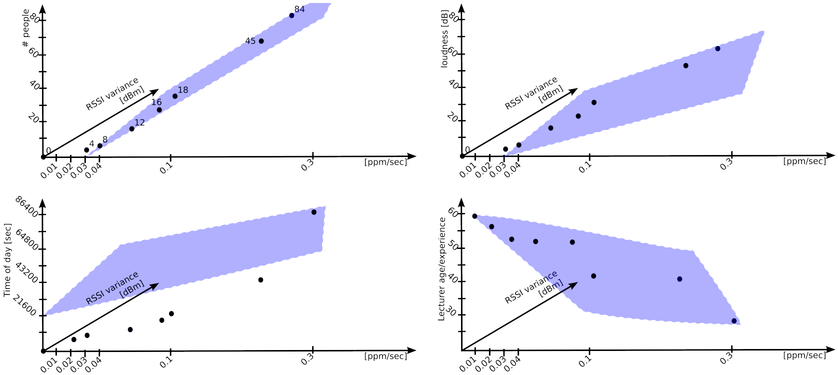
Multivariate linear regression

Multivariate linear regression describes a regression problem with multiple classes.



Multivariate linear regression

Multivariate linear regression describes a regression problem with multiple classes.



Multivariate linear regression

Regression model is extended to multiple responses: $Y_j = y_{j1}, \ldots, y_{jl}$

$$
\begin{aligned}
y_{j1} &= w_{01} + w_{11}x_{j1} + \cdots + w_{n1}x_{jn} \\
y_{j2} &= w_{02} + w_{12}x_{j1} + \cdots + w_{n2}x_{jn} \\
&\vdots \\
y_{jl} &= w_{0l} + w_{1l}x_{j1} + \cdots + w_{nl}x_{jn}
\end{aligned}
$$

Multivariate linear regression

Regression model is extended to multiple responses with respect to one class: $Y_j = y_{j1}, \ldots, y_{jl}$

$$
\begin{aligned}
y_{j1} &= w_{01} + w_{11}x_{j1} + \cdots + w_{n1}x_{jn} \\
y_{j2} &= w_{02} + w_{12}x_{j1} + \cdots + w_{n2}x_{jn} \\
\vdots & \qquad\qquad\quad \vdots \\
y_{jl} &= w_{0l} + w_{1l}x_{j1} + \cdots + w_{nl}x_{jn}
\end{aligned}
$$

Using least squares estimation it is then possible to estimate the regression coefficients associated with y_{ji} using only the i-th row of the matrix.

Multivariate linear regression

Using least squares estimation it is then possible to estimate the regression coefficients associated with y_{ji} using only the i-th row of the matrix.

$$W_i = \left(X^T X\right)^{-1} X^T Y_{(i)}$$

and collecting all univariate estimates into a matrix

$$W = \left(X^T X\right)^{-1} X^T Y$$

$Y_{(i)}$ is the vector of n measurements of the i-th variable
X^T denotes the transpose of X and X^{-1} its inverse

Multivariate linear regression - Proof

To minimize $E(W)$, we can simply solve for where its gradient is zero: $\nabla_W E(W) = 0$.

Multivariate linear regression - Proof

To minimize $E(W)$, we can simply solve for where its gradient is zero: $\nabla_W E(W) = 0$.

$$E(W) = \frac{1}{n}||\hat{y} - y||_2^2, \nabla_W E(W) = 0$$

Multivariate linear regression - Proof

To minimize $E(W)$, we can simply solve for where its gradient is zero: $\nabla_W E(W) = 0$.

$$E(W) = \frac{1}{n}||\hat{y} - y||_2^2, \nabla_W E(W) = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}||\hat{y} - y||_2^2 = 0$$

Multivariate linear regression - Proof

To minimize $E(W)$, we can simply solve for where its gradient is zero: $\nabla_W E(W) = 0$.

$$E(W) = \frac{1}{n}||\hat{y} - y||_2^2, \nabla_W E(W) = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}||\hat{y} - y||_2^2 = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}||XW - y||_2^2 = 0$$

Multivariate linear regression - Proof

To minimize $E(W)$, we can simply solve for where its gradient is zero: $\nabla_W E(W) = 0$.

$$E(W) = \frac{1}{n}||\hat{y} - y||_2^2, \nabla_W E(W) = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}||\hat{y} - y||_2^2 = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}||XW - y||_2^2 = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}(XW - y)^T(XW - y) = 0$$

Multivariate linear regression - Proof

To minimize $E(W)$, we can simply solve for where its gradient is zero: $\nabla_W E(W) = 0$.

$$E(W) = \frac{1}{n}||\hat{y} - y||_2^2, \nabla_W E(W) = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}||\hat{y} - y||_2^2 = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}||XW - y||_2^2 = 0$$

$$\Rightarrow \nabla_W \frac{1}{n}(XW - y)^T(XW - y) = 0$$

$$W = \left(X^T X\right)^{-1} X^T Y$$

Multivariable linear regression - Example (1/6)

Air Quality Sensor	RF Sensor	# People
210	3	400
160	3	330
240	3	370
140	2	230
300	4	500

1. The goal is to predict the number of people for air quality of 400 and RF activity level of 2.

Multivariable linear regression - Example (1/6)

Air Quality Sensor	RF Sensor	# People
210	3	400
160	3	330
240	3	370
140	2	230
300	4	500

1. The goal is to predict the number of people for air quality of 400 and RF activity level of 2.
2. First, we need to calculate w_0, w_1, and w_2 in $y = w_0 + w_1 x_1 + w_2 x_2$

Multivariable linear regression - Example (1/6)

Air Quality Sensor	RF Sensor	# People
210	3	400
160	3	330
240	3	370
140	2	230
300	4	500

1. The goal is to predict the number of people for air quality of 400 and RF activity level of 2.

3. Using $W = (X^T X)^{-1} X^T y$

Multivariable linear regression - Example (2/6)

Air Quality Sensor	RF Sensor	# People
210	3	400
160	3	330
240	3	370
140	2	230
300	4	500

$$X = \begin{bmatrix} 1 & 210 & 3 \\ 1 & 160 & 3 \\ 1 & 240 & 3 \\ 1 & 140 & 2 \\ 1 & 300 & 4 \end{bmatrix} \quad y = \begin{bmatrix} 400 \\ 330 \\ 370 \\ 230 \\ 500 \end{bmatrix}$$

Multivariable linear regression - Example (3/6)

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 210 & 160 & 240 & 140 & 300 \\ 3 & 3 & 3 & 2 & 4 \end{bmatrix} \begin{bmatrix} 1 & 210 & 3 \\ 1 & 160 & 3 \\ 1 & 240 & 3 \\ 1 & 140 & 2 \\ 1 & 300 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 1050 & 15 \\ 1050 & 236900 & 3310 \\ 15 & 3310 & 47 \end{bmatrix}$$

Multivariable linear regression - Example (4/6)

$$(X^T X)^{-1} = \begin{bmatrix} 4.95 & 0.0083333333 & -2.1666666667 \\ 0.008333333333 & 0.00027777778 & -0.02222222222 \\ -2.1666666667 & -0.0222222222 & 2.27777777778 \end{bmatrix}$$

Multivariable linear regression - Example (5/6)

$$X^T y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 210 & 160 & 240 & 140 & 300 \\ 3 & 3 & 3 & 2 & 4 \end{bmatrix} \begin{bmatrix} 400 \\ 330 \\ 370 \\ 230 \\ 500 \end{bmatrix} = \begin{bmatrix} 1730 \\ 386800 \\ 5460 \end{bmatrix}$$

Multivariable linear regression - Example (6/6)

$$(X^TX)^{-1}X^Ty = \begin{bmatrix} -43.166679742 \\ 0.52777864889 \\ 92.7777863278 \end{bmatrix}$$

$$\hat{y} = -43.166679742 + 0.52777864889 \times 400 + 92.7777863278 \times 4 = 539$$

Outline

Regression

Least squares estimation

Polynomial regression

Multivariable linear regression

Multivariate linear regression

Logistic regression

Logistic regression

Nominal classes

Classes might be nominal in real-world problems

Logistic regression

Nominal classes

Classes might be nominal in real-world problems

- **Weather** Sunny, rainy
- **Medical** positive diagnosis, negative diagnosis
- **Localisation** indoor, outdoor

Logistic regression

Nominal classes

Classes might be nominal in real-world problems

- **Weather** Sunny, rainy
- **Medical** positive diagnosis, negative diagnosis
- **Localisation** indoor, outdoor

In such case, classification is binary: $y \in \{0, 1\}$

Logistic regression

Nominal classes

Classes might be nominal in real-world problems

- **Weather** Sunny, rainy
- **Medical** positive diagnosis, negative diagnosis
- **Localisation** indoor, outdoor

In such case, classification is binary: $y \in \{0, 1\}$

Linear regression: $h(x)$ can be smaller than 0 or greater than 1

Logistic regression

Nominal classes

Classes might be nominal in real-world problems

Weather: Sunny, rainy

Medical: positive diagnosis, negative diagnosis

Localisation: indoor, outdoor

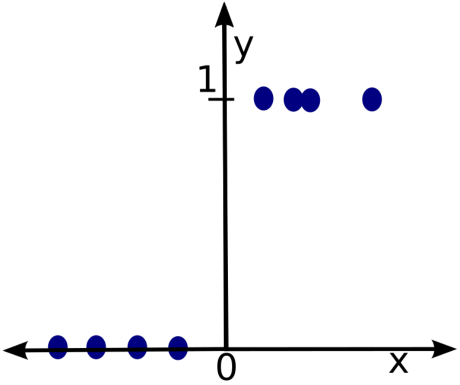
In such case, classification is binary: $y \in \{0, 1\}$

Linear regression: $h(x)$ can be smaller than 0 or greater than 1

Logistic regression: $0 \leq h(x) \leq 1$

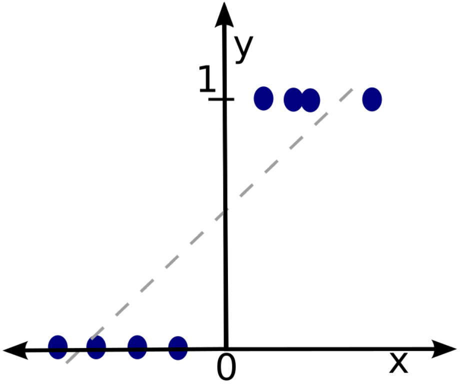
Logistic regression

Nominal classes



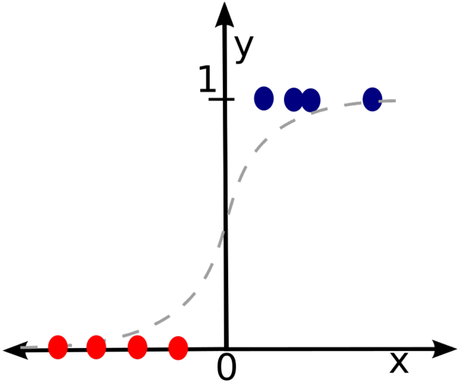
Logistic regression

Nominal classes



Logistic regression

Loss function



Logistic regression

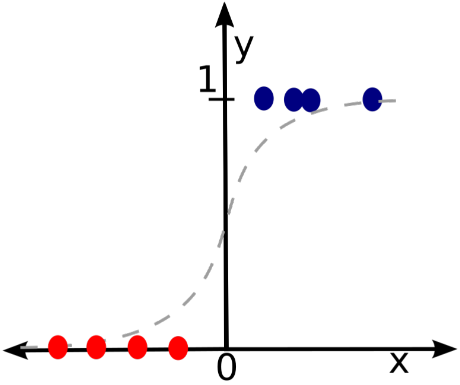
Loss function

Linear regression

$$h(x) = W^T x$$

Logistic regression

$$h(x) = \frac{1}{1+e^{-W^T x}}$$



Logistic regression

Loss function

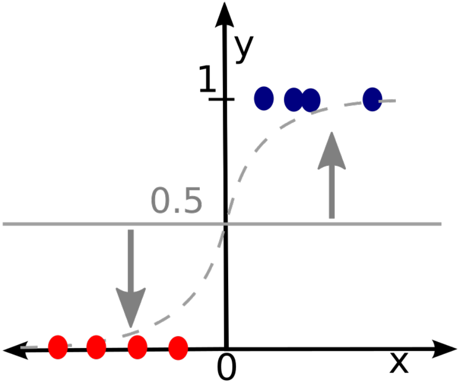
Linear regression

$$h(x) = W^T x$$

Logistic regression

$$h(x) = \frac{1}{1+e^{-W^T x}}$$

$$y = \begin{cases} 1 & \text{if } h(x) \geq 0.5 \\ 0 & \text{else} \end{cases}$$

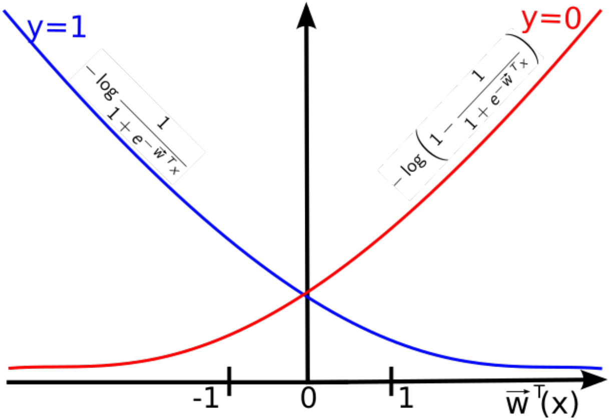


Logistic regression

Loss function

$$y = \begin{cases} 1 & \text{if } h(x) \geq 0.5 \\ 0 & \text{else} \end{cases}$$

$$E[h(x), y] = \begin{cases} -\log(h(x)) & \text{if } y = 1 \\ -\log(1 - h(x)) & \text{else} \end{cases}$$

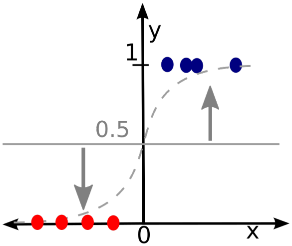


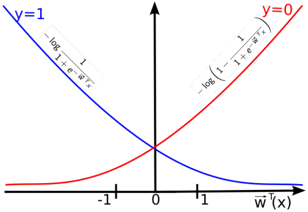
Logistic regression

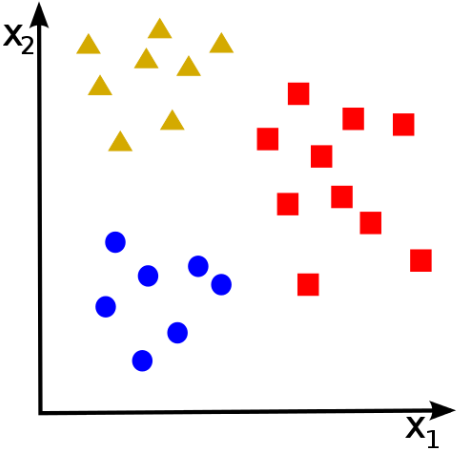
Multiclass classification

Multiple classes

Can we use logistic regression for problems with more than two classes?







Logistic regression

Multiclass classification

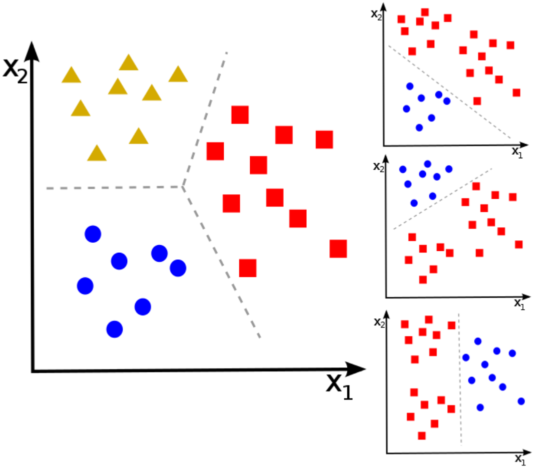
Multi-class: One-versus all:

Train classifiers for each class to obtain probability that x belongs to class i:

$$h_i(x) = P(y = i | \vec{x}, \vec{W})$$

then, choose

$$max_i \left(h_i(x) \right)$$



Questions?

Dariush Salami
dariush.salami@aalto.fi

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.