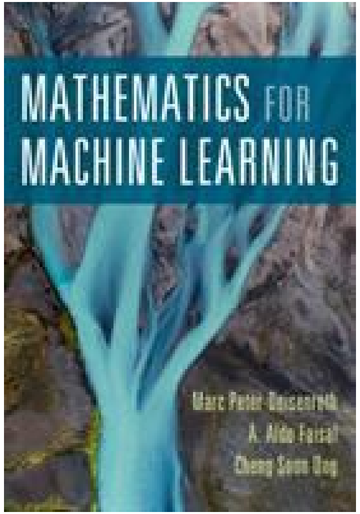
Regression Models

Mingjun Zhong
Department of Computing Science
University of Aberdeen

Goal

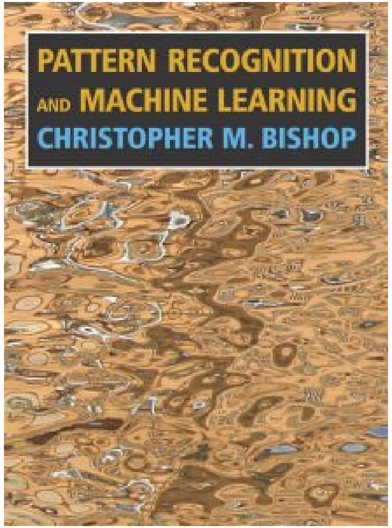
Regression models: are the “work horse” in statistics, supervised learning and data mining. This lecture is to introduce the fundamental concepts and popular regression models.

Textbooks



Chapter 2, 5 & 9

https://mml-book.github.io/book/mml-book.pdf



Chapter 3

http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf

Chapters 16-17

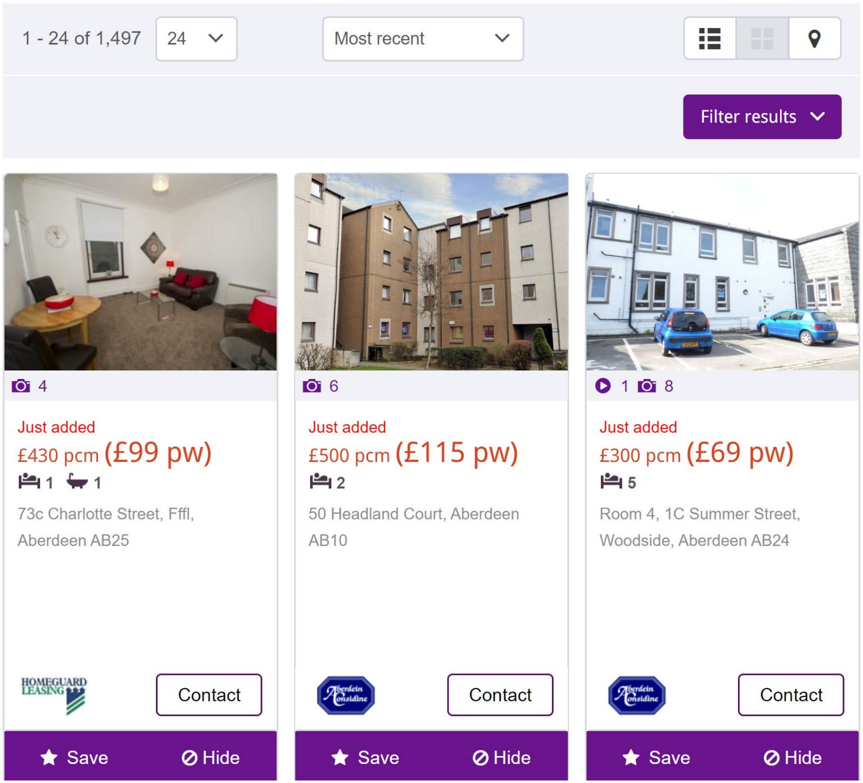
http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.HomePage

Roadmap

- Simple Linear Regression
- Multiple Linear Regression
- Evaluating Regression Model
- Generalized Linear Regression

Roadmap

- Simple Linear Regression
- Multiple Linear Regression
- Evaluating Regression Model
- Generalized Linear Regression



Data set

	Price	House size
1	420	5850
2	385	4000
3	495	3060
4	605	6650
5	610	6360
6	660	4160
7	660	3880
8	690	4160
9	838	4800
10	885	5500
...	...	...

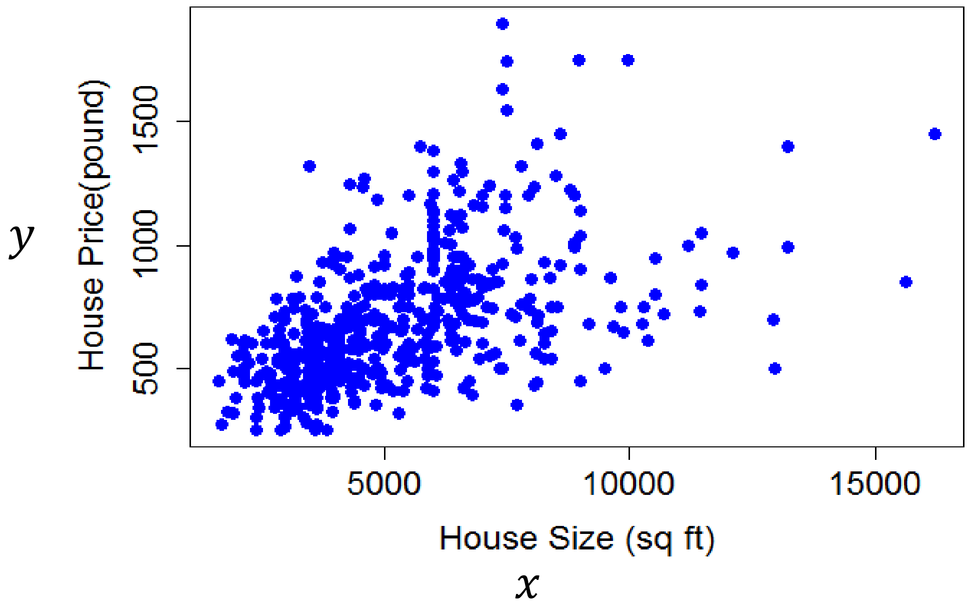
For this new house, give its size 4050 (sq ft), can we predict its rent price?

Simple Linear Regression

Data set

	Price	House size
1	420	5850
2	385	4000
3	495	3060
4	605	6650
5	610	6360
6	660	4160
7	660	3880
8	690	4160
9	838	4800
10	885	5500
…		

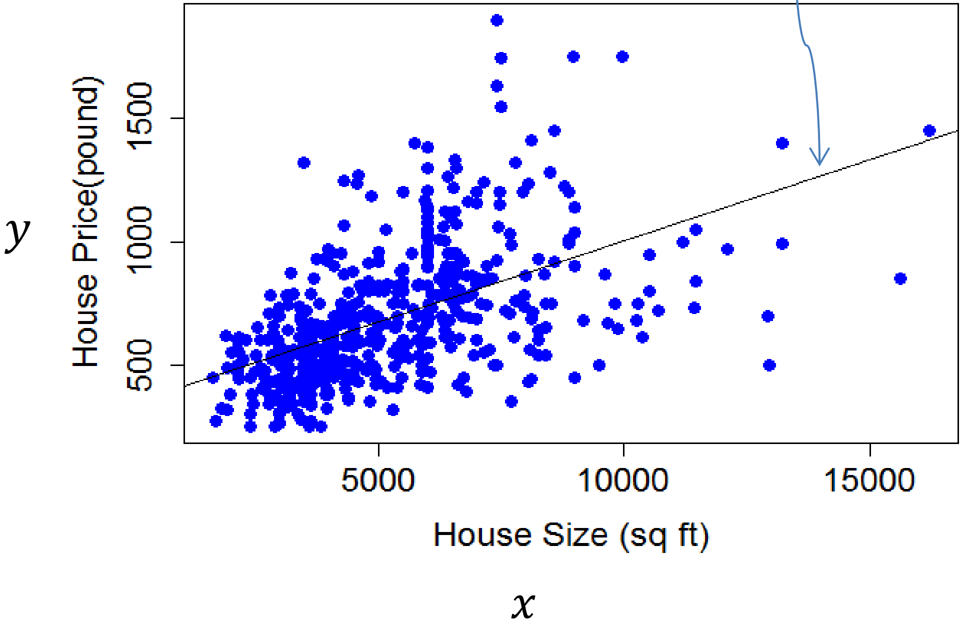
Data visualisation



Simple Linear Regression

$$y := f(x) = \underbrace{\overset{\text{bias}}{w_0} + \overset{\text{weight}}{w_1} x}_{\hat{y}} + \underset{\text{noise}}{\epsilon}$$

	Price	House size
	Target	Feature
1	420	5850
2	385	4000
3	495	3060
4	605	6650
5	610	6360
6	660	4160
7	660	3880
8	690	4160
9	838	4800
10	885	5500
…		



Naming and Notation Conventions

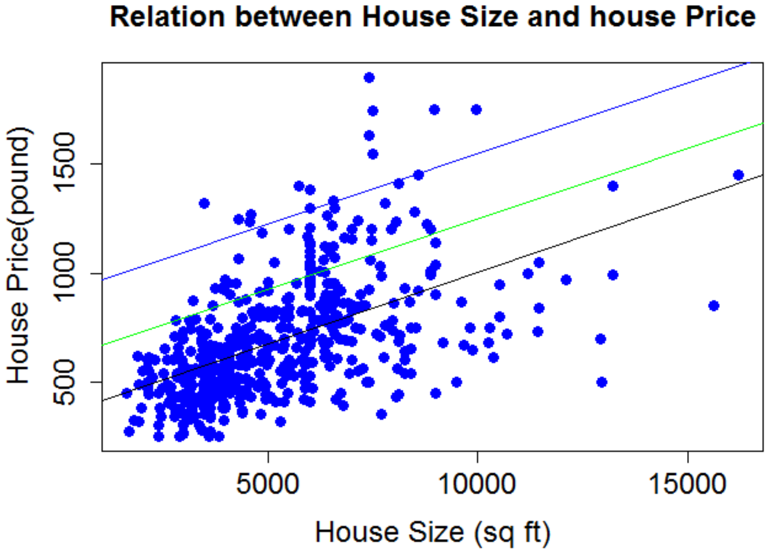
	Machine Learning & Data Mining	Statistics
y	Target	Response
x	Feature(s)	Predictor
w_0, w_1	Weights	Parameters
w_0	Bias	Intercept
	Model training	Parameters estimation

Names and notations are sometimes mix used in different books. In many statistic books, they use notation β_i represent weight/parameter.

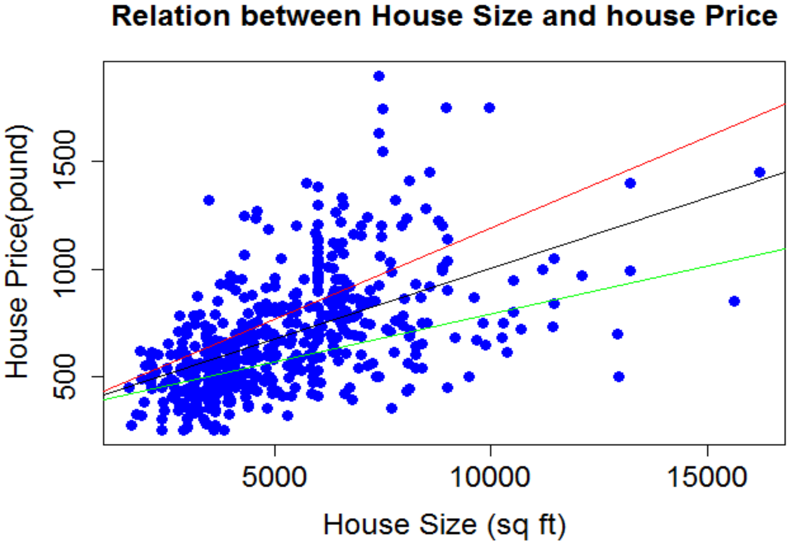
Simple Linear Regression Line

$$y := f(x) = w_0 + w_1 x + \epsilon$$

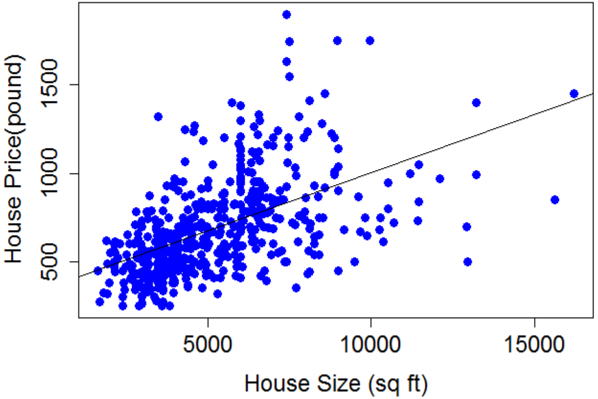
Effect of w_0

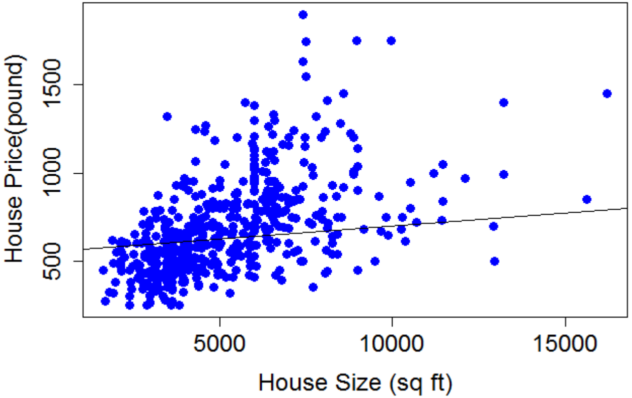


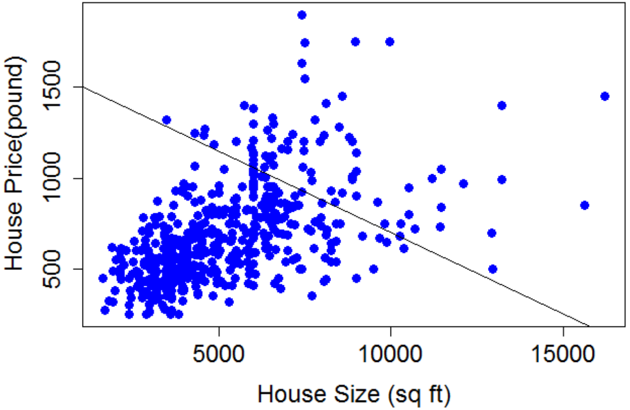
Effect of w_1



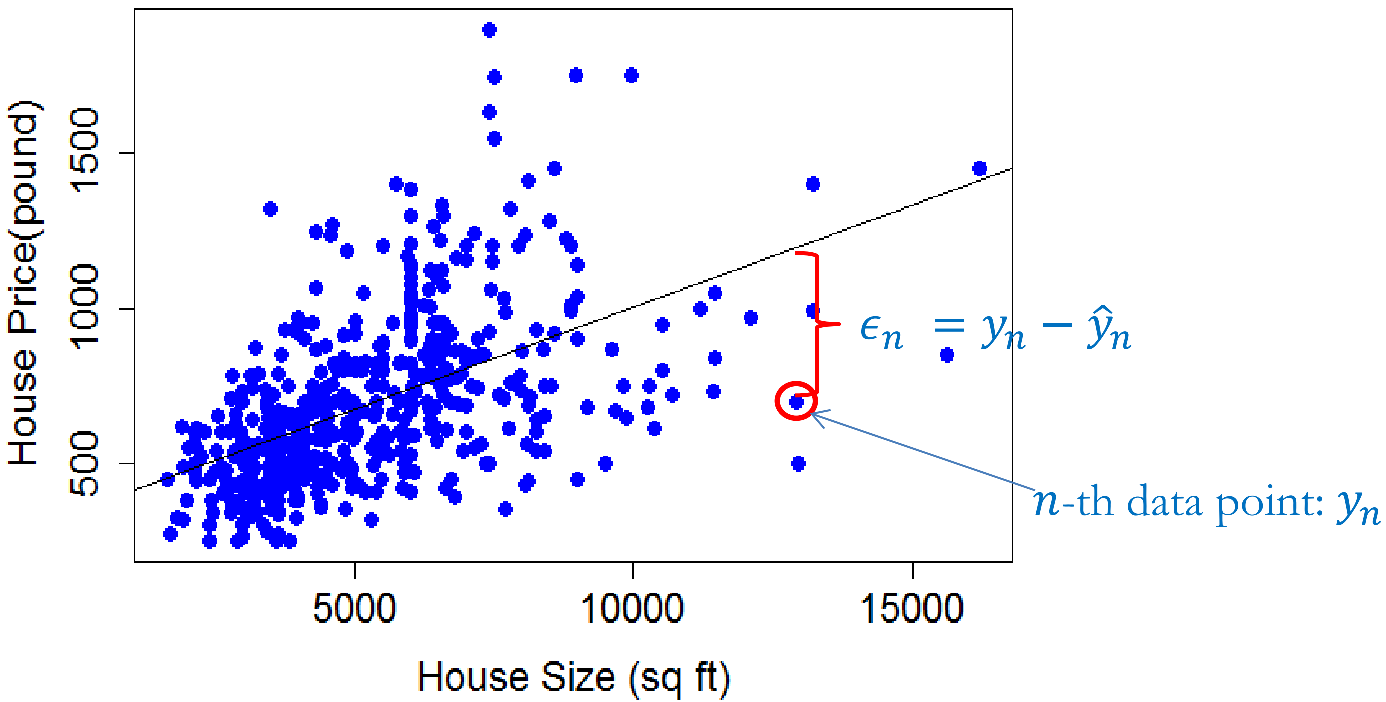
Which Line Fits the Data “Best”?







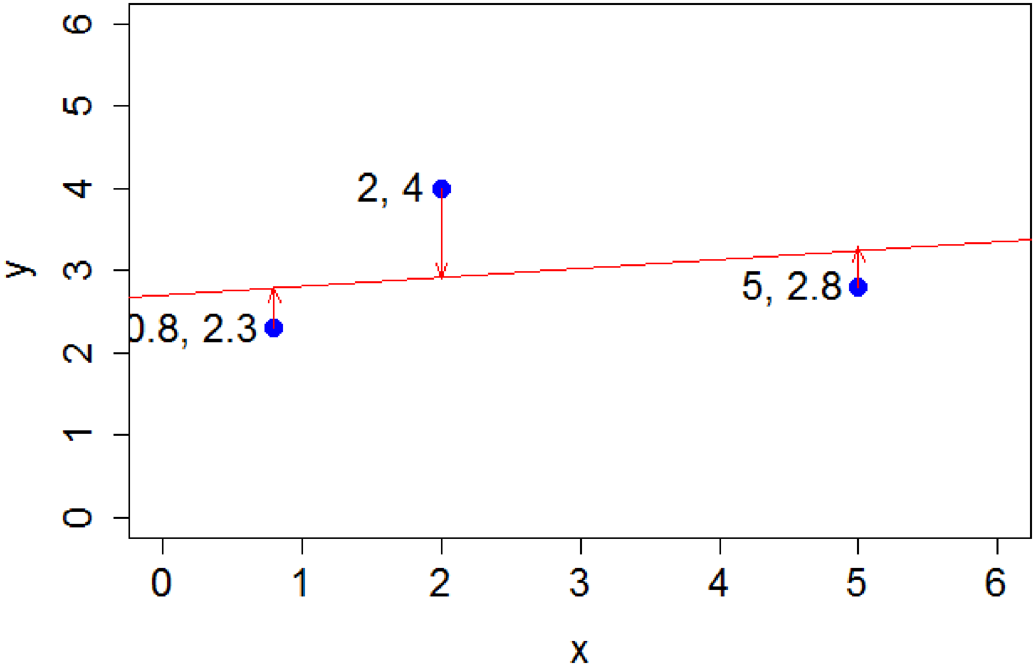
The Error



House Price(pound)
500
1000
1500
5000
10000
15000
House Size (sq ft)
$\epsilon_n = y_n - \hat{y}_n$
n-th data point: y_n

Sum of Squared estimate of Errors (SSE)

$$E = (2.3 - 2.8)^2 + (4 - 2.9)^2 + (2.8 - 3.4)^2 = 1.82$$



Different notations: Residual Sum of Squares (RSS); Sum of Squared Residuals (SSR)

How to find the line with the smallest SSE?

That’s the “best” line?!

This is called the Least Squares Estimation (LSE) method

Data pair

$$The\ linear\ regression\ model\colon\ \hat{y}_n = \hat{w}_0 + \hat{w}_1 x_n$$

Index (n)	Price (y)	House size (x)
1	420	5850
2	385	4000
3	495	3060
4	605	6650
5	610	6360
6	660	4160
7	660	3880
8	690	4160
9	838	4800
10	885	5500
…		

Data Pairs

(y_n, x_n)
(420, 5850)
(385, 4000)
…
…
…
…
…
…
…
(885,5500)
…

Expression of SSE

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

L_2 norm square

$$E = \epsilon_1^2 + \epsilon_2^2 + \ \ldots + \epsilon_N^2 = \sum_{n}^{N} \epsilon_n^2 = \|\boldsymbol{\epsilon}\|_2^2 \,,$$

where $\epsilon_n = y_n - \hat{y}_n = y_n - w_0 - w_1 x_n$. Then

$$E(w_0, w_1) = \sum_{n=1}^{N} (y_n - w_0 - w_1 x_n)^2$$

We can consider SSE is a function of w_0 and w_1

We see SSE is a quadratic function of w_0 and w_1

How to find the weights?

- We want to minimize the error, i.e., SSE.
- Find w_0, w_1 to minimize $E(w_0, w_1) = \sum_{n=1}^{N}(y_n - w_0 - w_1 x_n)^2$
  - Use directly Fermat's Theorem or Interior Extremum Theorem
  - Use Gradient Descent algorithm

- Indeed, you **do not** need to know Fermat's Theorem or Gradient Descent algorithm.
- You may just use ML tools like `sklearn.linear_model.LinearRegression`
- Those Python tool will do everything for you

Fermat's Theorem or Interior Extremum Theorem

Taking derivative of SSE with respect to w_0 and w_1 then gives

$$\frac{\partial E(w_0, w_1)}{\partial w_0} = 0, \qquad \frac{\partial E(w_0, w_1)}{\partial w_1} = 0.$$

Solving this system of linear equations, we have

$$\hat{w}_1 = \frac{\sum_{n=1}^{N} y_n x_n - \frac{\sum_{n=1}^{N} y_n \sum_{n=1}^{N} x_n}{N}}{\sum_{n=1}^{N} x_n^2 - \frac{\left(\sum_{n=1}^{N} x_n\right)^2}{N}},$$

$$\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}.$$

Gradient Descent

- Compute the first derivatives of SSE

$$\frac{\partial E(w_0, w_1)}{\partial w_0}, \qquad \frac{\partial E(w_0, w_1)}{\partial w_1}.$$

and then iteratively update the weights using Gradient Descent

- You can use ***Automatic Differentiation*** to implement Gradient Descent algorithm

- Reference: **Automatic differentiation in machine learning: a survey**

- https://arxiv.org/abs/1502.05767

Roadmap

- Simple Linear Regression
- Multiple Linear Regression
- Evaluating Regression Model
- Generalized Linear Regression

There Are Other Features of Houses

0: “No”; 1: “Yes”

Dummy variable

	Price	House size	Bedrooms	Bathrms	Stories	Driveway	Recroom	dummy
1	420	5850	3	1	2	1	0	1
2	385	4000	2	1	1	1	0	0
3	495	3060	3	1	1	1	0	0
4	605	6650	3	1	2	1	1	0
5	610	6360	2	1	1	1	0	0
6	660	4160	3	1	1	1	1	1
7	660	3880	3	2	2	1	0	1
8	690	4160	3	1	3	1	0	0
9	838	4800	3	1	1	1	1	1
10	885	5500	3	2	4	1	1	0
…	…	…	…	…	…	…	…	…

Categorical Coding Scheme (1 of K Coding Scheme)

	A	B	C	D	E	F	G	H	I	J
3	Age	Party	Gender	Income		Age	Party 1	Party 2	Gender 1	Income
4	20	Rep	Male	45000		20	1	0	1	45000
5	25	Dem	Male	39000		25	0	1	1	39000
6	45	Ind	Male	56000		45	0	0	1	56000
7	35	Rep	Female	49000		35	1	0	0	49000
8	50	Dem	Female	41000		50	0	1	0	41000
9	55	Ind	Female	42000		55	0	0	0	42000
10	39	Rep	Male	58000		39	1	0	1	58000
11	48	Dem	Male	55000		48	0	1	1	55000
12	30	Ind	Male	46000		30	0	0	1	46000
13	27	Rep	Female	42000		27	1	0	0	42000
14	47	Dem	Female	37000		47	0	1	0	37000
15	21	Ind	Female	25000		21	0	0	0	25000
16	48	Rep	Male	75000		48	1	0	1	75000
17	24	Ind	Male	43000		24	0	0	1	43000
18	28	Ind	Female	40000		28	0	0	0	40000
19	40	Dem	Female	31000		40	0	1	0	31000

http://www.real-statistics.com/multiple-regression/multiple-regression-analysis/categorical-coding-regression

Multiple Linear Regression

Simple expression:

$$y_n = w_0 + w_1 x_{n,1} + w_2 x_{n,2} + \cdots + w_D x_{n,D} + \epsilon_n$$

Vector expression:

$y_n = \boldsymbol{w}^T \boldsymbol{x_n} + \epsilon_n,$

where

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}, \qquad \boldsymbol{x}_n = \begin{bmatrix} 1 \\ x_{n,1} \\ \vdots \\ x_{n,D} \end{bmatrix},$$

Example (Real price = £495; House size = 3060; Number of Rooms = 3):

$$\boldsymbol{w} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix}, \qquad \boldsymbol{x}_n = \begin{bmatrix} 1 \\ 3060 \\ 3 \end{bmatrix}$$

$\hat{y}_n = 0.1 * 1 + 0.09 * 3060 + 20 * 3 =$ 335.5 (rent price)

$y_n - \hat{y}_n = \epsilon_n$

£495-£335.5=£159.5

Matrix expression:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon},$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \qquad \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}, \qquad \boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \cdots & x_{N,D} \end{bmatrix}, \qquad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

Estimation of $\boldsymbol{w}$ (Least Squares Estimation)

- The Sum of Squared Errors (SSE):

$$\begin{aligned} E(\boldsymbol{w}) &= \|\epsilon\|_2^2 = \epsilon^T \epsilon = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \\ &= \boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{y} + (\boldsymbol{X}\boldsymbol{w})^T \boldsymbol{X}\boldsymbol{w}. \end{aligned}$$

- We want to find the best w minimizing SSE:
  1) Gradient Descent
  2) Fermat's Theorem or Interior Extremum Theorem
- Require derivatives with respect to a vector
  - To compute derivatives of SSE with respect to weights

- Again, Python software can do all these things!

Estimation of $\boldsymbol{w}$ (Least Square Estimation)

$$
\begin{aligned}
E(\boldsymbol{w}) &= \|\boldsymbol{\epsilon}\|_2^2 = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}) \\
&= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + (\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{X}\boldsymbol{w}.
\end{aligned}
$$

Taking derivatives of SSE with respect to $\boldsymbol{w}$ gives

$$
\begin{aligned}
\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}\left\{\boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + (\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{X}\boldsymbol{w}\right\} \\
&= \frac{\partial}{\partial \boldsymbol{w}}\left\{\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}\right\} \\
&= -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{0}.
\end{aligned}
$$

Then $\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$.

Roadmap

- Simple Linear Regression
- Multiple Linear Regression
- Evaluating Regression Model
- Generalized Linear Regression

Evaluating Regression Model

- How accurate do you think the model is, typically for prediction?
- Do we have any evaluation metric, so that we can check this?

Errors on test data

- Suppose you have some test data
  - Test data are those observed pairs (x_i, y_i) where $i = 1,2,\cdots,M$
  - But these test data were not being used for estimating your model weights W.

- The error of the model on this test data is
  - $Mean\ Squared\ Error: MSE = \frac{1}{M}\sum_{i=1}^{M}(y_i - \widehat{y_i})^2$

- MSE is smaller, your model is better
- You can also use the Mean Absolute Error: $MAE = \frac{1}{M}\sum_{i=1}^{M}|y_i - \widehat{y_i}|$

Roadmap

- Simple Linear Regression
- Multiple Linear Regression
- Evaluating Regression Model
- Generalized Linear Regression

Simply Write the Model

Let's consider a data set contains N observation instances and D features. We can rewrite the model as follows:

$$y(\boldsymbol{x}) = w_0 + w_1 x_1 + w_2\, x_2 + \cdots + w_D x_D,$$

Note:

1. You can consider the full rigorous representation is (vector expression)

$$y_n = y(\boldsymbol{x}_n) + \epsilon_n$$

2. In Bishop's book, he used notation $y(\boldsymbol{w}, \boldsymbol{x})$ to represent $y(\boldsymbol{x})$.

General Form

In a more general writing, we could rewrite it as

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}),$$

Where $\boldsymbol{w}^T = [w_0, w_1, \cdots, w_D]$, and $\phi(\boldsymbol{x})$ is a vector valued function of the input vector $\boldsymbol{x}$. This is called a **linear parameter regression model** (LPM). (See Barber (2012) Bayesian Reasoning and Machine Learning Chapters 16-17). In Bishop's book, he called it **linear basis function models**, and $\phi_i(\boldsymbol{x})$ are known as **basis functions**.

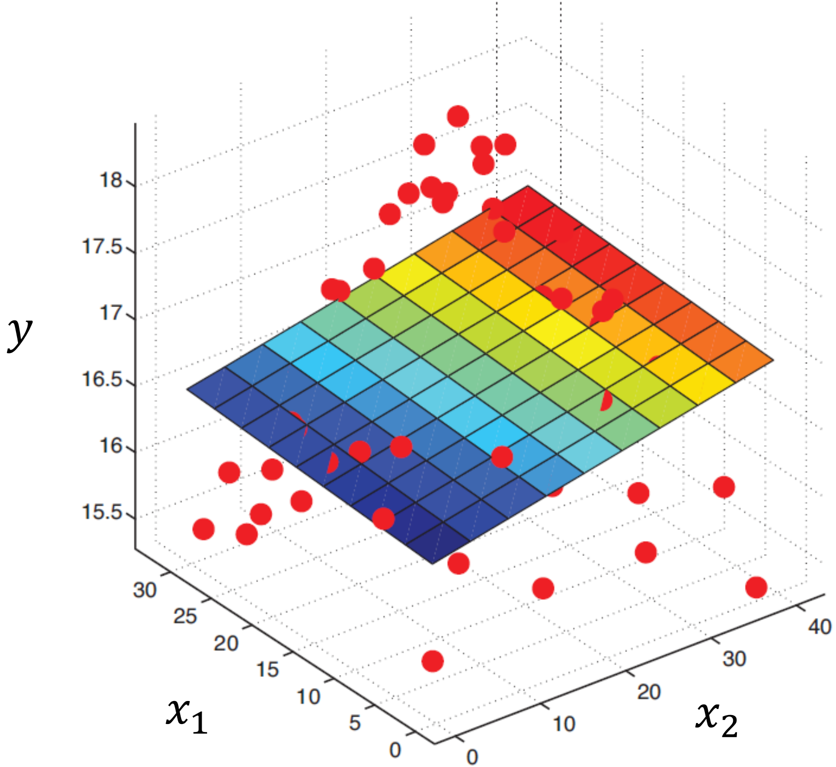
The model is linear in the parameter $\boldsymbol{w}$, not necessarily linear in $\boldsymbol{x}$.

Example of Multiple Linear Regression

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}),$$

where

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}, \qquad \phi(\boldsymbol{x}) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

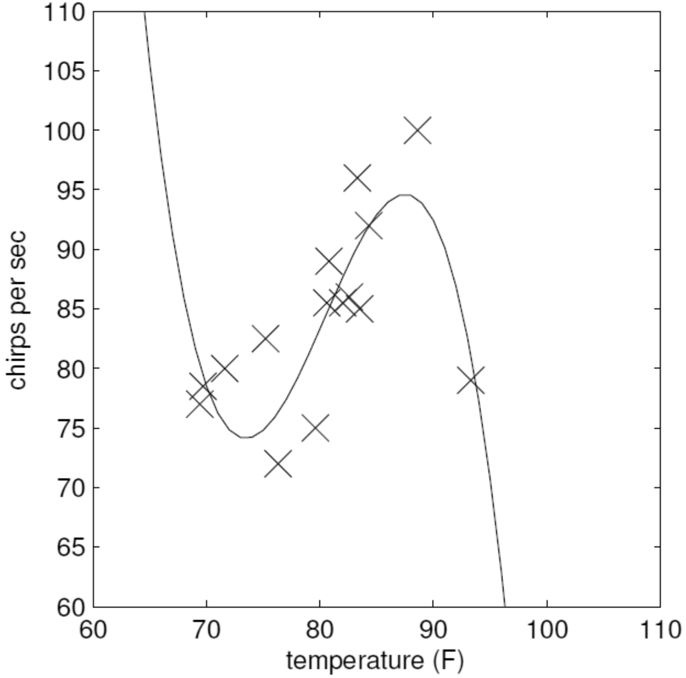


Example of Polynomial Curve Fitting

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}),$$

where

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}, \qquad \phi(\boldsymbol{x}) = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix}$$

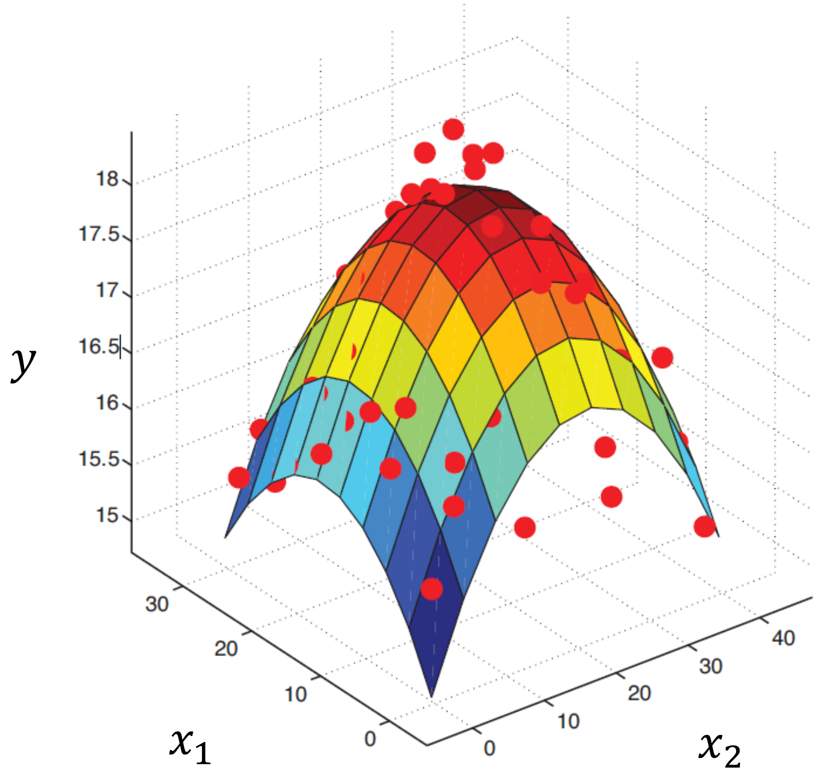


Example of Fitting Quadratic Form

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}),$$

where

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}, \qquad \phi(\boldsymbol{x}) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}$$



Estimation of $\boldsymbol{w}$

I do not give you details here. If you are interested, please check:

- Barber (2012) Bayesian Reasoning and Machine Learning, pp. 360-361 **(LSE)**
- Bishop (2007) Pattern Recognition and Machine Learning, pp. 140-142 **(MLE+LSE)**