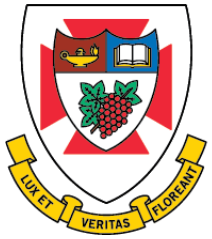
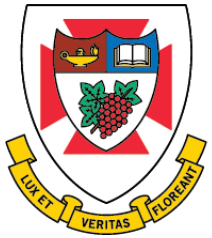


THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

INTRODUCTION TO MACHINE LEARNING

DIT 45100



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Module 2
Linear Regression

Multivariate Linear Regression

A Worked Example

- We are now in a position to build a linear regression model that uses all of the continuous descriptive features in the office rentals dataset.

- The general structure of the model is:

$$\text{RENTAL PRICE} = w[0] + w[1] \times \text{SIZE} + w[2] \times \text{FLOOR} + w[3] \times \text{BROADBAND RATE}$$

Multivariate Linear Regression

A Worked Example: office rental dataset

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

Multivariate Linear Regression

A Worked Example

- For this example let's assume that:
  - Learning rate
    - α = 0.00000002
  - Initial Weights
    - w[0] = -0.146
    - w[1] = 0.185
    - w[2] = -0.044
    - w[3] = 0.119

A Worked Example

Iteration 1

ID	Rental Price	Pred.	Error	Squared Error	errorDelta($\mathcal{D}$, w[i]) w[0]	w[1]	w[2]	w[3]
1	320	93.26	226.74	51411.08	226.74	113370.05	906.96	1813.92
2	380	107.41	272.59	74307.70	272.59	149926.92	1908.16	13629.72
3	400	115.15	284.85	81138.96	284.85	176606.39	2563.64	1993.94
4	390	119.21	270.79	73327.67	270.79	170598.22	1353.95	6498.98
5	385	134.64	250.36	62682.22	250.36	166492.17	2002.91	25036.42
6	410	130.31	279.69	78226.32	279.69	195782.78	1118.76	2237.52
7	480	142.89	337.11	113639.88	337.11	259570.96	3371.05	2359.74
8	600	168.32	431.68	186348.45	431.68	379879.24	5180.17	21584.05
9	570	170.63	399.37	159499.37	399.37	367423.83	5591.23	3194.99
10	620	187.58	432.42	186989.95	432.42	432423.35	3891.81	10378.16
			Sum	1067571.59	3185.61	2412073.90	27888.65	88727.43
Sum of squared errors (Sum/2)				533785.80				

A Worked Example

$$\mathbf{w}[j] \leftarrow \mathbf{w}[j] + \alpha \underbrace{\sum_{i=1}^{n} ((t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)) \times d_i[j])}_{errorDelta(\mathcal{D},\mathbf{w}[j])}$$

Initial Weights

w[0]:	-0.146	**w**[1]:	0.185	**w**[2]:	-0.044	**w**[3]:	0.119

Example

$$\mathbf{w}[1] \leftarrow 0.185 + 0.00000002 \times 2,412,074 = 0.23324148$$

New Weights (Iteration 1)

w[0]:	-0.146	**w**[1]:	0.233	**w**[2]:	-0.043	**w**[3]:	0.121

A Worked Example

Iteration 2

ID	Rental Price	Pred.	Error	Squared Error	errorDelta($\mathcal{D}$, w[i]) w[0]	w[1]	w[2]	w[3]
1	320	117.40	202.60	41047.92	202.60	101301.44	810.41	1620.82
2	380	134.03	245.97	60500.69	245.97	135282.89	1721.78	12298.44
3	400	145.08	254.92	64985.12	254.92	158051.51	2294.30	1784.45
4	390	149.65	240.35	57769.68	240.35	151422.55	1201.77	5768.48
5	385	166.90	218.10	47568.31	218.10	145037.57	1744.81	21810.16
6	410	164.10	245.90	60468.86	245.90	172132.91	983.62	1967.23
7	480	180.06	299.94	89964.69	299.94	230954.68	2999.41	2099.59
8	600	210.87	389.13	151424.47	389.13	342437.01	4669.60	19456.65
9	570	215.03	354.97	126003.34	354.97	326571.94	4969.57	2839.76
10	620	187.58	432.42	186989.95	432.42	432423.35	3891.81	10378.16
			Sum	886723.04	2884.32	2195615.84	25287.08	80023.74
Sum of squared errors (Sum/2)				443361.52				

A Worked Example

$$\mathbf{w}[j] \leftarrow \mathbf{w}[j] + \alpha \underbrace{\sum_{i=1}^{n} ((t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)) \times d_i[j])}_{errorDelta(\mathcal{D},\mathbf{w}[j])}$$

Initial Weights (Iteration 2)

w[0]:	-0.146	**w**[1]:	0.233	**w**[2]:	-0.043	**w**[3]:	0.121

Exercise

$\mathbf{w}[1] \leftarrow ?, \alpha = 0.00000002$

New Weights (Iteration 2)

w[0]:	?	**w**[1]:	?	**w**[2]:	?	**w**[3]:	?

A Worked Example

$$\mathbf{w}[j] \leftarrow \mathbf{w}[j] + \alpha \underbrace{\sum_{i=1}^{n} ((t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i)) \times d_i[j])}_{errorDelta(\mathcal{D},\mathbf{w}[j])}$$

Initial Weights (Iteration 2)

w[0]:	-0.146	**w**[1]:	0.233	**w**[2]:	-0.043	**w**[3]:	0.121

Exercise

$$\mathbf{w}[1] \leftarrow -0.233 + 0.00000002 \times 2195616.08 = 0.27691232$$

New Weights (Iteration 2)

w[0]:	-0.145	**w**[1]:	0.277	**w**[2]:	-0.043	**w**[3]:	0.123

A Worked Example

- The algorithm then keeps iteratively applying the weight update rule until it converges on a stable set of weights beyond which little improvement in model accuracy is possible.
- After 100 iterations the final values for the weights are:
  - w[0] = -0.1513
  - w[1] = 0.6270
  - w[2] = -0.1781
  - w[3] = 0.0714
- Which results in a sum of squared errors value of 2913.5

Multivariate Linear Regression

Interpreting Models

- The weights used by linear regression models indicate the effect of each descriptive feature on the predictions returned by the model.
- Both the **sign** and the **magnitude** of the weight provide information on how the descriptive feature effects the predictions of the model.

Multivariate Linear Regression

Interpreting Models

- The weights used by linear regression models indicate the effect of each descriptive feature on the predictions returned by the model.
- Both the **sign** and the **magnitude** of the weight provide information on how the descriptive feature effects the predictions of the model.

Table: Weights and standard errors for each feature in the office rentals model.

Descriptive Feature	Weight	Standard Error	*t*-statistic	*p*-value
SIZE	0.6270	0.0545	11.504	$<$0.0001
FLOOR	-0.1781	2.7042	-0.066	0.949
BROADBAND RATE	0.071396	0.2969	0.240	0.816

Multivariate Linear Regression

Interpreting Models

- It is tempting to infer the relative importance of the different descriptive features in the model from the magnitude of the weights
- However, direct comparison of the weights tells us little about their relative importance.
- A better way to determine the importance of each descriptive feature in the model is to perform a **statistical significance test**.

Multivariate Linear Regression

Interpreting Models

- The statistical significance test we use to analyze the importance of a descriptive feature **d** [j] in a linear regression model is the **t-test**.
- The null hypothesis for this test is that the feature does not have a significant impact on the model. The test statistic we calculate is called the t-statistic.

Multivariate Linear Regression

Interpreting Models

- The standard error for the overall model is calculated as

$$se = \sqrt{\frac{\sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2}{n-2}} \tag{18}$$

- A standard error calculation is then done for a descriptive feature as follows:

$$se(\mathbf{d}[j]) = \frac{se}{\sqrt{\sum_{i=1}^{n} \left(\mathbf{d}_i[j] - \overline{\mathbf{d}[j]}\right)^2}} \tag{19}$$

- The t-statistic for this test is calculated as follows:

$$t = \frac{\mathbf{w}[j]}{se(\mathbf{d}[j])} \tag{20}$$

Multivariate Linear Regression

Interpreting Models

- Using a standard t-statistic look-up table, we can then determine the p-value associated with this test (this is a two tailed t-test with degrees of freedom set to the number of instances in the training set minus 2).
- If the p-value is less than the required significance level, typically 0.05, we reject the null hypothesis and say that the descriptive feature has a significant impact on the model; otherwise we say that it does not.

Multivariate Linear Regression

Interpreting Models

- Using a standard t-statistic look-up table, we can then determine the p-value associated with this test (this is a two tailed t-test with degrees of freedom set to the number of instances in the training set minus 2).
- If the p-value is less than the required significance level, typically 0.05, we reject the null hypothesis and say that the descriptive feature has a significant impact on the model; otherwise we say that it does not.

Table: Weights and standard errors for each feature in the office rentals model.

Descriptive Feature	Weight	Standard Error	*t*-statistic	*p*-value
SIZE	0.6270	0.0545	11.504	$<$0.0001
FLOOR	-0.1781	2.7042	-0.066	0.949
BROADBAND RATE	0.071396	0.2969	0.240	0.816

Multivariate Linear Regression

Setting the Learning Rate

- **Learning rate decay** allows the learning rate to start at a large value and then decay over time according to a predefined schedule.
- A good approach is to use a decay schedule as follow:

$$\propto_{\tau} = \propto_{0} \frac{C}{C+\tau} \qquad (21)$$

Where,

$\propto_0$ and $\propto_\tau$ are initial learning rate and learning rate at iteration τ, respectively.

C is the decay constant

Setting the Learning Rate

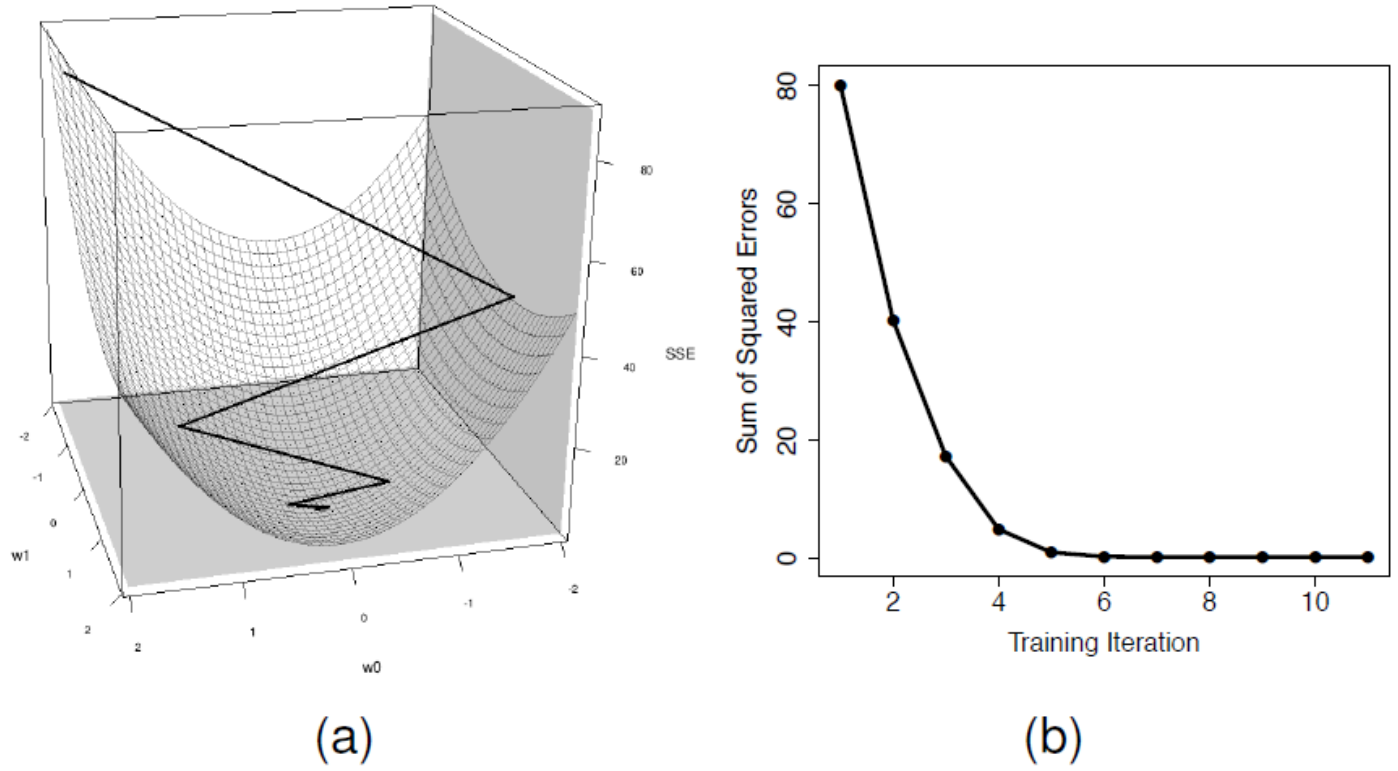


Figure: (a) The journey across the error surface for the office rentals prediction problem when learning rate decay is used ($\alpha_0 = 0.18$, $C = 10$); (b) a plot of the changing sum of squared error values during this journey.

Setting the Learning Rate

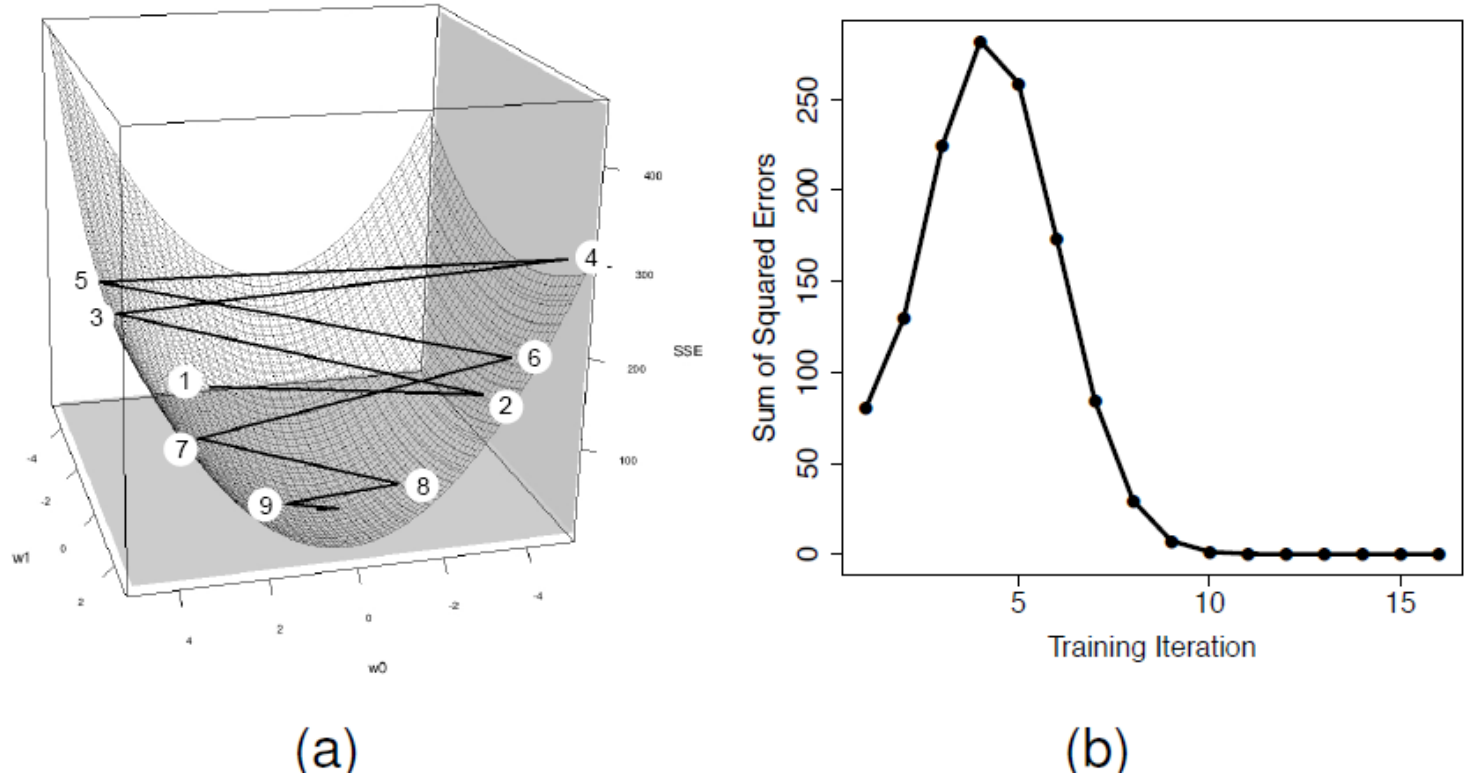
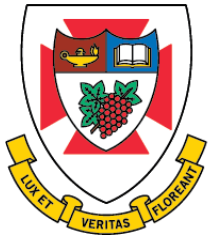


Figure: (a) The journey across the error surface for the office rentals prediction problem when learning rate decay is used ($\alpha_0 = 0.25$, $C = 100$); (b) a plot of the changing sum of squared error values during this journey.



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Handling the Non-Linear Relationships

Multivariate Linear Regression

Handling Non-Linear Relationships

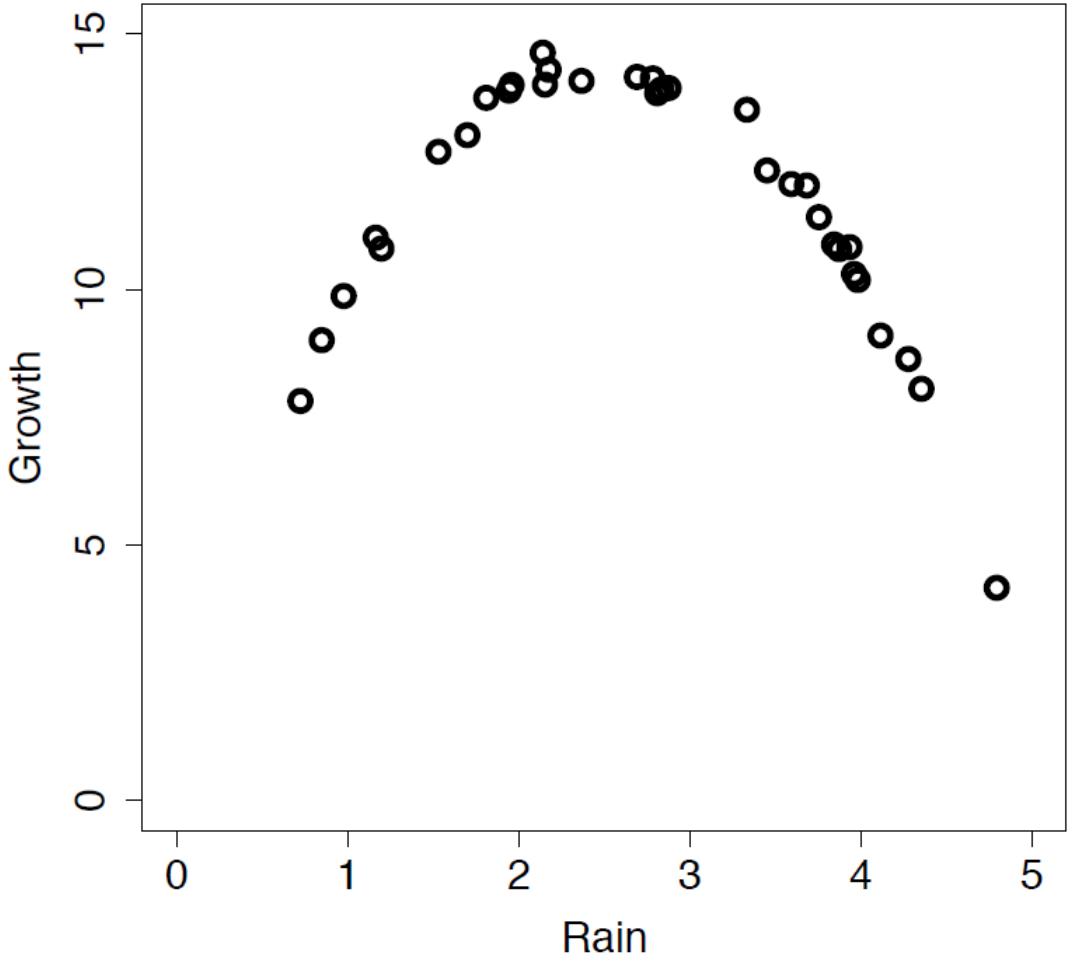
Table: A dataset describing grass growth on Irish farms during July 2012.

ID	Rain	Growth	ID	Rain	Growth	ID	Rain	Growth
1	2.153	14.016	12	3.754	11.420	23	3.960	10.307
2	3.933	10.834	13	2.809	13.847	24	3.592	12.069
3	1.699	13.026	14	1.809	13.757	25	3.451	12.335
4	1.164	11.019	15	4.114	9.101	26	1.197	10.806
5	4.793	4.162	16	2.834	13.923	27	0.723	7.822
6	2.690	14.167	17	3.872	10.795	28	1.958	14.010
7	3.982	10.190	18	2.174	14.307	29	2.366	14.088
8	3.333	13.525	19	4.353	8.059	30	1.530	12.701
9	1.942	13.899	20	3.684	12.041	31	0.847	9.012
10	2.876	13.949	21	2.140	14.641	32	3.843	10.885
11	4.277	8.643	22	2.783	14.138	33	0.976	9.876

Multivariate Linear Regression

Handling Non-Linear Relationships

- **Figure:** A scatter plot of the RAIN and GROWTH feature from the grass growth dataset.



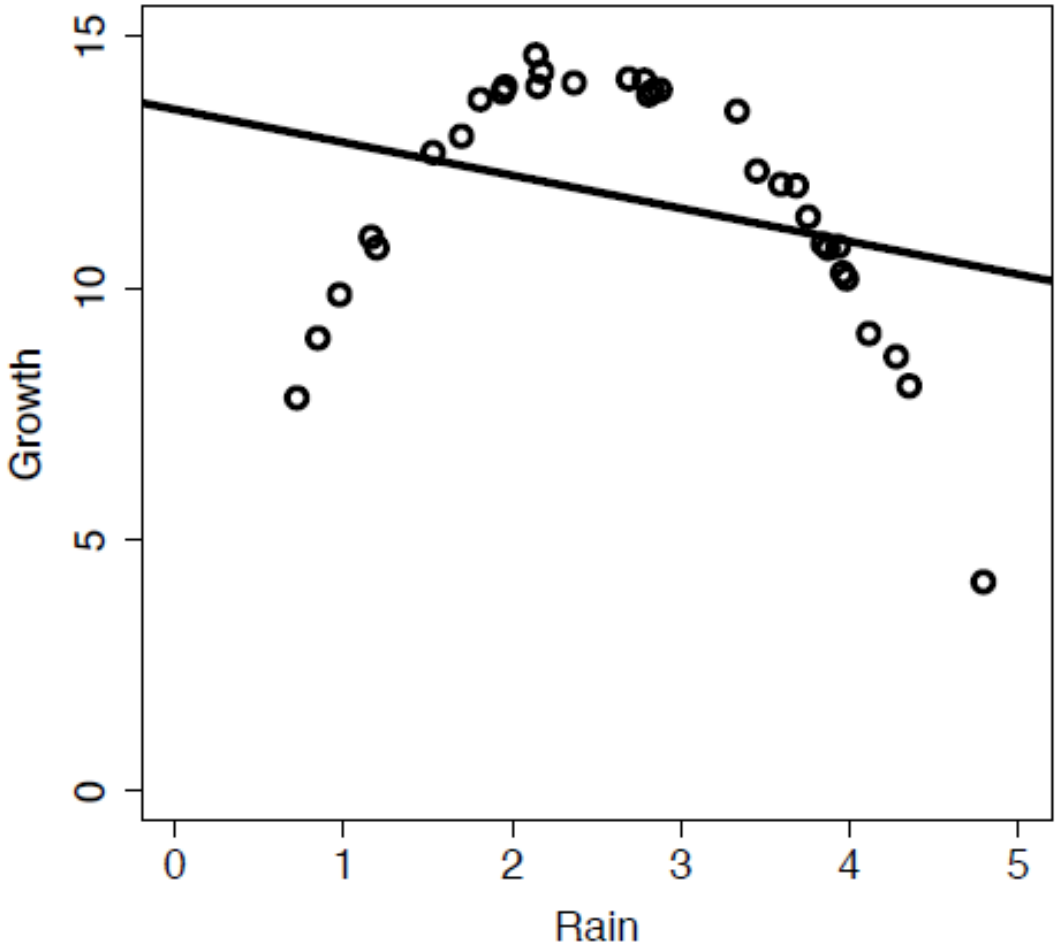
Multivariate Linear Regression

Handling Non-Linear Relationships

- **Figure:** A simple linear regression model trained to capture the relationship between the grass growth and rainfall.

- The best linear model we can learn for this data is:

$$GROWTH = 13{:}510 - 0.667 \times RAIN$$



Multivariate Linear Regression

Handling Non-Linear Relationships

- In order to handle non-linear relationships we transform the data rather than the model using a set of basis functions:

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \sum_{k=0}^{b} \mathbf{w}[k] \times \phi_k(\mathbf{d}) \quad (22)$$

  where ϕ_0 to ϕ_b are a series of b basis functions that each transform the input vector $\mathbf{d}$ in a different way.

- The advantage of this is that, except for introducing the mechanism of basis functions, we do not need to make any other changes to the approach we have presented so far.

Multivariate Linear Regression

Handling Non-Linear Relationships

- The relationship between rainfall and grass growth in the grass growth dataset can be accurately represented as a **second order polynomial** through the following model:

$$\text{GROWTH} = \mathbf{w}[0] \times \phi_0(\text{RAIN}) + \mathbf{w}[1] \times \phi_1(\text{RAIN}) + \mathbf{w}[2] \times \phi_2(\text{RAIN})$$

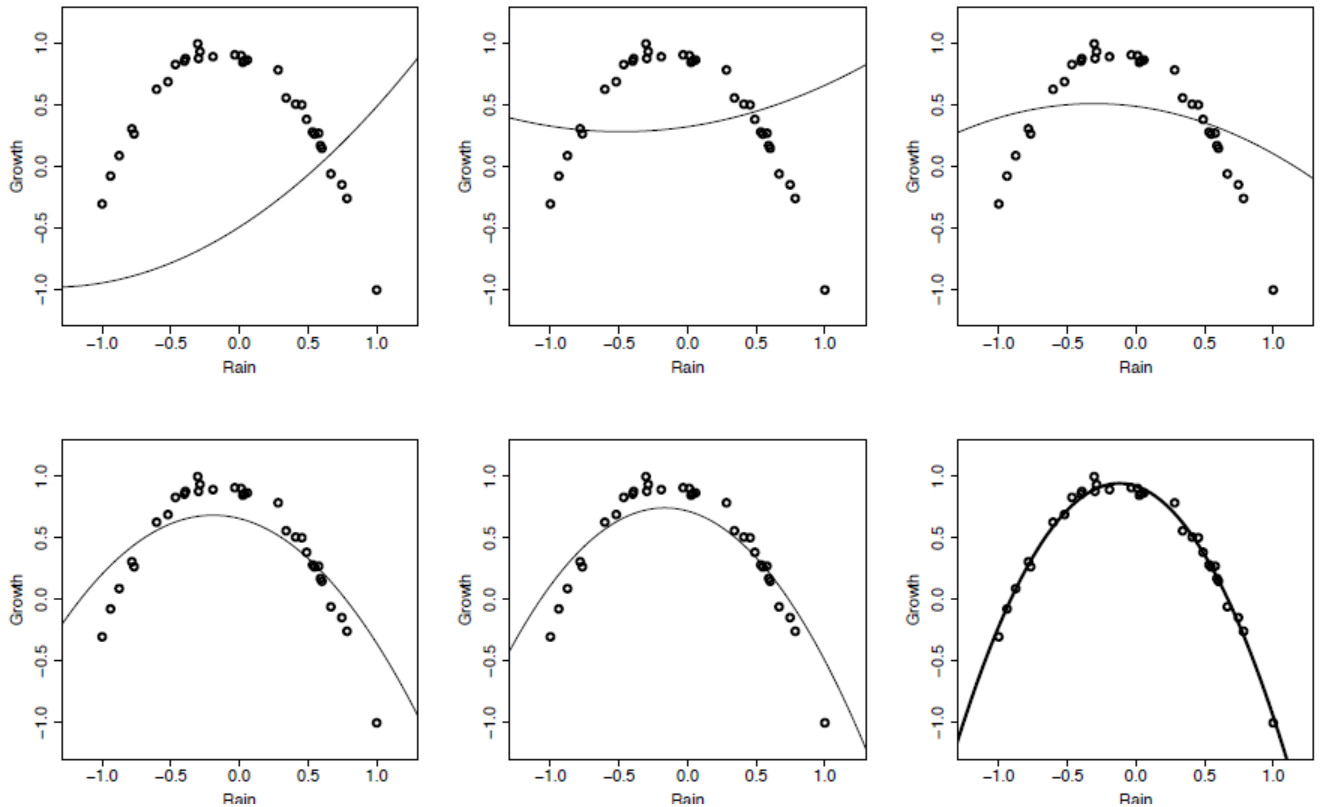
where

$$\begin{aligned} \phi_0(\text{RAIN}) &= 1 \\ \phi_1(\text{RAIN}) &= \text{RAIN} \\ \phi_2(\text{RAIN}) &= \text{RAIN}^2 \end{aligned}$$

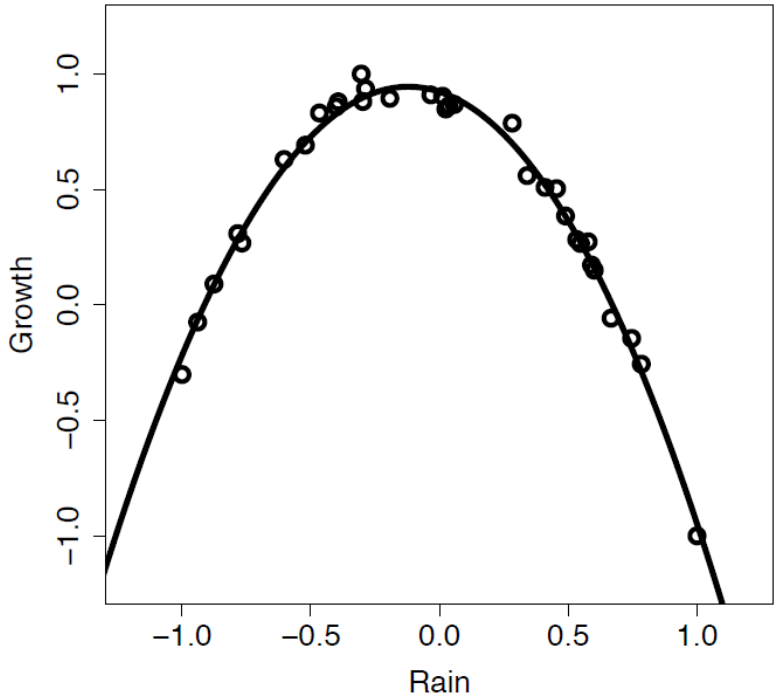
Multivariate Linear Regression

Handling Non-Linear Relationships

Figure: A selection of the models developed during the gradient descent process for the grass growth dataset. (Note that the RAIN and GROWTH features have been **range normalized** to the [-1, 1] range.)



Handling Non-Linear Relationships



$$\text{GROWTH} = \overset{0.8475}{\cancel{0.3707}} \times \phi_0(\text{RAIN}) + \overset{-0.3707}{\cancel{0.8475}} \times \phi_1(\text{RAIN}) + -1.717 \times \phi_2(\text{RAIN})$$

Handling Non-Linear Relationships

$$\text{GROWTH} = \overset{0.8475}{\cancel{0.3707}} \times \phi_0(\text{RAIN}) + \overset{-0.3707}{\cancel{0.8475}} \times \phi_1(\text{RAIN}) + -1.717 \times \phi_2(\text{RAIN})$$

$$\begin{aligned} \phi_0(\text{RAIN}) &= 1 \\ \phi_1(\text{RAIN}) &= \text{RAIN} \\ \phi_2(\text{RAIN}) &= \text{RAIN}^2 \end{aligned}$$

- What is the predicted growth for the following RAIN values:
  1. RAIN$= -0.75$
  2. RAIN$= 0.1$
  3. RAIN$= 0.9$

Multivariate Linear Regression

Handling Non-Linear Relationships

- Basis functions can also be used for multivariable linear regression models in the same way to train models for prediction problems that involve non-linear relationships.
- The only extra requirement being the definition of more basis functions.

Pros of Linear Regression

- Very fast
- No parameters to tune
- Easy to understand and highly interpretable models

Evaluation of Regression Models

- R-Squared (R^2)
- Error Measures
  - Mean Squared Error (MSE)
  - Mean Absolute Error (MAE)
  - Root Mean Squared Error (RMSE)

Summary

- **Big Idea**
- **Fundamentals**
  - Simple Linear Regression
  - Measuring Error
  - Error Surfaces
- **Standard Approach: Multivariate Linear Regression with Gradient Descent**
  - Multivariate Linear Regression
  - Gradient Descent
  - Learning Rates & Initial Weights
  - A Worked Example
  - Interpreting Models
  - Non-Linearities in the Data