Regression

Data Science Dojo

datasciencedojo
data science for everyone

Agenda

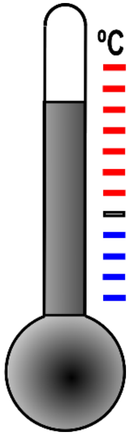
- Introduction
- Cost Function & Gradient Descent
  - Minimization
  - Implementation
- Hands-on Example
- Evaluating Regression Models
- Regularization

INTRODUCTION

Regression

Sales Forecasts

Housing Price
Predictions



Daily Temperature
Highs & Lows

datasciencedojo
data science for everyone

Regression vs Classification

- Classification
  - Target is discrete with finite value set
  - **Examples:** survived/dead, face/non-face, fraud/non-fraud
- Regression
  - Target is continuous
  - **Examples:** price, weight, height, temperature,

Notation

Passenger Id	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S

$$x_4^5$$

5: The passenger is in the 5th row
4: The passenger's name is the 4th column

Notation: Ozone Dataset

So how do we describe all the rows?

	ozone	radiation	temperature	wind
Row 1	41	190	67	7.4
Row 2	36	118	72	8.0
Row 3	12	149	74	12.6
	18	313	62	11.5
	23	299	65	8.6
	19	99	59	13.8

$x^1 = [190, 67, 7.4]$

$x^2 = [118, 72, 8.0]$

$x^3 = [149, 74, 12.6]$

Notation: Ozone Dataset

The ozone dataset uses radiation, temperature and wind to predict ozone levels.

	x_1	x_2	x_3
ozone	radiation	temperature	wind
41	190	67	7.4
36	118	72	8.0
12	149	74	12.6
18	313	62	11.5
23	299	65	8.6
19	99	59	13.8

Y X

Using this notation, we can describe all the columns of the dataset.

Notation Summary

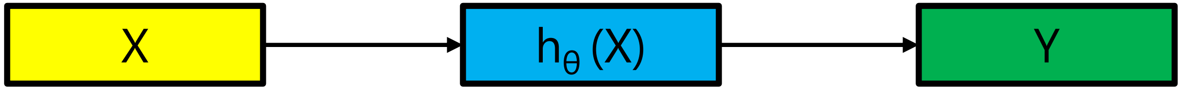
x^i – Each row of features
x_j – Each column of features
X – Set of all the feature columns
y^i – Each row of the target
Y – The target column
n – Number of rows in the dataset
m – Number of columns in the dataset

Features

Target

Function Notation

- θ – A vector of function parameters which define a specific hypothesis
- h_θ – A specific hypothesis (estimate) for the mapping X → Y



Regression Example

- Estimator (linear in this case) with n number of features

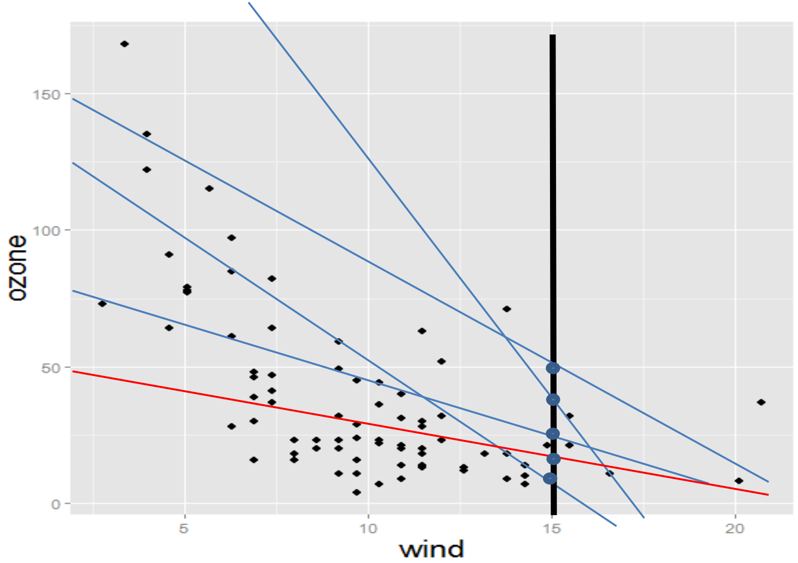
$$\theta = [\theta_0, \theta_1, \theta_2, \dots, \theta_m]$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m$$

COST FUNCTION AND GRADIENT DESCENT

What is a good regression line?

- Wind Speed=15 mph
- Ozone = ?
- Use the line that is **somewhere in the middle**
- How do we define "middle"?



$$h_\theta(x) = \theta_0 + \theta_1 x$$

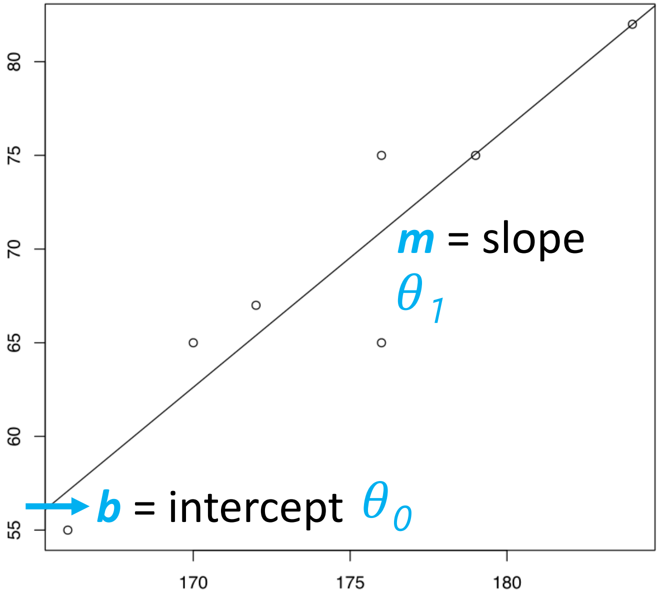
Defining a line

How do we define a line in slope-intercept notation?

- $y = \boldsymbol{m}x + \boldsymbol{b}$

In θ notation

- $h_\theta(\mathrm{x}) = \boldsymbol{\theta_1}\mathrm{x} + \boldsymbol{\theta_0}$



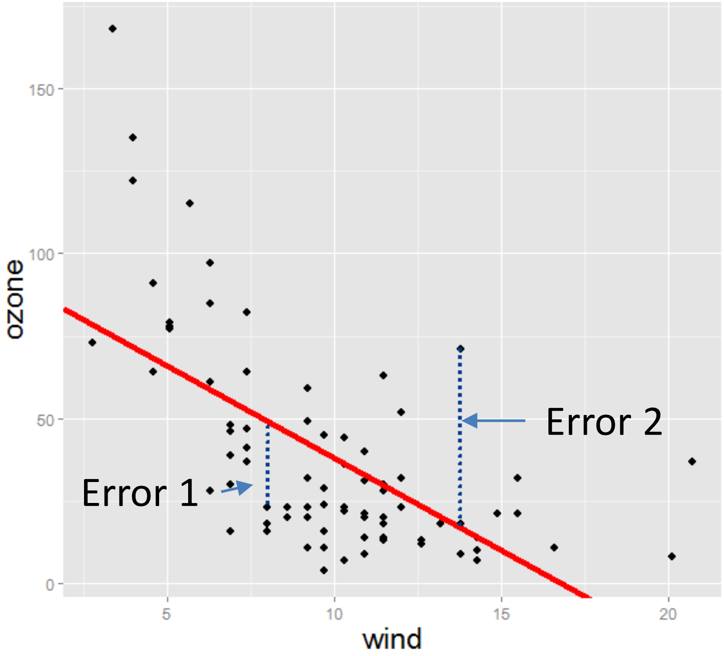
More Features

y	x_1	x_2	x_3
ozone	radiation	temperature	wind
41	190	67	7.4
36	118	72	8.0
12	149	74	12.6
18	313	62	11.5
23	299	65	8.6
19	99	59	13.8

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

Residuals

Difference between hypothesis $h_\theta(x)$ (predicted value) and true value (known target)

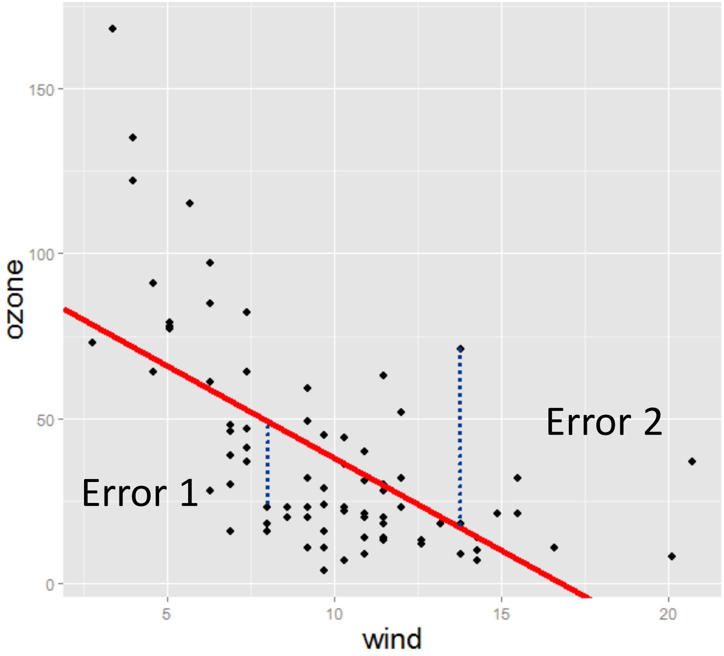


Cost Function

Minimize the 'cost' or 'loss' function – $J(\theta)$

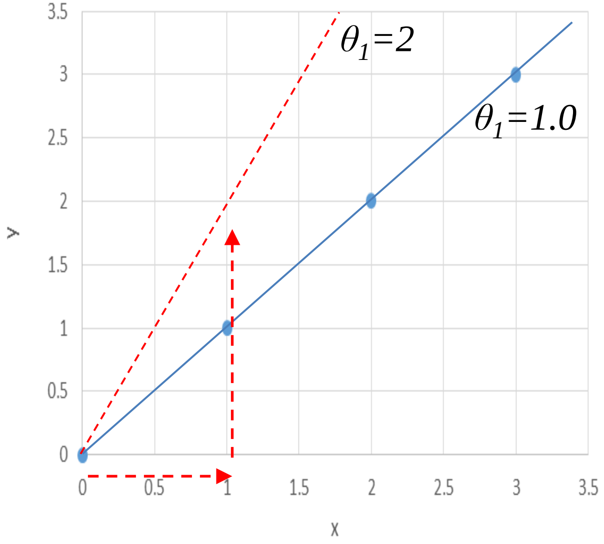
- Smaller for lower error
- Larger for higher error

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta\left(x^i\right) - y^i\right)^2$$



Cost Function

$$h_\theta(x) = \theta_0 + \theta_1 x$$

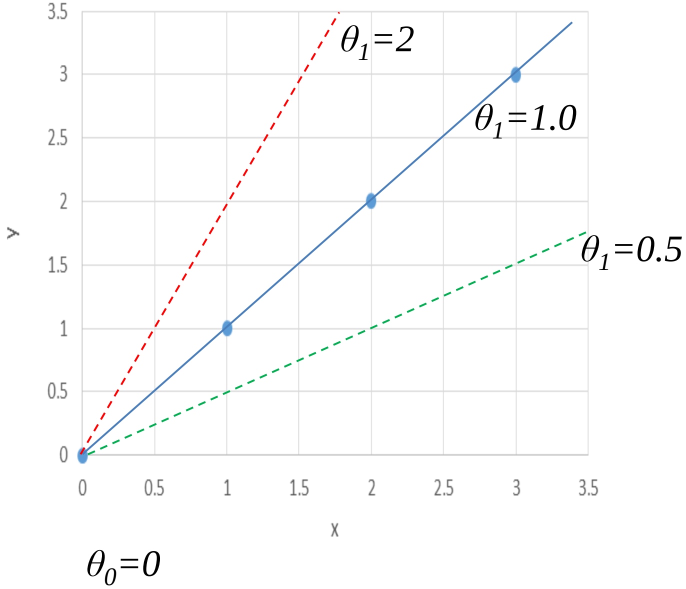


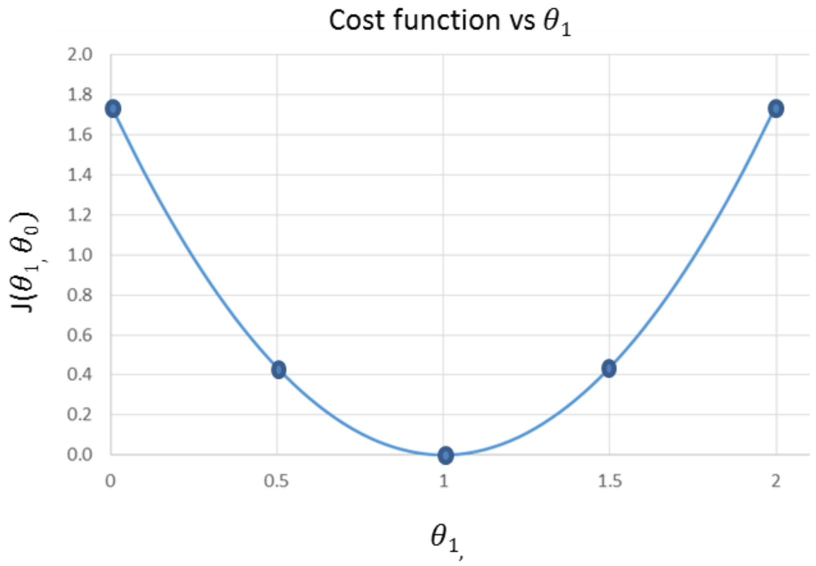
θ_0=0

Cost Function

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta(x^i) - y^i\right)^2$$

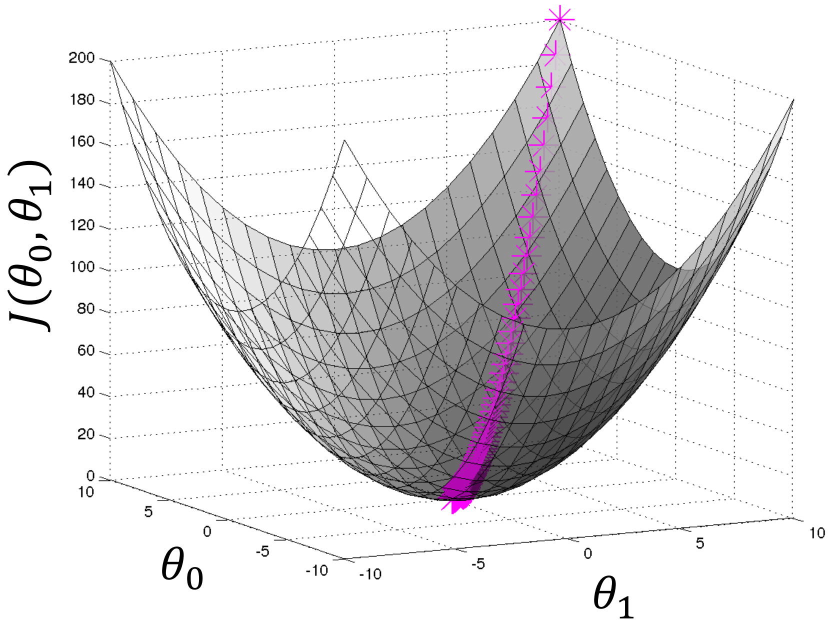




Cost function in three dimensions

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta(x^i) - y^i\right)^2$$



datasciencedojo
data science for everyone

HOW DO WE FIND OUT THE MINIMUM OF THE COST FUNCTION

Maximum/Minimum Problem

Find ***two non-negative*** *numbers whose* ***sum is 9*** *and so that the product of one number and the square of the other number is a* ***maximum****.*

Solution (1/2)

Sum of number is 9

$$9 = x + y$$

Product of two numbers is

$$P = x\,y^2$$

$$= x\,(9-x)^2$$

Solution (2/2)

$$P' = x\,(2)\,(9-x)(-1) + (1)\,(9-x)^2$$
$$= (9-x)\,[\,-2x + (9-x)\,]$$
$$= (9-x)\,[\,9-3x\,]$$
$$= (9-x)\,(3)[\,3-x\,]$$
$$= 0$$

x=9 or x=3

Maximum Problem

There are **50 apple trees** in an orchard.

Each tree produces **800 apples**. For each additional tree planted in the orchard, the apple output per tree drops by **10 apples**.

Question: *How many additional trees should be planted in the existing orchard in order to maximize the apple output of the orchard?*

Solution

$$A = (50 + t) \times (800 - 10t)$$

$$A = 40{,}000 + 300t - 10t^2$$

Solve for A′ and set to 0 to find maximum.

$$A' = -20t + 300 = 0$$

$$t = 15$$

Adding 15 trees will maximize apple production

Gradients

- **Derivative:** the slope in one direction
  - Also known as: f'(x)
- What about more features?
  - Most of our data will have numerous features
- **Gradient:** a multi-dimensional derivative
  - Also known as: f'(x,y,..) or **∇f**

Gradient Descent

- Goal : minimize $J(\theta)$
- Start with some initial θ and then perform an update on each θ_j in turn:

$$\theta_j^{k+1} := \theta_j^k - \alpha \frac{\partial}{\partial \theta_j} J(\theta^k)$$

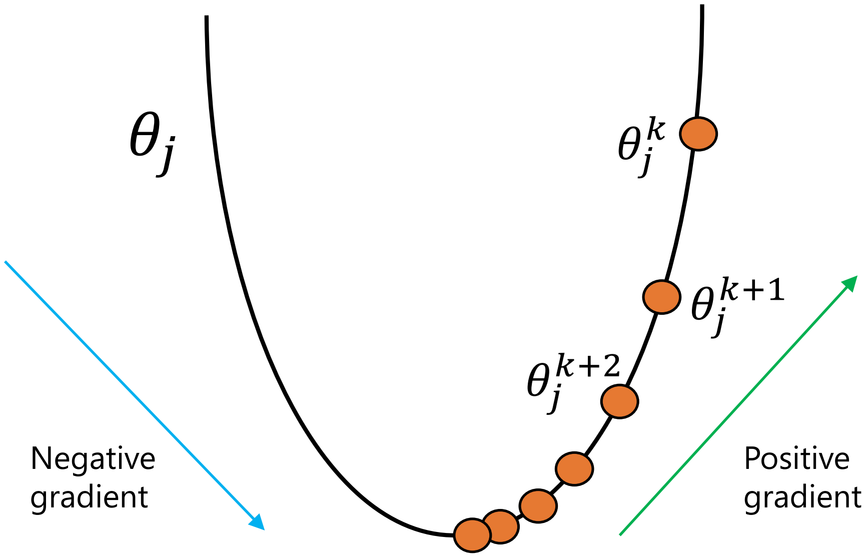
- Repeat until θ converges

Gradient Descent

$$\theta_j^{k+1} := \theta_j^k - \alpha \frac{\partial}{\partial \theta_j} J(\theta^k)$$

- α is known as the learning rate; set by user
- Each time the algorithm takes a step in the direction of the steepest descent and $J(\theta)$ decreases.
- α determines how quickly or slowly the algorithm will converge to a solution

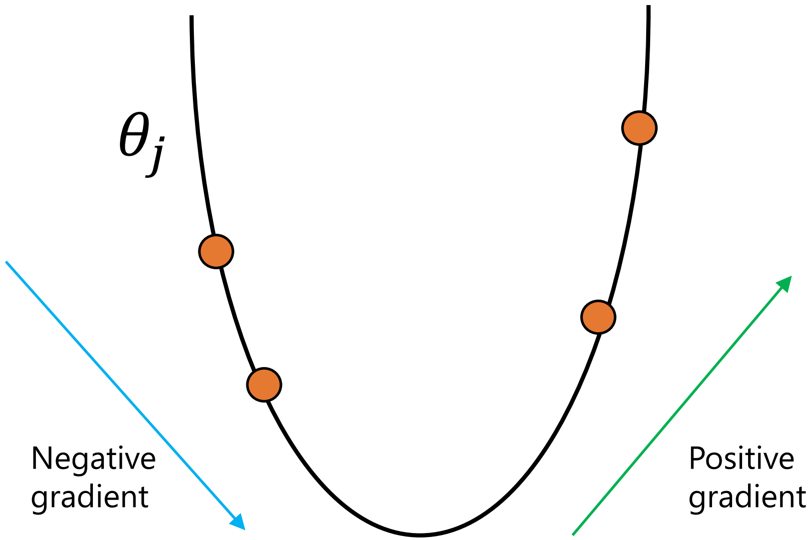
Intuition



$$\theta_j^{k+1} := \theta_j^k - \alpha \frac{\partial}{\partial \theta_j} J(\theta^k)$$

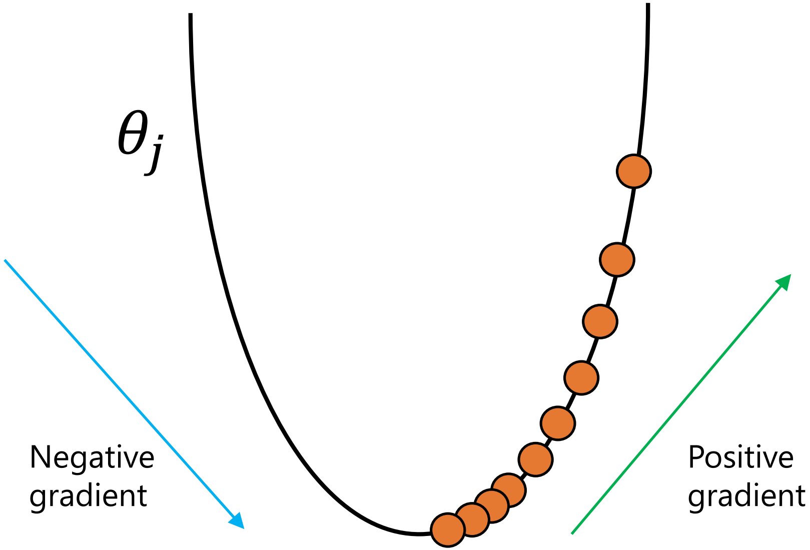
Learning Rate Effects Large α

$$\theta_j^{k+1} := \theta_j^k - \alpha \frac{\partial}{\partial \theta_j} J(\theta^k)$$



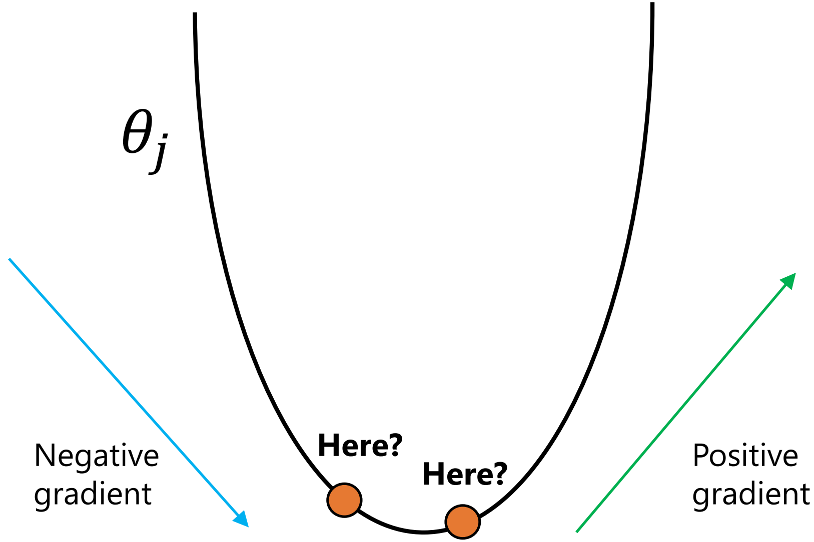
Learning Rate Effects Small α

$$\theta_j^{k+1} := \theta_j^k - \alpha \frac{\partial}{\partial \theta_j} J(\theta^k)$$



Gradient Descent Implementation

When do we stop updating?



- When θ_j^{k+1} is close to θ_j^k
- When $J(\theta^{k+1})$ is close to $J(\theta^k)$ [Error does not change]

Batch Gradient Descent

- How do we incorporate all our data?
- Loop!

For j from 0 to m:

$$\theta_j^{k+1} := \theta_j^k - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta\left(x^i\right) - y^i\right) x_j^i$$

- h_θ is updated only once the loop has completed
- Weaknesses?

Stochastic Gradient Descent

- Consider an alternative approach:

for i from 1 to n:

for j from 0 to m:

$$\theta_j^{k+1} := \theta_j^k - \alpha \frac{1}{n}\left(h_\theta\left(x^i\right) - y^i\right)x_j^i$$

- h_θ is updated when inner loop is complete
- If the training set is big, converges quicker than batch
- May oscillate around a minimum of $J(\theta)$ and never converge

Batch vs. Stochastic

Which is the best to use? It depends.

	Batch Gradient Descent	Stochastic Gradient Descent
Function	Updates hypothesis by scanning whole dataset	Updates hypothesis by scanning one training sample at a time
Rate of convergence	Slowly	Quickly (but may oscillate at minimum)
Appropriate Dataset Size	Small	Large

EVALUATING REGRESSION MODELS

Evaluation metrics for regression

- Mean Absolute Error (MAE)

- Root-Mean-Square Error (RMSE)
  - Root-Mean-Square Deviation

- Coefficient of Determination (R^2)

Mean Absolute Error

$$MAE(\theta) = \frac{\sum_{i=1}^{n} \left| h_\theta\left(x^i\right) - y^i \right|}{n}$$

- Mean of residual values
- "Pure" measure of error

Mean Absolute Error - Example

$$y = \{36, 19, 34, 6, 1, 45\}$$

$$h_\theta(x) = \{27, -2.6, 13, -7.3, -2.6, 48\}$$

$$|h_\theta(x) - y| = \{9, 21.6, 21, 13.3, 3.6, 3\}$$

$$MAE(\theta) = \frac{71.5}{6} = 11.9$$

Root-Mean-Square Error

$$RMSE(\theta) = \sqrt{\frac{\sum_{i=1}^{n}(h_\theta(x^i) - y^i)^2}{n}}$$

- Square root of mean of squared residuals
- Penalizes large errors more than small
- Good when large errors particularly bad

RMSE - Example

$$y = \{36, 19, 34, 6, 1, 45\}$$

$$h_\theta(x) = \{27, -2.6, 13, -7.3, -2.6, 48\}$$

$$(h_\theta(x) - y)^2 = \{81, 467, 441, 177, 13, 9\}$$

$$RSME(\theta) = \sqrt{\frac{1187}{6}} = 14.1$$

Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$SS_{res} = \sum_{i=1}^{n} \left(h_\theta\left(x^i\right) - y^i\right)^2 \qquad SS_{tot} = \sum_{i=1}^{n} \left(y^i - \bar{y}\right)^2$$

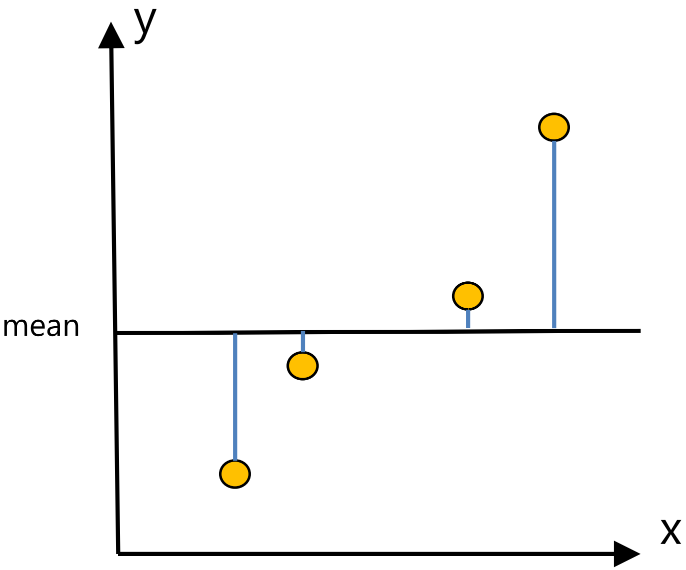
where

SS_{res} – Sum of squared residuals (i.e. total squared error)
SS_{tot} –Sum of squared differences from mean (i.e. total variation in dataset)

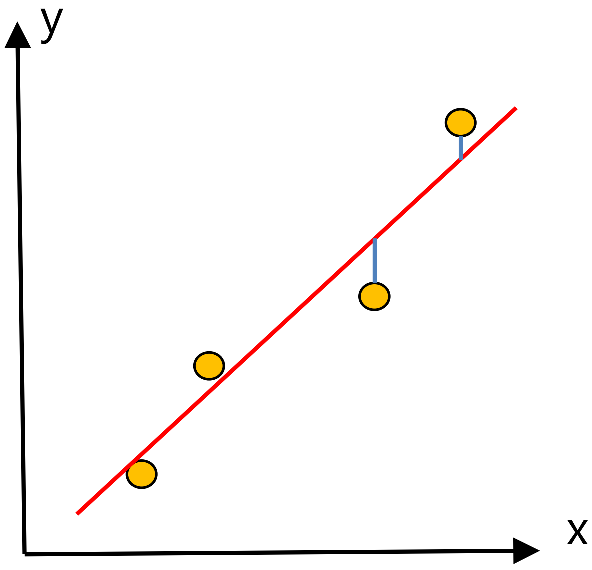
Result: Measure of how well the model explains the data

- "Fraction of variation in data explained by model"

Coefficient of Determination



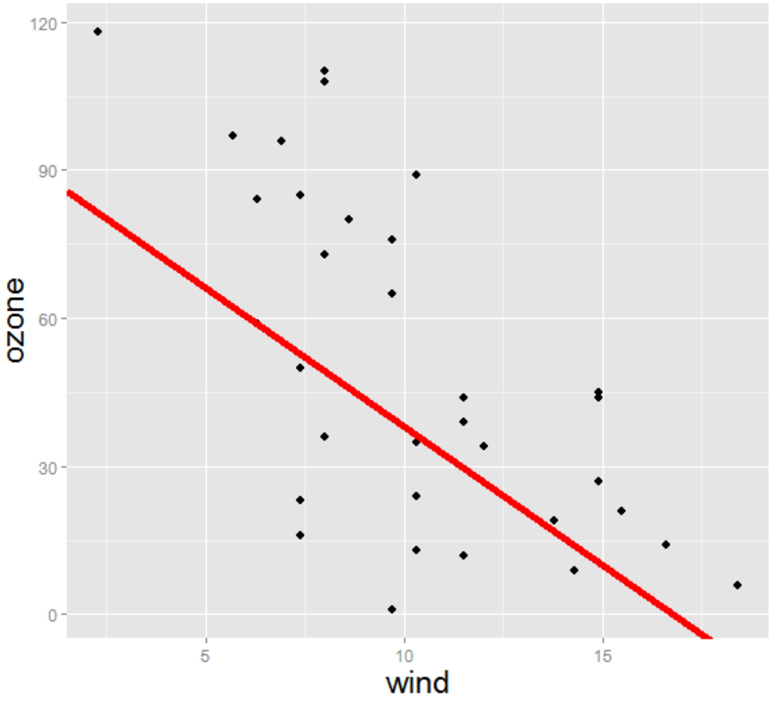
SS_{tot}



SS_{res}

R^2 Example

- $R^2 = 0.277$
- Want a much better model for real application
- $R^2 = 0.6$ can be a good model



datasciencedojo
data science for everyone

Adjusted R^2

- With a large number of features, some explanation of variation may occur by chance
- R^2 can be adjusted to account for large sets of features (or the independent variables)
- Adjustment typically done by dividing by factors proportional to the number of features

Note on p-values

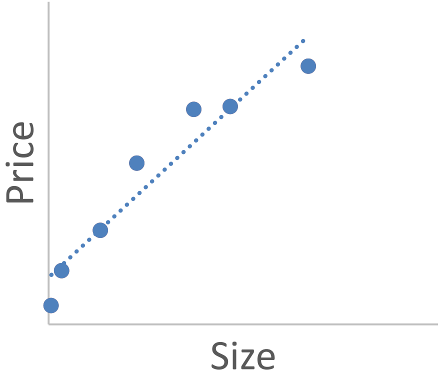
- $p < 0.05$
- Typically used to evaluate variables or features for statistical significance
- With enough features some may occur to be statistically significant by chance
- Beware of overfitting

REGULARIZATION

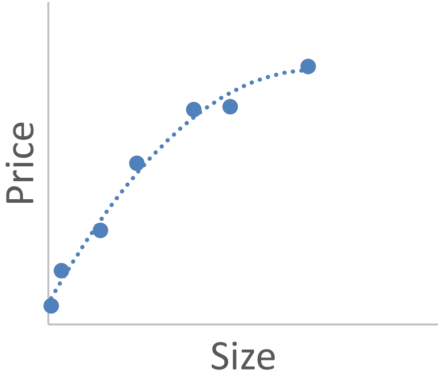
Overfitting

- Want to extract general trends
- Danger: "memorizing" the training set
- A model is **overfit** when model performance on test set is much worse than on training set.

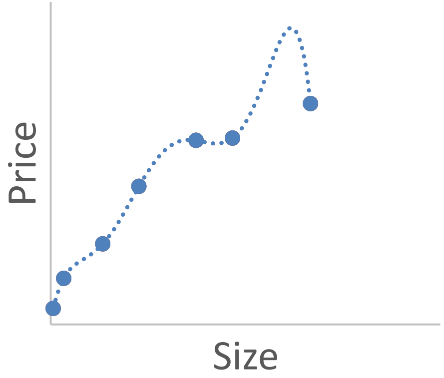
Overfitting



$$\theta_0 + \theta_1 x$$

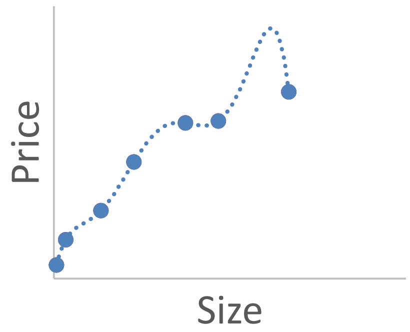


$$\theta_0 + \theta_1 x + \theta_2 x^2$$

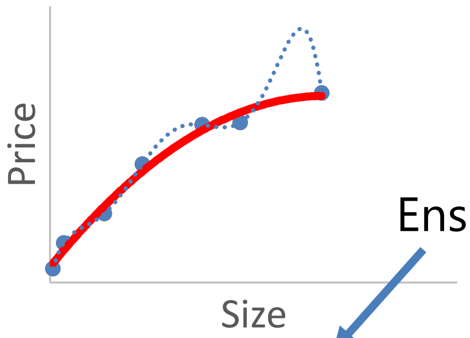


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Intuition



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$



Ensure Small

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

- Want to discourage complex models automatically – How?
- Adjust the cost function!
  - Penalize models with large high-order θ terms

$$J'(\theta) = J(\theta) + Penalty$$

Definitions

- Two most common
    - L1 regularization
        - lasso regression

$$J_{L1}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda\sum_{j=1}^{m}\left|\theta_j\right|$$

    - L2 regularization
        - ridge regression
        - weight decay

$$J_{L2}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda\sum_{j=1}^{m}\theta_j^2$$

Regularized Regression

$$J_{L1}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda\sum_{j=1}^{m}|\theta_j|$$

$$J_{L2}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda\sum_{j=1}^{m}\theta_j^2$$

- Find the best fit
- Keep the θ_j terms as small as possible.
- λ is a user-set parameter which controls the trade off

Regularized Regression

$$J_{L1}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda\sum_{j=1}^{m}\left|\theta_j\right|$$

$$J_{L2}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda\sum_{j=1}^{m}\theta_j^2$$

- Size of λ important
  - λ too high => no fitting
  - λ too low => no regularization

QUESTIONS

data science for everyone