Introduzione alla data science e al pensiero computazionale
Lezione 9: Regressione lineare

Giancarlo Succi

Dipartimento di Informatica – Scienza e Ingegneria

Università di Bologna

g.succi@unibo.it

Suppose that:

- I want to relate two random scalar phenomena, X and Y, to identify the relationships existing between them,
- I can measure their values several times i, so I can have a set of pairs (x_i, y_i) with i spanning the interval of observation, say $i \in [0 \dots n-1]$

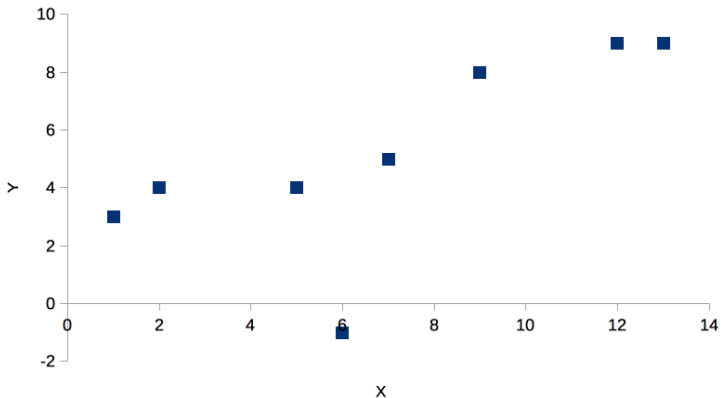| i	**X**	**Y**
0	1	3
1	2	4
2	5	4
3	6	-1
4	7	5
5	9	8
6	12	9
7	13	9

Using a simple and common approach, I may try to build a relationship between the two phenomena. However:

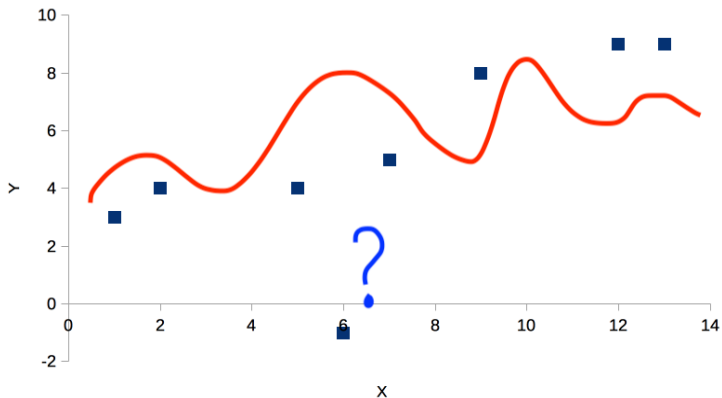- What kind of relationships I am going to look for?
- How do I build it?

In other words, I have this set of points:

How can I build a line that represent the relationships between these two sets?

Linear Regression – Definition

We need to define:

- A mean function that represents the relationship that I hypothesize between the phenomena X and Y
- A cost-minimization function to define the parameters of the mean function
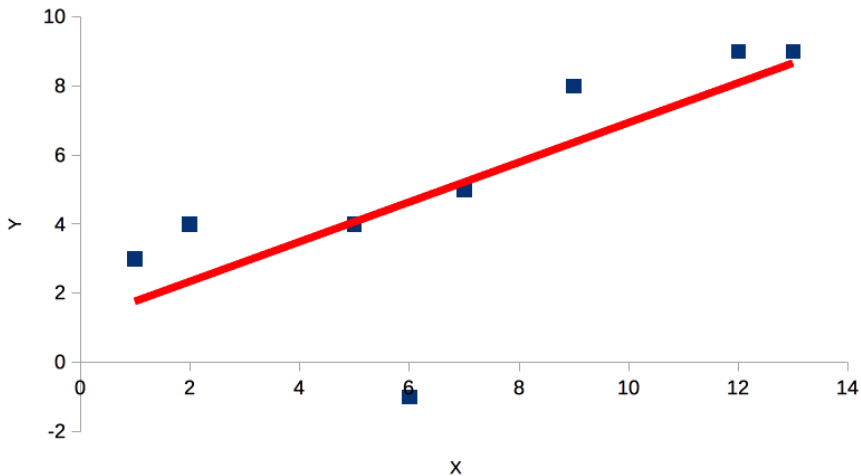
We will use initially:

- As mean function the simple line
- As cost function the square of the errors between the modeled values and the real values

We define Ordinary Least Squares (OLS) Linear Regression as a simple line that minimizes a square error between modelled values and real values.

This is what we would like to build:
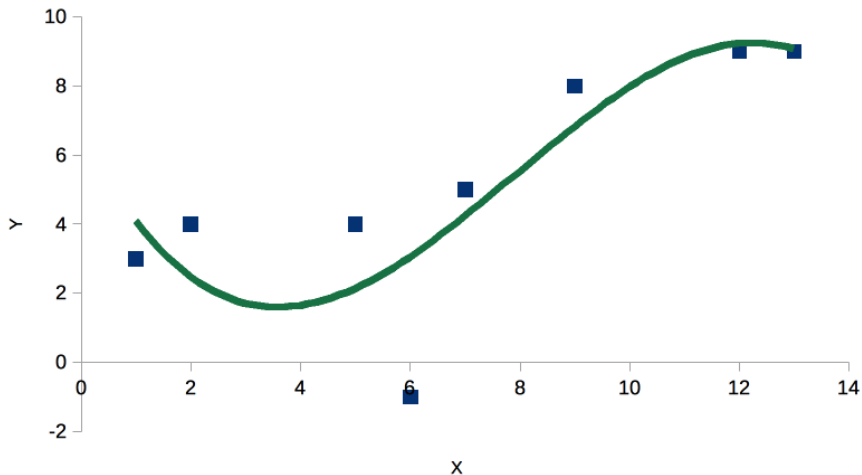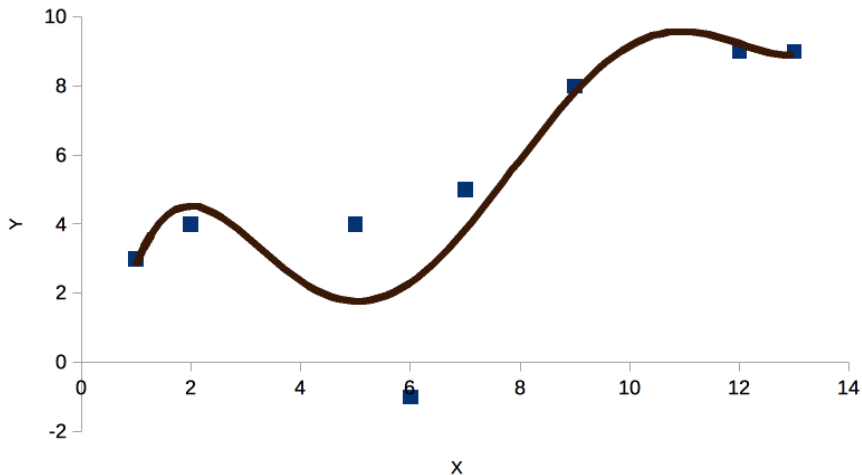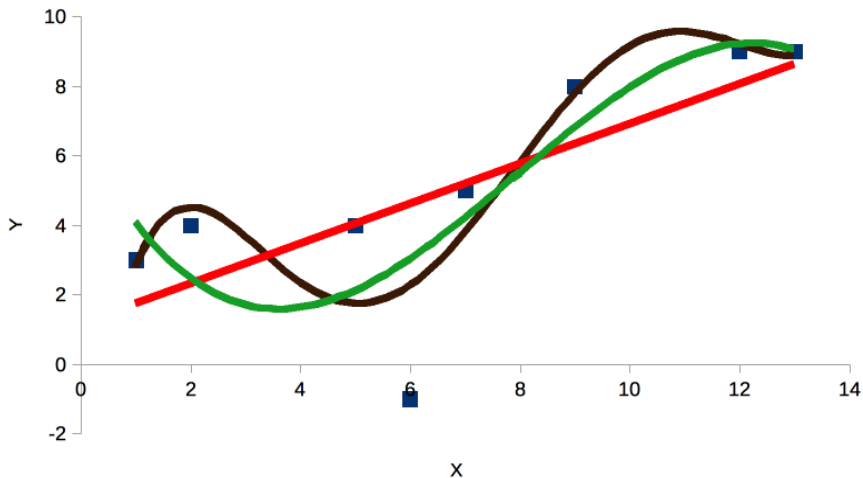
But we could have used as a mean function a cubic function:

But we could have used as a mean function a fifth order function:

What are the differences between all these 3?

I want to build a model of the kind:

$$Y = \theta_0 + \theta_1 X$$

Where X and Y are the phenomena that we are measuring.

Note:

- we know that there is no line passing for n arbitrary points with $n \leq 3$
- we need to introduce an approximation

$$\hat{Y} = \theta_0 + \theta_1 \hat{X} + \epsilon$$

- in our case ϵ is the error that is introduced by the approximation
- as we said, our cost function, our distance from the model, will be the square of the error ϵ^2
- θ_0 and θ_1 are called the regression coefficients

Altogether:

- we have a set of pairs (x_i, y_i) with $i \in [0 \ldots n-1]$
- we want to build n linear equations of the kind (the mean function):

$$y_i = \theta_0 + \theta_1 x_i + \epsilon_i$$

- and we start with an approximation of the kind:

$$\hat{y}_i = \theta_0 + \theta_1 x_i$$

Altogether:

- our goal is to compute θ_0 and θ_1 that minimize the quadratic error (the cost function)

$$\sum_{i=0}^{n-1} \epsilon_i^2$$

- notice that:
  - we will denote as (x_i, y_i) the original data
  - we will denote as $(\hat{x}_i, \hat{y}_i)$ the approximation that we obtain in the linear regression
  - x_i and $\hat{x}_i$ are the same
  - there could be errors in the slides and you get extra credits by finding them

Linear Regression – Computation 1

- Since

$$y_i = \theta_0 + \theta_1 x_i + \epsilon_i$$

- therefore

$$\epsilon_i = y_i - \theta_0 - \theta_1 x_i$$

- we need to minimize:

$$\sum_{i=0}^{n-1} \epsilon_i^2 = \sum_{i=0}^{n-1} (y_i - \theta_0 - \theta_1 x_i)^2$$

- we need to zero the two partial derivatives:

$$\frac{\partial \sum_{i=0}^{n-1} (y_i - \theta_0 - \theta_1 x_i)^2}{\partial \theta_i}$$

- so we have to solve two simple equations and then to check the Hessian

$$\frac{\partial \sum_{i=0}^{n-1} (y_i - \theta_0 - \theta_1 x_i)^2}{\partial \theta_0} = 0 \Rightarrow$$

$$2 \sum_{i=0}^{n-1} (y_i - \theta_0 - \theta_1 x_i) = 0 \Rightarrow$$

$$\sum_{i=0}^{n-1} (y_i - \theta_0 - \theta_1 x_i) = 0$$

$$\frac{\partial \sum_{i=0}^{n-1} (y_i - \theta_0 - \theta_1 x_i)^2}{\partial \theta_1} = 0 \Rightarrow$$

$$2 \sum_{i=0}^{n-1} x_i (y_i - \theta_0 - \theta_1 x_i) = 0 \Rightarrow$$

$$\sum_{i=0}^{n-1} x_i (y_i - \theta_0 - \theta_1 x_i) = 0$$

From the first equation:

$$\sum_{i=0}^{n-1} (\theta_0) = \sum_{i=0}^{n-1} (y_i - \theta_1 x_i) \Rightarrow$$

$$\sum_{i=0}^{n-1} (\theta_0) = \sum_{i=0}^{n-1} (y_i) - \theta_1 \sum_{i=0}^{n-1} (x_i) \Rightarrow$$

$$n\theta_0 = n\bar{y} - n\theta_1 \bar{x} \Rightarrow$$

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

$$\sum_{i=0}^{n-1} x_i(y_i - \theta_0 - \theta_1 x_i) = 0 \Rightarrow$$

$$\sum_{i=0}^{n-1} x_i y_i - \theta_0 \sum_{i=0}^{n-1} x_i - \theta_1 \sum_{i=0}^{n-1} x_i^2 = 0 \Rightarrow$$

$$\sum_{i=0}^{n-1} x_i y_i - n\theta_0 \bar{x} - n\theta_1 \bar{x^2} = 0 \Rightarrow$$

Substituting $\theta_0 = \bar{y} - \theta_1 \bar{x}$:

$$\sum_{i=0}^{n-1} x_i y_i - n(\bar{y} - \theta_1 \bar{x})\bar{x} - n\theta_1 \bar{x^2} = 0 \Rightarrow$$

$$\sum_{i=0}^{n-1} x_i y_i - n\bar{y}\bar{x} + n\theta_1 \bar{x}^2 - n\theta_1 \bar{x^2} = 0$$

$$n\theta_1(\bar{x^2} - \bar{x}^2) = \sum_{i=0}^{n-1} x_i y_i - n\bar{y}\bar{x}$$

$$\theta_1 = \frac{\sum_{i=0}^{n-1} x_i y_i - n\bar{y}\bar{x}}{n(\bar{x^2} - \bar{x}^2)}$$

$$\theta_1 = \frac{\frac{\sum_{i=0}^{n-1} x_i y_i}{n} - \bar{y}\bar{x}}{(\bar{x^2} - \bar{x}^2)}$$

$$\theta_1 = \frac{Cov(x, y)}{Var(x)}$$

Which we can also write as:

$$\theta_1 = \frac{\sum_{i=0}^{n-1}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=0}^{n-1}(x_i - \bar{x})^2}$$

Using the formula above we obtain that for the following dataset:

i	$\mathbf{X}$	$\mathbf{Y}$
0	1	3
1	2	4
2	5	4
3	6	-1
4	7	5
5	9	8
6	12	9
7	13	9

We have an equation:

$$\hat{Y} = \theta_0 + \theta_1 \hat{X}$$

with:

- $\theta_0 = 1.179$
- $\theta_1 = 0.574$

i	$\mathbf{X}$	$\mathbf{Y}$	$\hat{Y}$	ϵ
0	1	3	1.753	1.247
1	2	4	2.327	1.673
2	5	4	4.049	-0.049
3	6	-1	4.623	-5.623
4	7	5	5.197	-0.197
5	9	8	6.345	1.655
6	12	9	8.067	0.933
7	13	9	8.641	0.359

Build a linear regression for the following dataset:

X	Y
1	2
3	9
5	9
6	11
7	14
9	15
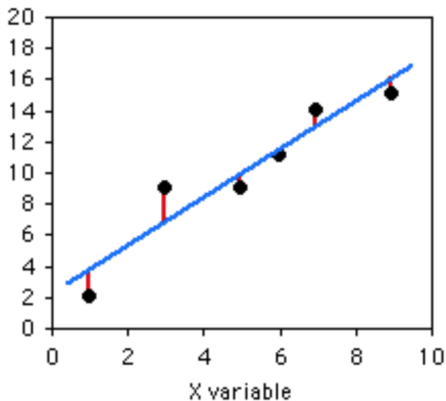
The regression equation for these numbers is $\hat{y} = 2.0286 + 1.5429x$.
Now, fill the blanks using such equation and calculate the sum of squared deviations (last column).

x	y	Predicted y ($\hat{y}$)	Deviate from predicted (abs.)	Squared deviate
1	2			
3	9			
5	9			
6	11			
7	14			
9	15			

Results. The sum of squared deviations: 10.8

x	y	Predicted y ($\hat{y}$)	Deviate from predicted (abs.)	Squared deviate
1	2	3.57	1.57	2.46
3	9	6.66	2.34	5.48
5	9	9.74	0.74	0.55
6	11	11.29	0.29	0.08
7	14	12.83	1.17	1.37
9	15	15.91	0.91	0.83

In fact, we might think to use linear regression to model phenomena, assuming a linear dependence between input (the collected parameters) and output.

Here are some "real world" examples (w.r.t. certain assumptions):

- - Impact of SAT Score (or GPA) on College Admissions;
- - Impact of product price on number of sales;
- - Impact of rainfall amount on the number of fruits yielded;
- - Impact of blood alcohol content on coordination.

We can evaluate the quality of linear regression, i.e. assess how good the model for the data that we have:

- - by the sum of squares of residuals;
- - by the coefficient of determination.

The sum of squares of residuals, also called the residual sum of squares:

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2$$

In the case above SS_{res} is equal to 39.751672.

The coefficient of determination describes the proportion of variance of the dependent variable explained by the regression model. If the regression model is "perfect", SS_{res} is zero, and R^2 is 1.

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

The total sum of squares:

$$SS_{tot} = \sum_i (y_i - \bar{y})^2$$

where

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

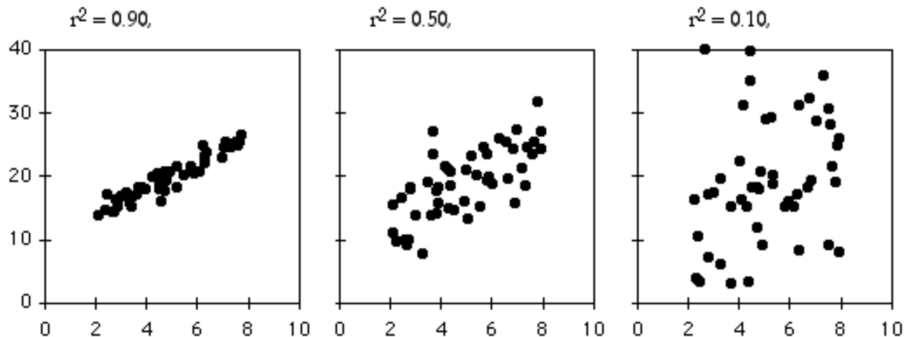$$SS_{tot} = \sum_i (y_i - \bar{y})^2 = 82.875$$

Remember that:

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2 = 39.751672$$

Therefore:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{39.751672}{82.875} = 0.5203$$

Multivariate Linear Regression

- The "X" variable is often called "feature" in machine learning.
- Indeed, we could have multiple features, say, n.
- If we also have m observations, we could build a system of m equations of the kind:

$$y_i = \boldsymbol{\theta}^T \cdot \boldsymbol{x}_i + \epsilon_i, i = 1 \ldots m$$

- and then we will build our linear regression (approximation) as:

$$\hat{y}_i = \boldsymbol{\theta}^T \cdot \hat{\boldsymbol{x}_i}, i = 1 \ldots m$$

- where $\boldsymbol{x}_i$ and $\hat{\boldsymbol{x}_i}$– are vectors of $n + 1$ features for the i-th observation

Question: Why here we use $n + 1$?

To find the value of $\boldsymbol{\theta}$, there is a closed-form solution, a mathematical equation that gives the result directly.

This is called the **Normal Equation**:

$$\theta = (\boldsymbol{X} \cdot \boldsymbol{X}^T)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y}$$

- We start considering a set of m equations of the form:

$$\hat{y}_i = \boldsymbol{\theta}^T \boldsymbol{x}_i, i = 1 \ldots m$$

where $\boldsymbol{x}_i$ has dimension $n + 1$

- We move all the model in matrix format:

$$\hat{\boldsymbol{y}} = \boldsymbol{X} \cdot \boldsymbol{\theta}$$

Notice that $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ have dimension (m,1), $\boldsymbol{X}$ (m,n+1), and θ (n+1,1). $\boldsymbol{X} \cdot \boldsymbol{\theta}$ has therefore dimension (m,1) as it should be.

- The error vector $\boldsymbol{\epsilon}$ is defined for each pair as:

$$\boldsymbol{\epsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y} = \boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y}$$

- And the square of the error is:

$$(\boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y})^T (\boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y})$$

- To determine the values of the parameters we take the partial derivatives and we null them:

$$\frac{\partial (\boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y})^T (\boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y})}{\partial \boldsymbol{\theta}} = 0$$

- Now we evaluate:

$$\frac{\partial (\boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y})^T (\boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y})}{\partial \boldsymbol{\theta}} =$$

$$= \frac{\partial ((\boldsymbol{X} \cdot \boldsymbol{\theta})^T (\boldsymbol{X} \cdot \boldsymbol{\theta}) - (\boldsymbol{X} \cdot \boldsymbol{\theta})^T \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{X} \cdot \boldsymbol{\theta} + \boldsymbol{y}^T \boldsymbol{y})}{\partial \boldsymbol{\theta}} =$$

$$= \frac{\partial ((\boldsymbol{X} \cdot \boldsymbol{\theta})^T (\boldsymbol{X} \cdot \boldsymbol{\theta}) - 2(\boldsymbol{X} \cdot \boldsymbol{\theta})^T \boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{y})}{\partial \boldsymbol{\theta}}$$

- Now we can consider that:

$$\frac{\partial(\boldsymbol{y}^T\boldsymbol{y})}{\partial\boldsymbol{\theta}} = 0$$

- that:

$$\frac{\partial((\boldsymbol{X}\cdot\boldsymbol{\theta})^T\boldsymbol{y})}{\partial\boldsymbol{\theta}} = \boldsymbol{X}^T\boldsymbol{y}$$

- Notice that $\boldsymbol{X}^T\boldsymbol{y}$ has dimension (n+1,m) · (m,1), that is, (n+1,1).

- and finally that:

$$\frac{\partial((\boldsymbol{X}\cdot\boldsymbol{\theta})^T(\boldsymbol{X}\cdot\boldsymbol{\theta}))}{\partial\boldsymbol{\theta}} = 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta}$$

- Notice that $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta}$ has dimension (n+1,m) · (m,n+1) · (n+1,1), that is, (n+1,1) as it should be.

- Substituting the results in the original formula:

$$2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta} - 2\boldsymbol{X}^T\boldsymbol{y} = 0 \Rightarrow$$

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{X}^T\boldsymbol{y} \Rightarrow$$

- Notice that $\boldsymbol{X}^T\boldsymbol{X}$ has dimension (n+1,m) · (m,n+1), that is, (n+1,n+1). Notice that m ≫ n, so we *hope* that $\boldsymbol{X}^T\boldsymbol{X}$ is invertible.

$$\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- QED.

The Normal Equation computes the inverse of $X^T \cdot X$, which is an n x n matrix (where n is the number of features).
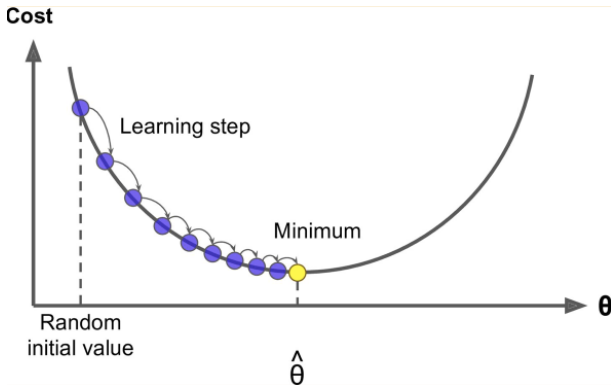
The computational complexity of inverting such a matrix is typically about $O(n^{2.4})$ to $O(n^3)$ (depending on the implementation).

In other words, if you double the number of features, you multiply the computation time by roughly $2^{2.4} = 5.3$ to $2^3 = 8$.

Gradient Descent is a very generic optimization algorithm capable of finding optimal solutions to a wide range of problems. The general idea of Gradient Descent is to tweak parameters iteratively in order to minimize a cost function.

To implement Gradient Descent, you need to compute the gradient of the MSE cost function with regards to each model parameter θ_j.
Mean squared error (MSE) cost function for a Linear Regression model:

$$MSE(\theta) = \frac{1}{m} \sum_{k=1}^{m} (\boldsymbol{\theta}^T \cdot \boldsymbol{x}^{(k)} - \boldsymbol{y}^{(k)})^2$$

$x^{(k)}$ - k-th observation vector ($x^{(k)}$ is an n-dimensional vector)

To implement Gradient Descent, you need to compute the gradient of the MSE cost function with regards to each model parameter θ_j.

$$\frac{\partial}{\partial \theta_j} MSE(\theta) = \frac{2}{m} \sum_{i=1}^{m} (\theta^T \cdot \boldsymbol{x}^{(i)} - \boldsymbol{y}^{(i)}) x_j^{(i)}$$

In vector form:

$$\nabla_\theta MSE(\theta) = \frac{2}{m} \boldsymbol{X}^T (\boldsymbol{X} \cdot \boldsymbol{\theta} - \boldsymbol{y})$$
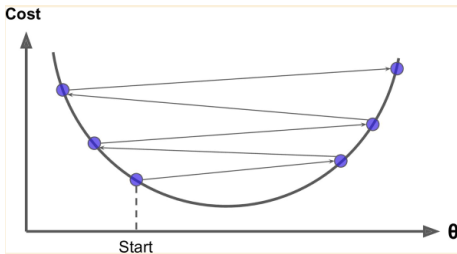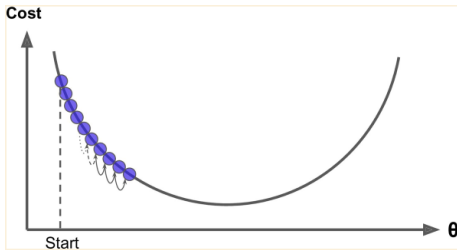
We update vector $\boldsymbol{\theta}$ step by step:

$$\boldsymbol{\theta}^{next} = \boldsymbol{\theta} - \eta \nabla_\theta MSE(\theta)$$
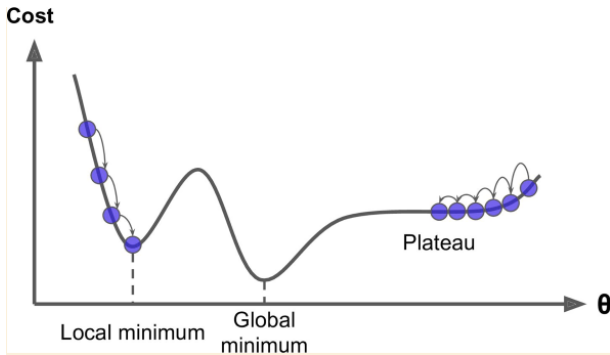
η – learning rate

Linear Regression is a statistical model developed in the field of Regression Analysis.

Later it was borrowed for the use of Machine Learning field.

Terminology difference

Regression analysis	Machine Learning
estimation, fitting	training, learning
regressors	features
response	target

References

1) http://www.cs.umd.edu/~djacobs/CMSC426/Convolution.pdf

2) https://www.researchgate.net/post/
Difference_between_convolution_and_correlation

3) https://www.tutorialspoint.com/signals_and_systems/
convolution_and_correlation.htm

Fine della lezione nove.