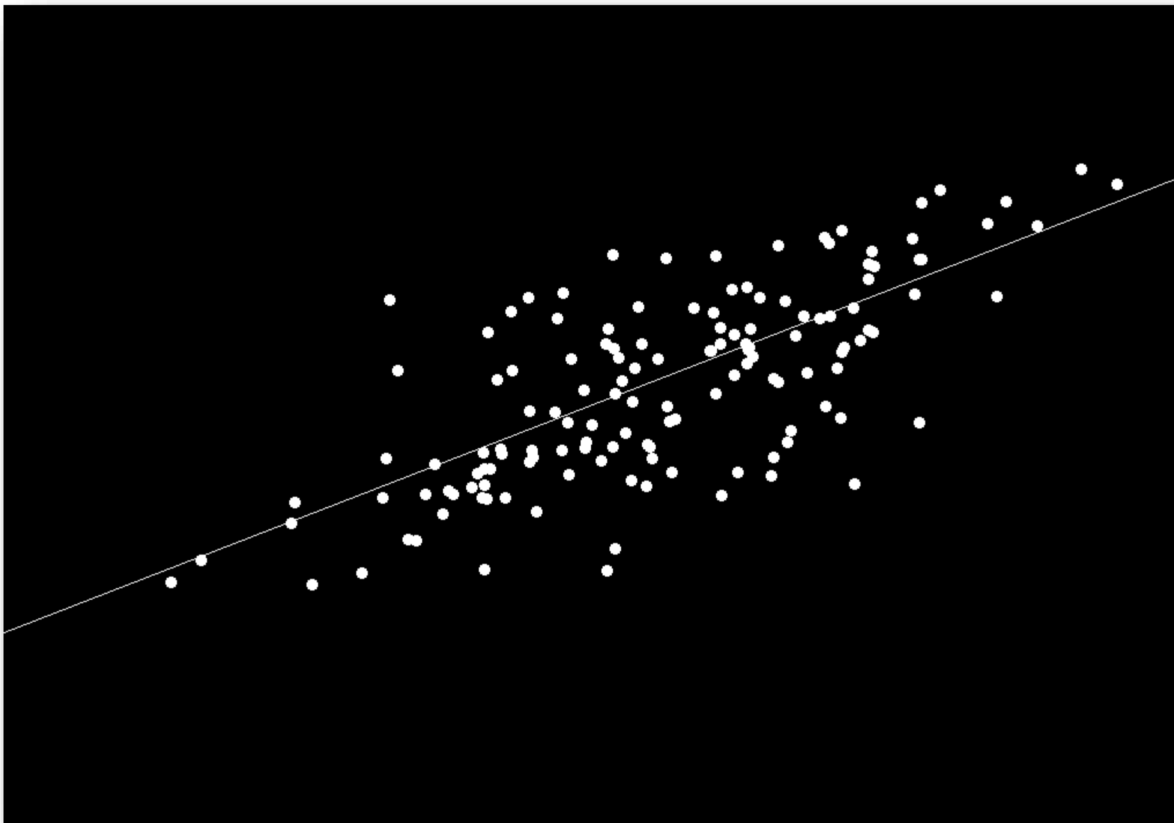
Machine-Learning Algorithms:

Simple Linear Regression & Polynomial Linear Regression

Least Squares Method & Stochastic Gradient Descent



. Mateus Branco .

. 2017 .

Index

Introduction

In this document I will explain the workings behind two Machine-Learning models known as “Simple Linear Regression”, using Ordinary Least Squares method for estimation method, and “Polynomial Linear Regression”, using “Stochastic gradient descent” as the optimization algorithm.

I will start by giving a more broad explanation in order to build intuition of the underlying mechanics of more simple and then general cases of regression such as “Simple Linear Regression” and “The Linear Predictor Function” respectively, so we can then dive smoothly into more complex concepts, namely “Polynomial Linear Regression” and “Stochastic gradient descent”.

Chapter 1 – Understanding the intuition behind *Simple Linear Regression*

Simple Linear Regression

First, let us decompose this concept into its components so that we can get a better feel for what it really means.

::Regression::

Starting from the end, "Regression" refers to the kind of statistical modeling process used for estimating the relationships between dependent (also called "explanatory variables") and independent variables. More specifically, it helps one understand how the values of the dependent variables change when hold fixed and any of the dependent variables is varied. As an example: how the rise in taxes(X) influences the number of cigarette packs consumed(Y).

-The dependent Variable (typically denoted as X) is usually a variable whose values we want to explain or predict and are dependent of other variables.

-The independent (typically denoted as Y) or explanatory variable is a variable that explains the other variables dependent on it while its values are independent.

::Linear::

"Linear" refers the approach we are using to tackle the regression-modeling problem. It being Linear means that our model will satisfy Linearity and that it can be graphically represented as a straight line.

::Simple::

"Simple" refers to the kind of Linear Regression method we are performing. In this case, it means that it concerns two-dimensional sample/data points were we only have one dependent and one independent variable, **Y** being dependent of **X**.

Chapter 2 – Defining *Simple Linear Regression* in more formal terms

Now that we have a good idea of what Linear Regression is, we can start talking about a way of rigorously define it.

We can think of linear regression as the "Line of best fit" in relationship to a set of Input Data.

Its mathematical definition can be represented by the linear equations:

$Y = A + BX$	|Or|	$Y = mX + b$	|Or|	$Y = B_0 + B_1X$

Let us look at the second example $Y = mX + b$

: **B** is the Intercept; it will move the intercept point up and down along the Y-axis, if changed.

: **m** is the Slope or gradient; the line will rotate around the intercept, if changed.

Chapter 3 – Least Squares Method

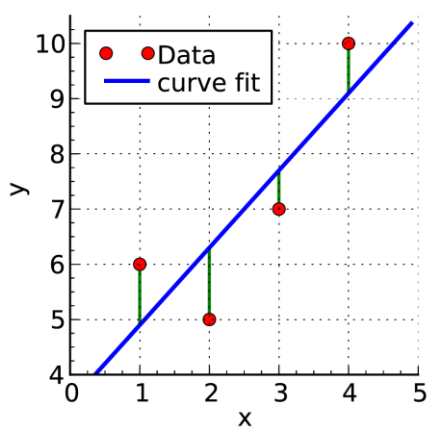
We can use the *Least Squares Method* as an estimation method to fit our data set and find the "Line of best fit". By using the following formulas, it is possible to calculate the values of α and β for the linear equation that will represent our fitting line:

$$y = \beta + \alpha x,$$

$$\alpha = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$\beta = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

```
void LeastSquares() {
	float xsum = 0;
	float ysum = 0;
	float xysum = 0;
	float xsqrsum = 0;
	int n = 0;

	for each (dataPoint p in PointsData)
	{
		n++;
		float x = p.x;
		float y = p.y;

		xsum += x;
		xsqrsum += pow(x, 2);
		ysum += y;
		xysum += x*y;
	}

	float num = 0;
	float den = 0;
	//y = β + αx
	//Beta
	num = (ysum)*(xsqrsum)-(xsum)*(xysum);
	den = n*(xsqrsum)-pow(xsum, 2);
	b[0] = (num / den);
	//Alpha
	num = n*(xysum)-(xsum)*(ysum);
	den = n*(xsqrsum) - pow(xsum, 2);
	b[1] = (num / den);
}
```



Chapter 4 – The Linear Predictor Function

: Basic Form

The kind linear equations showed above are one of the forms of the more general **Linear Predictor Function** written as:

$$f(i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p \, x_{ip},$$

Were $\boldsymbol{\beta_0}, \ldots, \boldsymbol{\beta_p}$ are the coefficients (regression coefficients, weights, etc.) indicating the relative effect of a particular explanatory variable on the outcome.

: Matrix Form

This kind of function can also be described using matrix notation as:

$$f(i) = \beta^T x_i = x_i^T \beta$$

Were $\boldsymbol{\beta}$ and $\boldsymbol{x_i}$ are assumed to be a **p-by-1** column vectors (as is standard when representing vectors as matrices);

$\boldsymbol{\beta^T}$ Indicates the Matrix transpose of $\boldsymbol{\beta}$ (which turns it into a **1-by-p** row vector);

And $\boldsymbol{\beta^T x_i}$ indicates matrix multiplication between the **1-by-p** row vector and the **p-by-1** column vector, producing a **1-by-1** matrix that is taken to be a scalar.

An example of the usage of such a linear predictor function is in **linear regression**, where each data point is associated with a continuous outcome ***yi***, and the relationship written:

$$y_i = f(i) + \epsilon_i = \beta^T x_i + \epsilon_i$$

Were $\boldsymbol{\epsilon_i}$ is a disturbance term or **error variable** – an unobserved random variable that adds noise to the linear relationship between the dependent variable and predictor function.

Chapter 5 – Simple VS multiple Linear Regression

The simplest case of a *Predictor Function* is one consistent of a single scalar dependent variable x and a single scalar independent variable y. It is known as **simple linear regression**.

The extension to multiple and/or vector-valued predictor variables (denoted with a capital X) is known as **multiple linear regression**, also known as **multivariable linear regression**.

Chapter 6 – The Polynomial Linear Regression

Polynomial Regression is a form of regression analysis in which the relationship between the independent variable X and the dependent variable Y is modelled as an nth degree polynomial in x.

Although *Polynomial Regression* fits a nonlinear relationship between the value of X and the corresponding conditional mean of Y, as a statistical estimation problem, it is linear.

We can model the expected value of y as an nth degree polynomial, yielding the general definition for a **Polynomial linear Regression** as following:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \epsilon$$

Computational and inferential problems of polynomial regression can be completely addressed using the techniques of multiple regression. This is done by treating x, x2, ... as being distinct independent variables in a *multiple regression model*.

Chapter 7 – Gradient Descent

When there are one or more different inputs / independent variables, as it is the case of *multiple regression models,* one can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on the training data.

We can imagine this problem as finding the lowest point of a curve, its local minimum. We could find this point by dropping a ball in one of the sides and let it roll down the slope of the curve until it reaches the bottom. This Lowest point will represent the minimum error value for our model and the best values for our Line’s coefficients.

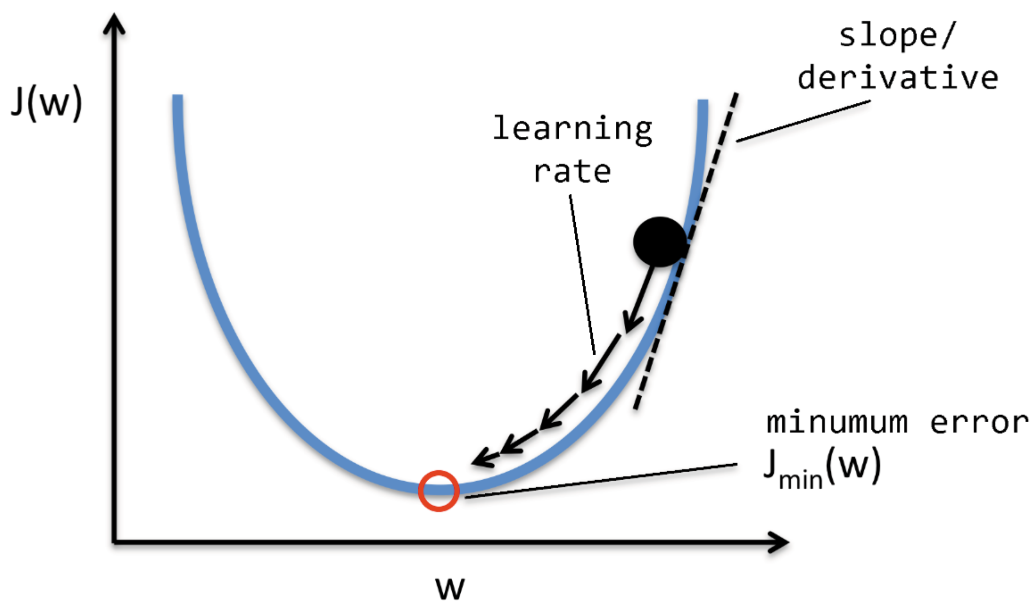
Now transcribing this intuition into rigorous definition, we can observe the slope of this curve as the derivative of a point. Knowing that at the most optimal position we have $\frac{\partial}{\partial\beta} = 0$, we can infer the error at any given time-step by finding the difference and change the β coefficients respectively, as follows:

$$\rho \in \mathbb{Z}$$

$$f(x) = \sum_{n=0}^{\rho+1} \beta_n x^n$$

$$\forall N \in \mathbb{Z} : \frac{\partial}{\partial\beta} = \frac{\theta}{N} \sum_{i=0}^{\rho+1} x_i^i (y_i - f(x_i))$$

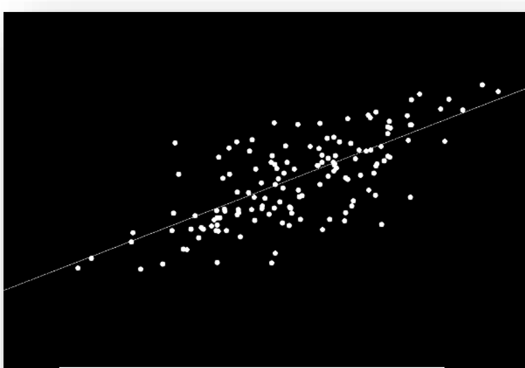
Were ρ represents the polynomial degree of the line; N represents the number of Data Points; θ represents the Learning rate of the model.



```
float fx(float X)
{
	float yfinal = 0;
	for (int n = 0; n < PolynDegree + 1; n++) {
		yfinal += (b[n] * pow(X, n));
	}
	return yfinal;
}

void gradientDescent(float learning_rate)
{
	for each (dataPoint p in PointsData)
	{
		float x = p.x;
		float y = p.y;

		float yHat = fx(x);
		float error = (y - yHat);

		for (int i = 0; i < PolynDegree + 1; i++) {

		b[i] += (error * pow(x,i) * learning_rate)/PointsData->size();

		}
	}
}
```

This algorithm is generalized to estimate the best fitting line for any nth degree polynomial function, this way working with *Simple Linear Regression* and *Multiple Linear Regression* if necessary.

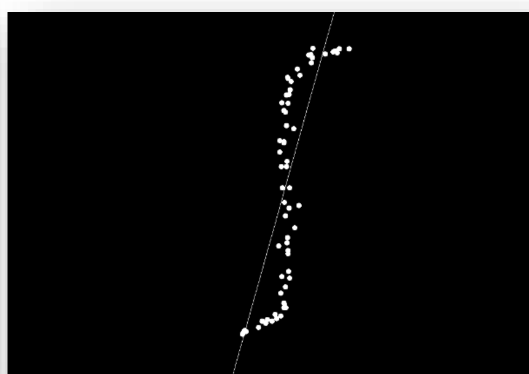
VISUALISATION DEMO



Least Squares Method

Simple Linear Reg.

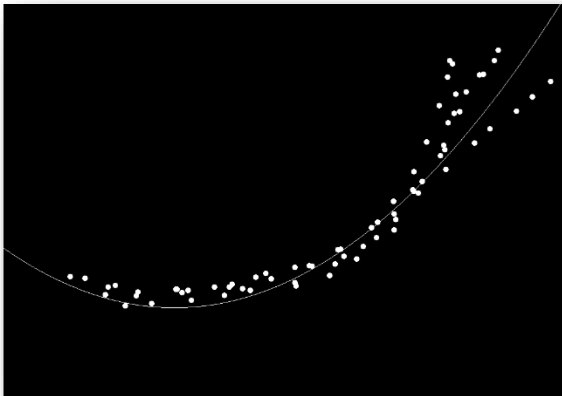
Example 1



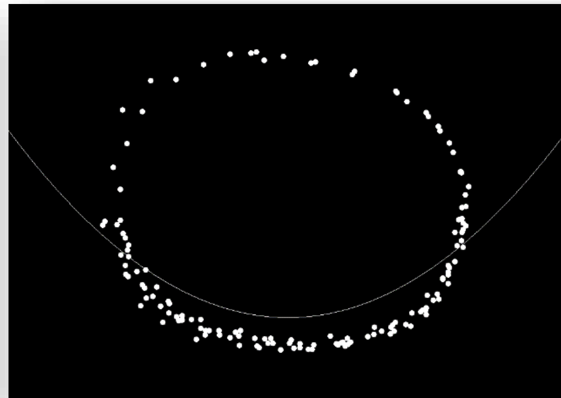
Least Squares Method

Simple Linear Reg.

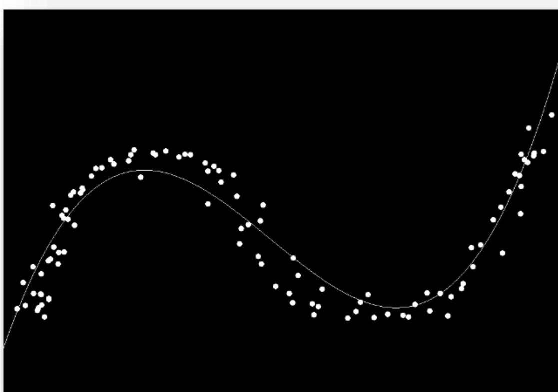
Example 2



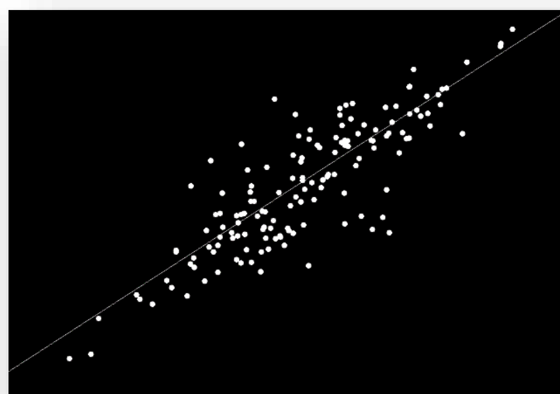
Gradient Descent
2^{nd} degree Polyn. Reg.
Example 3



Gradient Descent
2^{nd} degree Polyn. Reg.
Example 4



Gradient Descent
3^{rd} degree Polyn. Reg.
Example 5



Gradient Descent
1^{st} degree Polyn. Reg.
Example 6

Self-evaluation

Grade I think I deserve on this document: 5

References

Linear regression - Wikipedia

Polynomial regression - Wikipedia

Linear regression - Wikipedia

Conditional expectation - Wikipedia

https://upload.wikimedia.org/wikipedia/commons/b/b0/Linear_least_squares_example2.svg

Linear predictor function - Wikipedia

Random variable - Wikipedia

Simple linear regression - Wikipedia

Correlation and dependence - Wikipedia

(151) Linear Regression - Machine Learning Fun and Easy - YouTube

(151) How to Do Linear Regression using Gradient Descent - YouTube

Regression analysis - Wikipedia

Simple linear regression - Wikipedia

Gradient descent - Wikipedia

Introduction to Gradient Descent Algorithm along its variants

Polynomial regression - Wikipedia

Gradient descent - Wikipedia

Stochastic gradient descent - Wikipedia

Category:Stochastic optimization - Wikipedia

Category:Optimization algorithms and methods - Wikipedia

Why is polynomial regression considered a special case of multiple linear regression? - Cross Validated

Official Synonyms, Official Antonyms | Thesaurus.com

Account Synonyms, Account Antonyms | Thesaurus.com

An overview of gradient descent optimization algorithms

CS231n Convolutional Neural Networks for Visual Recognition

Category:Optimization algorithms and methods - Wikipedia

ball.png (1958×1061)

Gradient descent - Wikipedia