Simple Linear Regression

Jaya Nilakantan
Dawson College
jnilakantan@dawsoncollege.qc.ca

1. Introduction

In this exercise, we are introduced to both data sets and an algorithm in supervised learning.

Finding the line of best fit is one of the most intuitive algorithms in supervised learning. We will use a simple dataset to visualize the concept, and then extrapolate to a slightly more complex dataset.

1.1. Recall: Supervised learning

Supervised learning starts with a set of labeled data (called *training* data) which is used to make a mathematical or statistical model. The model is used for either prediction or classification purposes.

In order to validate the model, *test* data is used. The test data has the same formatting as the training data, and is chosen to properly reflect the different cases/demographics/conditions (called *features*) which are being looked at. Training and test data are important: they drive the correctness of the algorithm (the confidence with which you can make a conclusion).

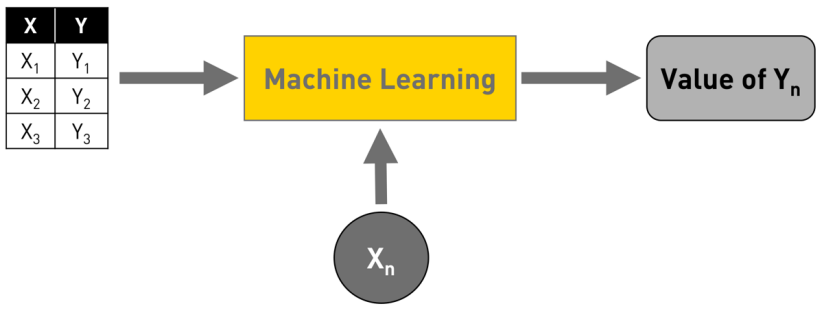

Figure 1. Predicting a value based on existing data

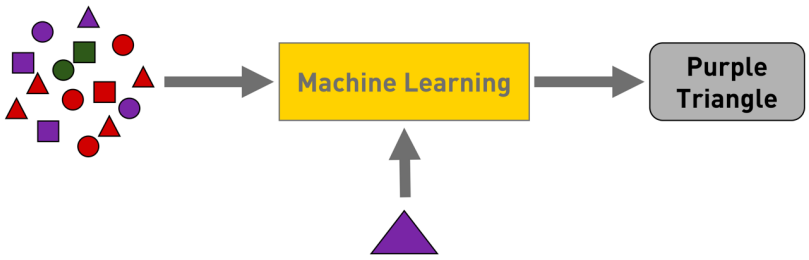

Figure 2. Classifying an object based on existing data

1.2. Where to find data?

There are various places where you can find data to create and test models:

- repositories of open data
  - websites like Kaggle (kaggle.com)
  - Microsoft Research open data (msropendata.com)
  - government open data (e.g., donnees.ville.montreal.qc.ca)
  - Classic data sets (list at https://en.wikipedia.org/wiki/Data_set#Classic_data_sets)
- APIs
  - for example, Reddit, Stack Overflow, NY Times, Wikipedia, …
- web scraping

In this exercise, we are first using a dataset representing salary and years of experience, followed by a dataset representing house prices.

1.3. Years of experience and salary

We are using a simple dataset originally from the Stack Overflow salary calculator. Here is a subset of the data:
Based on this data, how would you predict the salary after 4 years experience? You would probably look for a pattern: is the salary increasing by some predictable amount ever year. You are looking for the line of best fit.

The *features* are the variables that influence the result, or the *label*. With supervised learning, given the right answer (label) for some data (features), the algorithm is able to predict/classify a new set of data that it has never seen before with some level of confidence. In this simple dataset, we have a single feature, which is the years of experience, with the label being the salary.

What does the data look like?

Your instincts will tell you that there is a line that goes through this data, and any predicted value will be close to this line with some margin of error. This is an example of a *regression problem* where both the input (features) and output (label) are continuous (e.g. numerical values, such as someone's height

YearsExperience	Salary
1.1	39343
1.3	46205
1.5	37731
2	43525
2.2	39891
2.9	56642
3	60150
3.2	54445
3.2	64445

Figure 3. Subset of Stack Overflow salary data

or salary), as opposed to discrete (e.g. classifying cats vs. dogs - a *classification problem*).

1.4. Line of best fit

We want to create a *model* from the training data to use for future predictions. It is a straight line with a single feature. So it will have the form:

$$y = mx + b \tag{1}$$

Let's rewrite this equation in a different form

$$h_w(x) = w_0 + w_1 x \tag{2}$$

where $h_w(x)$ represents the *hypothesis* or the prediction function regarding the salary. w_0 (the y-intercept, sometimes called the *bias*) and w_1 (the slope, sometimes called the *weight* that we apply to the feature) change the straight line: they are the *parameters* to the model.

So how do you solve for w_0 and w_1 to get the best fitting line? Consider these two options:

Figure 5 shows what you know intuitively: the line of best fit is the one where the sum of all the distances from the datapoints to the line is minimized.

The training set (salary_data.csv) has 30 data points: we say m, which represents the number of training samples, is 30.

The x's are the input variable, or feature - in this case, the years of experience.

The y's are the output variable, or label. This is the target.

Figure 4. Scatter Chart of Stack Overflow salary data

So (x, y) denotes a single training sample, and $(x^{(i)}, y^{(i)})$ denotes the i^th training sample (i will be between 1 and 30). Notice that the superscript $^{(i)}$ represents which data point we are talking about, not an exponential! So $(x^{(2)}, y^{(2)})$ refers to the second data point (1.3, 46205).

So how do we find the best line? There are many ways to solve linear regression with one variable problems. One algorithm used often with simple linear regression (simple meaning that there is only 1 explanatory variable, or feature) to estimate the weights w_0 and w_1 is the ordinary least squares.

1.5. Ordinary Least Squares (OLS)

For each of the i data points in our training set of size m, we'll look at what our model would hypothesize the output to be given the input i.e., $h_w(x^{(i)})$, where x is the years of experience versus the actual output (e.g. the actual salary, $y^{(i)}$), and calculate the difference between the two (the *residual*). We then square the residual to give an extra large penalty to very erroneous predictions and then sum these squared-residuals.

Our *objective* is to minimize the sum of the squares. In mathematical notation, the squared residual of any point is:

$$(h_w(x^{(i)}) - y^{(i)})^2 \tag{3}$$

We are trying to find the line which ends up minimizing the OLS. There are two values that define the line, w_0 and w_1. So what we are trying to do is find the weights such that the OLS is the smallest.

The estimate of the values of the 2 weights (a simple regression model) requires a little calculus. There are errors associated with the estimates, which

tell you how confident you should be in the linear regression. You can find the derivation of the following formulas here. A discussion around the standard error in the weights can be found through this link. In the case of a simple linear regression model, the optimized w_0 and w_1 are calculated as:

$$w_1 = \frac{m\sum x^{(i)}y^{(i)} - \sum x^{(i)}\sum y^{(i)}}{m\sum (x^{(i)})^2 - (\sum x^{(i)})^2} \tag{4}$$

which can be rewritten as:

$$w_1 = \frac{\sum x^{(i)}y^{(i)} - m\bar{x}\bar{y}}{\sum (x^{(i)})^2 - m\bar{x}^2} \tag{5}$$

and

$$w_0 = \frac{1}{m}\sum_{i=1}^{m} y^{(i)} - w_1\frac{1}{m}\sum_{i=1}^{m} x^{(i)} \tag{6}$$

which can be rewritten as:

$$w_0 = \bar{y} - w_1\bar{x} \tag{7}$$

1.6. Exercise 1

Write a PHP script that:

1- reads in the salary_data.csv file. Beware: when you read in the file, don't add a null to the array of rows. In order words, your code should include a check like this:

```
while (!feof($file)) {
     $line = fgetcsv($file);
     if (is_array($line))
         $rows[] = $line;
}
```

2- calculates w_0 and w_1

3- predicts the salary after 7 years of experience.

2. PHP-ML

PHP-ML is a machine learning library for PHP. Documentation is at: https://php-ml.readthedocs.io/en/latest/. To use it:

1. Install with Composer:

```
composer require php-ai/php-ml
```

2. Make sure you require the autoloader

```
require_once __DIR__ . '/vendor/autoload.php';
```

3. Use it, and code away!

```
use Phpml\Regression\LeastSquares;

$samples = [[60], [61], [62], [63], [65]]; //the x(i) data
$targets = [3.1, 3.6, 3.8, 4, 4.1];  //the y(i) data

$regression = new LeastSquares();
$regression->train($samples, $targets);

$regression->predict([64]);
```

2.1. Exercise 2

Read in the salary_data.csv file, train the LeastSquares algorithm, and predict the salary after 7 years of experience. How close if your estimate to the ML library?

2.2. Exercise 3

Read in the housingdata.txt file. This file represents house sale data in Portland. The first column represents the square footage of the house, the second column represents the number of bedrooms, and the third column represents the price. This is a multiple linear regression, since there is more than one explanatory variable. Use the ML library to predict the price of a house with 1650 square feet and 3 bedrooms.

References

https://thenewstack.io/gentle-introduction-machine-learning/

http://caisplusplus.usc.edu/blog/curriculum/lesson2

https://php-ml.readthedocs.io/en/latest/

https://www.kaggle.com/rohankayan/years-of-experience-and-salary-dataset

https://www.kaggle.com/kennethjohn/housingprice

Figure 5. Best fit?

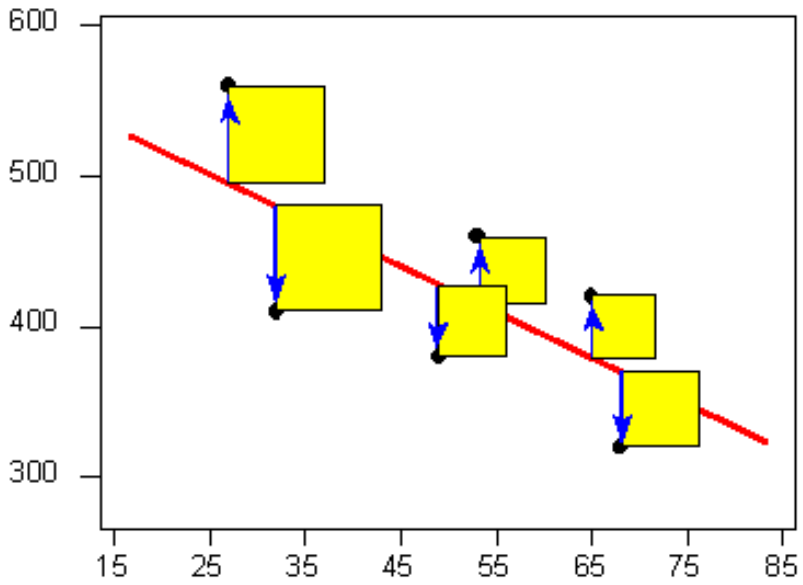

Figure 6. Visualizing Squared Residuals