Linear regression; class 2

15/01/2019

Reminder

* **Supervised learning**
  + **Regression**
  + Classification
* Unsupervised learning
  + Clustering
  + Dimension reduction
* Reinforcement learning

# Introduction to linear regression

Goal: predict output values

Example: training set of housing prices

Predict the final price (**Label**) depending on parameters (**Features**)

Learning the relationship between the input and the output, so between the features and the labels.

x(i) denotes the “input” variable

y(i) denotes the “output” or target that we are trying to predict

A pair ( x(i), y(i) ) is called a training example

A list of m training examples is called a training set

Use training set to learn algorithm to have a **hypothesis (function) h**

H: X → Y

# Simple linear regression

h(x) = ax + b

B is intercept

A is slope

h(x) is the hypothesis parameterized by t = (b, a)

There can be errors: distance between actual price and predicted price

Error e = y(i) - h(x(i))

Y = b + ax + e

We search to minimize the value of the error so the y value is as close as possible to the hypothesis.

Methods to know which model is the one with less errors

* Sum of error on all data points
  + Could have 0 if negative and positive values
  + So not a good method
* Sum abs(error) on all data points
  + > 0
* Sum error^2 on all data points
  + > 0
  + BEST
  + Sum of Squared Errors (SSE) is also called Residual Sum of Squares (RSS)
  + SSE = Sum of ( y(m) - (b + a \* x(m) )^2
    - B + a \* x(m) = h(x(m)) predicted value
    - y(m) actual value
  + Choose the one with the less SSE

## Cost function

The best hypothesis h(x) is the one that minimizes the cost function

J(b, a) = 1 / 2m Sum of i from 1 to m ( h(x(i)) - y(i) ) ^2

Divide by m to obtain the average of the sum

Divide by 2 to help with the derivation

The learning algorithm should find t\* = (b\*, a\*) that minimizes the cost.

Algorithms:

* Gradient descent
* Ordinary least square (OLS)

### Gradient Descent Algorithm

Let’s take a simplified h equation with h(x) = ax

We are calculating the Cost function with different values of a so it is the least possible

We take a fixed value of a and calculate its value

With its derivative we know if the parabola of the cost J is going down or up

So then we calculate a new a depending on the previous value of a, the derivative and a step alpha.

Alpha is called the learning step :

* Too small and it will be too long
* Too big and might miss the minimum, aka the point where the derivative is null

If have h(x) =ax + b

Repeat until convergence and calculate the derivatives for both a and b

Convergence means either

* Cost function is no longer changing by more than a defined value
* The number of iterations is reached

### Ordinary Least Square Algorithm

SSE = Sum ( y(i) - b - a\*x(i) )^2

Minimize the SSE by explicitly taking its derivatives with respect to the a and b and setting them to 0

SSE is not perfect

Larger SSE doesn’t necessarily mean worst fit -> need another metric

## R² Statistic

Answer the question : “ how much of variability in the output (y) is explained by the changed of input (x)”

TSS = Sum ( y(i) - avg(y) )²

R² = ( TSS - SSE ) / TSS = 1 - SEE / TSS

If close to 1 then indicated that a large portion of the variability has been explained by the regression

If close to 0 then indicates that the regression did not explained much of the variability response

# Multiple linear regression

Multiple features X = (x1, x2, … xn)

X(i) is a n dimensional feature vector

X(1) is the feature vector of the first training example

h(x) = bx0 + ax1 + cx2 + … + nxN where x0 = 1

Rewrite in matrix notation

# Evaluation of linear regression model

|  |  |
| --- | --- |
| Gradient descent | OLS (normal equation) |
| Need feature scaling  Normalisation of the values of theta |  |
| O(kn²)  Works well when n is large (10^4 - 10^5) | O(n³) need to calculate inverse of X^TX  Slow if n is very large |

### Polynomial regression also exists

There is no % between R² and the model. R² can be really big, but it doesn’t mean that your model is generalizable.

### Assessing Performance

Use a Test set to test if the R is still valable and if the values make sense

Training error vs model complexity

Bias-Variance tradeoff

* Underfit : High bias model : too simple, too much bias. Model didn’t learn well
* Overfit : High variance model

Looking for the sweet spot using the regularization

Finding the balance between how well the model fits the data and the magnitude of coefficients

Incorporating a penalty weight in the cost function

“Remember th at all models are wrong; the practical question is how wrong do they have to be to not be useful”

George Box, 1987

### Regularization

It’s about finding balance between:

How well the model fits the data

The magnitude of coefficients

This is achieved by incorporating a penalty on weights in the cost function

Ridge Regression

Lasso Regression

# Practical work