IMC-4302C – Statistical learning
1. Linear regression and stochastic gradient descent

Kevin Zagalo
<kevin.zagalo@inria.fr>

2019 – 2020

Use a `Jupyter` notebook and send it to the depository before the next session.
Any notebook sent after that won't be considered. Thank you !

Exercise 1 *Gradient descent*

We want to minimize a function $F : \mathbf{R}^d \to \mathbf{R}$, differential on $\mathbf{R}^d$. Let ∇F be its gradient. The gradient descent corresponds to the iterative algorithm :

— **Initialization** $x_0 \in \mathbf{R}^d$

— **Iteration** $x_{k+1} = x_k - \alpha \nabla F(x_k)$

The iteration is repeated until a stopping criterion is reached.

Question 1

Suggest some criterion to stop the algorithm.

Question 2

Implement this algorithm in Python. We will define a function which takes as input the function F and its gradient ∇F and others.

Question 3

What happens if F is not convex ?

Question 4

Discuss the influence of the x_0 point, in the non convex case, then in the convex case.

■

Exercise 2 *Least squared method*

A linear regression with n inputs $x_1, \dots, x_n \in \mathbf{R}^d$ and an output defined by $d + 1$ constants : the weights parameters $w_1, \dots, w_d$ and a bias b. We define the regression function $h_w(x) = \langle x, w \rangle - b$.

We define the local and global error, respectively :

$$e(x; w) = \frac{1}{2}(y_x - h_w(x))^2 \ ; \ E(x_1, \dots, x_n; w) = \frac{1}{n}\sum_{i=1}^{n} e(x_i; w) \tag{1}$$

To estimate the final weight parameters β, we want to minimize the global error.

We want to implement the algorithm :

```
— Input $X = (x_1, \dots, x_n)$ and the associated responses $(y_1, \dots, y_n)$, and $\epsilon > 0$ a parameter.
— $t = 0$
— $w(t) = \vec{0}$
— Repeat
  — Compute $L(w) = E(x_1, \dots, x_n; w)$
  — Update $w(t+1) = w(t) - \epsilon \nabla L(w(t))$
Until all data is explored and convergence of the weights sequence to $\beta$.
```

Question 1

We treated the bias b as a parameter. Show that we can consider it as a weight parameter. How to adapt the problem of regression ?

Question 2

Write a function taking the weights parameters, an observation and its associated response that updates the weight parameters.

Question 3

Implement the algorithm.

Question 4

Plot the global error over the iterations using `matplotlib`.

Question 5

Modify the algorithm by implementing SGD and plot the global error over the iterations.

Question 6

Compare execution times of GD and SGD with the library `time`.

■

Exercise 3

Download **the *house.csv* file** containing 3 columns that represent the area, the number of rooms and the price of 600 houses (one per row). Comment every result.

Linear regression with 1 feature

In this first part, we will train a linear model for house price prediction using only one feature the house area. We will start by implementing a cost function and the gradient of this cost function. Then, we will implement the gradient descent algorithm that minimizes this cost function and determine the linear model parameter θ in the equation $h_\theta(x) = \theta x$.

Question 1

Open this file with a file editor to understand more the data. Load the data in a `house_data` variable and check its size.

Hint: You could use `loadtxt` function from the `numpy` library.

Question 2

Extract n – the size of the sample. Extract the house area and price columns respectively in X and y arrays to visualize them. Normalize the area feature.

Hint: If you are using Numpy arrays, you need will need a $(600, 1)$-sized array. You can use `reshape` or newaxis.

Question 3

The cost function we will use for this linear model training is the **Mean Squared Error**. Implement the cost function defined by

$$J(\theta) = MSE(\theta; X, y) = \frac{1}{2n}\|X \cdot \theta - y\|_2^2$$

Question 4

Implement the gradient of the Mean Squared Error cost function :

$$\nabla J(\theta) = \partial_\theta MSE(\theta; X, y) = \frac{1}{n}X^\top(X \cdot \theta - y)$$

Question 5

The update equation of the gradient descent algorithm is given by:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla J(\theta^{(t)})$$

Where α represents the step or the **learning rate**. Compute twos or three steps of the gradient descent with different values of α. Choose the best one according to you. Comment what you see.

Question 6

- Implement the gradient descent algorithm, taking as input the gradient function, the learning rate, the stopping criterion and the initial parameter θ_0.
- Compute the optimal parameter θ_{opt}.
- Compare on the same axis the house prices in function of their areas and the prices predicted by the linear model in function of the areas.

Linear regression with 3 features

In this part, we will train a linear model for house price prediction using the house area, number of rooms and the bias term that represents the constant term in the linear model equation

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

Question 7

- Modify X to contain the number of rooms.
- This time don't normalize data. Be careful to what should happen to α, θ and/or your stopping criterion.
- Take the bias term into account.

Question 8

Use θ_{opt} to estimate the prices in the data set. Compare to real prices.

Question 9

You could also try to add other feature columns to the matrix X like $area^2$ or $area^{0.5}\ldots$ and see the effect on the model and the error.

Question 10

How much would cost a $330m^2$ flat with 5 rooms ?

■