BGU-ML Ex1, 2020a

November, 2019

In this exercise there is one open question, and one programming assignment.

A. Naïve Bayes

We learn in class about Naïve Bayes algorithm for classifications of samples with binary traits. Explain what is the reduction in sample's complexity needed in order to train a generative model with and w/o Naïve Bayes assumption (features' independence given the class). Sample complexity should be parameterized by the number of features (data's dimension). No need for formal proof, but a paragraph with intuition.

B. Logistic regression

In this part you will implement a logistic regression classifier with a gradient descent algorithm for parameters optimization. Here are some guidelines, please follow them and add more functions as needed. Try to avoid working with loops, and prefer matrices operation where possible. You are not supposed to use optimization functions, ML functions, or regression functions not implemented by you. Matrices representations and operations can be done with python/R functions/operators. Please make sure you have implemented all 4 functions as separated functions according the their definitions below.

1. Gradient function. Implement a function named 'gradient' that accept a matrix X, a vector y and a vector of parameters θ. The function returns a vector of partial derivatives.

   ```
   gradient(X, y, theta)
   ```

2. Gradient descent. Implement a function named 'gradient_descent' that accepts a matrix X, a vector y, step size alpha=0.1, maximal number of iterations max_iterations=1000, delta threshold=1e-5.

   ```
   gradient_descent(X, y, alpha=0.1, max_iterations=1000,
   threshold=1e-5)
   ```

   The function first adds a dummy variable of '1' as the first column in X to take care of the bias. Then, initialize the parameter θ to vector of zeros of the (new) length of features. In each iteration, update θ by subtracting α times the gradient. If the maximal number of iterations was reached, or the change in θ is settled, stop iterating. Change in θ is measured as $||\theta_{t+1} - \theta_t||_2$ (root square of sum of squares of differences).

The function returns the last θ.

3. Prediction function. Implement a function named 'predict' that accept a matrix X and the parameter θ. The function returns a binary vector of length of number of rows in X. Don't forget to add a dummy variable to X as the first column.
   `predict(X, theta)`
4. Main. Implement a main function, that will be called when file is executed. The function read a comma separated file text file (.csv) that was given as the only argument to the script. The file contains a header line, and the last column represent the class as 0 or 1. After loading the file, the function split it randomly to 80% rows as training, and the remaining 20% for testing.
   Then, the function train a logistic regression using the training set. Then, the function prints to the standard output the accuracy of the model estimated on the testing set. Accuracy is defined as the fraction of mislabeled elements.
   `main()`

You can test your implementation with the given datasets:

- ex1data1.csv
- ex1data2.csv
- ex1data3.csv

Running one of the next two commands (depends if you implemented with R or python):

```
> python ex1b.py ex1data1.csv
> Rscript ex1.r ex1data1.csv
```

Should print the accuracy. Make sure the accuracy is far from random (at least 0.5 for majority of runs) for any of the 3 given datasets.

Remarks:

You need to submit one file named ex1.tar. This is a tar file that contains two files:

`ex1a.pdf` and `ex1b.py` or `ex1b.r`

Where the first file contains your answer to the open question A above, and the second file is your implementation to the question B implemented in Python3 or R.

In the head of each file (`ex1a` and `ex1b`), please note your name and ID number.

The file `ex1.tar` should be submitted via Moodle. You can resubmit as many times as you like up to the deadline, but only the last registered submission will be checked and graded.

You can work in pairs. In such case note the names and ID numbers of both of you.

Don’t copy your answer or your code. Please note and refer to any resource you used (website, blog, book etc.).