5th April 2021

Logistic Regression

Advanced Machine Learning Project no. 1

Olaf Werner, 291139
Mateusz Szysz, 298911

1 Introduction

The goal of this report is to provide the results of experiments concerning different optimization algorithms for logistic regression. We decided to point only the most important conclusions not to make this article too long. All of the remaining experiments are included in the Jupyter notebook file.

2 Implementation details

One of the tasks was to propose and implement the stopping rule for each of three algorithms. We decided that the algorithms should stop when the largest change of coefficients is smaller than given tolerance level which is 0.0001 on default.
Also apart from including Gradient Descent, Stochastic Gradient Descent and Iteratively Weighted Least Squares we added Powell's Optimizing Method which is implemented in SciPy Optimize module.
In all of the experiments parameter estimation was done on the training set and all the metrics were calculated on the test set (for each method we used the same split). Splitting was prepared randomly with the proportion 4 to 1 meaning that 20% of the observations were included in the test set. However all the results are reproducible due to setting seed.

3 Results on the Cancer dataset

The first dataset that we tackled was Cancer dataset in which we predicted if a given patient suffers from cancer. The summary of the obtained results are presented in the table below.

	Accuracy	Precision	Recall	F-measure	R2 Score
SGD	0.982	0.977	0.977	0.977	0.926
GD	0.974	0.977	0.955	0.966	0.889
IWLS	0.93	0.909	0.909	0.909	0.704
Powell	0.93	0.909	0.909	0.909	0.704
Scikit-learn	0.965	0.976	0.932	0.953	0.852
LDA	0.956	1	0.887	0.94	0.815
QDA	0.974	0.936	1	0.967	0.889
KNN	0.982	0.977	0.977	0.977	0.926

The first three rows concern the algorithms implemented on our own. The fourth one is the result of optimization with Powell's algorithm. The fifth one is Scikit-learn logistic regression with default settings. The last three ones are Linear Discriminant Analysis, Quadratic Discriminant Analysis and K Nearest Neighbors.
As can be seen, we got the best results in case of R2 Score for the Logistic Regression with stochastic gradient descent optimization method and for K Nearest Neighbors. The worst ones were obtained with IWLS and Powell's optimization method. Probably it's due to the fact that stopping rules were chosen to obtain convergence for Gradient Descent and Stochastic Gradient Descent. IWLS and Powell's algorithms found the minimum faster so they adjusted to the training set too much and the problem of overfitting occured. The same conclusions can be drawn from analysing the chart below - the IWLS algorithm found local minima much faster.

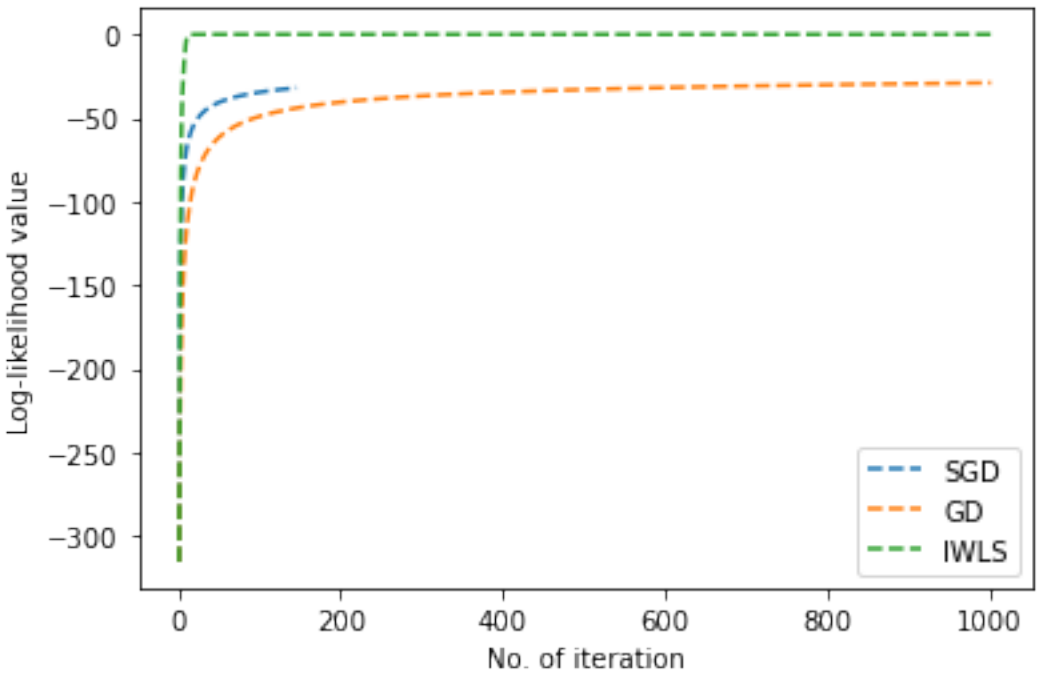

Fig. 1. Log-likelihood value depending on the number of iterations.

This chart shows also that the SGD algorithm converges faster than the GD algorithm. On the x-axis there is number of iterations, but to be precise we meant number of epochs. If the size of mini batch is much smaller than the length of the dataset but

still sufficiently large to possess 'stochastic properties' the SGD algorithm may be much faster than the GD one. If we plot the relation between the number of mini batches processed and the log-likelihood function, we would see than the trajectory of SGD would be very similar to the trajectory of GD but much more irregular.
Furthermore, we analysed the impact of setting different alpha coefficient values for Gradient Descent and and Stochastic Gradient Descent. The summary of the results is shown in the charts below.

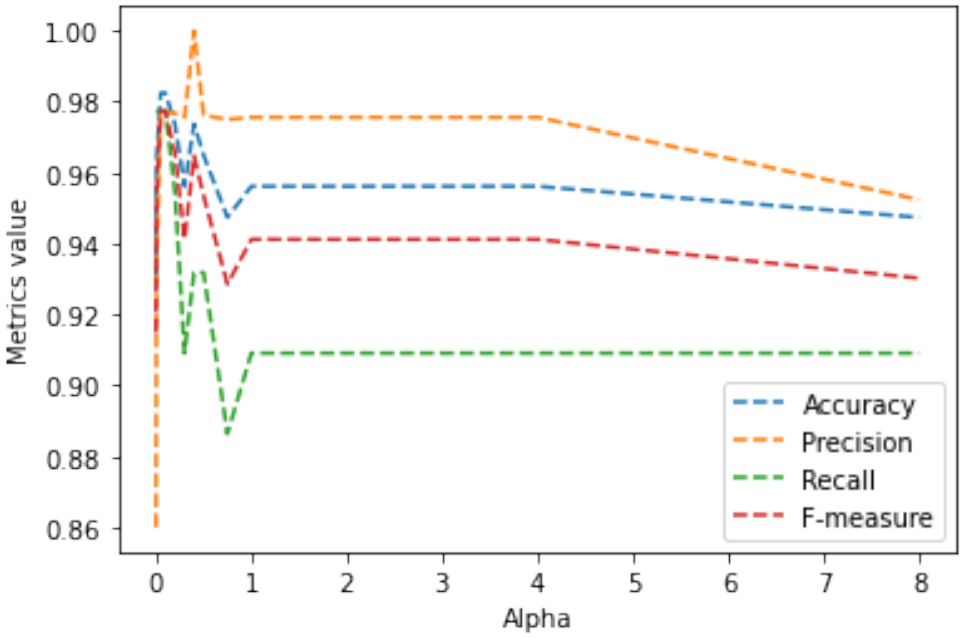

Fig. 2. Metrics values for SGD algorithm depending on value of the learning rate.

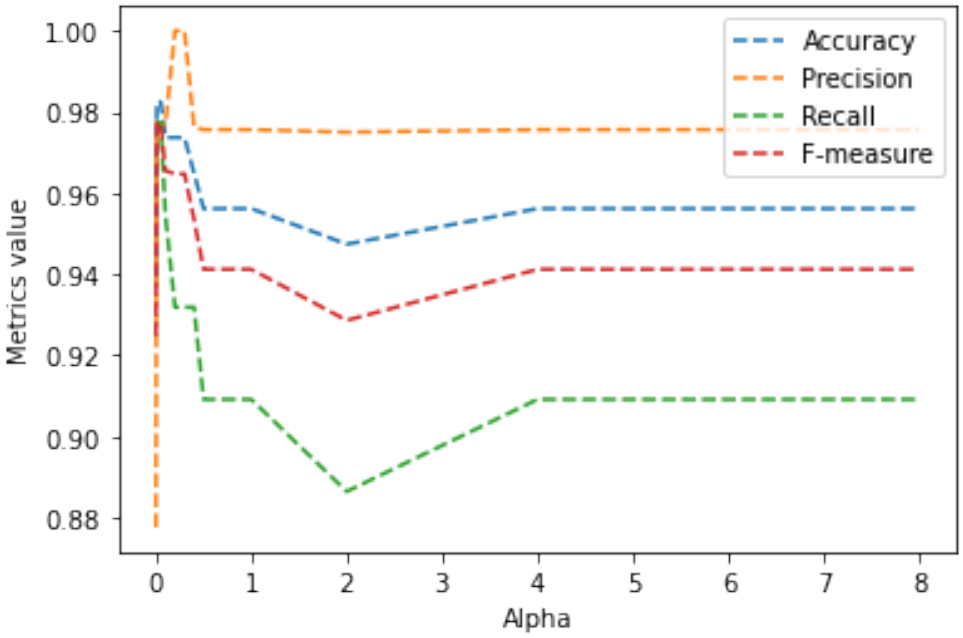

Fig. 3. Metrics values for GD algorithm depending on value of the learning rate.

As can be seen, in general very small values of the learning rate doesn't guarantee convergence in small number of iterations. On the other hand metrics values decrease if the step size is too large. It's due to the fact that the algorithm may 'jump over' the real solution and not be able to go into the minimum.

To sum it up, there are many classifiers and each of them may have different performance on the same dataset. What is more, even the same method may be very sensitive to using different optimization techniques (GD/SGD or IWLS/Powell), but also the same optimization algorithm may give different results for different parameters (different learning rates, sizes of mini-batches).

4 Final remarks

As we mentioned previously, we did not put all the experiments results in the report. We decided to focus on one of the five datasets, provide outcomes and brief interpretations. The results of using our implementations on each of four remaining data sets is printed in the Jupyter Notebook file.
We used other dataset that have attributes highly correlated with the dependent variable, but also there are some examples of datasets with attributes that has small or even none predictive power. However in all of them, the general conclusions are similar to ones that we presented in this report.

5 Bibliography

- lecture materials
- www.kaggle.com/rounakbanik/pokemon (Pokemon dataset)
- www.openml.org/d/37 (diabetes dataset)
- archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (cancer dataset)
- www.kaggle.com/brendan45774/test-file (Titanic dataset)
- archive.ics.uci.edu/ml/datasets/Bank+Marketing (Bank dataset)
- retostauffer.github.io/Rfoehnix/articles/logisticregression.html (IWLS algorithm)
- scikit-learn.org/stable/ (other classifiers and data preprocessing)
- machinelearningmastery.com/logistic-regression-with-maximum-likelihood-estimation/ (maximum likelihood concept overview)