Advanced Machine Learning

Project 1

Antoni Zajko, 313553,
Dawid Płudowski, 313452
Paweł Gelar, 313343

1 Introduction

The aim of this report is to introduce our results of the first project on the *Advanced Machine Learning* course. In our work, we implement three types of optimizers for logistic regression model. We compare their performance on 9 different datasets and compare it to baselines. In addition we analyze convergence of the implemented algorithms. Finally, we provide insight of how introducing interactions to the model can affect its performance.

2 Methodology

For the source of the datasets for our experiments, we choose OpenML[1]. In Table 1, we present the statistics describing chosen datasets, including their names and OpenML ids. We also provide labels describing whether the data was used by us in previous courses, however, this should be treated as suggestion only. As OpenML API allows to browse the data with respect to percentage of the missing data, we select only those datasets that have no missing values. We restrict our research to binary tasks so we do not perform any labels casting. Considering obligation to remove co-linearity from our data, we use VIF-based algorithm to remove attributes that can be explained by other attributes in iterative manner. We arbitrary choose $VIF = 5.0$ as the threshold to stop the algorithm. The schema of the algorithm is presented in Listing 1.

ID	**name**	**# Attr.**	**# Row**	**# Num. Attr.**	**# Cat. Attr.**	**type**	**Is new?**
37	diabetes	9	768	8	1	Small	no
719	veteran	8	137	3	5	Small	yes
333	monks-problems-1	7	556	0	7	Small	yes
3	kr-vs-kp	37	3196	0	37	Big	no
31	credit-g	21	1000	7	14	Big	no
40	sonar	61	208	60	1	Big	yes
1009	white-clover	32	63	27	5	Big	yes
53	heart-statlog	14	270	13	1	Big	no
59	ionosphere	35	351	34	1	Big	yes

Table 1: Description of used data. ID value refers to id of the data on OpenML platform.

For a stopping rule, we use a simple method based on a moving window. Precisely, we retain the value of the best achieved log-likelihood in the object attribute. Then in each epoch, we check if the value of log-likelihood improves by 10% in the last 30 iterations. If not, the stopping rule is triggered. The tolerance

[1] https://www.openml.org/

Algorithm 1: VIF-based co-linearity reduction.

Input: Dataset $\mathcal{D}$, threshold m

Output: Dataset without co-linearity $\mathcal{D}'$

while *There is no column with* $VIF > m$ **do**
  for $col \in \mathcal{D}$ **do**
    calculate VIF_{col}
  If there is any column with $VIF_{col} > m$, drop column with maximum VIF

parameter	**value**
number of CV folds	5
VIF threshold	5.0
Beta_1 (ADAM)	0.9
Beta_2 (ADAM)	0.99
number of batches (SGD & ADAM)	1
learning rate (SGD & ADAM)	0.1
stopping rule window	30
stopping rule minimal improvement	0.1

Table 2: Parameters of the experiments.

factor and patience were chosen in a trial-and-error manner which is not shown in the report. At the end of the training the weights corresponding to the highest log-likelihood are restored.

As the metric, we use balanced accuracy and all scores and plots in this report refers to it if not stated otherwise. All experiments were performed multiple times in cross-validation manner. For a performance reasons, we use only 5 splits but the code is prepared to easily change this value.

To summarize all values that can treated as a configuration of the performed experiments we present Table 2. We would like to emphasize that all values from this Table is prepared in a manner that allows their change only in one file and are not implementation-dependent.

3 Convergence analysis

In this section we present comparison of convergence of specific optimization algorithms. The results are presented on Figure 1. IWLS in all considered datasets converged to highest value of log-likelihood. The second best algorithm is ADAM which in six out of nine cases was better than SGD. The last algorithm in terms of opitimization result is SGD which managed to be better than ADAM in three datasests and four times managed to yield results similar to ones produced by IWLS.

IWLS also has the fastest convergence, i.e. in majority of cases stopping rule finished optimization after lowest number of iterations. Second fastest algorithm was SGD which in 6 out of 9 datasests was better than ADAM which out of implemented algorithms had lowest convergence.

IWLS had lowest variance of performance across folds which means that its results are the least influenced by the starting point of optimization and sampling of data. Two remaining algorithms had simmilar variance; however, in case of SGD there were some datasets where its variance was orders of magnitude higher than other approaches.

IWLS and ADAM were stable algorithms, i.e. the growth of log likelihood was in majority of cases monotonous. Different situation was in case of SGD where in some datasets optimization objective could decrease which can mean that this algorithm was converging in suboptimal directions.

Taking all into consideration IWLS is definetly the best algorithm in terms of all considered apects. It produces best results and has fastest convergence. Also it is robust to changes in starting point and sampling of data and finally it monotonically increases its optimization objective.

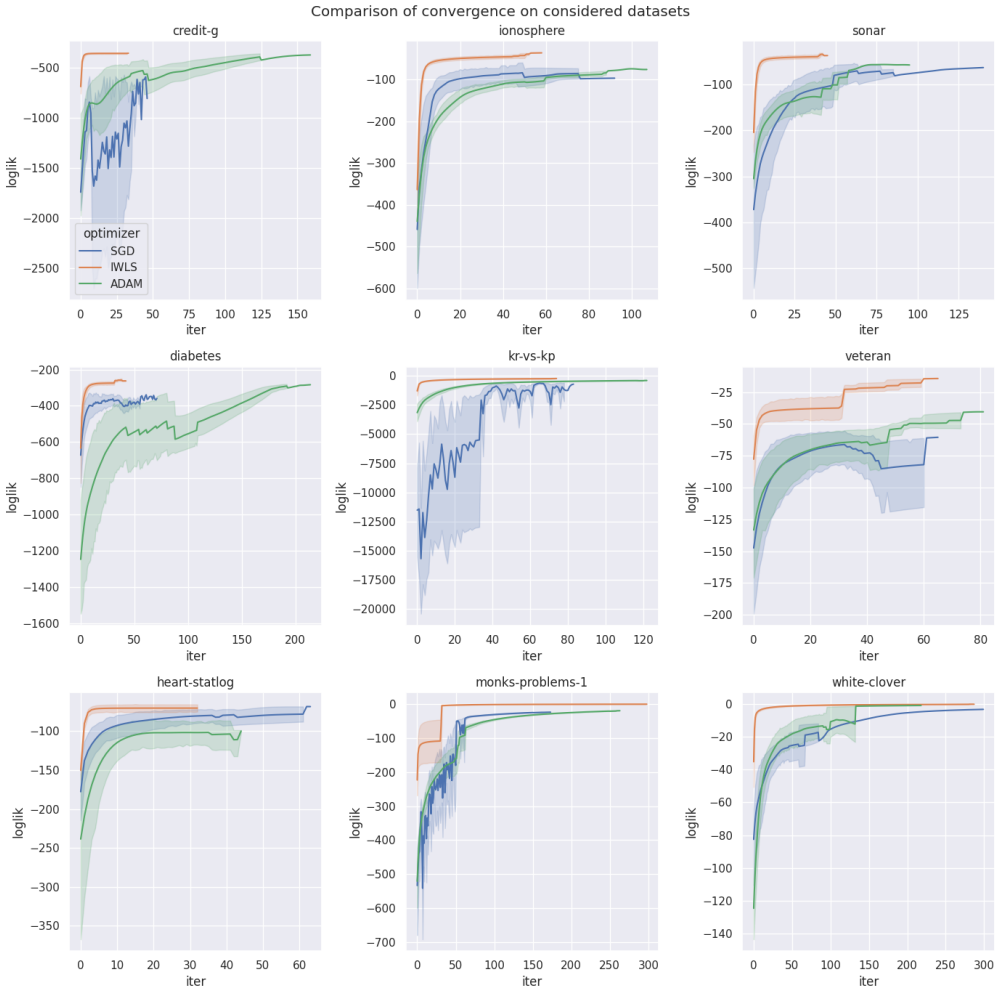

Figure 1: Trajectories of log-likelihood of implemented algoriths. Confidence intervals were obtained by 5-fold cross-validation. IWLS has the fastest convergence and yields best results.

4 Comparison of classifiers

In this section, we would like to introduce the results of the performed experiments. We also present baselines which is listed in Table 3. As a presentation of the results, we propose violin plot of ranks and critical distance (CD) plots. Both plots are robust to the difference in achievable metrics for different datasets and thus, allow to present the scores without the need to standardize values of balanced accuracy across tasks.

First, we present violin plots with respect to achieved ranks (Figure 2.). The ranks are calculated with respect to a single CV-split on a single dataset. Because we use 5 splits and 8 datasets, each violin is created

based on 40 observations. One can observe that IWLS optimizer is the best among all introduced optimizers (orange color in the plot). However, the best method is the random forest. It is reasonable results, as the random forest is the model which has non-linear decision boundary. This fact makes it capable to adjust its predictions to more complex tasks.

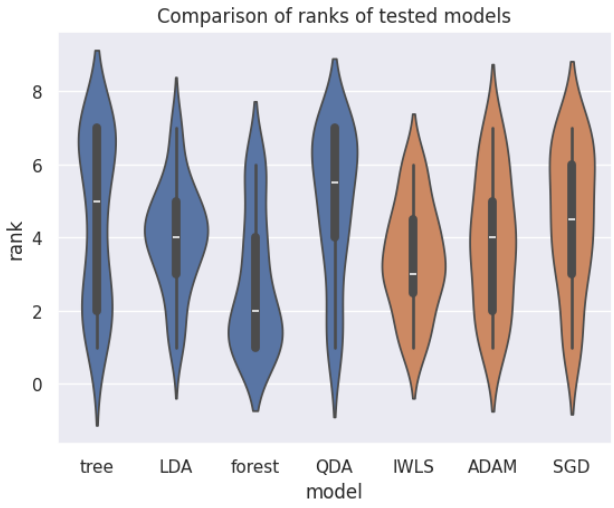

Figure 2: Violin plots. Our implementation is colored in orange, while baselines is colored blue.

Next, we present the CD plot (Figure 3.). Here, we analyse whether the difference in mean value of ranks that the models achieved is significant in statistical sense. The position on the line denotes the average rank that the model achieve while the horizontal line connecting the models denotes that the difference between them is not significant. As present on the Figure, the is no single model that can be statistically distinguish from all the other.

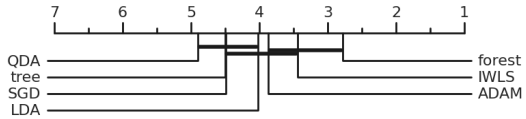

Figure 3: CD plots. Models connected by horizontal line is statistically indistinguishable in terms of balanced accuracy.

5 Analysis of interactions influence

In this section we present comparison of influence of adding two-feature interactions. The results are presented on Figure 4.

Model	LDA	QDA	Tree	Forest
Type	linear	non-linear	non-linear	non-linear

Table 3: Baselines used throughout the experiments.

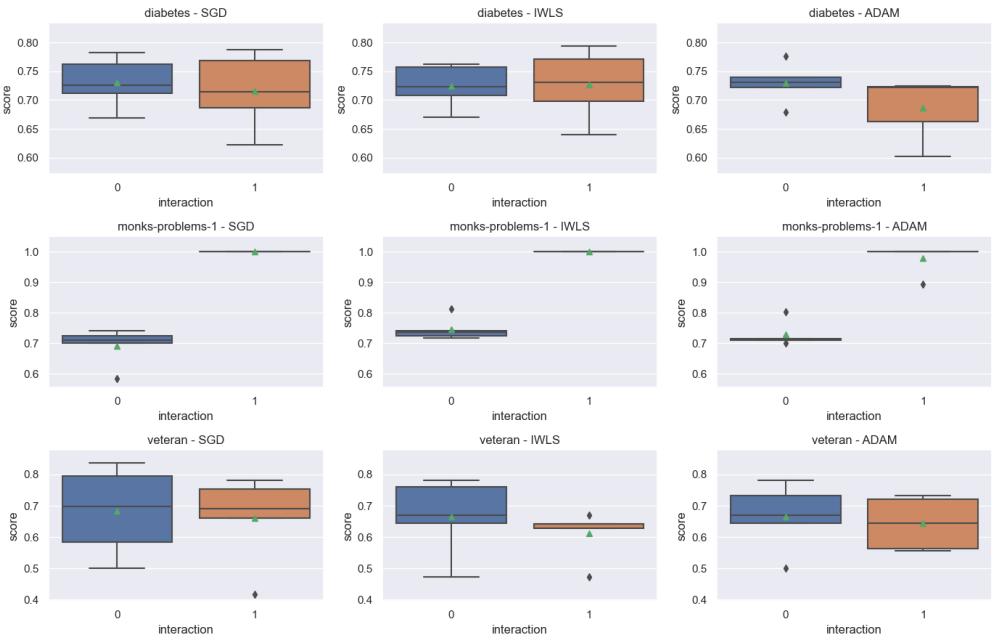

Figure 4: Boxplots showing comparison of models trained with and without interactions. The score on y axis is balanced accuracy. Mean scores are marked by green triangles.

The *monks-problem-1* dataset stands out, as the interactions vastly improve the score. After a closer inspection, it turns out that this dataset is artificial, and the labels are deterministically defined as follows

$$y := (a_1 = a_2) \vee (a_5 = 1)$$

so the classes are not linearly separable, but become linearly separable upon adding $a_1 * a_2$ variable. Because of that the models with interactions can achieve perfect accuracy.

In *diabetes* and *veteran* datasets, the models with interactions perform on average slightly worse, than those without. Because the number of samples in those datasets was quite low ($<$1000), the added variables made the model overfit (the tested models were not regularized).

Looking at the results we have no basis to assume that the optimization method changes how the interactions influence the final score.

In conclusion, on natural datasets, the model with interactions performed on average slightly worse than the one without. Nevertheless, we think that it is worth comparing the results for these two variants on a validation dataset, when using a logistic regression model, as it is quick to train. Regularization could also improve performance when using interactions.

6 Summary

During our experiments we analyse different types of optimizers. The state-of-the-art optimizer, IWLS, occurs to be better than the gradient based methods. Considering the baselines, IWLS is only worse than random forest but this is easily explained by the fact that logistic regression models has less capability than the tree-based models when it comes to interaction in data.

It is also worth to mention that ADAM-based gradient optimizer achieved slightly better results than the SGD optimizer which is also correct according to the theory. However, one need to remember that all

optimizer has the same ability to find the global minima for the logistic regression and the main difference is in the time of the optimization, not the final results.