Optimization algorithms for logistic regression
Advanced Machine Learning - Project 1

Jakub Piwko, 313451
Kacper Skonieczka, 313505
Grzegorz Zakrzewski, 313555

02.04.2024

Contents

1 Methodology

1.1 Data

Data used in this project will be presented in a form of list below. For every set we provide size after transformations in brackets, short description, information about target variable and source which is available after clicking the name of the dataset.

1. **Prima Indians Diabetes** (768 x 9) - The objective of the dataset is to predict whether or not a patient has diabetes, having certain diagnostic measurements. Target variable: *Outcome*.
2. **Banknote Authentication** (1372 x 5) - Data contains features statistically extracted from images that were taken from genuine and forged banknote-like specimens. Target variable: *Class*.
3. **Abalone** (4177 x 6) - set containing physical characteristics of sea shell creatures. Target variable: *Rings*.
4. **Online News Popularity** (39644 $\times$ 56) - set predicts level of news popularity based on statistics about content and activity related to online articles. Target variable: *Shares*
5. **Credit Card Fraud Detection** (284807 $\times$ 31) - original data set contains information about credit card transactions, but due to confidentiality, the set was transformed by PCA and delivered in numerical feature form. Target variable: *class*
6. **Malware Detection** (100000 $\times$ 16) - data set contains information about processes and its aim is to detect malware activity. Target variable: *Classification*.
7. **Students' dropout and academic success** (4424 $\times$ 32) - objective of data set is to predict if student will dropout of studies based on some personal information and academic performance. Target variable: *Target*
8. **Drugs Consumption** (1885 $\times$ 13) - data set contains personal information and measures concerning substances abusing. We have to predict if person is addicted. Target variable: *amphet*.
9. **Higgs Bosson Detection** (98049 $\times$ 29) - data set concerns detection of higgs bosson particle signal based on simulated physical properties. Target variable: *class*

For some data sets we needed to perform small preprocessing. We deleted columns that where highly correlated. We decided to delete features if the correlation was higher than 0.8, and if we had enough columns left. In some cases we needed to delete features that had single value or were a form of record identification. Not every set was originally dedicated to binary classification task. In such cases we coded target variable to present binary values based on information carried by this column. Transformation details are included in dedicated file.

1.2 Experiments

Our `LogisticRegression` class uses a stopping rule with three key parameters:

- `iterations` - caps the number of optimization iteration to set value.
- `min_delta` - minimum required change in cost function to consider progress.
- `patience` - defines the number of iterations to wait for an improvement in the cost function that meets or exceeds the `min_delta`. If such an improvement is not observed for set `patience` value consecutive iterations, the algorithm triggers early stopping.

For evaluation, balanced accuracy was the chosen metric. Model performance was assessed on the test set and averaged over five train-test splits selected randomly. If an algorithm failed to converge within 500 iterations, the last iteration's solutions were used.

Basic evaluation of experiment results is presented in Figure 1. The bar plots show the average value of balanced accuracy for all three optimization algorithms across all datasets. Although the conclusions will be described more specifically in the following sections, there are some observations visible at first glance. In general, IWLS, SGD, and ADAM algorithms performed similarly. However, the 5th and 7th datasets were exceptions, where the ADAM optimization algorithm performed much better than the other two algorithms. In 8 out of 9 cases, the IWLS algorithm achieved the worst or nearly the worst results.

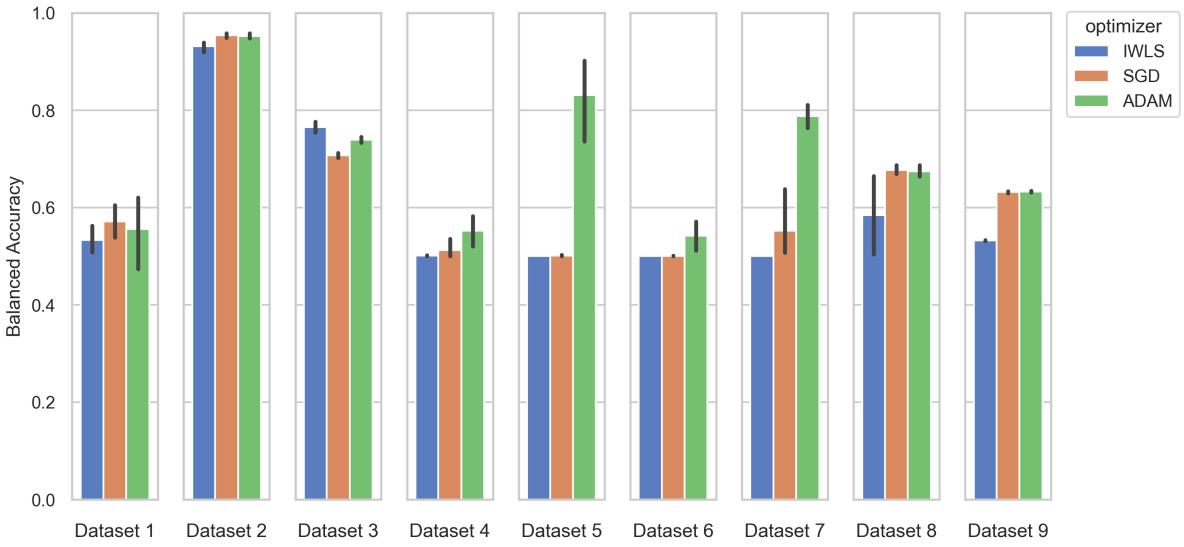

Figure 1: Bar plots presenting the average value of balanced accuracy for different optimization algorithms across nine datasets.

2 Convergence Analysis

Before we delve into convergence analysis, there are two noteworthy points to mention. Firstly, according to the instructions, the value of the log-likelihood function on the training data was measured for every iteration of the learning process, with a maximum of 500 iterations. However, some optimization algorithms managed to converge within a smaller number of iterations, and the value of the log-likelihood function was not calculated for the remaining iterations up to the limit. Secondly, the behaviour of a specific optimization algorithm on a selected dataset is difficult to predict and mostly depends on the values of hyper-parameters. It's unlikely that the same set of hyper-parameters will work well on completely different datasets. The default values of parameters were selected once and did not change during the experiments.

The number of iterations needed for each optimization algorithm to converge is presented in Table 1. The number of iterations is averaged across five splits. Several conclusions can be drawn from these numbers. Firstly, it is evident that the proposed stopping rule was effective a significant number of times. The stopping rule was triggered five times for the ADAM optimizer and four times for IWLS and SGD. Interestingly, the stopping rule wasn't triggered at all for datasets 4, 6, and 7, but for datasets 2, 3, and 8 the learning process stopped before reaching the limit of iterations in every case. This could indicate that some datasets are simpler than others, or perhaps the conditions of the proposed stopping rule should be adjusted. When the stopping rule was triggered for the IWLS algorithm, it was also triggered for the SGD algorithm. However, the behaviour of the ADAM optimizer wasn't as consistent. Notably, the IWLS optimization algorithm stopped after very few iterations - less than 30 - for datasets 3 and 5.

	dataset								
optimizer	1	2	3	4	5	6	7	8	9
ADAM	415	191	193	500	500	500	500	320	415
IWLS	500	321	27	500	28	500	500	406	500
SGD	500	415	141	500	397	500	500	63	500

Table 1: The average number of iterations needed for each optimization algorithm to converge on a specific dataset, with a limit of 500 iterations.

The main insight into how the value of the log-likelihood function depends on the number of iterations is presented in the line plots in Figure 2. Three different situations can be distinguished from these plots.

In several cases, the values of the log-likelihood function change in the most expected way, that is, they monotonously decrease towards zero. This can be observed for the ADAM optimization algorithm for dataset no. 1, and for SGD and ADAM optimizers for dataset no. 3.

However, a more frequent situation is when the value of the log-likelihood function rapidly decreases within a small number of initial iterations, and then remains almost constant. Examples can be seen for datasets number 2, 5, and 9. These are problematic cases because we would prefer that the stopping rule interrupt the learning process. Since the results cannot improve further, we would like to end the computation earlier. These examples show that there is room for improvement in the proposed stopping rule.

Lastly, the worst situation occurs when the values of the log-likelihood function do not behave in a stable manner from the very first iteration, but rather jump (with high variance) around some value. This is a clear sign that the optimization algorithm struggles with the specific problem and cannot progress towards the solution. Improper values of hyper-parameters may be responsible for these cases.

To sum up, the convergence of the algorithm strongly depends on the problem, the specification of the stopping rule, the hyper-parameters, and the optimizer itself. While the algorithms successfully converged for some datasets, they did not for others. Adjustments to the stopping rule and the hyper-parameters are necessary for each specific problem to unlock the full potential of the optimization algorithm.

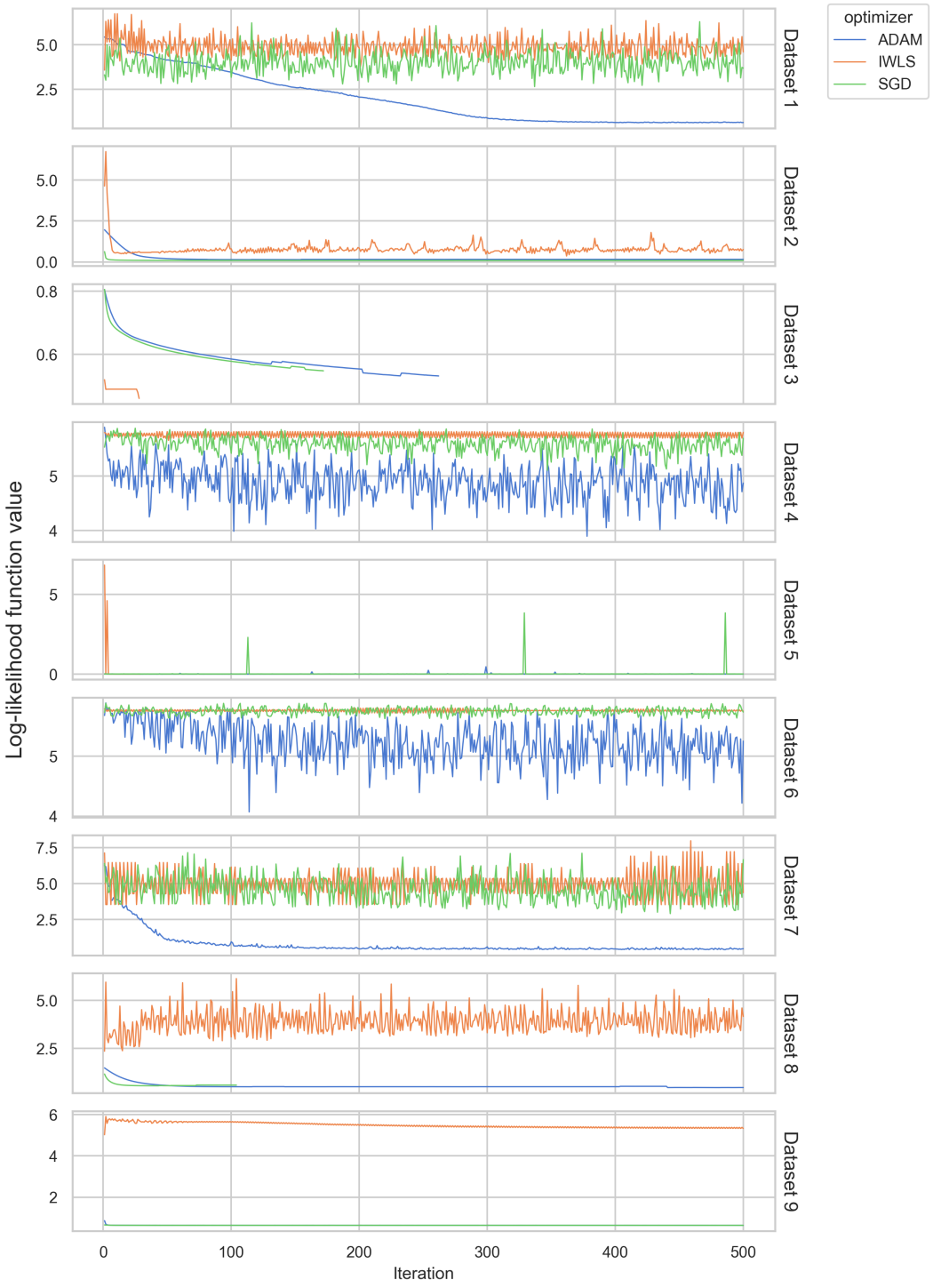

Figure 2: Line plots showing the value of the log-likelihood function for every epoch in comparison among three implemented optimization algorithms and all datasets.

3 Comparison of classification performance

In the next part of the conducted experiments, the performance of logistic regression with IWLS, SGD, and ADAM optimization algorithms was compared to the performance of other classification methods: LDA, QDA, Decision Tree, and Random Forest.

For a fair comparison, other classification methods were trained and tested within the same train/test splits as the IWLS, SGD, and ADAM optimizers. Implementations from the scikit-learn library were used. Default parameters were used for the LDA and QDA models. In the case of Decision Tree and Random Forest algorithms, the hyper-parameters were slightly modified to prevent expected overfitting. The values of the modified hyper-parameters are presented in Table 2. The hyper-parameters and the stopping rule of the IWLS, SGD, and ADAM optimization algorithms remained unchanged.

Decision Tree	Random Forest
`max_depth=16`	`max_depth=8`
	`n_estimators=50`
`min_samples_split=8`	
`min_samples_leaf=8`	
`max_features="log2"`	

Table 2: The values of the modified hyper-parameters for Decision Tree and Random Forest models.

For precise information on which classification method performed best across all the datasets, one should refer to Table 3. This table indicates whether a specific classification method achieved the 1st, 2nd, ..., or 7th ranking position in terms of balanced accuracy. For example, in all but one case, the IWLS optimization algorithm had the worst balanced accuracy metric values. In contrary, for every dataset, the Random Forest models were among the top three classifiers. The last row in Table 3 also shows the average ranking position for each classification method. It suggests that IWLS, SGD, and ADAM optimizers performed the worst, LDA, QDA, and Decision Tree models were in the middle, and the Random Forest models were clearly the best.

	classification method						
dataset	IWLS	SGD	ADAM	LDA	QDA	DecisionTree	RandomForest
1	7	5	6	1	2	4	3
2	7	5	6	4	2	3	1
3	2	7	5	4	6	3	1
4	7	6	4	2	5	3	1
5	7	6	5	2	1	4	3
6	7	6	5	3	4	1	2
7	7	6	4	3	2	5	1
8	7	3	4	5	1	6	2
9	7	5	3	4	6	2	1
Avg. position	6.4	5.4	4.7	3.1	3.2	3.4	1.7

Table 3: The ranking of the classification methods in terms of the balanced accuracy for each dataset, as well as the average ranking position.

One may want to examine the magnitude of differences in the performance of classification methods. Figure 3 presents nine bar plots displaying the average balanced accuracy (and confidence intervals) for each classifier. Drawing specific conclusions regarding all datasets and algorithms may be challenging, but some general patterns emerge. Typically, IWLS, SGD, and ADAM optimizers perform significantly worse than the other methods. Sometimes, this rule applies to all three optimization algorithms, as in the case of datasets no. 1 and 6, sometimes to IWLS and SGD (dataset no. 7), and sometimes only to the IWLS optimizer (dataset no. 8). The differences are not as significant in the case of datasets no. 2 and 3. Random Forest is clearly the best model in terms of balanced accuracy, but its performance is not significantly better than the others.

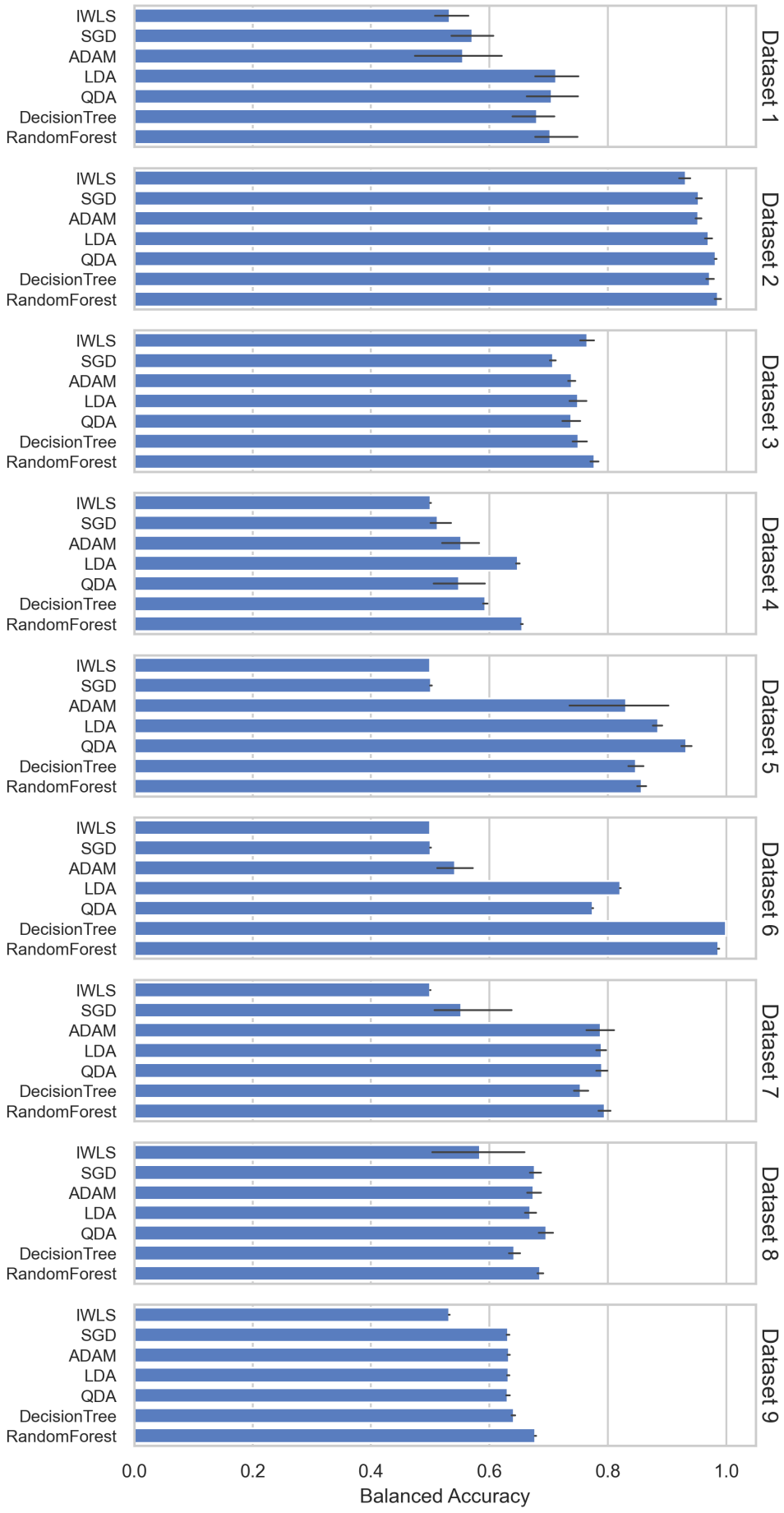

Figure 3: Average balanced accuracy score when comparing different classification methods and datasets.

4 Comparison of classification performance with and without interactions

The three small datasets were utilized to compare two versions of logistic regression: models without interactions and models with interactions. Experiments were conducted with the same hyper-parameters, stopping rule, and train/test splits as in the previous sections.

Little or no meaningful insights can be derived from the results of the last set of experiments. The average balanced accuracy scores are presented in Table 4. For all datasets, for SGD and ADAM optimization algorithms, the results achieved by models with interactions were slightly better than those without interactions, but the differences are not larger than 0.04. In the case of the IWLS optimizer, this trend only applies to dataset no. 2, while the balanced accuracy score is higher for the IWLS algorithm without interactions for datasets no. 1 and 3.

	variant of logistic regression					
dataset	IWLS	SGD	ADAM	IWLS+INT	SGD+INT	ADAM+INT
1	0.533	0.571	0.556	0.529	0.573	0.581
2	0.931	0.953	0.952	0.982	0.990	0.989
3	0.765	0.708	0.739	0.739	0.745	0.744

Table 4: Average balanced accuracy scores computed for the small datasets and all variants of logistic regression.

No additional conclusions can be drawn from the bar plots shown in Figure 4. It is worth noting that in the case of variants with interactions, the performance of SGD and ADAM optimizers is slightly better than that of the IWLS optimization algorithm, but the gains are negligible. A similar trend was observed in previous sections.

We must consider the possibility that the lack of significant differences between variants with and without interactions in the obtained results may be due to the characteristics of the specific datasets. It's likely that in all three small datasets, the target variable does not depend on the variables created by multiplying the existing ones. To make a fair comparison of classification performance with and without interactions, it would be preferable to use artificially created datasets.

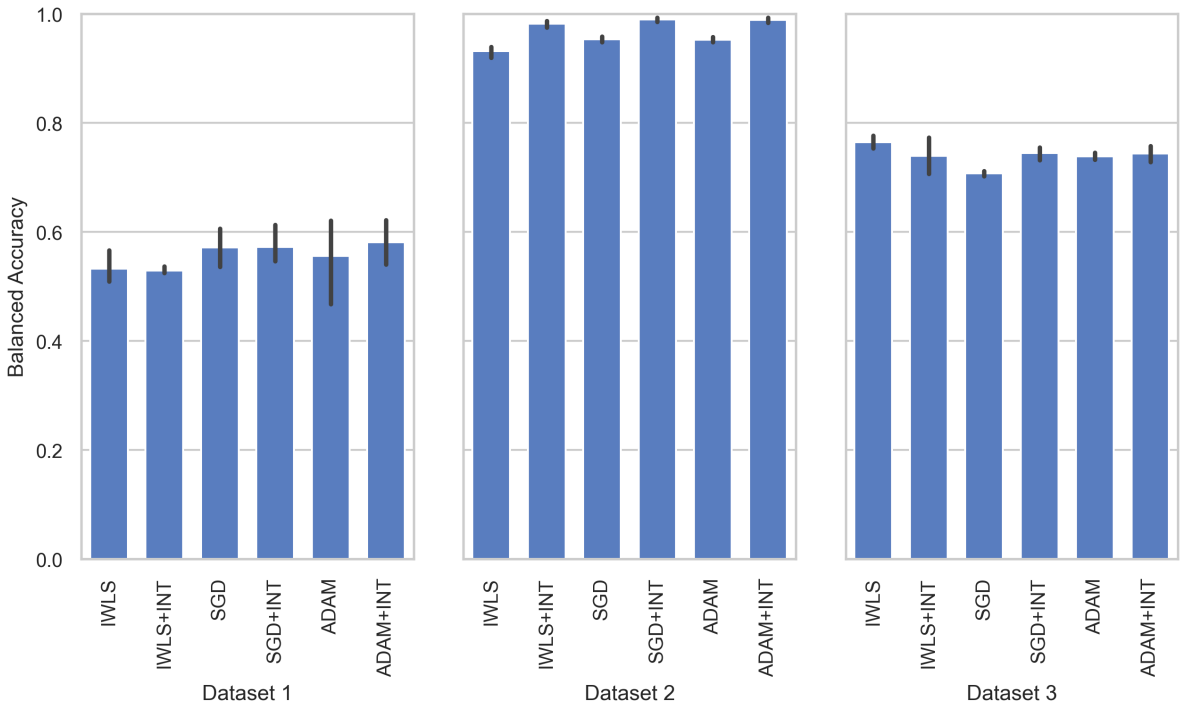

Figure 4: Bar plots presenting the average value of balanced accuracy for different optimization algorithms with or without interactions across three small datasets.