Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence

ELIANA PASTOR, Politecnico di Torino, Italy
LUCA DE ALFARO, UCSC, USA
ELENA BARALIS, Politecnico di Torino, Italy

1 SOURCE CODE

Repository. All the materials to reproduce the results (source code, datasets and experiments) are available in our repository: https://github.com/elianap/divexplorer_SIGMOD21_experiments.

The source code of the DivExplorer tool is also available as a python package in the PyPI repository [4]. Additional information is available in the DivExplorer project page [3].

Programming Language. DivExplorer has been developed in Python.

Packages/Libraries Needed. ipywidgets>=7.2.1, matplotlib>=3.1.1, numpy>=1.16.4, mlxtend>=0.17.1, pandas>=0.24.2, plotly>=4.5.0, python_igraph>=0.8.3, scikit_learn>=0.23.2, psutil, requests

2 DATASET

In the experiments, we used the following datasets: the COMPAS dataset [1] and the Adult, German Credit Data, Bank Marketing, and heart datasets from the UCI repository [2]. We also used an artificial dataset. The source code for its generation is available in the script *E03_artificial.py*.

All the datasets are already available in the *datasets* folder. The reference links are also available in *datasets/datasets.txt*. The applied processing is available in the script *import_dataset.py*.

3 HARDWARE

The experiments were performed on a PC with Ubuntu 16.04.1 LTS 64 bit, 16 GB RAM, 2.40GHz×4 Intel Core i7, SSD storage.

4 EXPERIMENTS

4.1 Setting the environments

DivExplorer is implemented in Python. The experiments were performed with Python 3.6.10.

To set up the environment, the user can follow the instruction in `prepare_conda.txt`. The instructions (i) create the virtual environment and (ii) install the required libraries using conda and pip. The file *requirements.txt* contains the list of the required packages.

4.2 Reproduce paper results

Run all experiments. The user can reproduce all the figures and the tables in the paper by running the python script *run_experiments.py*. Once the environment is activated, we can run it with

```
python run_experiments.py
```

By default, the results are stored in the *output* folder in the current directory. Specifically, the figures (in pdf format) and the table (in csv format) will be available in the *figures* and *tables* subdirectories respectively.

Run experiments of interest. To ease the script understanding, we also categorize the scripts to reproduce the results based on the target of the experiment. As a result, rather than running all the experiments, the user can run specific experiments of interest. We can run each experiment with

```
python experiment-name.py
```

The scripts are categorized as follows.

Script	Description	Ex. time
`run_experiments.py`	Runs all experiments, generate all figures and tables in the paper	≈ 7m30s
`E01_compas.py`	Runs COMPAS experiments	≈ 4s
`E02_adult.py`	Runs adult experiments	≈ 10s
`E03_artificial.py`	Runs experiments with artificial dataset	≈ 10s
`E04_redundancy.py`	Runs experiments with redundancy pruning	≈ 1m
`E05a_compute_performance.py`	Computes performance results	≈ 6m
`E05b_plot_performance.py`	Visualizes performance results (it requires to run `E05a` first)	≈ 2s

Table 1. Summary of experiments with execution time.

COMPAS experiments. The script `E01_compas.py` runs all the experiments related with the COMPAS dataset. It generates the following tables and figures of the paper: "table_1", "table_2", "table_3", "figure_1", "figure_2", "figure_3", "figure_5".

Adult experiments. The script `E02_adult.py` runs all the experiments related with the adult dataset. It generates the following results: "table_5", "table_6", "figure_8", "figure_9", "figure_11".

Our approach reveals peculiar behaviors of a classification model. In the experiments, we used the labels of a random forest classifier with cross-validation. This process may be affected by small differences in the final labeling due to its randomicity. To avoid differences due to model training, we provide in *./datasets/processed* the pre-processed labeled dataset to reproduce results. To also re-train a random forest model and re-generating the labeled data (rather than the precomputed ones) by specifying `-retrain` as a parameter of the script.

Artificial data experiments. The script `E03_artificial.py` generates the artificial dataset and runs the divergence analysis. It generates "figure_4".

Redundancy pruning results. The script `E04_redundancy.py` runs the redundancy pruning experiments for the COMPAS and adult datasets. It generates "figure_10".

Performance results. The script `E05a_compute_performance.py` runs the divergent pattern extraction for all the datasets under analysis varying the minimum support threshold. The results in terms of execution time and the number of extracted patterns will be stored (by default) in the *performance_results* folder. The script `E05b_plot_performance.py` uses the output of the `E05a` script and generates "figure_6" and "figure_7".

5 ADDITIONAL MATERIAL

Notebooks. We also provide two python notebooks to run the experiments on the COMPAS and Adult dataset, respectively named *NB1_compas.ipynb* and *NB2_adult.ipynb*. The user can run the notebook and interactively inspect peculiar behavior in subgroups.

Python package. The source code of DivExplorer is available as a python package [4]. The documentation offers examples of usage.

REFERENCES

[1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[2] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[3] DivExplorer Project Page. 2021. https://divexplorer.github.io

[4] DivExplorer pip package. 2021. https://pypi.org/project/divexplorer/ Repo: https://github.com/elianap/divexplorer.