Testing Balancedness of ML Algorithms

Arnab Sharma[1], Heike Wehrheim[2]

Abstract: With the increased application of machine learning (ML) algorithms to decision-making processes, the question of *fairness* of such algorithms came into the focus. Fairness testing aims at checking whether a classifier as "learned" by an ML algorithm on some training data is biased in the sense of discriminating against some of the attributes (e.g. gender or age). Fairness testing thus targets the *prediction* phase in ML, not the learning phase.

In our approach, we investigate fairness for the *learning* phase. Our definition of fairness is based on the idea that the learner should treat all data in the training set equally, disregarding issues like names or orderings of features or orderings of data instances. We term this property *balanced data usage*. We have developed a (metamorphic) testing approach called TiLe for checking balanced data usage and report on some experiments of using TiLe to check classifiers from the `scikit-learn` library for balancedness.

1 Overview

In supervised machine learning, an ML algorithm is presented with a set of labelled training data, each consisting of a feature vector and a label (describing e.g. a class). From this, it learns a *predictive model* generalizing from the data as to later make predictions of labels for unseen data instances. A predictor is considered to be unfair if a change to the value of a feature in a data instance leads to a change in the prediction. Fairness testing [GYBM17] aims at checking such discrimination in ML predictors.

Complementary to that, we are interested in finding "unfairness" or "biasedness" in the learning phase. Basically, we aim at investigating whether all the data instances are used in the same way during training, and we formalize this using a number of metamorphic (mm) transformations [CCY98]: (a) permutation of training data instances (row permutation), (b) permutation of feature ordering (column permutation) and (c) shuffling of feature names. We consider an ML algorithm to be *balanced* when an application of mm-transformations on the training data does not change the predictive model learned.

For checking balancedness, we have implemented a framework called TiLe. TiLe contains a training data repository which consists of 4 artificial and 9 real-world datasets. Given an ML classifier as input, TiLe tests it for balancedness by training two predictors (per training data

[1] Universität Paderborn, Institut für Informatik, arnab.sharma@uni-paderborn.de
[2] Universität Paderborn, Institut für Informatik, wehrheim@upb.de

doi:10.18420/se2019-48

Tab. 1: Sensitivity to metamorphic transformations

Classifiers	*FN shuffling*	*Row perm.*	*Column perm.*
k-NN	✗	✓	✗
Decision Tree	✗	✓	✓
Naive Bayes	✗	✗	✗
SVM	✗	✓	✗
Neural Network	✗	✓	✓
Logistic Regression	✗	✓	✓
Random Forest	✗	✓	✓
Ada Boost	✗	✓	✓
Bagging Classifier	✗	✓	✓
Extra Trees	✗	✗	✓
Gradient Boosting	✗	✓	✓
Gaussian	✗	✗	✗
Stochastic Gradient	✗	✓	✗
Linear Regression	✗	✓	✓
Elastic Net	✗	✗	✓

and per mm-transformation): one with the original data and one with an mm-transformation applied on it. Afterwards, the predictors are checked for equality. Equality checking proceeds either by comparing the representations of the predictive models or by testing. A comparison of representations is only possible for some models (e.g. for support vector machines); often unequivalent representations define equal predictors. In a large number of cases we therefore employ testing, using different strategies for test case selection.

In our experiments, we used TiLe to check 13 classifiers from the `scikit-learn` library. Table 1 show the results of the experiments: none of the classifiers are sensitive to shuffling of feature names (i.e., feature names play no role for learning), but all are sensitive to either row or column permutation. Given that a software developer employing an off-the-shelf machine learning algorithm would expect it to learn some fixed predictive model when run on a given data set, this is a surprising result. To see whether our approach is also able to detect intentionally biased algorithms, we have furthermore applied it on 4 classifiers which are by design unbalanced (e.g. the ones of Zafar et al. [Za17]). All such unbalancedness was detected by TiLe.

References

[CCY98] Chen, T.Y.; Cheung, S.C.; Yiu, S.M.: Metamorphic testing: a new approach for generating next test cases. Technical report, Technical Report HKUST-CS98-01, 1998.

[GYBM17] Galhotra, S.; Y. Brun, Yuriy; Meliou, A.: Fairness testing: testing software for discrimination. In: ESEC/FSE. ACM, pp. 498–510, 2017.

[Za17] Zafar, M.B.; Valera, I.; Gomez-Rodriguez, M.; Gummadi, K.P.: Fairness Constraints: Mechanisms for Fair Classification. In: AISTATS. pp. 962–970, 2017.