Journal of Machine Learning Research 7 (2016) 1-5 Submitted 7/16; Published -

A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning

Guillaume Lemaître G.LEMAITRE58@GMAIL.COM
LE2I UMR6306, CNRS, Arts et Métiers, Univ. Bourgogne Franche-Comté
12 rue de la Fonderie, 71200 Le Creusot, France
ViCOROB, Universitat de Girona
Campus Montilivi, Edifici P4, 17071 Girona, Spain

Fernando Nogueira FMFNOGUEIRA@GMAIL.COM
theScore, Inc.
500 King Street West 4^{th} Floor Toronto, Ontario M5V1L9 Canada

Christos K. Aridas CHAR@UPATRAS.GR
University of Patras
University Campus, 26504 Patras, Greece

Editor: -

Abstract

`imbalanced-learn` is an open-source python toolbox aiming at providing a wide range of methods to cope with the problem of imbalanced dataset frequently encountered in machine learning and pattern recognition. The implemented state-of-the-art methods can be categorized into 4 groups: (i) under-sampling, (ii) over-sampling, (iii) combination of over- and under-sampling, and (iv) ensemble learning methods. The proposed toolbox only depends on `numpy`, `scipy`, and `scikit-learn` and is distributed under MIT license. Furthermore, it is fully compatible with `scikit-learn`.Documentation, unit tests as well as integration tests are provided to ease usage and contribution. The toolbox is publicly available in GitHub `https://github.com/fmfn/UnbalancedDataset`.

Keywords: Imbalanced Dataset, Over-Sampling, Under-Sampling, Ensemble Learning, Machine Learning, Python.

1. Introduction

Real world datasets commonly show the particularity to have a number of samples of a given class under-represented compared to other classes. This imbalance gives rise to the "class imbalance" problem (Prati et al., 2009) (or "curse of imbalanced datasets") which is the problem of learning a concept from the class that has a small number of samples.

The class imbalance problem has been encountered in multiple areas such as telecommunication managements, bioinformatics, fraud detection, and medical diagnosis, and has been considered one of the top 10 problems in data mining and pattern recognition (Yang and Wu, 2006; Rastgoo et al., 2016). Imbalanced data substantially compromises the learning process, since most of the standard machine learning algorithms expect balanced class distribution or an equal misclassification cost (He and Garcia, 2009). For this reason, several approaches have been specifically proposed to handle such datasets. Such standalone

methods have been implemented mainly in R language (Torgo, 2010; Kuhn, 2015; Dal Pozzolo et al., 2013). Up to our knowledge, however, there is no python toolbox allowing such processing while cutting edge machine learning toolboxes are available (Pedregosa et al., 2011; Sonnenburg et al., 2010).

In this paper, we present the `imbalanced-learn` API, *a python toolbox to tackle the curse of imbalanced datasets in machine learning*. The following sections present the project vision, a snapshot of the API, an overview of the implemented methods, and finally, the conclusion of this paper, including future functionalities for the `imbalanced-learn` API.

2. Project management

Quality insurance In order to ensure code quality, a set of unit tests is provided leading to a coverage of 99 % for the release 0.1 of the toolbox. Furthermore, the code consistency is ensured by following `PEP8` standards and each new contribution is automatically checked through landscape, which provides metrics related to code quality.
Continuous integration To allow user and developer to either use or contribute to this toolbox, Travis CI is used to easily integrate new code and ensure back-compatibility.
Community-based development All the development is performed in a collaborative manner. Tools such as git, GitHub, and gitter are used to ease collaborative programming, issue tracking, code integration, and idea discussions.
Documentation A consistent API documentation is provided using `sphinx` and `numpydoc`. An additional installation guide and examples are also provided and centralized on GitHub[1].
Project relevance At the edition time, the repository is visited no less than 2,000 per week, attracting about 300 unique visitors per week.

3. Implementation design

```
from sklearn.datasets import make_classification
from sklearn.decomposition import PCA
from imblearn.over_sampling import SMOTE

# Generate the dataset
X, y = make_classification(n_classes=2, weights=[0.1, 0.9],
                           n_features=20, n_samples=5000)

# Apply the SMOTE over-sampling
sm = SMOTE(ratio='auto', kind='regular')
X_resampled, y_resampled = sm.fit_sample(X, y)
```

Listing 1: Code snippet to over-sample a dataset using SMOTE.

The implementation relies on `numpy`, `scipy`, and `scikit-learn`. Each sampler class implements three main methods inspired from the `scikit-learn` API: (i) `fit` computes the parameter values which are later needed to resample the data into a balanced set; (ii) `sample` performs the sampling and returns the data with the desired balancing ratio; and (iii) `fit_sample` is equivalent to calling the method `fit` followed by the method `sample`. A

1. `http://fmfn.github.io/UnbalancedDataset/`

class `Pipeline` is inherited from `scikit-learn` toolbox to automatically combine `samplers`, `transformers`, and `estimators`.

4. Implemented methods

The `imbalanced-learn` toolbox provides four different strategies to tackle the problem of imbalanced dataset: (i) under-sampling, (ii) over-sampling, (iii) a combination of both, and (iv) ensemble learning. The following subsections give an overview of the techniques implemented.

4.1 Notation and background

Let χ an imbalanced dataset with χ_{min} and χ_{maj} being the subset of samples belonging to the minority and majority class, respectively. The balancing ratio of the dataset χ is defined as:

$$r_{\chi} = \frac{|\chi_{min}|}{|\chi_{maj}|} \ , \tag{1}$$

where $|\cdot|$ denotes the cardinality of a set. The balancing process is equivalent to resample χ into a new dataset χ_{res} such that $r_{\chi} > r_{\chi_{res}}$.

4.2 Under-sampling

Under-sampling refers to the process of reducing the number of samples in χ_{maj}. The implemented methods can be categorized into 2 groups: (i) fixed under-sampling and (ii) cleaning under-sampling.

Fixed under-sampling refer to the methods which perform under-sampling to obtain the appropriate balancing ratio $r_{\chi_{res}}$. The implemented methods perform the under-sampling based on different criteria such as: (i) random selection, (ii) clustering, (iii) nearest neighbours rule (i.e., `NearMiss` (Mani and Zhang, 2003)), and (iv) classification accuracy (i.e., `instance hardness threshold` (Smith et al., 2014)).

In the contrary to the previous methods, *cleaning under-sampling* do not allow to reach specifically the balancing ratio $r_{\chi_{res}}$, but rather clean the feature space based on some empirical criteria. These criteria are derived from the nearest neighbours rule, namely: (i) `condensed nearest neighbours` (Hart, 1968), (ii) `edited nearest neighbours` (Wilson, 1972), (iii) `one-sided selection` (Kubat et al., 1997), (iv) `neighbourhood cleaning rule` (Laurikkala, 2001), and (v) `Tomek links` (Tomek, 1976).

4.3 Over-sampling

In the contrary to under-sampling, data balancing can be performed by over-sampling such that new samples are generated in χ_{min} to reach the balancing ratio $r_{\chi_{res}}$. Two methods are currently available: (i) `Random over-sampling` is performed by randomly replicating the samples of χ_{min} to obtain the appropriate balancing ratio $r_{\chi_{res}}$ and `SMOTE` which randomly generate new samples between tuple of nearest neighbours of χ_{min} (Chawla et al., 2002). Different variants of this algorithm have been proposed: `SMOTE borderline 1 & 2` (Han et al., 2005) and `SMOTE SVM` (Nguyen et al., 2011).

4.4 Combination of over- and under-sampling

`SMOTE` over-sampling can lead to over-fitting which can be avoided by applying cleaning under-sampling methods (Prati et al., 2009). In that regard, Batista et al. (2003) combined `SMOTE` either with `Tomek links` or `edited nearest neighbours`.

4.5 Ensemble learning

Under-sampling methods imply that samples of the majority class are lost during the balancing procedure. Ensemble methods offer an alternative to use most of the samples. In fact, an ensemble of balanced sets is created and used to later train any classifier. Two methods are available to build such ensemble proposed by Liu et al. (2009): `EasyEnsemble` and `BalanceCascade`. The former is based on iteratively applying the `random under-sampling` method to build several sets, each of them with a desired balancing ratio $r_{\chi_{res}}$. The latter differs such that a classifier is used at each iteration to determine the class of the randomly selected samples. Misclassified samples are kept and propagated in the next subset.

5. Future plans and conclusion

In this paper, we shortly presented the foundations of the `imbalanced-learn` toolbox vision and API. As avenues for future works, additional methods based on prototype/instance selection, generation, and reduction will be added as well as additional user guides.

References

G. E. Batista, A. L. Bazzan, and M. C. Monard. Balancing training data for automated annotation of keywords: a case study. In *WOB*, pages 10–18, 2003.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, pages 321–357, 2002.

A. Dal Pozzolo, O. Caelen, S. Waterschoot, and G. Bontempi. Racing for unbalanced methods selection. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 24–31. Springer, 2013.

H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, pages 878–887. Springer, 2005.

P. Hart. The condensed nearest neighbor rule. *Information Theory, IEEE Transactions on*, 14(3):515–516, May 1968.

H. He and E. Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284, 2009.

M. Kubat, S. Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *International Conference in Machine Learning*, volume 97, pages 179–186. Nashville, USA, 1997.

M. Kuhn. Caret: classification and regression training. *Astrophysics Source Code Library*, 1:05003, 2015.

J. Laurikkala. *Improving identification of difficult small classes by balancing class distribution*. Springer, 2001.

X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

I. Mani and I. Zhang. knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of Workshop on Learning from Imbalanced Datasets*, 2003.

H. M. Nguyen, E. W. Cooper, and K. Kamei. Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 3(1):4–21, 2011.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

R. C. Prati, G. E. Batista, and M. C. Monard. Data mining with imbalanced class distributions: concepts and methods. In *Indian International Conference Artificial Intelligence*, pages 359–376, 2009.

M. Rastgoo, G. Lemaitre, J. Massich, O. Morel, F. Marzani, R. Garcia, and F. Meriaudeau. Tackling the problem of data imbalancing for melanoma classification. In *Bioimaging*, 2016.

M. R. Smith, T. Martinez, and C. Giraud-Carrier. An instance level analysis of data complexity. *Machine learning*, 95(2):225–256, 2014.

S. C. Sonnenburg, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, V. Franc, et al. The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, 11(Jun):1799–1802, 2010.

I. Tomek. Two modifications of CNN. *Systems, Man, and Cybernetics, IEEE Transactions on*, 6:769–772, 1976.

L. Torgo. *Data mining with R: learning with case studies*. Chapman & Hall/CRC, 2010.

D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *Systems, Man and Cybernetics, IEEE Transactions on*, (3):408–421, 1972.

Q. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.