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

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 depends only on numpy, scipy, and scikit-learn and is distributed under MIT license. Further- more, it is fully compatible with scikit-learn and is part of the scikit-learn-contrib supported project. Documentation, unit tests as well as integration tests are provided to ease usage and contribution. Source code, binaries, and documentation can be downloaded from https://github.com/scikit-learn-contrib/imbalanced-learn.

Keywords: ImbalancedDataset,Over-Sampling,Under-Sampling,EnsembleLearning, 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. Some of these methods have been implemented mainly in R language (Torgo, 2010; Kuhn, 2015; Dal Pozzolo et al., 2013). Up to our knowledge, 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, we conclude this work by including future functionalities for the imbalanced-learn API.

2. Project management

Quality assurance In order to ensure code quality, a set of unit tests is provided leading to a coverage of 99 % for the release 0.2 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 both the user and the 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 GitHub1.

Project relevance At the edition time, the repository is visited no less than 2, 000 times per week, attracting about 300 unique visitors per week. Additionally, the toolbox is supported by scikit-learn through the scikit-learn-contrib projects.

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 several statistics which are later needed to resample the data into a balanced set; (ii)

[1] https://github.com/scikit- learn- contrib/imbalanced- learn

Metho d	Over-sampling		Under-sampling	
	Binary	Mutli-class	Binary	Multiclass
ADASYN (He et al., 2008)	3	7	7	7
SMOTE (Chawla et al., 2002; Han et al., 2005; Nguyen et al., 2011)	3	7	7	7
ROS	3	3	7	7
CC	7	7	3	3
CNN (Hart, 1968)	7	7	3	3
ENN (Wilson, 1972)	7	7	3	3
RENN	7	7	3	3
AKNN	7	7	3	3
NM (Mani and Zhang, 2003)	7	7	3	3
NCL (Laurikkala, 2001)	7	7	3	3
OSS (Kubat et al., 1997)	7	7	3	3
RUS	7	7	3	3
IHT (Smith et al., 2014)	7	7	3	7
TL (Tomek, 1976)	7	7	3	7
BC (Liu et al., 2009)	7	7	3	7
EE (Liu et al., 2009)	7	7	3	3
SMOTE + ENN (Batista et al., 2003)	3	7	3	7
SMOTE + TL (Batista et al., 2003)	3	7	3	7

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 class Pipeline is inherited from the scikit-learn toolbox to automatically combine samplers, transformers, and estimators. Additionally, we provide some specific state- of-the-art metrics to evaluate classification performance.

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 be an imbalanced dataset with χmin and χmaj being the subset of samples belonging to the minority and majority class, respectively. The balancing ratio of the dataset χ is defined as:

|χmin|rχ=, (1)

|χmaj |

where

| · | denotes the cardinality of a set. The balancing process is equivalent to resample χ into a new dataset χres such that rχ > rχres.

Under-sampling Under-samplingreferstotheprocessofreducingthenumberofsamples in χmaj. 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 rχres. Contrary to the previous methods, cleaning under-sampling do not allow to reach specifically the balancing ratio rχres, but rather clean the feature space based on some empirical criteria.

Over-sampling Contrarytounder-sampling,databalancingcanbeperformedbyover- sampling such that new samples are generated in χmin to reach the balancing ratio rχres.

Combination of over- and under-sampling Over-sampling can lead to over-fitting which can be avoided by applying cleaning under-sampling methods (Prati et al., 2009).

Ensemblelearning Under-samplingmethodsimplythatsamplesofthemajorityclass 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.

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.

Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pages 1322–1328. IEEE, 2008.

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.