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, Université 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

Dayvid V. R. Oliveira DVRO@CIN.UFPE.BR
VIISAR Research Group, Centro de Informática - Universidade Federal de Pernambuco
Av. Jornalista Anbal Fernandes, s/n - Cidade Universitria - PE, 50740-560, Brazil

Editor: -

Abstract

UnbalancedDataset 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 state-of-the-art methods implemented can be categorized into 4 different sampling strategies: (i) under-sampling, (ii) over-sampling, (iii) combination of over- and under-sampling, and (iv) ensemble learning methods. The proposed toolbox only depends of *numpy*, *scipy*, and *scikit-learn* and is distributed under MIT license. The implementation pattern is identical to the *scikit-learn* API to offer compatibility. Documentation, unit tests as well as integration tests are provided to allow new developers to contribute. The toolbox is publicly available in GitHub `https://github.com/fmfn/UnbalancedDataset` as well as the documentation `http://fmfn.github.io/UnbalancedDataset/`.

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

1. Introduction

Many real world datasets have many samples of some classes (majority classes), and only a few samples the other class (minority 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 compared to the other classes.

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). Medical data, for example, are prone to suffer from class imbalance due to the fact that the portion of diseased patients is far lower than healthy patients, and the

detection and classification of minority diseased patients are highly essential so that they can be treated as soon as possible.

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 et al. (2009). For this reason, several approaches have been specifically proposed to handle such datasets.

In this paper, we present the **UnbalancedDataset** API, *a python toolbox to tackle the curse of imbalanced datasets in machine learning*. The following sections present the implemented methods, implemented design, sustainability and continuous integration details, and finally, the conclusion of this paper, including future functionalities for the UnbalancedDataset 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 check through landscape, which provides metrics related to code quality.

Continuous integration To allow user and developer to either use or contribute 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 as git, GitHub, and gitter are used to ease collaborative programming, issue tracking, code integration, and ideas discussions.

Documentation A consistent API documentation is provided using *sphinx* and *numpydoc*. Additional installation guide, examples, and tutorial are also provided and centralized on GitHub[1].

3. Implementation design

The implementation rely on *numpy*, *scipy*, and *scikit-learn*. Each class implements 3 main functions inspired from the *scikit-learn* API: (i) *fit* computes the parameter values which are later needed to transform the data into a balanced set; (ii) *transform* performs the sampling and return the data with the desired balancing ratio; and (iii) *fit_transform* is equivalent of calling the function *fit* follow the function *transform*.

4. Implemented methods

The **UnbalancedDataset** 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 sections give an overview of the techniques implemented.

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

4.1 Under-sampling

Considering the problem of imbalanced, under-sampling is performed such that the number of samples of the majority class is reduced. The following methods are considered to perform such balancing.

Random under-sampling is performed by randomly selecting without replacement a subset of samples from the majority class such that the number of samples is then equal in both minority and majority classes.

Cluster centroids method refers to the use of a k-means to cluster the feature space such that k is set to be equal to the number of samples composing the minority class. Hence, the centroids of these clusters define the new samples of the majority class.

Condensed nearest neighbours Something here.

Edited nearest neighbours Something here.

Instance hardness neigbours

NearMiss offers three different methods to under-sample the majority class Mani and Zhang (2003). In NearMiss-1, samples from the majority class are selected such that for each sample, the average distance to the k nearest neighbour samples from the minority class is minimum. NearMiss-2 diverges from NearMiss-1 by considering the k farthest neighbours samples from the minority class. In NearMiss-3, a subset M containing samples from the majority class is generated by finding the m nearest neighbours from each sample of the minority class. Then, samples from the subset M are selected such that for each sample, the average distance to the k nearest neighbour samples from the minority class is maximum. In our experiment, k and m are fixed to 3.

One-sided selection Something here.

Neighbourhood cleaning rule consists of applying two rules depending on the class of each sample Laurikkala (2001). Let define x_i as a sample of the dataset with its associated class label y_i. Let define y_m as the class of the majority vote of the k nearest neighbours of the sample x_i. If y_i corresponds to the majority class and $y_i \neq y_m$, x_i is rejected from the final subset. If y_i corresponds to the minority class and and $y_i \neq y_m$, then the k nearest neighbours are rejected from the final subset. In our experiment k is fixed to 3.

Tomek links can be used to under-sample the majority class of the original dataset Tomek (1976). Let define a pair of nearest neighbour samples (x_i, x_j) such that their associated class label $y_i \neq y_j$. The pair (x_i, x_j) is defined as a Tomek link if, by relaxing the class label differentiation constraint, there is no other sample x_k defined as the nearest neighbour of either x_i or x_j. Under-sampling is performed by removing the samples belonging to the majority class and forming a Tomek link. It can be noted that this under-sampling strategy does not enforce a strict balancing between the majority and the minority classes.

4.2 Over-sampling

In the contrary, the data balancing can be performed by over-sampling in which the new samples belonging to the minority class are generated aiming at equalizing the number of samples in both classes. Two different methods are considered.

Random over-sampling is performed by randomly replicating the samples of the minority class such that the number of samples is equal in both minority and majority classes.

SMOTE is a method to generate synthetic samples in the feature space Chawla et al. (2002). Let define x_i as a sample belonging to the minority class. Let define x_{nn} as a randomly selected sample from the k nearest neighbours of x_i, with k set to 3. Therefore, a new sample x_j is generated such that $x_j = x_i + \sigma\,(x_{nn} - x_i)$, where σ is a random number in the interval $[0, 1]$. Three other variants of this algorithm exist: (i) SMOTE borderline 1, (ii) SMOTE borderline 2, and (iii) SMOTE SVM.

4.3 Combination of over- and under-sampling

Subsequently, over-sampling methods can be combined with under-sampling methods to clean the subset created. In that regard, two different combinations are tested.

SMOTE + Tomek links are combined to clean the samples created using SMOTE Batista et al. (2003). SMOTE over-sampling can lead to over-fitting which can be avoided by removing the Tomek links from both majority and minority classes Prati et al. (2009).

SMOTE + edited nearest neighbours are combined for the same aforementioned reason Batista et al. (2004).

4.4 Ensemble learning

Under-sampling methods implies that samples of the majority class will be lost during the balancing procedure. Ensemble methods can offer an alternative to use most of the samples. In fact, an ensemble of balanced set will be created and used to later train any classifier. Two methods are available to build such ensemble.

Balance cascade

Easy ensemble

5. Conclusion

Acknowledgments

We would like to acknowledge support for this project from git, GitHub, Travis CI, and Gitter.

References

Gustavo EAPA Batista, Ana LC Bazzan, and Maria Carolina Monard. Balancing training data for automated annotation of keywords: a case study. In *WOB*, pages 10–18, 2003.

Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1):20–29, 2004.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, pages 321–357, 2002.

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

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

Inderjeet 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.

Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. Data mining with imbalanced class distributions: concepts and methods. In *IICAI*, pages 359–376, 2009.

Ivan Tomek. Two modifications of cnn. *IEEE Trans. Syst. Man Cybern.*, 6:769–772, 1976.

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