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 4th 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. 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

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 (Rastgoo et al., 2016; Yang and Wu, 2006). 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 `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 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:

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

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} to obtain the appropriate balancing ratio $r_{\chi_{res}}$. The following methods are considered to perform such balancing.

Random under-sampling is performed by randomly selecting without replacement a subset of samples from χ_{maj} to obtain the desired balancing ratio $r_{\chi_{res}}$.

Cluster centroids method refers to the use of a k-means to cluster the feature space. k corresponds to the number of samples in $\chi_{res_{maj}}$ defined by the desired balancing ratio $r_{\chi_{res}}$. Note that the samples in $\chi_{res_{maj}}$ do not correspond to the original samples of χ_{maj} and are synthetically generated.

Condensed nearest neighbours Something here.

Edited nearest neighbours Something here.

Instance hardness threshold

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

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 χ_{maj} and $y_i \neq y_m$, x_i is rejected from the final subset. If y_i corresponds to χ_{min} class and and $y_i \neq y_m$, then the k nearest neighbours are rejected from the final subset.

Tomek links can be used to under-sample χ_{maj} (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 χ_{maj} and forming a Tomek link.

4.3 Over-sampling

In the contrary of under-sampling, data balancing can be performed by over-sampling in which new samples are generated in χ_{min} to reach the balancing ratio $r_{\chi_{res}}$. The following methods are currently available.

Random over-sampling is performed by randomly replicating the samples of χ_{min} to obtain the appropriate balancing ratio $r_{\chi_{res}}$.

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 $x_j = x_i + \sigma\left(x_{nn} - x_i\right)$, 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.

4.4 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).

4.5 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 Something here

Easy ensemble Something here

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.

Andrea Dal Pozzolo, Olivier Caelen, Serge Waterschoot, and Gianluca Bontempi. Racing for unbalanced methods selection. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 24–31. Springer, 2013.

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

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

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.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

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

Mojdeh Rastgoo, Guillaume Lemaitre, Joan Massich, Olivier Morel, Franck Marzani, Rafael Garcia, and Fabrice Meriaudeau. Tackling the problem of data imbalancing for melanoma classification. In *Bioimaging*, 2016.

SĆ Sonnenburg, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, Vojtä Franc, et al. The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, 11(Jun):1799–1802, 2010.

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

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

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