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

Guillaume Lemaître
Parietal team, Inria, CEA, Université Paris-Saclay
1 Rue Honoré d’Estienne d’Orves, 91120 Palaiseau, France

Fernando Nogueira
ShoppeAI
488 Wellington Street West, Suite 304, Toronto, Ontario M5V 1E3, Canada

Christos K. Aridas
Computational Intelligence Laboratory
Department of Mathematics
University of Patras
GR-26504 Patras, Greece

Editor: Geoff Holmes

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. Furthermore, 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: 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. 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 GitHub.1

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

```python
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)
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 $\chi$ be an imbalanced dataset with $\chi_{\text{min}}$ and $\chi_{\text{maj}}$ being the subset of samples belonging to the minority and majority class, respectively. The balancing ratio of the dataset $\chi$ is defined as:

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

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

**Under-sampling** Under-sampling refers to the process of reducing the number of samples in $\chi_{\text{maj}}$. 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 $r_{\chi_{\text{res}}}$, Contrary to the previous methods, cleaning under-sampling do not allow to reach specifically the balancing ratio $r_{\chi_{\text{res}}}$, but rather clean the feature space based on some empirical criteria.

<table>
<thead>
<tr>
<th>Method</th>
<th>Over-sampling</th>
<th>Under-sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Binary</td>
<td>Multi-class</td>
</tr>
<tr>
<td>ADASYN (He et al. 2008)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SMOTE (Chawla et al. 2002, Han et al., 2005, Nguyen et al. 2011)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ROS</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>CC</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>CNN (Hart, 1968)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ENN (Wilson 1972)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>RENN</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>AKNN</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NM (Mani and Zhang 2003)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NCL (Laurikkala 2001)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>OSS (Kubat et al. 1997)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>RUS</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>IHT (Smith et al. 2014)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>TL (Tomek 1976)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>BC (Lau et al. 2009)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>EE (Liu et al. 2009)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SMOTE + ENN (Batista et al. 2003)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SMOTE + TL (Batista et al. 2003)</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Over-sampling  Contrary to under-sampling, data balancing can be performed by over-sampling such that new samples are generated in $\chi_{\text{min}}$ to reach the balancing ratio $r_{\chi_{\text{res}}}$. 

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).

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.

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


