geometric-smote - A Package for Flexible and Efficient Over-Sampling

Georgios Douzas GDOUZAS@NOVAIMS.UNL.PT
NOVA Information Management School
Universidade Nova de Lisboa
Campus de Campolide, 1070-312 Lisboa, Portugal

Fernando Bacao BACAO@NOVAIMS.UNL.PT
NOVA Information Management School
Universidade Nova de Lisboa
Campus de Campolide, 1070-312 Lisboa, Portugal

Editor:

Abstract

Learning from class-imbalanced data continues to be a frequent and challenging problem in machine learning. To mitigate this problem several approaches have been proposed. A popular approach is the generation of artificial data for the minority classes, known as over-sampling. Geometric SMOTE is a state-of-the-art over-sampling algorithm that has been shown to outperform other standard over-samplers in a large number of data sets. In order to make available Geometric SMOTE to the machine learning community, we provide a Python implementation with source code and documentation found at `https://github.com/georgedouzas/geometric-smote` and `https://geometric-smote.readthedocs.io`, respectively. The implementation integrates seamlessly with the `scikit-learn` ecosystem.

Keywords: machine learning, classification, imbalanced learning, over-sampling, Python

1. Introduction

The imbalanced learning problem is defined as a machine learning classification task using data sets with binary or multi-class targets where one of the classes, called the majority class, outnumbers significantly the remaining classes, called the minority class(es) (Chawla et al., 2003). The imbalance learning problem can be found in multiple domains such as chemical and biochemical engineering, financial management, information technology, security, business, agriculture or emergency management (Haixiang et al., 2017).

Standard machine learning classification algorithms induce a bias towards the majority class during training. This results in low performance when metrics suitable for imbalanced data are used for the classifier's evaluation.

In this paper, we present the `geometric-smote` software project, a Python implementation of the Geometric-SMOTE (Douzas and Bacao, 2019) algorithm. The following sections provide a description of the algorithm's properties as well as a presentation of the software architecture and functionalities.

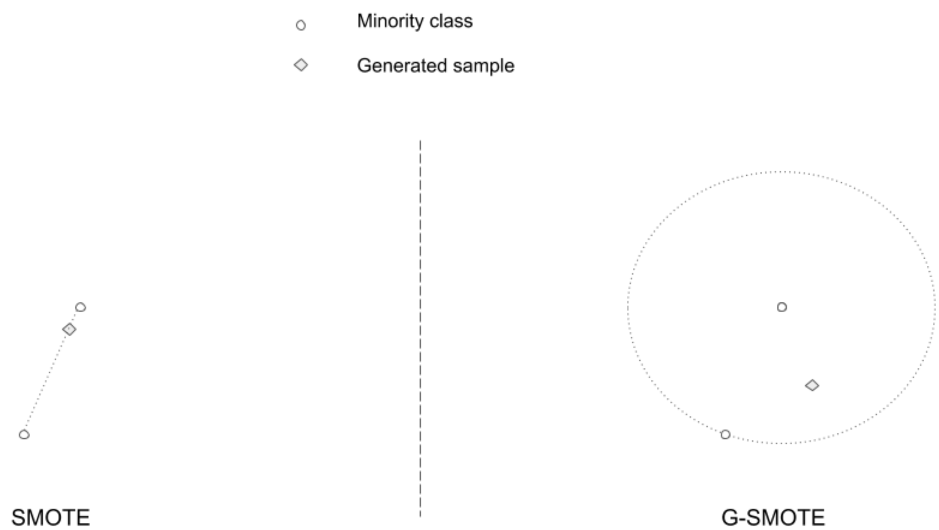

Figure 1: Comparison between the data generation mechanisms of SMOTE and G-SMOTE. SMOTE uses linear interpolation, while G-SMOTE defines a circle as the permissible data generation area.

2. Geometric SMOTE algorithm

A general approach to deal with the imbalanced learning problem is the modification at the data level by over-sampling the minority class(es) (Fernández et al., 2013). Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002), the first informed over-sampling algorithm proposed, generates synthetic instances along a line segment that joins minority class samples. Many variants of SMOTE have been proposed to deal with some of its limitations (He and Garcia, 2009). A Python implementation of SMOTE and several of its variants is available in the `imbalanced-learn` (Lemaitre et al., 2016) toolbox, which is fully compatible with the popular machine learning library `scikit-learn` (Pedregosa et al., 2011).

Geometric SMOTE (G-SMOTE) uses a different approach compared to the existing SMOTE's variations. More specifically, G-SMOTE over-sampling algorithm substitutes the data generation mechanism of SMOTE by defining a flexible geometric region around each minority class instance and generating synthetic instances inside the boundaries of this region. The algorithm requires the selection of the hyperparameters `truncation_factor`, `deformation_factor`, `selection_strategy` and `k_neighbors`. The first three of them, called geometric hyperparameters, control the shape of the geometric region while the later adjusts its size. Figure 1 presents a visual comparison between the data generation mechanisms of SMOTE and G-SMOTE.

G-SMOTE algorithm has been shown to outperform SMOTE and its variants across 69 imbalanced data sets for various classifiers and evaluation metrics (Douzas and Bacao, 2019).

3. Software architecture

The `geometric-smote` software project is written in Python 3. It contains an object-oriented implementation of G-SMOTE as well as an extensive online documentation found at `https://geometric-smote.readthedocs.io`. The provided API is compatible with `scikit-learn` and `imbalanced-learn` libraries, therefore it makes full use of various features that support standard machine learning functionalities. For instance, `GeometricSMOTE` objects can be used in a machine learning pipelines, through `imbalanced-learn`'s class `Pipeline`, that automatically combines `samplers`, `transformers` and `estimators`.

The main module of `geometric-smote` is called `geometric-smote.py`. It contains the class `GeometricSMOTE` that implements the G-SMOTE algorithm. The initialization of a `GeometricSMOTE` instance includes G-SMOTE's geometric hyperparameters that control the generation of synthetic data i.e. `truncation_factor`, `deformation_factor` and `selection_strategy`. The implementation also supports the use of categorical features through the `categorical_features` initialization parameter.

`GeometricSMOTE` inherits from the `BaseOverSampler` class of `imbalanced-learn` library and implements its `_fit_resample` abstract method. Consequently, an instance of the `GeometricSMOTE` class provides the `fit` and `fit_resample` methods, the two main methods for resampling. Both of them take as input parameters the `X` and `y`. The first method computes various statistics which are used to resample `X`, while the second method does the same but additionally returns a resampled version of `X` and `y`. Figure 2 provides a visual representation of the above classes and functions hierarchy while Listing 1 presents an example of over-sampling am imbalanced 3-class data set.

```python
from collections import Counter from gsmote
import GeometricSMOTE from sklearn.datasets import make_classification

# Generate an imbalanced 3-class data set
X, y = make_classification(
        random_state=23,
        n_classes=3,
        n_informative=5,
        n_samples=500,
        weights=[0.8,0.15, 0.05]
)

# Create a GeometricSMOTE object with default hyperparameters
gsmote = GeometricSMOTE()

# Resample the imbalanced data set using G-SMOTE
X_res, y_res = gsmote.fit_resample(X, y)
```

Listing 1: Code snippet to over-sample a data set using G-SMOTE.

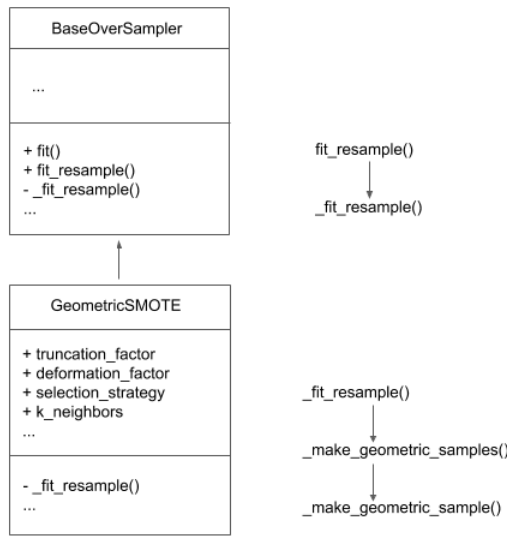

Figure 2: UML class diagrams and callgraphs of main classes and methods.

4. Project management

Releases of the `geometric-smote` package are available via PyPI and conda-forge. Collaboration on the project is possible via GitHub where users can open new issues or reply to current issues and make pull requests. Continuous integration with GitHub Actions is also supported. The `PEP8` style standards are followed while extensive unit testing of the code is applied. The documentation includes installation instructions, a detailed description of the API and a user guide with various examples. Finally, the package is distributed under the MIT license.

5. Impact and conclusions

The `geometric-smote` project provides the only Python implementation, to the best of our knowledge, of the state-of-the-art over-sampling algorithm G-SMOTE. A significant advantage of this implementation is that it is built on top of the `scikit-learn`'s ecosystem. Therefore, using the G-SMOTE over-sampler in typical machine learning workflows is an effortless task for the user. Also, the public API of the main class `GeometricSMOTE` is identical to the one implemented in `imbalanced-learn` for all over-samplers. This means that users of `imbalanced-learn` and `scikit-learn`, that apply over-sampling on imbalanced data, can integrate the `gsmote` package in their existing work in a straightforward manner or even replace directly any `imbalanced-learn`'s over-sampler with `GeometricSMOTE`.

References

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. ISSN 10769757. doi: 10.1613/jair.953.

Nitesh V Chawla, Aleksandar Lazarevic, Lawrence Hall, and Kevin Boyer. SMOTEBoost: improving prediction of the minority class in boosting. *Principles of Knowledge Discovery in Databases, PKDD-2003*, pages 107–119, 2003. ISSN 03029743. doi: 10.1007/b13634. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.80.1499`.

Georgios Douzas and Fernando Bacao. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information Sciences*, 501:118–135, oct 2019. ISSN 0020-0255. doi: 10.1016/J.INS.2019.06.007. URL `https://www.sciencedirect.com/science/article/pii/S0020025519305353?via{%}3Dihub`.

Alberto Fernández, Victoria López, Mikel Galar, María José del Jesus, and Francisco Herrera. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 42:97–110, 2013. ISSN 0950-7051. doi: http://dx.doi.org/10.1016/j.knosys.2013.01.018. URL `http://www.sciencedirect.com/science/article/pii/S0950705113000300`.

Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, May 2017. doi: 10.1016/j.eswa.2016.12.035. URL `https://doi.org/10.1016/j.eswa.2016.12.035`.

Haibo He and Edwardo A Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. ISSN 10414347. doi: 10.1109/TKDE.2008.239.

Guillaume Lemaitre, Fernando Nogueira, and Christos K Aridas. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18:1–5, 2016. ISSN 15337928. doi: http://www.jmlr.org/papers/volume18/16-365/16-365.pdf. URL `http://arxiv.org/abs/1609.06570`.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. *Scikit-learn: Machine learning in Python*, volume 12. 2011. ISBN 9781783281930. doi: 10.1007/s13398-014-0173-7.2. URL `http://dl.acm.org/citation.cfm?id=2078195`.