MDPI

Article

Imbalanced Learning in Land Cover Classification: Improving minority classes' prediction accuracy using the Geometric SMOTE algorithm

Georgios Douzas [1], Fernando Bacao[1]*, Joao Fonseca [1,†] and Manvel Khudinyan [1,†]

[1] NOVA Information Management School; {gdouzas, bacao, jpfonseca, mkhudinyan}@novaims.unl.pt
* Correspondence: bacao@novaims.unl.pt; Tel.: +351 21 382 8610
† These authors contributed equally to this work.

Version December 4, 2019 submitted to Remote Sens.

Abstract: The automatic production of Land Use/Land Cover maps continues to be a challenging problem, with important impacts on the ability to promote sustainability and good resource management. The ability to build robust automatic classifiers and produce accurate maps can have a significant impact on the way we manage and optimize natural resources. The difficulty in achieving these results comes from many different factors, such as data quality and uncertainty. In this paper, we address the imbalanced learning problem, a common and difficult conundrum in remote sensing that affects the quality of classification results by proposing Geometric-SMOTE, a novel oversampling method, as a tool for addressing the imbalanced learning problem in remote sensing. Geometric-SMOTE is a sophisticated oversampling algorithm which increases the quality of the generated instances over previous methods, such as the Synthetic Minority Oversampling TEchnique. The performance of Geometric- SMOTE, in the LUCAS (Land Use/Cover Area frame Survey) dataset, is compared to other oversamplers using a variety of classifiers. The results show that Geometric-SMOTE significantly outperforms all the other oversamplers and improves the robustness of the classifiers. These results indicate that, when using imbalanced datasets, remote sensing researchers should consider the use of these new generation oversamplers to increase the quality of the classification results.

Keywords: Imbalanced learning; LULC classification; Oversampling; Geometric-SMOTE; Class imbalance;

1. Introduction

The production of accurate Land Use/Land Cover (LULC) maps offers unique monitoring capabilities within the remote sensing domain [1]. LULC maps are being used for a variety of applications, ranging from environmental monitoring, land change detection, natural hazard assessment up to agriculture and water/wetland monitoring [2], therefore, accurate and timely production of LULC maps is of great significance. LULC maps are usually produced by two main procedures: photo-interpretation by the human eye, which is time and resource consuming and is not suitable for operational LULC-mapping over large areas, and second, automatic mapping using remotely sensed data and different classification algorithms.

The availability and a swift update of high-quality satellite remote sensing data has brought tremendous progress in providing up-to-date and accurate land cover information. Multispectral images, particularly, are an essential resource to build LULC maps, allowing for the use of classification algorithms to automate their production. Although significant progress has been made in the use of supervised learning techniques for automatic image classification [3], the acquisition of labeled training

sets continues to be a bottleneck [4]. In order to build accurate and robust supervised classifiers it is crucial to have a large enough training dataset. Often, the problem is that different land cover types have very different levels of area coverage, which causes some of them to be frequent in the training dataset, while others are limited [5].

A particular case where this phenomenon happens is the LUCAS dataset: Land Use and Coverage Area frame Survey coordinated by The Statistical Office of the European Commission (Eurostat) [6]. LUCAS surveys have been carried out every three-years since 2006 and are freely accessible For this statistical sampling survey a 2 km regular grid is implemented, and over 1,000,000 points were observed in the European Union territory for the year of 2015. Although, the LUCAS dataset is designed for statistical estimation, some existing studies are using this data for training machine learning classifiers for land cover classification successfully [7,8], since each of the observation is empirically registered in the field (in-situ). This sampling strategy is particularly interesting for this research, as it causes uneven representation of different land cover classes in the dataset for the given area.

The above-mentioned asymmetry in class distribution affects the performance of classifiers negatively. In the machine learning community, the problem is known as imbalanced learning problem [9]. The imbalanced learning problem generally refers to a skewed distribution of data across classes in both binary and multi-class problems [10].The latter, in particular, appears to be an even more challenging task [11]. In both cases, during the learning phase, the minority class(es) contribute less to the minimization of accuracy, the typical objective function, inducing a bias towards the majority class. Consequently, as typical classification algorithms are designed to work with reasonably balanced datasets, learning the decision boundaries between different classes becomes a very difficult task [12].

The possible approaches to deal with the class imbalance problem can be divided into three main groups [13]:

1. Cost-sensitive solutions. They introduce a cost matrix that applies higher misclassification costs for the examples of the minority class.
2. Algorithmic level solutions. They modify the algorithmic procedure to reinforce the learning of the minority class.
3. Resampling solutions. They rebalance the class distribution either by removing instances from the majority class or by generating artificial data for the minority class(es).

The latter method constitutes a more general approach since it can be used for any classification algorithm and it does not require any type of domain knowledge in order to construct a cost matrix.

There are several resampling solutions to deal with the imbalanced learning problem, which also can be divided into three categories:

1. Undersampling algorithms reduce the size of the majority class.
2. Oversampling algorithms attempt to even the distributions by generating artificial data for the minority class(es).
3. Hybrid approaches use both oversampling and undersampling techniques to ensure a balanced dataset.

In this paper, we compare the performance of various oversampling algorithms on the publicly available EUROSTAT's Land Use/Cover Area Statistical Survey (LUCAS) dataset [14] with Landsat 8 data. The experimental procedure includes a comparison of five oversamplers using five classifiers and three evaluation metrics. Specifically, the oversampling algorithms are Geometric SMOTE (G-SMOTE) [15], the Synthetic Minority Oversampling TEchnique (SMOTE) [16], Borderline SMOTE (B-SMOTE) [17], thr Adaptive Synthetic Sampling Technique (ADASYN) [18] and Random Oversampling (ROS), while no oversampling is included as a baseline method. Results show that G-SMOTE outperforms every other oversampling technique, for the selected evaluation metrics.

This paper is organized in five sections: Section 2 analyzes the resampling methods, section 3 describes the proposed methodology, section 4 shows the results and discussion and section 5 presents the conclusions drawn from this study.

2. Resampling methods

Data modification through resampling has been the most popular approach to deal with the imbalanced learning problems in machine learning in general and remote sensing in particular [5]. As mentioned above, by decoupling the imbalance problem from the classification algorithms, resampling allows the users to apply any standard algorithm once the resampling preprocessing step is done. This stratagem is especially convenient for users that are not machine learning experts and want to use several classifiers. Additionally, resampling methods can be naturally applied to multi-class imbalanced data, which is relevant for LULC classification. In this section, we present the most relevant applications of resampling methods for remote sensing imbalanced data classification.

2.1. Random resampling

Random resampling refers to non-informed strategies that remove instances from the majority class or replicate instances from the minority class. As such, the selection of the data occurs randomly without exploiting any additional information.

Some of the existing remote sensing studies implement the Random Undersampling (RUS) method [19], which randomly reduces the number of the majority class training samples. However, this method has the disadvantage of information loss as it discards samples from the majority class [5]. Contrary to RUS, ROS is a method that can be considered as equivalent to Bootstrapping, as it avoids information loss. However, ROS simply replicates randomly selected instances of the minority class, increasing the risk of overfitting [20]. [21] reports that balancing data with ROS affects the classification performance differently for various classifiers. In their paper, land cover classification with highly imbalanced data is carried out with six different models. The application of ROS slightly improved the performance of the Random Forest (RF) and Support Vector Machine (SVM) classifiers. On the other hand, it reduced the classification accuracy for classifiers such as Decision Tree (DT), Artificial Neural Network (ANN), K-Nearest Neighbors (KNN) and Boosted DT.

2.2. Informed resampling

In the above section, the disadvantages of RUS and ROS have been pointed out. Informed resampling methods aim to overcome these insufficiencies. More specifically, they use the local or global information of the class distribution to remove or generate instances. Our focus is on oversampling algorithms, since the size of the LUCAS dataset does not favor the use of undersampling approaches. Additionally, [22] carried out a comparative analysis of undersamplers' and oversamplers' performance for land cover classification with the Rotation Forest ensemble classifier, showing that oversampling methods outperform undersampling methods.

SMOTE is the most popular informed oversampling method, and it has been used to successfully deal with the class imbalance problem in land cover classification [23]. In this approach, the minority class is oversampled by randomly selecting a minority class instance and generating synthetic examples along the line segment joining it with one of its minority class neighbors. A number of studies report significant improvements in LULC mapping accuracy with the use of SMOTE oversampling. For instance, the variational semi-supervised learning (VSSL) proposed by [23] aims to deal with the imbalance problem in LULC mapping. VSSL is a semi-supervised learning framework consisting of a deep generative model. It allows learning successfully from both labeled and unlabeled samples while using SMOTE to balance the data. [24] used OpenStreetMap crowdsourced data and Landsat time series for LULC classification. Similarly, the application of SMOTE improved the classification results. Other examples of the successful application of SMOTE in remote sensing can be found in [25], [26].

Although recent studies demonstrate the usefulness of SMOTE for remote sensing applications, it still has some drawbacks. The SMOTE algorithm has the disadvantage of generating noisy data [27]. In order to mitigate this problem many variations of SMOTE have been developed. B-SMOTE is one of the most popular SMOTE-based oversamplers. Similarly to SMOTE, it uses the k- nearest neighbors selection strategy. The main difference to the original algorithm is that it modifies the data generation mechanism by generating samples closer to the decision boundary. B-SMOTE has also been reported to perform better than SMOTE in a number of studies [28,29]. ADASYN is another well-known variation of SMOTE. It is based on the idea of adaptively generating minority class instances according to their weighted distribution: more instances are generated for those minority class instances that are harder to learn compared to ones that are easier to learn [18].

The SMOTE algorithm can be decomposed into two parts: the selection strategy for the minority class instances and the data generation mechanism. The first part is related to the generation of noisy instances since the SMOTE selection strategy considers all the minority samples as equivalent. The above-mentioned SMOTE variations (B-SMOTE and ADASYN) aim to deal with this problem. On the other hand, the second part is responsible for the diversity of the artificial instances. There are scenarios where the linear interpolation mechanism used in SMOTE generates nearly duplicate instances that may lead to overfitting. The G- SMOTE algorithm is an extension of SMOTE that aims to deal with both problems. G-SMOTE defines a flexible geometric region around each minority class instance for synthetic data generation. The shape of this area is controlled by a set of hyperparameters. This element significantly increases the diversity of generated instances. Furthermore, G-SMOTE is designed to avoid noisy sample generation since it modifies the SMOTE selection strategy. G-SMOTE has been shown to outperform SMOTE and its above mentioned variations across 69 imbalanced datasets for various classifiers and evaluation metrics. Figure 1 depicts the data generation mechanism of both SMOTE and G-SMOTE using a deformed geometric region.

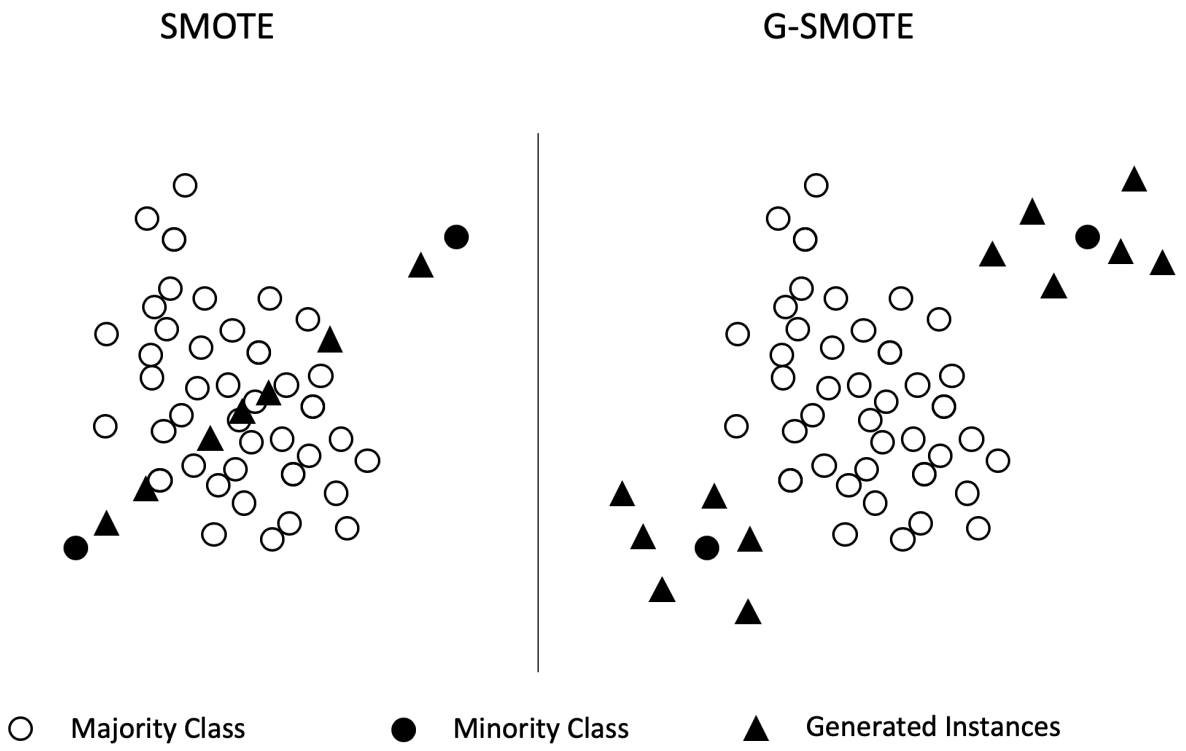

Figure 1. Example of minority class oversampled by SMOTE and G-SMOTE algorithms. G-SMOTE generates non-noisy samples with greater variety than SMOTE.

3. Methodology

This section describes the evaluation process of G-SMOTE's performance. A description of the study area, dataset, oversamplers, classifiers, evaluation metrics as well as the experimental procedure is provided. Figure 2 represents the flowchart of the steps applied in this experiment.

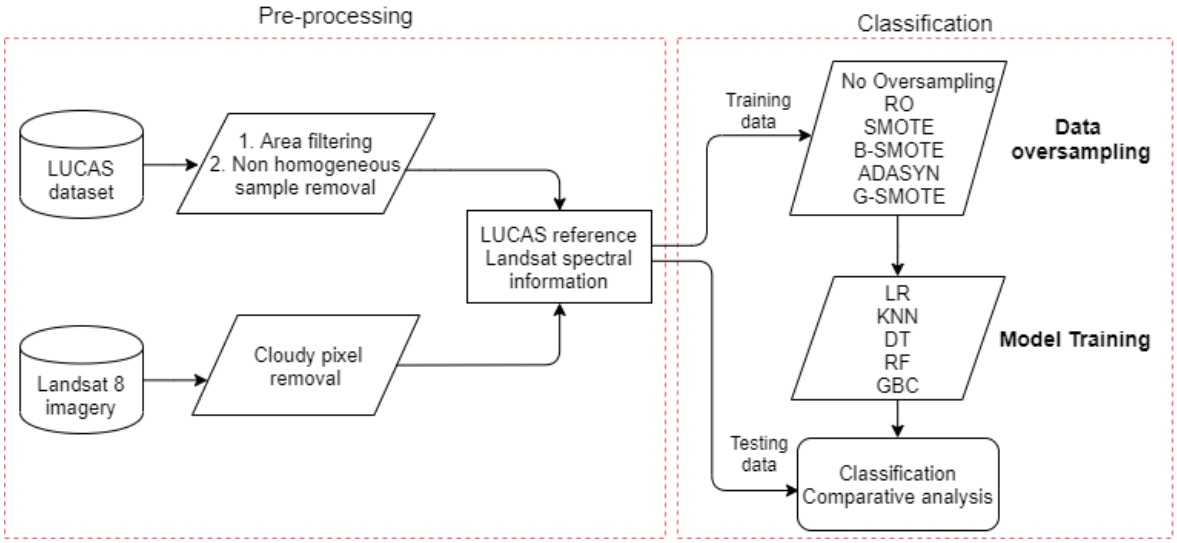

Figure 2. Flowchart containing the steps applied in the entire method.

3.1. Study area

The area of study is in north-western Portugal, corresponding to the area covered by the Landsat 8 image from track 204 and row 32, shown in figure 3. The area contains all eight main land cover types defined by LUCAS 2015: artificial land, cropland, woodland, shrubland, grassland, bare land, water and wetlands.

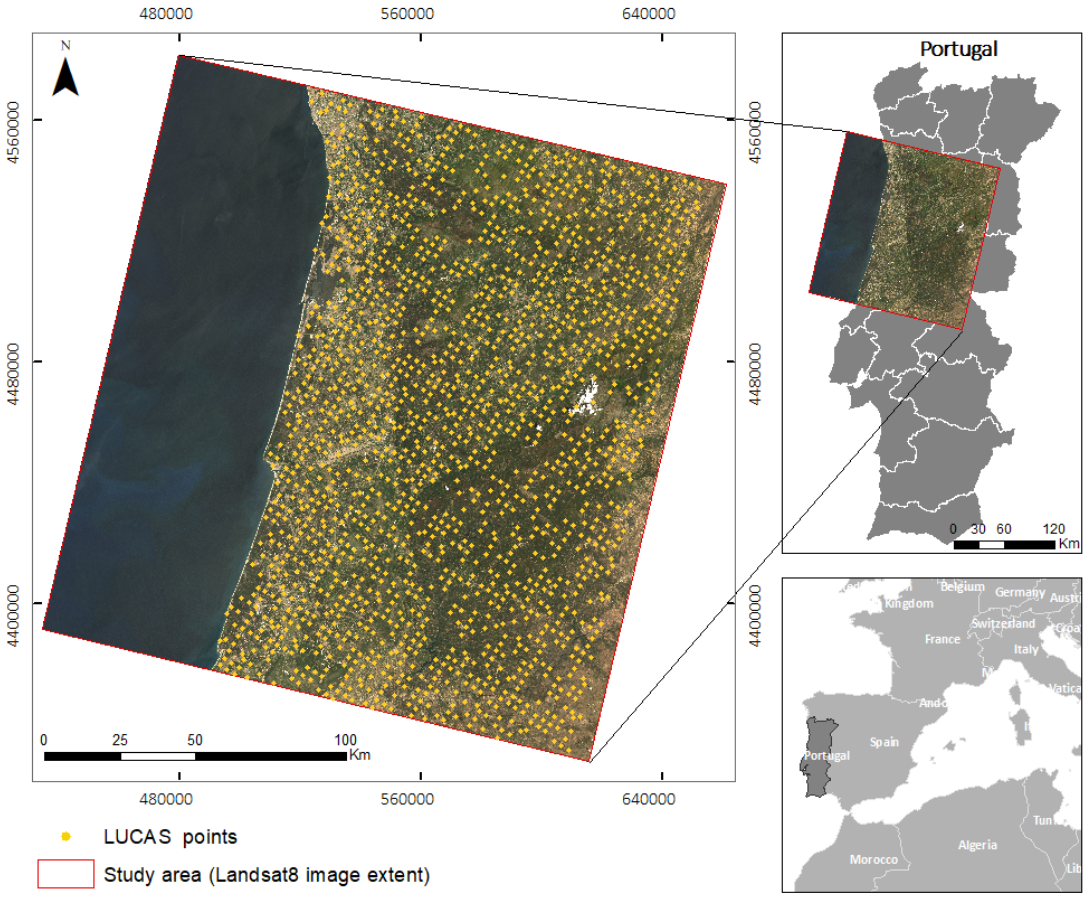

Figure 3. Study area and LUCAS reference data (coordinate system: WGS-84 UTM Zone 29, projection: Transverse-Mercator, Landsat image acquisition date: 2015/05/25).

3.2. Remote sensing data

The remotely sensed data includes eight images from the moderate-resolution Landsat 8 multi-spectral sensor. The images are Level-2 surface reflectance product (OLI/TIRS); One image was acquired each month from February to September 2015. The acquisition mode is Descending. Data were pre-processed in order to remove pixels with cloud cover. Only bands 2, 3, 4, 5, 6, and 7 are used from each image. Accordingly, each reference point from the LUCAS dataset has 48 features, representing pixel values from each spectral band from each image.

3.3. LUCAS dataset

The 2015 LUCAS data was used as reference data for both model training and validation. The LUCAS point label represents the corresponding land cover/use type within the radius of 1.5 m for homogeneous classes and a 20 m radius extent ("extended window") for heterogeneous classes (e.g., shrubland), gathered by field observation and a very high-resolution photo interpretation [6]. In order to reduce the risk of having Landsat pixel information represented wrongly in the field, we only kept points observed in-situ from a close distance (<100 m). With the same objective we removed the points which had linear features in the observation (e.g., roads). This procedure was not solely applied to the class of "artificial land," as this would remove most parts of the samples. Furthermore, points with cloudy pixels in the Landsat data were also excluded. This way, 1694 out of 2060 LUCAS points were retained. This dataset contains eight classes that represent the main land cover types for the study area

This pixel selection excluded a large number of unacceptable reference points, and we assumed the remaining ones to be suitable enough to represent the land cover type in a Landsat pixel coverage

area of 30x30 m. Further, we surmised that classifiers are capable of overcoming the noise caused by pixels having mixed land cover representation if such pixels are still available in the dataset.

The number of samples per class and the Imbalance Ratio (IR), defined as the ratio of the number of samples for the majority class over the number of samples for any of the minority classes, is presented in Table 1.

LUCAS Category	Land cover type	Instances	IR
A	Artificial land	131	5.81
B	Cropland	270	2.81
C	Woodland	761	1.00
D	Shrubland	296	2.61
E	Grassland	185	4.11
F	Bareland	37	20.56
G	Water	10	76.10
H	Wetlands	4	190.25

Table 1. LUCAS nomenclature and classes distribution.

Table 2 presents a description of the LUCAS dataset, including information about the majority class C and the smallest minority class H to emphasize the imbalance character of the dataset:

Dataset	LUCAS
Features	47
Instances	1694
Instances of class C	761
Instances of class H	4
IR of class H	190.25

Table 2. Description of the LUCAS dataset.

3.4. Evaluation metrics

Amongst the possible choices existing for classifier's performance evaluation, *Accuracy*, user's accuracy (or *Precision*) and producer's accuracy (or *Recall*) are the most common in LULC classification [30,31]. For a binary classification task, their calculation is given in terms of the true positives TP, true negatives TN, false positives FP, and false negatives FN [30]. More specifically, $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$. For the multi-class case, the average value across classes is used, as explained below.

The LUCAS dataset is highly imbalanced, having a wide range of IRs for the different minority classes. Therefore, the use of the metrics above is not an appropriate choice since they are mainly determined by the majority class contribution [32]. An appropriate evaluation metric should consider the classification accuracy of all classes. A simple approach for the multi-class case is to select a binary class evaluation metric, apply it to each binary sub-task of the multi-class problem, i.e., consider each class versus the rest and finally average its values. For this purpose, *F-score* and *G-mean* metrics are used as the primary evaluation methods while *Accuracy* is provided for discussion:

- The *Accuracy* is the number of correctly classified samples divided by the sum of all samples. Assuming that the various classes are labeled by the index c, *Accuracy* is given by the following formula:

$$Accuracy = \frac{\sum_c \mathrm{TP}_c}{\sum_c (\mathrm{TP}_c + \mathrm{FP}_c)}$$

– The *F-score* is the harmonic mean of *Precision* and *Recall*. The *F-score* for the multi-class case can be calculated using their average per class values [32]:

$$F\text{-}score = 2\frac{\overline{Precision} \times \overline{Recall}}{\overline{Precision} + \overline{Recall}}$$

– The *G-mean* is the geometric mean of *Sensitivity* and *Specificity*. *Sensitivity* is identical to the *Recall* while *Specificity* is given by the formula $Specificity = \frac{TN}{TN+FP}$. Therefore, they are equal to the true positive and true negative rates, respectively. The *G-mean* for the multi-class case can be calculated using their average per class values:

$$G\text{-}mean = \sqrt{\overline{Sensitivity} \times \overline{Specificity}}$$

3.5. Machine learning algorithms

The main objective of the paper is to show the effectiveness of G-SMOTE when it is used on multi-class highly imbalanced data of a remote sensing application as well as to compare its performance to other oversampling methods. Four oversampling algorithms were used in the experiment along with G-SMOTE. ROS was chosen for its simplicity. SMOTE was selected for being the most widely used oversampler. ADASYN and B-SMOTE were selected for representing popular modifications of the original SMOTE algorithm. Finally, no oversampling was also applied as an additional baseline method.

For the evaluation of the oversampling methods, the classifiers Logistic Regression (LR) [33], K-Nearest Neighbors (KNN) [34], Decision Tree (DT) [35], Gradient Boosting Classifier (GBC) [36] and Random Forest (RF) [37] were selected. The choice of classifiers was made according to the following criteria: learning type, training time, and popularity within the remote sensing community. All these algorithms were found to be computationally efficient and commonly used for the proposed task, with the exception of LR, which is rarely used in remote sensing applications [2,21].

3.6. Experiment settings

In order to evaluate the performance of each oversampler, every possible combination of oversampler, classifier, and metric is formed. The evaluation score for each of the above combinations is generated through an n-fold cross-validation procedure with $n = 3$. Before starting the training of each classifier, and in each stage $i \in \{1, 2, ..., n\}$ of the n-fold cross-validation procedure, synthetic data S_i were generated using the oversampler, based on the training data T_i of the $n - 1$ folds, such that the resulting $S_i \cup T_i$ training set becomes perfectly balanced. This enhanced training set, in turn, was used to train the classifier. The performance evaluation of the classifiers was done on the validation data V_i of the remaining fold, where $V_i \cup T_i = D$, $V_i \cap T_i = \emptyset$ while D represents the dataset. The process above is repeated three times and the results are averaged.

The range of hyperparameters used for each classifier and oversampler are presented in table 3:

Classifier	Hyperparameters	Values
LR	maximum iterations	10000
KNN	number of neighbors	3, 5
DT	maximum depth	3, 6
GBC	maximum depth	3, 6
	number of estimators	50, 100
RF	maximum depth	None, 3, 6
	number of estimators	50, 100
Oversampler		
G-SMOTE	number of neighbors	3, 5
	selection strategy	combined, minority, majority
	truncation factor	-1.0, -0.5, .0, 0.25, 0.5, 0.75, 1.0
	deformation factor	0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0
SMOTE	number of neighbors	3, 5
BORDERLINE SMOTE	number of neighbors	3, 5
ADASYN	number of neighbors	2, 3

Table 3. Hyperpameters grid.

3.7. Software implementation

The implementation of the experimental procedure was based on the Python programming language, using the Scikit-Learn [38], Imbalanced-Learn [39], and Geometric-SMOTE libraries. All functions, algorithms, experiments and results reported are provided at the GitHub repository of the project. Additionally, the Research-Learn library provides a framework to implement comparative experiments, also being fully integrated with the Scikit-Learn ecosystem.

4. Results and discussion

This section presents the results and analyses of oversamplers comparisons on the LUCAS dataset. The classification results are shown for all combinations of oversamplers and classifiers used in the experiment. The next subsection covers their interpretation in detail.

4.1. Results

For each combination of classifier and metric, a cross-validation score for all oversamplers is provided in Table 4:

Classifier	Metric	NONE	ROS	SMOTE	B-SMOTE	ADASYN	G-SMOTE
LR	Accuracy	**0.57**	0.50	0.50	0.53	0.48	0.51
LR	F-score	0.30	0.29	0.29	0.30	0.28	**0.31**
LR	G-mean	0.51	0.53	0.52	0.53	0.52	**0.57**
KNN	Accuracy	**0.56**	0.45	0.43	0.49	0.42	**0.56**
KNN	F-score	0.27	0.24	0.25	0.26	0.24	**0.28**
KNN	G-mean	**0.50**	0.48	0.49	**0.50**	0.48	**0.50**
DT	Accuracy	**0.51**	0.43	0.42	0.47	0.42	0.48
DT	F-score	0.24	0.24	0.25	**0.27**	0.25	**0.27**
DT	G-mean	0.49	0.48	0.49	0.51	0.49	**0.52**
GBC	Accuracy	**0.58**	0.56	0.56	0.57	0.55	0.57
GBC	F-score	0.31	0.31	0.31	0.31	0.31	**0.33**
GBC	G-mean	0.53	0.54	0.54	0.55	0.54	**0.56**
RF	Accuracy	**0.59**	0.58	0.56	0.57	0.55	0.58
RF	F-score	0.31	0.31	0.32	0.31	0.31	**0.34**
RF	G-mean	0.53	0.54	0.55	0.55	0.54	**0.57**

Table 4. Cross-validation scores of oversamplers.

A ranking score was assigned to each oversampling method with the best and worst performing methods receiving scores from 1 to 6, respectively. Table 5 presents the ranking scores per classifier and evaluation metric:

Classifier	Metric	NONE	ROS	SMOTE	B-SMOTE	ADASYN	G-SMOTE
LR	Accuracy	**1**	4	5	2	6	3
LR	F-score	3	4	5	2	6	**1**
LR	G-mean	6	3	4	2	5	**1**
KNN	Accuracy	**1**	4	5	3	6	2
KNN	F-score	2	6	4	3	5	**1**
KNN	G-mean	3	6	4	2	5	**1**
DT	Accuracy	**1**	4	5	3	6	2
DT	F-score	5	6	4	**1**	3	2
DT	G-mean	5	6	4	2	3	**1**
GBC	Accuracy	**1**	4	5	3	6	2
GBC	F-score	3	5	4	2	6	**1**
GBC	G-mean	6	5	3	2	4	**1**
RF	Accuracy	**1**	3	5	4	6	2
RF	F-score	6	5	2	3	4	**1**
RF	G-mean	6	5	3	2	4	**1**

Table 5. Ranking of oversamplers.

The percentage difference between G-SMOTE and NONE, ROS and SMOTE respectively for every combination of metric and classifier is calculated from the following formula:

$$\text{Percentage Difference} = 100 \times \frac{Score(G-SMOTE) - Score(Oversampler)}{Score(Oversampler)}$$

.

For each combination of an oversampler, classifier and metric, a positive (negative) value of the above formula indicates the G-SMOTE's relative performance gain (loss) compared to the oversampler. Table 6 presents the results of the above calculation:

Classifier	Metric	NONE	ROS	SMOTE
LR	Accuracy	-11.96	1.34	2.11
LR	F-score	5.74	6.79	8.48
LR	G-mean	10.19	7.0	7.85
KNN	Accuracy	-0.11	24.98	30.93
KNN	F-score	1.98	15.24	12.92
KNN	G-mean	1.51	5.46	3.5
DT	Accuracy	-6.93	10.95	14.23
DT	F-score	10.02	10.18	6.98
DT	G-mean	6.36	7.48	6.01
GBC	Accuracy	-1.75	2.39	2.46
GBC	F-score	5.26	6.26	5.3
GBC	G-mean	5.07	4.08	3.49
RF	Accuracy	-1.34	0.55	4.03
RF	F-score	11.65	8.88	7.6
RF	G-mean	8.36	5.63	4.98

Table 6. Percentage difference between G-SMOTE and other popular methods.

Wilcoxon signed-rank test is used as an alternative to the paired Student's t- test when the distribution of the differences between the two samples cannot be assumed to be normally distributed. In our case, it is applied to test the null hypothesis that the pairwise difference between G-SMOTE's scores and the scores of the remaining oversampling methods follows a symmetric distribution around zero, i.e. G-SMOTE performs similarly to them. The values for the *Accuracy* metric are excluded in the NONE case while for the remaining oversampling methods all metrics are used. This choice will be justified in the next section. Table 7 presents the p-values for the Wilcoxon tests:

Oversampler	p-value	Significance
NONE	5.1e-03	True
ROS	6.5e-04	True
SMOTE	6.5e-04	True
B-SMOTE	9.0e-03	True
ADASYN	6.5e-04	True

Table 7. Wilcoxon test.

4.2. Discussion

From table 4, we can observe that G-SMOTE outperforms all other oversampling methods for both *F-score* and *G-mean* metrics on all classifiers. The absolute best results are achieved when G-SMOTE is combined with LR and RF. It is vital to notice that the *Accuracy* scores show the well-known bias towards the majority class as discussed in 3.4. In a multi-class classification problem with an imbalanced dataset, where the prediction of all the classes are of equal importance as in many remote sensing applications, *Accuracy* should be of secondary importance compared to more robust metrics, such as *F- score* and *G-mean*. Nevertheless, even for the *Accuracy* metric, G-SMOTE shows the best performance amongst the oversamplers.

In table 5, the rankings of the oversamplers are presented and show the superiority of G-SMOTE. Although ROS and SMOTE are the most popular oversampling methods in remote sensing applications, it is clear from the tables that they produce suboptimal results. Table 6 directly compares the performance of G-SMOTE with ROS and SMOTE, including also NONE as a baseline method.

Table 7, provides a statistical confirmation of the previous conclusions. Using the Wilcoxon signed-rank test, the null hypothesis that the pairwise difference of scores between G-SMOTE and any of the remaining oversampling methods follows a symmetric distribution around zero is rejected at a significance level of $\alpha = 0.01$.

This study is the first to present a systematic comparison of oversampling algorithms in remote sensing. However, several previous studies reported results consistent with our findings. [25] reported an increase in *F-score* and *G-mean* when oversampling was applied, while classification overall accuracy did not improve. Similarly, results obtained in [5] demonstrate increased classification performance when using SMOTE. According to our experiment, performance can be further increased by using G-SMOTE. A number of other studies [21,23] do not use imbalance specific metrics, therefore it cannot be directly compared to our results.

5. Conclusions

In this paper we applied G-SMOTE, a novel oversampling algorithm, on a LULC classification problem, using a highly imbalanced multi-class dataset (LUCAS). G-SMOTE's performance was evaluated and compared with other oversampling methods. More specifically, ROS, SMOTE, B-SMOTE and ADASYN were the selected oversamplers while LR, KNN, DT, GBC and RF were used as classifiers.

The experimental results show that using a G-SMOTE oversampler can significantly improve the classification performance, resulting in higher values of *F- score* and *G-mean*. Therefore, readers should consider using G-SMOTE when accurately predicting the minority classes is of equal or higher

importance compared to the accurate prediction of the majority class. Examples of the above case are the land cover change detections and rare land cover type classification.

G-SMOTE can be a useful tool for remote sensing researchers and practitioners, as it systematically outperforms the previous widely used oversamplers. G-SMOTE is easily accessible to the users through an open source implementation.

Author Contributions: Conceptualization, F.B.; Methodology, G.D.; Software, G.D.; Validation, F.B., G.D.; Formal Analysis, J.F and M.K.; Writing - Original Draft Preparation, M.K., J.F.; Writing - Review & Editing, F.B., G.D., J.F., M.K.; Supervision, F.B.; Funding Acquisition, F.B.

Funding: This research was funded by "Fundação para a Ciência e Tecnologia" (Portugal), grants' number PCIF/SSI/0102/2017 and DSAIPA/AI/0100/2018 - IPSTERS.

Acknowledgments: The authors would like to thank Direção Geral do Território (DGT) for supporting the data used in this study.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

OS	Oversampling
CV	Cross-Validation
LULC	Land Use/Land Cover
LUCAS	Land Use/Cover Area Statistical Survey
SMOTE	Synthetic Minority Over-sampling Technique
ADASYN	Adaptive Synthetic Sampling Technique
G-SMOTE	Geometric Synthetic Minority Over-sampling Technique
B-SMOTE	Borderline Synthetic Minority Over-sampling Technique
ROS	Random Oversampling
NONE	No Oversampling
LR	Logistic Regression
KNN	K-Nearest Neighbors
DT	Decision Trees
GBC	Gradient Boosting Classifier
RF	Random Forest

References

1. Mellor, A.; Boukir, S.; Haywood, A.; Jones, S. Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin. *ISPRS Journal of Photogrammetry and Remote Sensing* **2015**, *105*, 155–168. doi:10.1016/J.ISPRSJPRS.2015.03.014.
2. Khatami, R.; Mountrakis, G.; Stehman, S.V. A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. *Remote Sensing of Environment* **2016**, *177*, 89–100. doi:10.1016/J.RSE.2016.02.028.
3. Tewkesbury, A.P.; Comber, A.J.; Tate, N.J.; Lamb, A.; Fisher, P.F. A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sensing of Environment* **2015**, *160*, 1–14. doi:10.1016/J.RSE.2015.01.006.
4. Rajan, S.; Ghosh, J.; Crawford, M. An Active Learning Approach to Hyperspectral Data Classification. *IEEE Transactions on Geoscience and Remote Sensing* **2008**, *46*, 1231–1242. doi:10.1109/TGRS.2007.910220.
5. Feng, W.; Huang, W.; Bao, W. Imbalanced Hyperspectral Image Classification With an Adaptive Ensemble Method Based on SMOTE and Rotation Forest With Differentiated Sampling Rates. *IEEE Geoscience and Remote Sensing Letters* **2019**, pp. 1–5. doi:10.1109/LGRS.2019.2913387.
6. LUCAS, E.E. LUCAS 2015 (Land Use / Cover Area Frame Survey) **2015**. *Technical reference document C1. Instructions for Surveyors*, 140.

7. Pflugmacher, D.; Rabe, A.; Peters, M.; Hostert, P. Mapping pan-European land cover using Landsat spectral-temporal metrics and the European LUCAS survey. *Remote Sensing of Environment* **2019**, *221*, 583–595. doi:10.1016/j.rse.2018.12.001.
8. Mack, B.; Leinenkugel, P.; Kuenzer, C.; Dech, S. A semi-automated approach for the generation of a new land use and land cover product for Germany based on Landsat time-series and Lucas in-situ data. *Remote Sensing Letters* **2017**, *8*, 244–253. doi:10.1080/2150704X.2016.1249299.
9. Chawla, N.V.; Japkowicz, N.; Kotcz, A. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter* **2004**, *6*, 1. doi:10.1145/1007730.1007733.
10. Abdi, L.; Hashemi, S. To Combat Multi-Class Imbalanced Problems by Means of Over-Sampling Techniques. *IEEE Transactions on Knowledge and Data Engineering* **2016**, *28*, 238–251. doi:10.1109/TKDE.2015.2458858.
11. García, S.; Zhang, Z.L.; Altalhi, A.; Alshomrani, S.; Herrera, F. Dynamic ensemble selection for multi-class imbalanced datasets. *Information Sciences* **2018**, *445-446*, 22–37. doi:10.1016/J.INS.2018.03.002.
12. Sáez, J.A.; Krawczyk, B.; Woźniak, M. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition* **2016**, *57*, 164–178. doi:10.1016/J.PATCOG.2016.03.012.
13. Fernández, A.; López, V.; Galar, M.; del Jesus, M.J.; Herrera, F. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems* **2013**, *42*, 97–110. doi:10.1016/J.KNOSYS.2013.01.018.
14. LUCAS, E.E. LUCAS 2015 (Land Use / Cover Area Frame Survey) **2015**. *Technical reference document C3 Classification (Land cover and Land use)*, 93.
15. Douzas, G.; Bacao, F. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information Sciences* **2019**, *501*, 118–135. doi:10.1016/J.INS.2019.06.007.
16. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **2002**, *16*, 321–357. doi:10.1613/jair.953.
17. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. International Conference on Intelligent Computing. Springer, Berlin, Heidelberg, 2005, pp. 878–887. doi:10.1007/11538059_91.
18. Haibo He.; Yang Bai.; Garcia, E.A.; Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE, 2008, pp. 1322–1328. doi:10.1109/IJCNN.2008.4633969.
19. Azadbakht, M.; Fraser, C.; Khoshelham, K. Improved urban scene classification using full-waveform LiDAR. *Photogrammetric Engineering & Remote Sensing* **2016**, *82*, 973–980.
20. Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence* **2016**, *5*, 221–232.
21. Maxwell, A.E.; Warner, T.A.; Fang, F. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing* **2018**, *39*, 2784–2817.
22. Feng, W.; Huang, W.; Ye, H.; Zhao, L. Synthetic Minority Over-Sampling Technique Based Rotation Forest for the Classification of Unbalanced Hyperspectral Data. IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2018, pp. 2651–2654.
23. Cenggoro, T.W.; Isa, S.M.; Kusuma, G.P.; Pardamean, B. Classification of imbalanced land-use/land-cover data using variational semi-supervised learning. Proceedings - 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech 2017. IEEE, 2018, Vol. 2018-January, pp. 1–6. doi:10.1109/INNOCIT.2017.8319149.
24. Johnson, B.A.; Iizuka, K. Integrating OpenStreetMap crowdsourced data and Landsat time-series imagery for rapid land use/land cover (LULC) mapping: Case study of the Laguna de Bay area of the Philippines. *Applied Geography* **2016**, *67*, 140–149.
25. Bogner, C.; Seo, B.; Rohner, D.; Reineking, B. Classification of rare land cover types: Distinguishing annual and perennial crops in an agricultural catchment in South Korea. *PloS one* **2018**, *13*, e0190476.
26. Panda, A.; Singh, A.; Kumar, K.; Kumar, A.; Swetapadma, A.; others. Land Cover Prediction from Satellite Imagery Using Machine Learning Techniques. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE, 2018, pp. 1403–1407.
27. Douzas, G.; Bacao, F. Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning. *Expert Systems with Applications* **2017**, *82*, 40 – 52. doi:https://doi.org/10.1016/j.eswa.2017.03.073.

28. Nguyen, H.M.; Cooper, E.W.; Kamei, K. Borderline over-sampling for imbalanced data classification **2009**. *2009*, 24–29.
29. Ramentol, E.; Caballero, Y.; Bello, R.; Herrera, F. SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and information systems* **2012**, *33*, 245–265.
30. Liu, C.; Frazier, P.; Kumar, L. Comparative assessment of the measures of thematic classification accuracy. *Remote Sensing of Environment* **2007**, *107*, 606–616. doi:10.1016/j.rse.2006.10.010.
31. Olofsson, P.; Foody, G.M.; Stehman, S.V.; Woodcock, C.E. Making better use of accuracy data in land change studies: Estimating accuracy and area and quantifying uncertainty using stratified estimation. *Remote Sensing of Environment* **2013**, *129*, 122–131. doi:10.1016/j.rse.2012.10.031.
32. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* **2009**, *21*, 1263–1284, [arXiv:1011.1669v3]. doi:10.1109/TKDE.2008.239.
33. McCullagh, P.; Nelder, J. *Generalized Linear Models*; 1989; p. 532, [arXiv:1011.1669v3]. doi:10.1007/978-1-4899-3242-6.
34. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **1967**, *13*, 21–27. doi:10.1109/TIT.1967.1053964.
35. Salzberg, S.L. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning* **1994**, *16*, 235–240. doi:10.1007/BF00993309.
36. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* **2001**, *29*, 1189–1232, [arXiv:1011.1669v3]. doi:DOI 10.1214/aos/1013203451.
37. Liaw, A.; Wiener, M.; others. Classification and regression by randomForest. *R news* **2002**, *2*, 18–22.
38. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
39. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* **2017**, *18*, 1–5.