# Chapter 15
# SMOTE-Cov: A New Oversampling Method Based on the Covariance Matrix

**Ireimis Leguen-deVarona, Julio Madera, Yoan Martínez-López, and José Carlos Hernández-Nieto**

**Contents**



## 15.1 Introduction

Real-world data often present characteristics that affect classification: noise, missing values, inexact or incorrect values, inadequate data size, poor representation in data sampling, etc. The imbalanced dataset problem represents a field of interest as it occurs when the number of instances that represent one class(rare events) [1] is much larger than the other classes, a common problem in certain areas such as fraud detection, cancer gene expressions, natural disasters, software defects, and risk management [2]. Rare events are difficult to detect because of their infrequency and casualness; misclassification of rare events could often result in heavy costs. For example, for smart computer security threat detection [3], dangerous connection attempts may only appear out of hundreds of thousands log records, but failing to identify a serious vulnerability breach would cause enormous losses.

Then, in the case of the datasets with binary class, it can be defined that it is balanced if it has an approximately equal percentage of examples in the concepts to be classified, that is, if the distribution of examples by classes is uniform, otherwise it is unbalanced. To measure the degree of imbalance of a problem, [4] defined the imbalance ratio (IR) as

I. Leguen-deVarona · J. Madera (✉) · Y. Martínez-López · J. C. Hernández-Nieto
University of Camagüey, Camagüey, Cuba
e-mail: ireimis.leguen@reduc.edu.cu; julio.madera@reduc.edu.cu; yoan.martinez@reduc.edu.cu; jose.hernandez@reduc.edu.cu



P. Vasant et al. (eds.), *Data Analysis and Optimization for Engineering and Computing Problems*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-48149-0_15

$$IR = \frac{|C+|}{|C-|} \geq 1.5 \tag{15.1}$$

where

$C+$ is the number of instances that belongs to the majority class
$C-$ is the number of instances that belongs to the minority class

Therefore, a dataset is imbalanced when it has a marked difference (IR $\geq$ 1.5) between the examples of the classes. This difference causes low predictive accuracy for the infrequent class as classifiers try to reduce the global error without taking into account the distribution of the data. In imbalanced sets, the original knowledge is usually labeled as oddities or noise, focusing exclusively on global measurements [5]. The problem with the imbalance is not only the disproportion of representatives but also the high overlap between the classes. To overcome this problem, diverse strategies have been developed and can be divided into four groups: at the data level [6, 7], at the learning algorithm level [8], cost-sensitive learning [9], and based on multiple classifiers [10]; where the techniques at the data level are the most used because its use is independent of the classifier that is selected.

One of the best-known algorithms within data-level techniques is the Synthetic Minority Oversampling Technique (SMOTE) [7, 11] for the generation of synthetic instances. One of SMOTE's shortcomings is that it generalizes the minority area without regard to the majority class leading to a problem commonly known as overgeneralization; this has been solved with the use of cleaning methods such as SMOTE—Tomek links (TL) [6, 11], SMOTE—ENN [6, 11], Borderline—SMOTE1 [11, 12], SPIDER [13], SMOTE—RSB* [14], ADASYN [6], among others. These algorithms have been designed to operate with values of both discrete and continuous features for problems with imbalances in their two classes; most of them use the KNN to obtain the synthetic instances, and although this is a method that offers good results, it does not take into account the dependency relationships between attributes, which can influence the correct classification of the examples of the minority class.

A way to obtain the dependency relation of the attributes is Probabilistic Graphical Models (PGMs) [15] that represent joint probability distributions where nodes are random variables and arcs are conditional dependence relationships. Generally, PGMs have four fundamental components: semantics, structure, implementation, and parameters. As part of the PGMs, there are Gaussian Networks that are graphic interaction models for the multivariate normal distribution [16], and some use the Covariance Matrix (CM) to analyze the relationships between variables.

This chapter proposes an algorithm based on SMOTE and the Covariance Matrix estimation to balance datasets with continuous attributes and binary class, exploding the dependency relationships between attributes and obtaining AUC [17] values similar to the algorithms of the state-of-the-art.

An experimental study was performed ranking two SMOTE-Cov variants, SMOTE-CovI (which generates new values within the interval of each attribute) and SMOTE-CovO (which allows some values to be outside the interval of the

attributes), against SMOTE, SMOTE-ENN, SMOTE-Tomek Links, Borderline-SMOTE, ADASYN, SMOTE-RSB*, and SPIDER, using 7 datasets from the UCI repository [18] with different imbalance ratios and using C4.5 as a classifier. The performance of the classifier was evaluated using AUC and hypothesis testing techniques as proposed by [19, 20] for statistical analysis of the results.

## 15.2 Oversampling Based on the Covariance Matrix

This section introduces oversampling based on the Covariance Matrix. First, we describe the Covariance Matrix that allows the computation of variable dependency. Then, we give an overview of our proposed algorithm. Finally, we describe our experimental setup in four steps: tool, dataset selection, evaluation methodology, and classifier used.

### *Covariance Matrix*

In statistics and probability theory, the covariance matrix is a matrix that contains the covariance between the elements of a vector, where it measures the linear relationship between two variables. If the vector-column entries are

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \tag{15.2}$$

then the covariance matrix $\sum_{ij}$ is the matrix, whose $(i, j)$ entry is the covariance

$$\sum_{ij} = \mathbf{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right] \tag{15.3}$$

where the operator $\mathbf{E}$ denotes the expected value (mean) of its argument

$$\mu_i = \mathbf{E}(X_i) \tag{15.4}$$

The Covariance Matrix allows determining if there is a dependency relationship between the variables and it is also the data necessary to estimate other parameters. In addition, it is the natural generalization to higher dimensions of the concept of the variance of a scalar random variable [20].

### *SMOTE-Cov*

The Algorithm 5 shows the steps of SMOTE-Cov to balance datasets. During the loading of the dataset in the first step, the algorithm expects continuous valued attributes and a binary class. Then, it uses the formula 1 to verify whether the dataset is balanced or not. If it is imbalanced, the algorithm computes the Covariance Matrix. The Covariance Matrix allows the detection of the dependency relationship between attributes. Then, from the estimated covariance matrix, new synthetic instances are generated to balance the minority class. This process stops when an equilibrium between the two classes is reached. The algorithm checks that all the new values generated from the covariance are obligatorily within the interval of each attribute, in the case that some are outside the interval, what is done is to take it to the minimum or maximum, making a kind of REPAIR of the value.

**Algorithm 5:** SMOTE-Cov steps

```
Input: Dataset X,inRange[Boolean]
Output: Balanced dataset X
Data: Dataset X
Step 1: Load dataset X;
Step 2: Compute X IR using Eq. 15.1;
if IR ≥ 1.5 then
    Step 3: Estimate covariance matrix using Eq. 15.3, this will provide us with a
    probabilistic distribution of the dataset;
    Step 4: For each attribute, a range is determined by it min-max value;
    while X is not in equilibrium do
        Step 5: Generate new instance y according to the covariance matrix;
        if range ≠ true then
            add y to X;
        else
            for i ← 0 to Y_i do
                if Y_i < min Y_i then
                    Y_i = min Y_i;
                else if Y_i > max Y_i then
                    Y_i = max Y_i;
                else
                    continue;
                end
            end
        end
    end
else
    return X;
end
```

## 15.3 Tools and Experimental Setup

The algorithm was developed using the R language because it is designed for statistical processing and has the cov() function for calculating the covariance. In order to evaluate the behavior of the proposed algorithm, it was compared against the state-of-the-art algorithms of oversampling data balancing; two variants are taken into account: when the attributes are inside or outside of the dependence range. Seven datasets from the UCI repository were chosen with IR $\geq 1.5$, see Table 15.1, with continuous attributes and binary class. This experiment uses fivefold cross-validation, and the data are split into two subsets: the training/calibration set (80%) and the test set (20%). The final result is the mean of the 5 result sets. The partitions were made using KEEL in such a way that the number of instances per class remained uniform. The partitioned datasets are available on the KEEL website [21].

The training datasets are balanced, generating new synthetic instances from the minority class to complete the quantities of the majority class and using a sample of the control test, which remains imbalanced and without any modification. The new datasets are generated from the obtained instances, using the SMOTE-Cov algorithm, and a classifier is used as a mean to measure the performance using other techniques.

The classifier used for the experimental study is C4.5 (implemented in the Weka package as J48) [22], which has been referred to as a statistical classifier and one of the top 10 algorithms in Data Mining that is widely used in imbalance problems [14].

The Area Under the Curve (AUC) (15.5) is used to measure the performance of classifiers over imbalanced datasets using the graph of the Receiver Operating Characteristic (ROC) [17]. In these graphics, the trade-off between the benefits (TPrate) and cost(FPrate) can be visualized, which represent the fact that the capacity of any classifier cannot increase the number of true positives without also increasing the false positives. AUC summarizes the performance of the learning algorithm in a number.

$$AUC = \frac{1 + TPrate - FPrate}{2} \tag{15.5}$$

**Table 15.1** Description of the datasets used in the experiments

| Dataset | Instances | Attributes | IR |
|---|---|---|---|
| ecoli2 | 336 | 7 | 5.4 |
| glass-0-1-2-3_vs_4-5-6 | 274 | 9 | 3.20 |
| glass1 | 214 | 9 | 1.81 |
| Iris | 150 | 4 | 2 |
| newthyroid2 | 215 | 5 | 5.14 |
| Pima | 768 | 8 | 1.86 |
| vehicle3 | 846 | 18 | 2.99 |

where

$TPrate$ are the correctly classified positive cases that belong to the positive class
$FPrate$ are the negative cases that were misclassified as positive examples

## *Experimental Study*

The AUC result values are studied with this already balanced dataset. Table 15.2 shows that the AUC results of the data-balancing algorithm applying the Covariance Matrix with its CovI and CovO variants are similar or comparable with respect to the state-of-the-art oversampling algorithms, using C4.5 as a classifier.

For the statistical analysis of the results, hypothesis-testing techniques were used [19, 20]. In both experiments, the Friedman and Iman-Davenport tests were used [23], in order to detect statistically significant differences between groups of results. The Holms test was also carried out [24] with the aim of finding significantly higher algorithms. These tests are suggested in the studies presented in [19, 20, 23], where it is stated that the use of these tests is highly recommended for the validation of results in the field of automated learning. Table 15.3 shows the ranking obtained by the Friedman test for the experiment. Although the algorithm with the best ranking was ADASYN, Holm's test performed below will demonstrate to what extent this algorithm can be significantly superior to the one proposed in the research.

Table 15.4 summarizes the results of Holms test, taking ADASYN as a control method, all hypotheses with $p$–values $\leq 0.05$ are rejected, showing that ADASYN is significantly superior to the SMOTE-CovI and Borderline-SMOTE algorithms.

**Table 15.2** AUC of the data balancing algorithms with the generation of oversampling classes of the state-of-the-art, CovI and CovO

| Algorithms | Iris | glass1 | Pima | vehicle3 | glass-0-1-2-3_vs_4-5-6 | ecoli2 | new thyroid2 |
|---|---|---|---|---|---|---|---|
| ADASYN | **1** | 0.74 | 0.73 | 0.74 | 0.88 | 0.91 | **0.98** |
| Borderline–SMOTE | 0.99 | **0.77** | 0.70 | 0.65 | 0.82 | 0.89 | 0.95 |
| SMOTE–ENN | 0.99 | 0.74 | 0.74 | 0.71 | **0.93** | 0.89 | 0.92 |
| SMOTE–RSB | 0.97 | 0.72 | **0.75** | 0.73 | 0.90 | 0.89 | 0.96 |
| SMOTE–TL | 0.99 | 0.74 | 0.72 | **0.79** | 0.90 | 0.89 | 0.93 |
| SMOTE | **1** | **0.77** | 0.74 | 0.72 | 0.84 | **0.92** | 0.92 |
| SPIDER | 0.99 | 0.74 | 0.72 | 0.71 | 0.92 | 0.89 | 0.95 |
| Original | **1** | 0.72 | **0.75** | 0.72 | 0.90 | 0.85 | 0.96 |
| SMOTE–CovO | **1** | 0.71 | 0.72 | 0.71 | 0.92 | 0.86 | 0.95 |
| SMOTE–CovI | 0.95 | 0.72 | 0.70 | 0.72 | 0.86 | 0.86 | 0.96 |

The bold values represent the best AUC obtained by each algorithm for each dataset

**Table 15.3** Friedman's test

| Algorithms | Ranking |
|---|---|
| ADASYN | 3.4286 |
| Borderline–SMOTE | 6.9286 |
| SMOTE–ENN | 5.4286 |
| SMOTE–RSB | 4.9286 |
| SMOTE–TL | 5.2857 |
| SMOTE | 4.5714 |
| SPIDER | 5.6429 |
| Original | 5 |
| SMOTE–CovO | 6.3571 |
| SMOTE–CovI | 7.4286 |

**Table 15.4** Holms test with $\alpha = 0.05$, taking ADASYN as a control method

| i | Algorithms | $Z = \frac{(R_o - R_i)}{SE}$ | $p$-value | Holm | Hypothesis |
|---|---|---|---|---|---|
| 9 | SMOTE–CovI | 2.47 | 0.01 | 0.005 | Reject |
| 8 | Borderline–SMOTE | 2.16 | 0.03 | 0.006 | Reject |
| 7 | SMOTE–CovO | 1.80 | 0.07 | 0.007 | Accept |
| 6 | SPIDER | 1.36 | 0.17 | 0.008 | Accept |
| 5 | SMOTE–ENN | 1.23 | 0.21 | 0.01 | Accept |
| 4 | SMOTE–TL | 1.14 | 0.25 | 0.012 | Accept |
| 3 | Original | 0.97 | 0.33 | 0.01 | Accept |
| 2 | SMOTE–RSB | 0.92 | 0.35 | 0.02 | Accept |
| 1 | SMOTE | 0.70 | 0.48 | 0.05 | Accept |

In the case of SMOTE-CovO, SPIDER, SMOTE_ENN, SMOTE_TL, Original, SMOTE-RSB and SMOTE, the null hypothesis is accepted, which means that there are no significant differences between ADASYN and them, so it can be concluded that they are as effective.

## 15.4 Conclusions and Future Work

In this chapter, a new algorithm is proposed to generate synthetic instances of the minority class, using the Covariance Matrix. The experimental study carried out shows the effectiveness of the proposed algorithm compared to eight recognized state-of-the-art algorithms. SMOTE-Cov showed similar or comparable results, taking into account the results of the AUC curve of the C4.5 classifier and using nonparametric tests to demonstrate that there are no significant differences between them, with the exception of the ADASYN versus the SMOTE-CovI variant. This can be influenced because the attributes present in the studied datasets come from other intervals and not from the actual attribute within the dataset.

Having results comparable to those of the state-of-the-art, these datasets allow extending the experimentation in the future to datasets with tens, hundreds or thousands of attributes and with strong dependency relationships. It is also intended to use covariance regularization (Shrinkage) to balance data, where the number of positive instances is less than the number of attributes.

**Acknowledgments** We would like to thank VLIR (Vlaamse Inter Universitaire Raad, Flemish Interuniversity Council, Belgium) for supporting this work under the Networks 2019 Phase 2 Cuba ICT and the Play! for food: Improving food production and social welfare in the province of Camagüey project.

## References

1. M. Maalouf, T.B. Trafalis, Robust weighted kernel logistic regression in imbalanced and rare events data. Comput. Stat. Data Anal. **55**(1), 168–183 (2011)
2. G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: review of methods and applications. Exp. Syst. Appl. **73**, 220–239 (2017)
3. K. Alzhrani, E.M. Rudd, C.E. Chow, T.E. Boult, Automated big security text pruning and classification, in *2016 IEEE International Conference on Big Data (Big Data)* (IEEE, Piscataway, 2016), pp. 3629–3637
4. A. Fernández, S. García, M.J. del Jesus, F. Herrera, A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets Syst. **159**(18), 2378–2398 (2008)
5. N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, Smoteboost: improving prediction of the minority class in boosting, in *European Conference on Principles of Data Mining and Knowledge Discovery* (Springer, Berlin, 2003), pp. 107–119
6. G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsl. **6**(1), 20–29 (2004)
7. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
8. Y.-M. Huang, C.-M. Hung, H.C. Jiau, Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. Nonlinear Anal. Real World Appl. **7**(4), 720–747 (2006)
9. Z.-H. Zhou, X.-Y. Liu, On multi-class cost-sensitive learning. Comput. Intell. **26**(3), 232–257 (2010)
10. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. C **42**(4), 463–484 (2011)
11. A. More, Survey of resampling techniques for improving classification performance in unbalanced datasets. (2016, preprint). arXiv:1608.06048
12. H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in *International Conference on Intelligent Computing* (Springer, Berlin, 2005), pp. 878–887
13. J. Stefanowski, S. Wilk, Selective pre-processing of imbalanced data for improving classification performance, in *International Conference on Data Warehousing and Knowledge Discovery* (Springer, Berlin, 2008), pp. 283–292
14. E.R. Martínez, F. Herrera, R.B. Pérez, Y.C. Mota, Y.S. López, Edición de conjuntos de entrenamiento no balanceados, haciendo uso de operadores genéticos y la teoría de los conjuntos aproximados

15. D. Koller, N. Friedman, *Probabilistic Graphical Models: Principles and Techniques* (MIT Press, Cambridge, 2009)
16. R.M. Sanner, J.-J.E. Slotine, Gaussian networks for direct adaptive control. IEEE Trans. Neural Netw. **3**(6), 837–863 (1992)
17. A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognit. **30**(7), 1145–1159 (1997)
18. UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/index.php. Accessed 14 Feb 2019
19. S. García, F. Herrera, Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol. Comput. **17**(3), 275–306 (2009)
20. D.J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures* (CRC Press, Boca Raton, 2003)
21. KEEL-dataset repository, http://www.keel.es/. Accessed 14 Feb 2019
22. J.R. Quinlan, *C4. 5: Programs for Machine Learning* (Elsevier, Amsterdam, 2014)
23. R.L. Iman, J.M. Davenport, Approximations of the critical region of the fbietkan statistic. Commun. Stat. Theory Methods **9**(6), 571–595 (1980)
24. S. Holm, A simple sequentially rejective multiple test procedure. Scand. J. Stat. pp. 65–70 (1979)