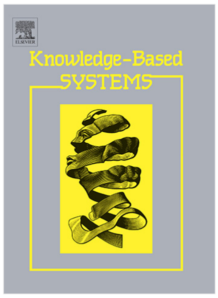
Accepted Manuscript



Multi-Imbalance: An open-source software for multi-class imbalance learning

Chongsheng Zhang, Jingjun Bi, Shixin Xu, Enislay Ramentol, Gaojuan Fan, Baojun Qiao, Hamido Fujita

PII: S0950-7051(19)30104-2
DOI: https://doi.org/10.1016/j.knosys.2019.03.001
Reference: KNOSYS 4697

To appear in: *Knowledge-Based Systems*

Received date : 15 September 2018
Revised date : 1 March 2019
Accepted date : 2 March 2019

Please cite this article as: C. Zhang, J. Bi, S. Xu et al., Multi-Imbalance: An open-source software for multi-class imbalance learning, *Knowledge-Based Systems* (2019), https://doi.org/10.1016/j.knosys.2019.03.001

Multi-Imbalance: an open-source software for multi-class imbalance learning

Chongsheng Zhang[a], Jingjun Bi[a], Shixin Xu[a], Enislay Ramentol[b], Gaojuan Fan[a], Baojun Qiao[a*], Hamido Fujita[c]

[a] *The Big Data Research Center, Henan University, 475001 KaiFeng, China*

[b] *SICS Swedish ICT, Isafjordsgatan 22, Box 1263, SE-164 29 Kista, Sweden*

[c] *Faculty of Software and Information Science, Iwate Prefectural University, Iwate, Japan*

Abstract

Imbalance classification is one of the most challenging research problems in machine learning. Techniques for two-class imbalance classification are relatively mature nowadays, yet multi-class imbalance learning is still an open problem. Moreover, the community lacks a suitable software tool that can integrate the major works in the field. In this paper, we present Multi-Imbalance, an open source software package for multi-class imbalanced data classification. It provides users with seven different categories of multi-class imbalance learning algorithms, including the latest advances in the field. The source codes and documentations for Multi-Imbalance are publicly available at `https://github.com/chongshengzhang/Multi_Imbalance`.

Keywords: multi-class imbalance leaning, imbalanced data classification

1. Introduction

Imbalance learning has become one of the major research topics in machine learning. It has important applications in credit card fraud detection [8], fault diagnosis [36], medical diagnosis [6], pattern recognition [32], etc. It also has strong potentials in security, such as malicious Apps detection [29, 30, 31]. Until now, there are several software tools for analyzing two-class imbalanced data, at data or algorithm level. However, for multi-class imbalanced data, there are very few available software packages, even though many researchers have proposed various algorithms and techniques to address this issue [1, 22, 4]. In this work, we develop the "Multi-Imbalance" (Multi-class Imbalanced data classification) software package and share it with the community to boost research in this field.

The developed Multi-Imbalance software contains 18 different algorithms for multi-class imbalance learning, which are depicted in Figure 1. We divide these algorithms into 7 modules (categories). We will introduce the framework and functionality of this software in the next sections.

Multi-Imbalance enables researchers to directly re-use our implementations on multi-class imbalanced data classification, thus avoid coding them from scratch. Hence, it will be very helpful for researchers and engineers in this field.

The remainder of this paper is organized as follows: Section 2 provides the background about imbalance classification, Section 3 describes the software framework, Section 4 presents an illustrative example and Section 5 concludes the paper.

2. Background

In this section, we provide an overview of the decomposition strategies and introduce the seven modules in Multi-Imbalance. We will also present existing software tools for two-class imbalance learning.

2.1. Classification in imbalance learning field

Imbalanced (skewed) data presents in many real-world applications. However, conventional machine learning methods do not focus on the prediction accuracy of the minority class(es), which may be more interesting for certain applications. In the past years, many two-class imbalanced data classification algorithms have been proposed [5, 3], which can be divided into four categories: solutions at data level or at algorithm level, methods that are cost-sensitive or ensemble-based.

However, many of the above algorithms cannot directly handle multi-class imbalanced data, hence significant effort has been invested in this issue in recent years [1, 22, 37, 4]. These algorithms are typically combinations of binarization techniques that transform the original multi-class data into binary subsets, with a two-class imbalance classification algorithm. Figure 2 describes the overall procedure of these algorithms. The multi-class imbalanced data is first split into (balanced) dichotomies, then a corresponding binary classifier is trained on each dichotomy. These binary classifiers are

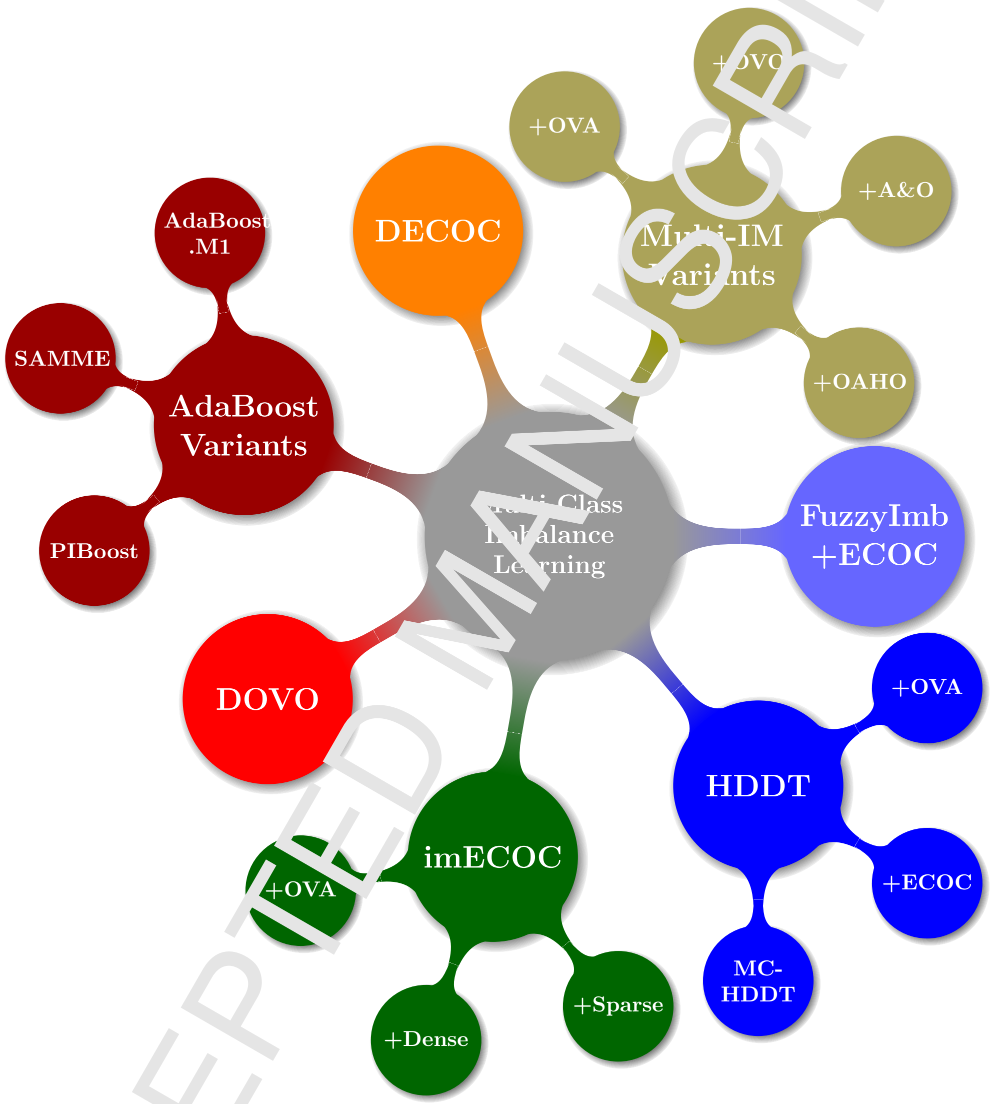


Figure 1: The 7 main modules and 18 algorithms in Multi-Imbalance

then integrated using ensemble learning methods such as majority voting to make predictions. The four commonly adopted accuracy metrics for imbalanced data are Acc, AUC, G-Mean, and F-Measure.

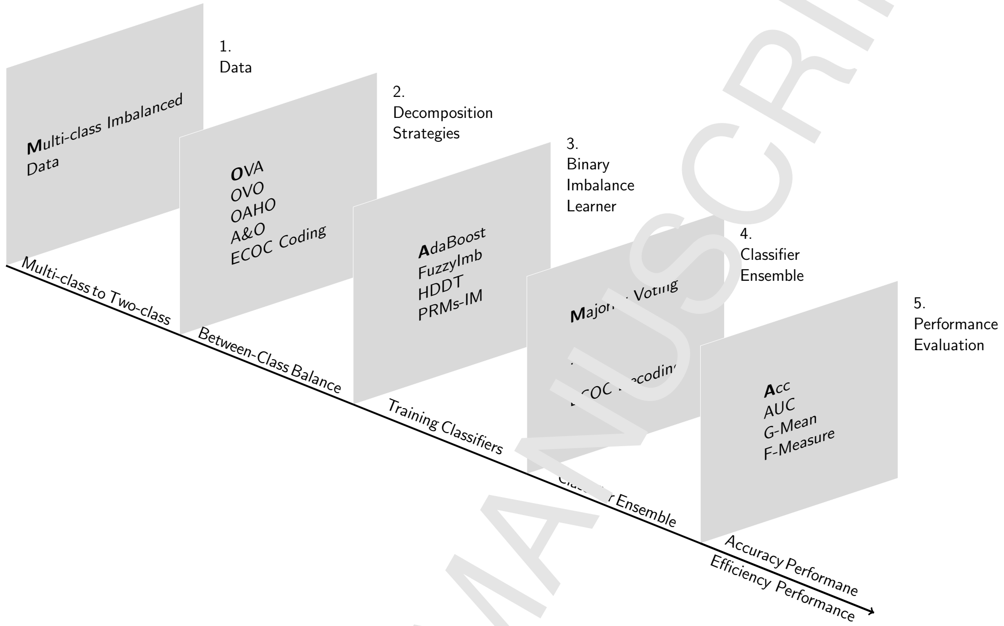


Figure 2: The general procedure of Multi-Imbalance

2.2. Related imbalance classification algorithms and binarization techniques

As shown in Figure 2, we first need to transform the original multi-class data into binary subsets via decomposition strategies. Multi-Imbalance implements 5 different binarization techniques:

- One-vs-One Approach (OVO) [14]: OVO trains a binary classifier for each possible pair of classes ignoring the examples that do not belong to the pair classes.

- One-vs-All (OVA) [26]: OVA trains a single classifier for each class, considering the current class as the minority one and the remaining classes as a majority one.

- One-Against-Higher-Order (OAHO) [24]: OAHO first sorts the classes by the number of samples in descending order $\{C_1, C_2, \ldots, C_n\}$, where C_1 has the largest number of samples. Starting from C_1 until C_{n-1}, it sequentially labels the current class as 'positive class' and all the remaining classes with lower ranks as 'negative classes', then trains a binary classifier over each resulting dataset.

- All-and-One (A&O) [11]: A&O is a combination of OVO and OVA. For a new prediction, it first uses OVA to get the top-2 prediction results (c_i, c_j), then adopts the OVO classifier previously trained for the pair of classes containing c_i and c_j to make the final prediction.

- The Error Correcting Output Codes (ECOC Coding) [7]: ECOC uses the idea of error correction output coding to classify the multi-class data. It first builds a codeword for each class to obtain the largest distance between various classes, thus transforms the classes of the multi-class data into c codewords.

Multi-Imbalance contains the following imbalanced data classification algorithms, but only AdaBoost variants support multi-class data classification.

- FuzzyImb (Imbalanced Fuzzy-Rough Ordered Weighted Average Nearest Neighbor Classification, IFROWANN for short): Proposed in [25], this algorithm is a powerful classifier for two-class imbalanced data based on fuzzy rough set theory and ordered weighted average aggregation.

- imECOC: Proposed in [21], it is an adapted ECOC [7] method for imbalance learning, for each binary classifier it simultaneously considers the between-class and within-class imbalance. It decodes with weighted distance to find the closest codeword (class).

- HDDT (Hellinger distance decision trees): It is a decision tree technique that uses the Hellinger distance as the splitting criterion [15].

- PRMs-IM: It randomly divides the majority samples into m parts (m is the ratio between the number of majority and minority samples), next combines each part with all the minority instances, then trains a corresponding binary classifier [13].

- AdaBoost variants. AdaBoost (Adaptive Boosting) [10] is originally a binary classification algorithm that integrates multiple weak classifiers to build a stronger classifier. In Multi-Imbalance, we incorporate five AdaBoost based algorithms for handling multi-class imbalanced data.

  - AdaBoost.M1 and SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function): they extend AdaBoost in both the updating of the samples' weights and the classifier combination strategy [38]. The main difference between them lies in the method for updating the weights of the samples.

– AdaC2.M1: proposed in [27], this method derives the best cost setting through the genetic algorithm (GA) for the subsequent boosting.

– AdaBoost.NC: proposed in [28], it deprecates GA since it is very time-consuming, but emphasizes on the ensemble diversity during training.

– PIBoost: proposed in [9], it uses a margin-based exponential loss function to classify multi-class imbalanced data.

2.3. Combining two-class imbalance classification algorithms with binarization techniques

In subsection 2.2, we have described the most representative methods for imbalanced classification on two classes and the binarization techniques used to transform the multi-class problem into a set of binary problems. The combination of the 5 categories of baseline algorithms and the 5 binarization techniques results in 18 algorithms implemented in Multi-Imbalance.

As an example, Figure 3 shows the 3 corresponding algorithms from the combination of OVO, OVA and OAHO with Multi-IM as the binary classifier. The multi-class data is first decomposed into several dichotomies, using OVO, OVA, or OAHO. It next trains a specific classifier on each dichotomy. These binary classifiers are finally integrated using majority voting or other ensemble methods. The Multi-IM, HDDT and AdaBoost variants modules in Figure 1 (a total of 12 algorithms) belong to this type of procedure.

Figure 4 gives another example, which is the combination of imECOC with different ECOC encoding methods, where three different ECOC encoding methods for handle multi-class data are considered, which are ECOC (sparse), ECOC (dense), and ECOC (OVA). Let us split a training dataset into two parts, which are data matrix and label vector. In the training phase, for the multi-class data, imECOC first generates the codewords for each class, using different ECOC encoding methods presented above. Let m be the bit number of the codewords. Next, imECOC replaces the original class label of the training dataset with the corresponding codewords, so that the label vector part is now transformed into codeword matrix part. Next, for each combination between the data matrix part and every column of the codeword matrix part (which is actually a dichotomy), imECOC trains a specific classifier using a certain baseline classification algorithm. Finally, m classifiers will be obtained by imECOC. Each of these m classifiers will be assigned a normalized weight W to represent its importance. In the testing/prediction phase, for a new instance x in the test dataset, imECOC first uses the m classifiers to make predictions on x, each prediction will also be multiplied with the weight of the corresponding classifier. Finally, an $m-bit$ prediction vector will be derived on x. imECOC then calculates the weighted distance between $m-bit$ prediction vector and each of the codewords, and the class (codeword) with the minimal distance will be assigned to x. The imECOC and FuzzyImb modules in Figure 1 (a total of 4 algorithms) belong to this type of procedure.

We note that, Figure 3 and Figure 4 can represent the two general types of procedures (combinations) for multi-class imbalance classification in Multi-Imbalance. Besides, another type of procedure in Multi-Imbalance is ensemble methodology, which is described in the subsection below.

2.4. Ensemble Methods for Multi-class imbalanced data classification

Recently, two ensemble-based algorithms were proposed which were specifically designed for multi-class imbalance classification. They have been included in our software owing to their excellent results.

- Diversified One-against-One(DOVO) [17]: It aims to find the best classification algorithm for each dichotomy when applying the OVO decomposition method to decompose the multi-class data. It integrates the heterogeneous classifiers to make predictions on the test instances.
- Diversified Error Correcting Output Codes (DECOC) [4]: It uses ECOC to decompose the multi-class data, next adopts DOVO like algorithm to find the best classification algorithm for each decomposed binary data and train a corresponding binary classifier. These classifiers are integrated with the same weighted distance function as imECOC [21].

2.5. Existing Software Tools for Two-class Imbalance Learning

There are very few open-source implementations for imbalanced data classification, although significant advances have been achieved in recent years. In the literature, we find the following software:

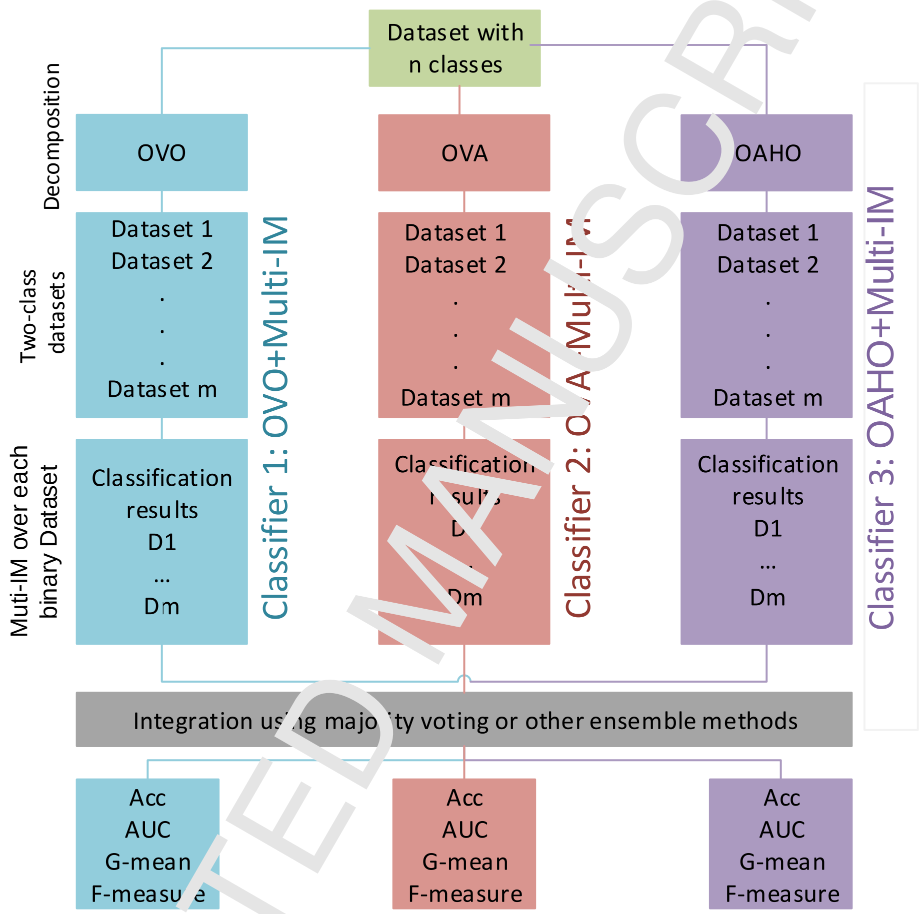


Figure 3: Example of resulting algorithms using OVO, OVA and OAHO combined with Multi-IM

KEEL [2]: A software tool developed in Java to assess evolutionary algorithms for Data Mining problems of various kinds including regression, classification, unsupervised learning. KEEL includes a module named "Imbalanced Learning". This module contains many algorithms for two-class imbalanced data, but it rarely includes algorithms for multi-class imbalanced data classification.

Imbalanced-learn [20]: A Python toolbox to tackle imbalanced data (de-

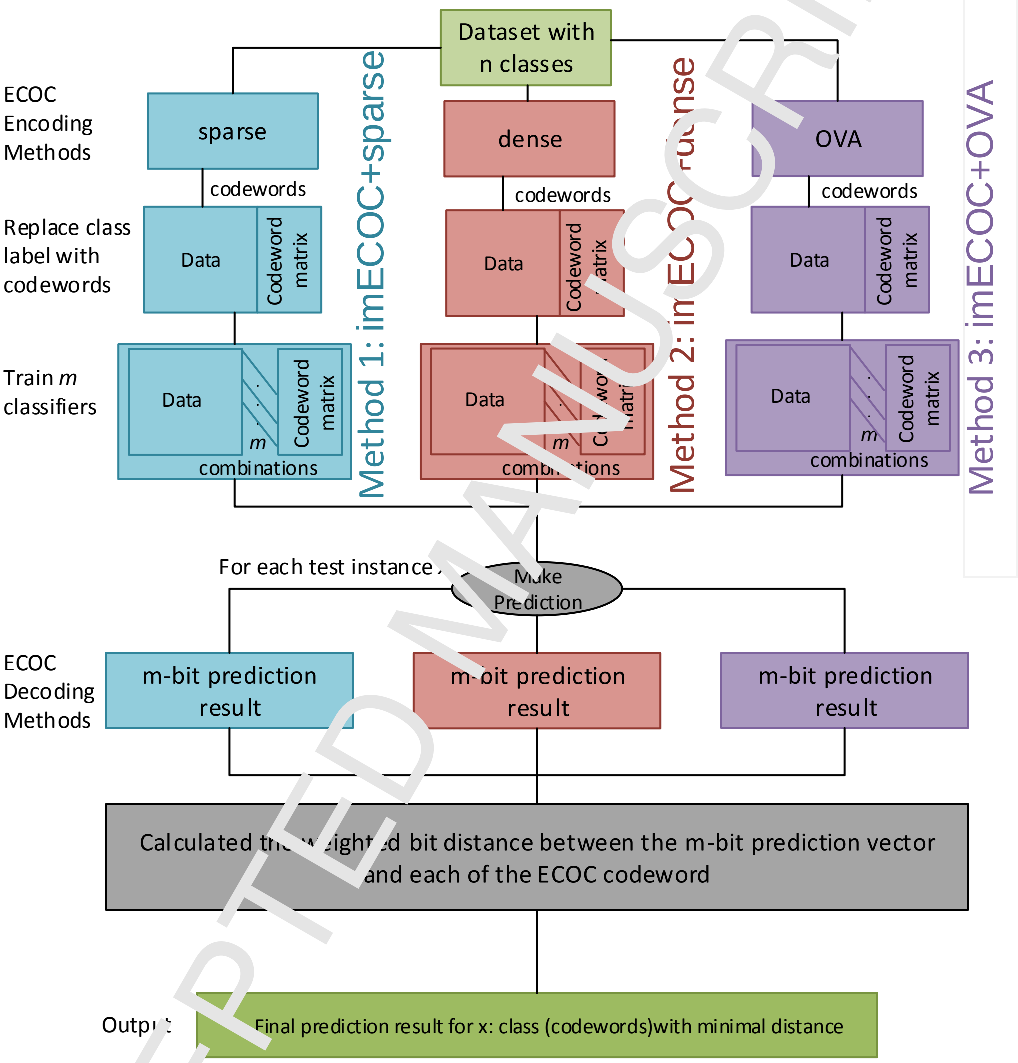


Figure 4: Example of resulting algorithms using ECOC combined with ImECOC

veloped to be compatible with scikit-learn[1]). The implementations of this toolbox are at data level (re-sampling methods), and a few algo-

[1] scikit-learn.org and available in scikit-learn-contrib projects.

rithms support multi-class imbalance learning. However, many latest advances on imbalance learning are not available in this software.

CRAN: An R package for imbalance learning, it only includes resampling methods for two-class problems. The package also includes a useful interface to perform oversampling.

climbR package[23]: A repository of functions designed for model evaluation and learning on imbalanced data.

To validate our implementations in Multi_Imbalance, we use the AdaBoost-SAMME algorithm from scikit-learn to test the Wine_data_set_indx_fixed dataset and report its accuracy performance. The 5-fold cross-validation overall accuracy is 0.9494; while in Multi_Imbalance, the overall accuracy of SAMME on the same dataset is 0.9698. On the thyroid_data_set_indx_fixed dataset, the accuracy of scikit-learn AdaBoost-SAMME is only 0.9764, whereas the reported accuracy of SAMME in Multi_Imbalance is 0.9806. The accuracy difference between the two SAMME algorithms may be caused by the detailed implementations of the decision tree algorithms and the choices of related parameters.

However, none of the existing software tools provides a comprehensive solution for the multi-class imbalance learning problem, hence the need of Multi-Imbalance. We open-source this software to advance research in this field. The source codes and documentation are available at the GitHub repository: *https://github.com/chongshengzhang/Multi_Imbalance*.

3. Software Framework

3.1. Software Architecture

Multi-Imbalance is composed of seven main modules, which represent seven categories of multi-class imbalanced data classification algorithms, which are shown in Figure 1.

The first class of algorithms are variants of AdaBoost. This module consists of five specific algorithms, i.e., AdaBoost.M1, SAMME, AdaC2.M1, AdaBoost.NC, and PIBoost.

The second module contains three variants of HDDT, which are HDDTova, HDDTecoc, and MCHDDT.

The third module includes a class of three algorithms, which are imECOC+Sparse, imECOC+Dense, and imECOC+OVA. These three algorithms are based upon imECOC, which uses the ECOC decomposition strategy to support multi-class imbalanced data classification.

The fourth module has a class of four algorithms based upon Multi-IM, which are Multi-IM+OVA, Multi-IM+OVO, Multi-IM+OAHO, and Multi-IM+A&O.

Multi-IM [12] extends the two-class PRMs-IM algorithm to the multi-class scenario, by combining it with the A&O decomposition strategy. PRMs-IM. It uses weighted voting to ensemble these classifiers for the prediction phase. Besides A&O, other decomposition methods such as OVA, OVO and OAHO are also supported in our implementation.

Each of the three remaining modules contains a single algorithm. In the fifth module, the corresponding algorithm is FuzzyImbECOC, where we extend the FuzzyImb algorithm from two-class to multi-class imbalance learning using the ECOC (sparse) decomposition strategy.

The sixth module contains the DOVO, which has demonstrated outstanding accuracy performance. It exhaustively finds the best classifier for each decomposed dichotomy, then integrates the corresponding two-class classifiers with ensemble learning strategies.

The seventh module presents our proposed DECOC algorithm, which is a combination of DOVO and imECOC. It has shown the best overall accuracy performance than state-of-the-art, including DOVO.

3.2. Implementation Platforms

Multi-Imbalance is developed using Matlab. We also provide alternative implementations of the major algorithms in OCTAVE which is a free software for replacing Matlab. The baseline algorithms used in our implementation are called from WEKA[2] to avoid implementing them from scratch. WEKA is a very powerful data mining tool implemented in Java, there is an Matlab wrapper that enables Matlab to communicate with WEKA.

3.3. Software Functionality

The main entry of our software is the *testall.m* file, where the main functions of the 18 state-of-the-art classification algorithms from the above seven modules can be chosen (called). In our implementations, we keep the seven modules very independent to facilitate users to reuse them conveniently.

[2]http://www.cs.waikato.ac.nz/ml/weka/

The output of each algorithm is the prediction results on the test instances. The above-mentioned four accuracy evaluation metrics are provided (implemented) in the *accuracyPerf()* function.

4. An Illustrative Example

The on-line documentation and user manuals in our software's GitHub repository provides a illustrative example for each of the 18 multi-class imbalanced data classification algorithms (please see "User_manual_Matlab.pdf" and "User_manual_Octave.pdf" for user manuals and examples of the main functionality of the software). As mentioned above, the main entry of our software is the *testall.m* file.

Let us take the class of HDDT algorithms as an example, where we want to use the *HDDTova* (*HDDTecoc*) function. As shown in Figure 2, the multi-class imbalanced data is first decomposed into dichotomies, using strategies such as OVA, ECOC Coding, etc. Each dichotomy is then artificially balanced to emphasize the minority class. After that, a binary classifier is trained on each dichotomy, using HDDT as the base learner. These binary classifiers are then integrated using majority voting or other ensemble methods. The ensemble of classifiers will then be used to predict the labels of the test samples. Finally, we can adopt the *accuracyPerf()* function to obtain the accuracy results of HDDTova, which include Acc, AUC, G-Mean, and F-Measure.

5. Conclusions

This paper presents "Multi-Imbalance", which is an open source software for the multi-class imbalanced data classification. It contains 18 algorithms, which are very flexible and easy to use. This software should be helpful for researchers and practitioners who need to tackle the multi-class imbalanced data classification problems.

One of the future directions is spatial-constrained imbalance learning, with applications in route planning and recommendation[16, 18, 19]. Another more challenging direction is imbalance learning from 3D data, with applications in 3D reconstruction and virtual reality [33, 34, 35]. It is also an interesting topic to explore the relationship between few-shot learning and imbalance learning.

Required Metadata

Current executable software version

Table 1 gives the information about the software release.

Nr.	(executable) Software metadata description	Please fill in this column
S1	Current software version	1.1
S2	Permanent link to executables of this version	*https://github.com/chongshengzhang /Multi_Imbalance*
S3	Legal Software License	GPLv3
S4	Computing platform/Operating System	Microsoft Windows, Mac OSx.
S5	Installation requirements & dependencies	Matlab or Octave
S6	If available, link to user manual - if formally published include a reference to the publication in the reference list	*https://github.com/chongshengzhang /Multi_Imbalance/tree/master/doc/*
S7	Support email for questions	Prof. Chongsheng Zhang (chongsheng.zhang@yahoo.com)

Table 1: Software metadata

Current code version

Table 2 describes the metadata about the source codes of Multi-Imbalance.

6. Acknowledgements

The research of Enislay Ramentol is funded by the European Research Consortium for Informatics and Mathematics (ERCIM) Alain Bensoussan Fellowship Programme.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	1.1
C2	Permanent link to code/repository used of this code version	*https://github.com/chongshengzhang/Multi_Imbalance/releases/tag/v1.1*
C3	Legal Code License	GPL v3
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	Matlab or Octave
C6	Compilation requirements, operating environments & dependencies	Matlab or Octave
C7	If available Link to developer documentation/manual	Not available
C8	Support email for questions	Prof. Chongsheng Zhang (chongsheng.zhang@yahoo.com)

Table 2: Code metadata

7. References

[1] L. Abdi and S. Hashemi. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Trans Knowl Data Eng*, 28(1):238–251, 2016.

[2] J. Alcalá, L. Sánchez, S. García, M.J. del Jesús, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, and F. Herrera. KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing*, 13:3:307–318, 2009.

[3] G. E. A. P. A. Batista, R. C. Prati, and M.C. Monard. A study of the behaviour of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1):20–29, 2004.

[4] Jingjun Bi and Chongsheng Zhang. An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. *Knowledge-Based Systems*, 158:81–93, 2018.

[5] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligent Research*, 16:321–357, 2002.

[6] Y. Chen. An empirical study of a hybrid imbalanced-class dt-rst classification procedure to elucidate therapeutic effects in uremia patients. *Med Biol Eng Comput*, 54(6):983–1001, 2016.

[7] T. Dietterich and G. Bakiri. Error-correcting output codes: A general method for improving multiclass inductive learning programs. *AAAI*, pages 395–395, 1994.

[8] T.E Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 3:291–316, 1997.

[9] B.A. Fernández and L. Baumela. Multi-class boosting with asymmetric binary weak-learners. *Pattern Recognition*, 47(5):2080–2090, 2014.

[10] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci*, 55(1):119–139, 1997.

[11] P.N. García and B.D. Ortiz. Improving multiclass pattern recognition by the combination of two strategies. *IEEE Trans. Pattern Anal Mach. Intell*, 28 (6):1001–1006, 2006.

[12] A.S. Ghanem, S. Venkatesh, and G. WesT. Multi-class pattern classification in imbalanced data. *International Conference on Pattern Recognition*, pages 2881–2884, 2010.

[13] S. Ghanem, S. Venkatesh, and G. West. Learning in imbalanced relational data. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[14] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Ann. Statist*, 26(2):451–471, 1998.

[15] T. R. Hoens, Q. Qian, N.V. Chawla, and et al. Building decision trees for the multi-class imbalance problem. *Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg*, pages 122–34, 2012.

[16] Lei Huang, Bowen Ding, Aining Wang, Yuedong Xu, Yipeng Zhou, and Xiang Li. User behavior analysis and video popularity prediction on a large-scale vod system. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 14(3s):67:1–67:24, 2018.

[17] S. Kang, S. Cho, and P. Kang. Constructing a multi-class classifier using one-against-one approach with different binary classifiers. *Neurocomputing*, 149:677–682, 2015.

[18] Yongxuan Lai, Zheng Lv, Kuan-Ching Li, and Minghong Liao. Urban traffic coulomb's law: A new approach for taxi route recommendation. *IEEE Transactions on Intelligent Transportation Systems*, DOI: 10.1109/TITS.2018.2870990, 2019.

[19] Yongxuan Lai, Lu Zhang, Fan Yang, Lv Zheng, Tian Wang, and Kuanching Li. Casq: Adaptive and cloud-assisted query processing in vehicular sensor networks. *Future Generation Computer Systems*, 94:237–249, 2019.

[20] Guillaume Lemaître, 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(17):1–5, 2017.

[21] X. Y Liu, Q.Q Li, and ZH Zhou. Learning imbalanced multi-class data with optimal dichotomy weights. In *IEEE 13th International Conference on Data Mining (IEEE ICDM)*, 2013.

[22] L.Yijing, G.Haixiang, L. Xiao, L. Yanan, and L. Jinling. Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. *Knowledge-Based Systems*, 94:88–104, 2016.

[23] Lawrence Mosley. *A balanced approach to the multi-class imbalance problem*. PhD thesis, Iowa State University Capstones, 2013.

[24] Y.L. Murphey, H. Wang, G. Ou, and et al. Oaho: an effective algorithm for multi-class learning from imbalanced data. In *International Joint Conference on Neural Networks, IEEE*, pages 406–411, 2007.

[25] E. Ramentol, S. Vluymans, N. Verbiest, Y. Caballero, R. Bello, C. Cornelis, and F. Herrera. Ifrowann: Imbalanced fuzzy-rough ordered weighted average nearest neighbor classification. *IEEE Trans. Fuzzy Systems*, 23(5):1622–1637, 2015.

[26] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Machine Learning Research*, 5:101–145, 2004.

[27] Y. Sun, M.S Kamel, and Y. Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Proceedings of the 6th IEEE International Conference on Data Mining*, pages 592–602, 2006.

[28] S. Wang, H. Chen, and X. Yao. Negative correlation learning for classification ensembles. *Proc. Int. Joint Conf. Neural Netw*, pages 2893–2900, 2010.

[29] Wei Wang, Yuanyuan Li, Xing Wang, Jiqiang Liu, and Xiangliang Zhang. Detecting android malicious apps and categorizing benign apps with ensemble of classifiers. *Future Generation Computer Systems*, 78:987–994, 2018.

[30] Wei Wang, Jiqiang Liu, Georgios Pitsilis, and Xiangliang Zhang. Abstracting massive data for lightweight intrusion detection in computer networks. *Information Sciences*, 433-434:417–430, 2018.

[31] Wei Wang, Xing Wang, Dawei Feng, Jiqiang Liu, Zhen Han, and Xiangliang Zhang. Exploring permission-induced risk in android applications for malicious application detection. *IEEE Trans. Information Forensics and Security*, 9(11):1869–1882, 2014.

[32] X.Gao, Z.Chen, S.Tang, Y. Zhang, and J. Li. Adaptive weighted imbalance learning with application to abnormal activity recognition. *Neurocomputing*, 173:1927–1935, 2016.

[33] Gang Xu, Tsz Ho Kwok, and Charlie C.L. Wang. Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization. *Computer-Aided Design*, 91:1 – 13, 2017.

[34] Gang Xu, Ming Li, Bernard Mourrain, Timon Rabczuk, Jinlan Xu, and Stephane P.A. Bordas. Constructing iga-suitable planar parameterization from complex cad boundary by domain partition and global/local optimization. *Computer Methods in Applied Mechanics and Engineering*, 328:175 – 200, 2018.

[35] Gang Xu, Bernard Mourrain, Regis Duvigneau, and Andre Galligo. Analysis-suitable volume parameterization of multi-block computational

domain in isogeometric applications. *Computer-Aided Design*, 45(2):395 – 404, 2013.

[36] Z. Yang, W. Tang, A. Shintemirov, and Q. Wu. Association rule mining based dissolved gas analysis for fault diagnosis of power transformers. *IEEE Trans. Syst.,Man, Cybern. C, Appl. Rev*, 39:597–610, 2009.

[37] Chongsheng Zhang, Changchang Liu, Xiangliang Zhang, and George Almpanidis. An up-to-date comparison of state-of-the-art classification algorithms. *Expert Syst. Appl.*, 82:128–150, 2017.

[38] J. Zhu, H. Zou, S. Rosset, and et al. Multi-class adaboost. *Statistics & Its Interface*, 2(3):349–360, 2006.