SMOTE-GBM: An Improved Classification Model for Early Folding Residues During Protein Folding

Isra Al-Turaiki

College of Computer and Information Sciences, Information Technology Department King Saud University, Riyadh, Saudi Arabia

Summary
Proteins are fundamental molecules that play important roles in the cell. The function and behavior of proteins are determined by their native structure. However, the protein folding process is not well understood. Machine learning algorithms have been widely used to solve bioinformatics problems. Building predictive models from early folding residues (EFRs) has recently been investigated. However, the datasets used suffer from the class imbalance problem. This renders the classification task difficult. In this paper, we address the class imbalance problem in an EFR dataset using the synthetic minority oversampling technique (SMOTE). We trained an ensemble model, the gradient boosted machine (GBM), using the balanced dataset. We then compared the performance of our trained model with that of other models in the literature. Our experimental results indicate that better classification performance is obtained when oversampling is used to overcome the class imbalance problem. In particular, better improvement was observed in terms of precision, recall, and F-measure values.

Key words:
Early folding residue (EFR), Machine learning, Synthetic minority oversampling technique (SMOTE), Ensemble, Gradient boosted machine (GBM)

1. Introduction

Over the past decades, many significant advances have been made in the field of bioinformatics. With the advent of high-throughput technology, the exponential growth of biological data has made computational approaches an integral part of biological data analysis and research.

Proteins are diverse biological molecules that play fundamental roles in the cell. They are made of long chains of amino acids that fold into three-dimensional structures. The process of protein folding is complex and can cause disease if unsuccessful. Understanding how proteins assume three-dimensional structures as well as how these structures remain stable are important for determining protein behavior and function. However, the protein folding process is still not well understood [1].

In order to understand how a protein folds, it is vital to understand the early stages of the folding process—that is, the early interactions between local structural elements. Early folding residues (EFRs) are key residues that initiate and guide the folding process [1]. They are identified experimentally using the pulse-labeling hydrogen–deuterium exchange method. This method examines the folding process with spatial and temporal resolution [1]. Thus, EFRs can be used to help researchers understand how a protein folds. EFRs are difficult to obtain experimentally and have been obtained for a small number of proteins. Currently, the Start2Fold database [2][3] contains curated and classified residue- or segment-level data on the folding and/or stability proteins. This database can be used to develop a classification model for EFRs.

Due to their capability to deal with multidimensional bioinformatics data, machine learning methods have been applied successfully to solve many bioinformatics problems [4][5][6], including the protein folding problem in terms of EFRs. Raimondi et al. [7] developed a support vector machine (SVM) model with a radial basis function kernel for the classification of EFRs. The model was trained using data from 30 proteins available in the Start2Fold database. The dataset consisted of 25 features for 3398 records. Using 27-fold cross validation, the developed model achieved an accuracy of 73.4% and a low value for precision of 36.1%. Bittrich et al. [1] created a dataset of EFRs also extracted from the Start2Fold database. Their dataset was composed of 3266 residues with 27 features. The authors used the dataset to develop a classification model based on generalized matrix learning vector quantization (GMLVQ) [8]. The performance evaluation of the developed model showed a maximum accuracy of 77.4% with a low precision of 29.7%. The implemented model was compared to state-of-the-art classifiers: naïve Bayes (NB), random forest (RF), and SVM. The GMLVQ model was augmented with a visualization tool that allows the user to interpret the resulting model.

Previous research on EFR classification was based on datasets that suffered from the class imbalance problem, i.e., the minority class was poorly represented. This non-uniform distribution of classes can affect the performance of the classification algorithm. This means that the resulting model will be biased toward the majority class, will exhibit poor classification performance, and will yield more false negatives. We believe that the

Manuscript received March 5, 2020
Manuscript revised March 20, 2020

performance of EFR classification models could be improved if the class imbalance problem is addressed.

In this study, we addressed the protein folding problem in terms of EFRs. In particular, we focused on improving the classification of EFRs by applying the synthetic minority oversampling technique (SMOTE) [9] to overcome the class imbalance problem. Since its introduction in 2002, SMOTE has been applied successfully to different types of real-world problems, including bioinformatics problems such as the classification of the molecular functions of proteins [10]. The technique has demonstrated both simplicity and robustness. We used the SMOTE-balanced dataset to train a gradient boosted machine (GBM) classifier. The GBM is an ensemble machine learning algorithm that sequentially builds a set of trees for classification. The main assumption of ensemble algorithms is that combining the predictions of multiple models improves the performance of the classification task.

The paper is organized as follows: Section 2 describes the methodology used in the present work, including data description and preprocessing, SMOTE, and classification using a GBM. Section 3 describes the results and provides a relevant discussion. Section 4 presents the conclusions drawn from the study.

2. Methodology

To improve the classification of EFRs, we built a GBM classification model trained using a SMOTE-balanced dataset. In this section, we briefly describe the dataset used and the basic idea of SMOTE and GBM.

2.1 Dataset Description

We used the dataset prepared by Bittrich et al. [1], which was extracted from the Start2Fold database [2][3]. The dataset consisted of 3266 residues (records) with two class labels: late and early. It was also imbalanced, with the early class constituting only 14.8% of the dataset. Each residue was described using a set of 27 features to capture properties such as energy profiling, secondary structure, relative accessible surface area, non-covalent contacts, graph representation, and topological descriptors. All the features were numeric except for the folds feature, which had the class label. Table 1 summarizes the dataset's features [1].

Table 1: The features of the Bittrich et al. [1]. dataset

Feature	Description
e	Computed energy values
ePred	Predicted energy values
SecSize	Size of the surrounding secondary structure elements
LF	Fraction of surrounding unordered secondary structure elements
Rasa	Relative accessible surface area
PlipLC	Absolute count of local PLIP contacts
PlipHbLC	Absolute count of local PLIP hydrogen bonds
PlipHpLC	Absolute count of local PLIP hydrophobic interactions
PlipBbLC	Absolute count of local PLIP backbone contacts
PlipLR	Absolute count of long-range PLIP contacts
PlipHbLR	Absolute count of long-range PLIP hydrogen bonds
PlipHpLR	Absolute count of long-range PLIP hydrophobic interactions
PlipBbLR	Absolute count of long-range PLIP backbone contacts
PlipBN	Betweenness using all PLIP contacts
PlipCL	Closeness using all PLIP contacts
PlipCC	Clustering coefficient using all PLIP contacts
PlipHbBN	Betweenness using PLIP hydrogen bonds
PlipHbCL	Closeness using PLIP hydrogen bonds
PlipHbCC	Clustering coefficient using PLIP hydrogen bonds
PlipHpBN	Betweenness using PLIP hydrophobic interactions
PlipHpCL	Closeness using PLIP hydrophobic interactions
PlipHpCC	Clustering coefficient using PLIP hydrophobic interactions
ConvBN	Betweenness using the distance-based contact definition
ConvCL	Closeness using the distance-based contact definition
ConvCC	Clustering coefficient using the distance-based contact definition

2.2 Dataset Normalization

Normalization is used to scale the values of features such that they fall within a smaller range—for example, from 0 to 1 [11]. This is a very important preprocessing step, since it eliminates the effect of measurement unit on the final data mining results. Normalizing feature values allows all features to be given an equal weight when performing the data mining task.

z-score normalization is a data normalization technique whereby the values of a feature, a, are normalized based on its mean and standard deviation. The z-score denotes the number of standard deviations a feature value is from the mean of all feature values. The z-score is calculated as follows:

$$a' = \frac{a - \bar{x}}{s} \quad (1)$$

where a' is the normalized value of a, x' is the mean, and s is the standard deviation.

2.3 Oversampling Using SMOTE

Our dataset in this study was class-imbalanced, as there were 482 records of class early and 2784 records of class late. Performing the classification task on the dataset as such would yield poor performance on the minority class, since the classifier would not see enough records in the training phase. Thus, it is of crucial importance to handle the imbalanced dataset problem before proceeding to the classification task. There are many oversampling techniques for dealing with the class imbalance problem, of which SMOTE [9] is one of the most widely used. In SMOTE, new records are synthesized from the minority class. Basically, SMOTE works as follows: a record, *x*, is randomly selected from the minority class in the dataset. Then, the *k* neighbors of *x* are determined, with *k* usually set to 5. One of the identified neighbors, *y*, is then chosen. A new synthetic record, *z*, is generated at a randomly selected point between *x* and *y* in the feature space. Research has demonstrated the success of SMOTE when applied in a variety of applications [12]. The technique has also been shown to be robust and to perform better than simple oversampling.

2.4 Gradient Boosted Machines (GBMs)

To perform the classification task, we used the GBM algorithm proposed by Friedman [13], which is typically used for regression and classification tasks [14][15]. A GBM is an ensemble method that sequentially builds a set of trees. In each iteration, a tree is improved based on the performance of the tree in the previous iteration. A GBM is composed of three elements: a loss function (e.g., mean square error), a weak learner (e.g., decision trees), and an additive model. The GBM algorithm finds a final model that will minimize the loss function. As shown in Algorithm 1, it starts with an initial prediction—for example, in regression, the initial prediction is the mean of the observed values. Then, residuals, which are the differences between observed values and the predicted value, are calculated. A model is then built to predict the residuals. Models are built sequentially, with each model seeking to correct errors in the previous model. The obtained model is added to the previous model, and the process is repeated for a user-defined number of iterations. The *learning rate* refers to the fraction of the current predicted value that is added to the value predicted in the previous iteration. The learning rate can take any value between 0 and 1. Research suggests that small values (< 0.01) lead to better performance of GBMs [16]. According to Kuhn and Johnson [17], the value of the learning rate is inversely proportional to the computation time required to reach the optimal model. GBMs are highly flexible and can be customized to deal with any data-driven task [18].

Algorithm 1 Simple Gradient Boosting Algorithm

Inputs:
M: number of iterations
Choice of loss function
Choice of weak classifier
Algorithm
Initialize model f_0 with a predicted value for each record (e.g., average of response value)
for i=1 to M
  compute residuals R_i
  build a model f_i to predict R_i
  use f_i to predict response values for all records
  update model f_i predictions to generate model f_{i+1}
end for
return f_m

3. Experimental Results

3.1 Experimental Setup

In this study, we used the GBM implementation in RapidMiner Studio 9.6 [19]. The algorithm was run using a learning rate of 0.01, and the maximum depth of trees was equal to 10. For SMOTE, we used the Operator Toolbox [20] extension. We ran our experiments using a MacBook Pro, with the macOS Catalina operating system, version 10.15.3, and a 2.3 GHz 8-Core Intel Core i9 with 16 GB RAM.

We applied 10-fold cross validation. The performance of the classification model was evaluated in terms of accuracy, precision, recall, F-measure, and area under the curve (AUC).

Accuracy here refers to the percentage of correctly classified records, which was calculated as shown in Equation 2.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

Precision refers to the percentage of records correctly classified as belonging to the early class out of all the records predicted to belong to the early class. This percentage was calculated as shown in Equation 3.

$$PR = \frac{TP}{TP + FP} \quad . \quad (3)$$

Recall is the percentage of records correctly classified as belonging to the early class out of the total number of records belonging to the early class. This percentage was calculated according to Equation 4.

$$RE = \frac{TP}{TP + FN} \quad (4)$$

where TP represents true positives, i.e., the number of records correctly classified as belonging to the positive class (early); FP represents false positives, i.e., the number of records incorrectly classified as positive records; FN represents false negatives, i.e., the total number of residues incorrectly classified as belonging to the negative class (late); and TN represents true negatives, i.e., the total number of residues correctly classified as late.

The F-measure is the harmonic mean of precision and recall, and it was calculated via Equation 5 below.

$$F_1 = \frac{2 * precision * recall}{precision + recall} \quad . \quad (5)$$

The receiver operating characteristic curve (ROC) is a graphical representation of the performance of a classification model. It plots the true positive rate against the false positive rate. The AUC measures the overall performance of a binary classifier. The value of the AUC ranges from 0.5 (for a random classifier) to 1 (perfect classifier).

3.2 Experimental Results

Here, we evaluate the performance of the proposed framework and compare it to published results of NB, RF, SVM, and GMLVQ using the Bittrich et al. dataset [1]. For the GMLVQ model, we compare the best values obtained in the different configurations presented in [1].

First, we investigated the effect of oversampling on the performance of the GBM classifier. Fig. 1 shows the performance measures for the GBM classifier with and without applying oversampling using SMOTE. As shown in the figure, we observed improved performance of the classifier in all measures when oversampling was applied. More specifically, better values for precision, recall, and F-measure are obtained. These values improved by a large margin.

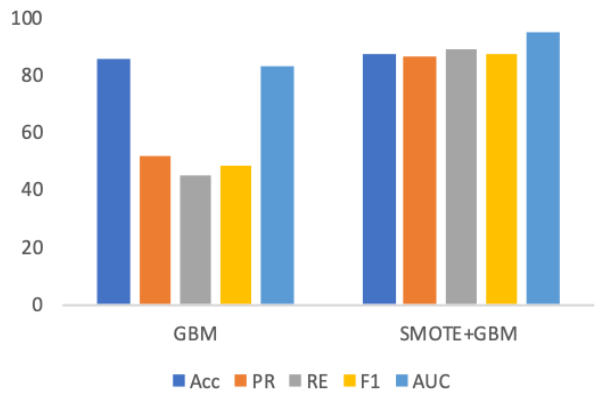

Fig. 1 The performance measures of GBM with and without SMOTE.

Then, we looked at the performance of SMOTE+GBM as compared to other approaches in the literature. Fig. 2 shows the performance of GBM as compared to classifiers presented in [1]. Again, the GBM with SMOTE outperformed all the compared classifiers. In particular, precision and recall were improved by a large margin. This indicates that oversampling allows the classifier to better learn to identify early folding residues.

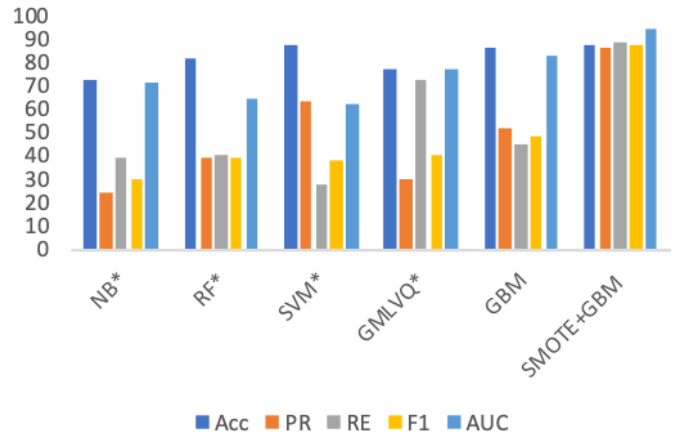

Fig. 2 The performance measures of SMOTE+GBM compared to other classifiers in the literature. * indicates results from [1].

We also wanted to investigate whether the performance of the previously published classifier could also be improved using oversampling. Fig. 3 shows the performance of NB, RF, and SVM when oversampling was applied. For NB, RF, and SVM, significant improvement was observed in terms of precision, recall, and F-measure.

In contrast to the results reported in [1], where it was difficult to determine which classifier performed best, our results indicate that the performance of machine learning algorithms is superior when oversampling is applied.

Overall, the SMOTE+GBM model is found to outperform all the compared models in all measures, except recall. The best recall value of 97.41% is obtained by the SMOTE+RF model. In addition, with the 27 features representing the EFRs, machine learning models are able to better separate between early folding residues and late folding residues.

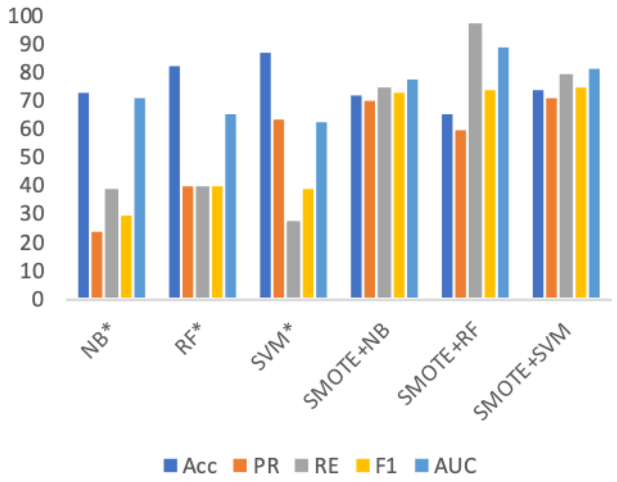

Fig. 3 The performance measures for NB, RF, and SVM when oversampling was applied.

4. Conclusion

Proteins are dynamic molecules that assume native structures in order to perform their functions. The folding process of proteins is complex. There have been many attempts to improve our understanding of this process. Recently, the classification of ERPs has gained the interest of researchers. Machine learning-based models were developed using real datasets. However, the datasets used suffer from the class imbalance problem. In this research, we investigated the potential of oversampling using SMOTE to improve the classification of EFR. A GBM was trained using the balanced dataset. Experimental results indicate that SMOTE leads to a superior classification task when using GBM, NB, RF, and SVM. The improvement was observed in terms of high precision and recall values. Further research on this problem could focus on studying other methods for dealing with the class imbalance problem. In addition to improving classification accuracy, the interpretability of the trained models is crucial. The potential of interpretable machine learning models, such as: decision trees, decision, rules, etc. could be investigated.

References

[1] S. Bittrich, M. Kaden, C. Leberecht, F. Kaiser, T. Villmann and D. Labudde, “Application of an interpretable classification model on Early Folding Residues during protein folding,” BioData Min., vol.12, p.1, 2019.
[2] “Start2Fold.” http://bio2byte.be/start2fold/ (accessed Mar. 28, 2020).
[3] R. Pancsa, M. Varadi, P. Tompa and W. F. Vranken, “Start2Fold: a database of hydrogen/deuterium exchange data on protein folding and stability,” Nucleic Acids Res., vol.44, Database issue, pp.D429–D434, Jan. 2016.
[4] A. Sohail and F. Arif, “Supervised and unsupervised algorithms for bioinformatics and data science,” Prog. Biophys. Mol. Biol., vol.151, pp.14–22, Mar. 2020.
[5] E. Naresh, B. P. Vijaya Kumar Ayesha and S. P. Shankar, “Impact of machine learning in bioinformatics research,” in Statistical Modelling and Machine Learning Principles for Bioinformatics Techniques, Tools, and Applications, K. G. Srinivasa, G. M. Siddesh and S. R. Manisekhar, Eds. Singapore: Springer, 2020, pp.41–62.
[6] Q. Zou and Q. Liu, “Advanced Machine Learning Techniques for Bioinformatics,” IEEE/ACM Trans. Comput. Biol. Bioinform., vol.16, no.4, pp.1182–1183, Jul. 2019.
[7] D. Raimondi, G. Orlando, R. Pancsa, T. Khan and W. F. Vranken, “Exploring the sequence-based prediction of folding initiation sites in proteins,” Sci. Rep., vol.7, no.1, p.8826.
[8] A. Sato and K. Yamada, “Generalized learning vector quantization,” in Proceedings of the 8th International Conference on Neural Information Processing Systems, Denver, Colorado, Nov. 1995, pp.423–429.
[9] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Oversampling Technique,” J. Artif. Intell. Res., vol.16, pp.321–357, Jun. 2002.
[10] D. Hwangt, F. Fotouhit and R. L. Finley, “Predictive model for yeast protein functions using modular neural approach,” in Third IEEE Symposium on Bioinformatics and Bioengineering, 2003. Proceedings, Mar. 2003, pp.436–440.
[11] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, 3rd edition. Haryana, India; Burlington, MA: Morgan Kaufmann.
[12] A. Fernandez, S. Garcia, F. Herrera and N. V. Chawla, “SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary,” J. Artif. Intell. Res., vol.61, pp.863–905, Apr. 2018.
[13] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” Ann. Stat., vol.29, no.5, pp.1189–1232, 2001.
[14] L. Deng, J. Pan, X. Xu, W. Yang, C. Liu and H. Liu, “PDRLGB: precise DNA-binding residue prediction using a light gradient boosting machine,” BMC Bioinformatics, vol.19, no.19, p.522, Dec. 2018.
[15] P. Xuan, C. Sun, T. Zhang, Y. Ye, T. Shen and Y. Dong, “Gradient boosting decision tree-based method for predicting interactions between target genes and drugs,” Front. Genet., vol.10, 2019.
[16] G. Ridgeway, “Generalized boosted models: a guide to the GBM package,” p.15, Jan. 2019.
[17] M. Kuhn and K. Johnson, Applied Predictive Modeling. New York: Springer-Verlag, 2013.

[18] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," Front. Neurorobotics, vol.7, p.21, 2013.
[19] "RapidMiner | Best Data Science & Machine Learning Platform," RapidMiner. https://rapidminer.com/ (accessed Mar. 15, 2020).
[20] "Operator Toolbox," RapidMiner Marketplace. facesContext.externalContext.requestURL (accessed Mar. 28, 2020).

Isra AL-Turaiki is an assistant professor of Computer Science at King Saud University. She received her PhD degree in 2014 from the College of Computer Sciences at King Saud University. Her research interests include data mining, machine learning, and bioinformatics.