Introduction

Class imbalance is a common recurring problem in most bioinformatics data as the data is usually highly skewed towards the majority than the minority class. Class imbalance affects various bioinformatics fields, including disease research and medical diagnosis.The majority class has a far higher number of occurrences than the minority class(es). When skewed datasets are used to build models, it creates a learning bias towards the majority class. For example, when creating a system for anomaly detection, if the ratio of the classes is 9,000:1 (usually the case in real-world applications), there is an issue of over-classification of the majority class. The increased prior probability causes the misclassification of the minority class. This defeats the purpose of most medical diagnoses, as the rare class(e.g., medical diagnosis) is the class that needs to be classified the most accurately.

Most classic Machine Learning models trained on an imbalanced dataset will work on increasing overall performance rather than focusing on the significance of each class. A traditional Machine Learning model trained on a data set to detect cancer(6500:1) can result in 95% accuracy without correctly classifying all the detected cases. The class imbalance problem needs to be addressed in bioinformatics data.

Data sampling is a standard solution to handle class imbalance. Data sampling requires adding or removing class instances to obtain a balanced dataset. Data sampling can extrapolate or populate the cases of minority classes. Data sampling can also balance the dataset by drawing samples from the majority class. Both techniques will result in a much more balanced dataset. Class balance is achievable on different ratios- 50:50,35:65 based on the requirement. There are several ways to achieve balance - adding duplicates from the minority class and randomly removing the samples from the majority class. From the original samples, synthetically adding samples to the minority class is also possible. For most highly unbalanced cases achieving a 50:50 ratio could be better but is only sometimes necessary.

The majority class's instances are eliminated by undersampling. Because it randomly excludes samples from the majority class, random under-sampling (RUS) is a common technique. In contrast, adding cases to the minority class is termed Over-sampling. Based on if addition is random, the method is called Random over-sampling (ROS). Both ways of data sampling have their drawbacks. Under-sampling may lead to the removal of samples that can lead to the removal of excessive majority class Samples, which in turn can cause issues. Over-sampling, on the other hand, requires duplication and can cause overfitting of the model. The SMOTE( Synthetic Minority over-sampling technique) consists of synthetically generating samples close to the ones in the minority class and avoids overfitting.

Another strategy for addressing the issue of class imbalance is ensemble learning. It combines the predictions of multiple base learners to improve classification performance. The three ensemble learning techniques - Bagging, Boosting, and Random Forest - are commonly used in various domains, including bioinformatics, to address the class imbalance challenge.

1. Bagging: Bootstrap Aggregating, also known as bagging, entails training several base learners using various subsets of the training data. The subsets are created by randomly sampling the training data with replacement (i.e., allowing the same instance to be sampled multiple times), hence the term "bootstrap." The predictions of the base learners are combined by averaging (for regression) or by taking a majority vote (for classification) to obtain the final prediction. Bagging can help reduce overfitting by averaging out the individual predictions of the base learners and can improve classification performance by reducing variance.
2. Boosting: Boosting is an iterative ensemble technique that adjusts the weights of the training instances to give more importance to misclassified instances. The base learners are trained sequentially, with each subsequent base learner focusing on correcting the mistakes made by the previous base learners. This way, boosting can improve the accuracy of the ensemble by emphasizing the samples that are difficult to classify. Popular boosting algorithms include AdaBoost, XGBoost, and gradient boost.
3. Random Forest: Random Forest adds randomness to the tree-building process by randomly choosing a subset of features at each split in addition to bootstrapping subsets of the training data. This helps in reducing correlation among the base learners and can lead to better generalization performance. High precision, noise resistance, and capacity for handling large-scale data are all strengths of Random Forest.

Ensemble learning methods, including Bagging, Boosting, and Random Forest, can be instrumental in bioinformatics, where class imbalance is a common challenge. Combining the predictions of multiple base learners, ensemble methods can potentially improve the classification performance for minority classes and make the models more robust to imbalanced data. With the help of several base learners and the aggregated judgments of those learners, ensemble learning tries to produce numerous classification models. Additionally, ensemble learning frequently outperforms any single base learner, is less likely to overfit, and is effective at minimizing the bias of classification findings.

Existing methodologies & their critiques

Comparing data sampling strategies to ensemble learning methods, it has been discovered that the latter are more successful at enhancing classification performance for imbalanced bioinformatics data[3].Using 15 unbalanced and high-dimensional gene expression datasets, the study assessed nine classification techniques, three feature rankers, and four feature subset sizes. The results show that ensemble learning approaches consistently improve classification outcomes over data sampling methods, with Select-Bagging utilizing Naive Bayes (NB) and Random Forest being the two best ensemble learning strategies. Based on these findings, it is advised to utilize either Random Forest with 100 trees (RF100) or Select-Bagging with NB for imbalanced datasets. RF100 has the advantage of not being dependent on the choice of the base learner, unlike Select-Bagging, which uses NB as the base learner. These findings offer valuable insights for bioinformatics data analysis, suggesting that ensemble learning methods, particularly Select-Bagging with NB or Random Forest with 100 trees, can improve classification performance in imbalanced gene expression datasets.

RUSBoost

A novel hybrid algorithm called RUSBoost combines data sampling and boosting techniques to improve models' performance on imbalanced data[2].RUSBoost is a variation of another hybrid algorithm called SMOTEBoost.The data sampling component of the AdaBoost algorithm is used by both RUSBoost and SMOTEBoost, however their methods differ.The synthetic minority class examples are produced by extrapolating between existing examples using the SMOTE (Synthetic Minority Over-sampling Technique) oversampling technique, which is used by SMOTEBoost. In contrast, RUSBoost employs random undersampling (RUS), a more simple method that randomly omits samples from the majority class. Despite its simplicity, it shows that RUS often performs well, even outperforming SMOTE.

The motivation for proposing RUSBoost as an alternative to SMOTEBoost is because of its drawbacks, complexity, training time, and potential loss of information. SMOTEBoost is more complex and time-consuming than RUSBoost because SMOTE is a more difficult and time-consuming method than RUS. Moreover, SMOTE increases the training time since it requires the construction of an ensemble of models, which can be computationally expensive. RUSBoost, on the other hand, shortens the time necessary to build a model, which is useful when building an ensemble of models. Although random undersampling may result in some loss of information, this drawback mitigates by combining it with boosting. Information missing while constructing one model is likely to be included in other models within the boosting ensemble. The results demonstrate that the more straightforward and quicker RUSBoost algorithm often performs favorably compared to SMOTEBoost, resulting in significantly better classification performance.

Select-Boost

The Select-Boosting strategy, shown in Figure 1, combines feature selection with the boosting technique to improve the performance of weak classifiers on high-dimensional bioinformatics data.

By including feature selection into the boosting process, Select-Boosting addresses the issue of high dimensionality in bioinformatics data. Each instance in the training data has an equal starting weight given to it (for example, 1/a, where a is the total number of instances in the original dataset). The weights of the cases are then changed in accordance with the classification performance of the weak classifier that was trained on the training data. With a data-level strategy, Select-Boosting selects instances from the training data based on how heavily they are weighted. A new training dataset is generated for the current boosting iteration. Instances previously misclassified (i.e., have higher weights) are more likely present in the new training data. In contrast, instances that were correctly classified (i.e., have lower weights) are less likely to be present. This sampling with replacement process allows some cases to appear multiple times, while others may not, depending on the weight distribution.

Feature selection is applied to the new training datasets generated with each boosting iteration.This entails decreasing the number of features required to train the classification model by choosing a subset of pertinent features from the high-dimensional bioinformatics data. This phase is essential to reducing the danger of overfitting and enhancing the model's applicability to new data.Subsequently, a classification model with the selected features is trained on these derived datasets.The posterior probabilities of all the models are averaged to get the final classification model. The models' weights are determined based on their classification performance during the boosting process, where models that perform better receive higher weights. This ensemble of models with selected features and weighted averaging synergistically enhance the classification performance of the weak classifiers on bioinformatics data, leveraging the benefits of both boosting and feature selection playfully and initially.

Select-RUS Boost

Select-RUSBoost can be used as an ensemble approach for addressing the issue of class imbalance in highly imbalanced bioinformatics data. To enhance the performance of supervised classification models, the boosting approach combines feature selection and data sampling. Select-RUSBoost applies random under-sampling (RUS) to the majority class at each boosting iteration to create a fully-balanced dataset.RUS includes gradually eliminating members of the dominant class until the minority class's size is equal to that of the dominant class. This helps balance the class distribution, which is often necessary for bioinformatics data, where one type may be significantly more prevalent.After data sampling, feature selection is applied to the fully-balanced dataset to select the most informative features for building the classification model. Filtering, wrapping, and embedding strategies are methods for choosing the relevant features that are most beneficial to the categorization process. Feature selection is crucial in bioinformatics data, as it can help identify the essential genes, proteins, or other molecular features relevant to the classification problem.

The boosted ensemble trains on the fully-balanced dataset with the selected features. The boosting algorithm, such as AdaBoost or gradient boosting, is used to sequentially train a series of weak classifiers on the fully-balanced dataset, emphasizing misclassified instances. The weight parameter for each sample is adjusted at each iteration based on its classification error. The boosting algorithm focuses on the misclassified instances and gradually improves the classification performance.After boosting iterations, the posterior probabilities of all the models trained on the fully-balanced datasets with the selected features are combined through weighted averaging to construct the final classification model. This integrated model can provide a more accurate and robust prediction on the highly imbalanced bioinformatics data, as it considers the results of multiple weak classifiers trained on balanced datasets with informative features.By incorporating data sampling and feature selection within the boosting algorithm, Select-RUSBoost can effectively address the class imbalance issue in highly imbalanced bioinformatics data and potentially improve the performance of supervised classification models in this context.

The following was found after a thorough empirical research on 15 highly skewed gene expression datasets utilizing two innovative hybrid boosting algorithms, Select-Boosting and Select-RUSBoost, along with two distinct learners, three feature rankers, and four feature subset sizes[1]. The observed performance benefit is minimal, hence it may not be essential to incorporate data sampling methods like Select-Boosting and Select-RUSBoost into the boosting framework. The findings indicate that the difference in performance between Select-Boosting and Select-RUSBoost is not statistically significant. Therefore, including data sampling techniques in the boosting framework may not significantly improve classification performance for highly imbalanced informatics data.

Conclusion and Future work

Ensemble learning methods, combining multiple models to make predictions or decisions, can have drawbacks when applied to machine learning tasks. Ensemble learning methods often require combining various models, increasing complexity in implementation, training, and maintenance. It can make the overall system more challenging to understand and manage. Ensemble learning methods may require more than single-model approaches, as multiple models must be trained and combined. It can increase the time and computational power required for training and inference, which may only be practical in some settings, particularly those with limited computing resources. The issue with ensemble learning methods is overfitting. On unknown data, it could lead to subpar generalization performance.

Ensemble models can be more difficult to interpret than single-model approaches, as they involve combining the outputs of multiple models. It can make it harder to explain the predictions or decisions made by the ensemble, which may be undesirable in specific contexts where interpretability and explainability are essential, such as in regulatory or legal settings. Ensemble learning methods can be computationally expensive, requiring more training, inference time, and resources than single-model approaches. It can increase the cost of implementing and maintaining the ensemble system, which may not be feasible in resource-constrained environments. Ensemble learning methods may need more clarity, as they may involve combining multiple models with different architectures or techniques, making it harder to understand the contribution of each model to the overall prediction. It can reduce the transparency and trustworthiness of the ensemble system, particularly in sensitive or critical applications where model interpretability and accountability are essential. Ensemble learning methods may include similar or redundant models, adding little value to the ensemble. It can increase the system's complexity without necessarily improving its performance, leading to inefficiencies in computational resources and model maintenance. Ensemble learning methods may not always guarantee improved performance over single-model approaches. In some cases, the ensemble may not achieve better accuracy or generalization performance than a single well-tuned model, which may result in wasted effort and resources.

While ensemble learning methods have several potential drawbacks, they also have many benefits and can be highly effective in improving machine learning models' accuracy, robustness, and generalization performance. However, using sampling and ensemble learning together can provide several benefits.Ensemble learning involves combining multiple base models to make predictions, and sampling techniques such as bootstrapping or random undersampling/oversampling can further diversify the training data used for each base model. It can reduce overfitting and increase the generalization ability of the ensemble model, resulting in improved predictive performance.Sampling techniques can help address challenges associated with noisy or imbalanced datasets, where certain classes or samples are underrepresented. Ensemble learning can then combine the predictions of different base models trained on different instances, resulting in a more robust and accurate ensemble model that can handle such data characteristics effectively. Ensemble models can be more stable than individual base models, as they are less likely to be affected by small changes in the training data. When combined with sampling techniques, which can reduce the sensitivity of the models to specific samples, the resulting ensemble model can exhibit improved stability and consistency in its predictions.

Ensemble learning, combined with sampling techniques, can improve prediction performance by leveraging the strengths of multiple base models. Sampling can help identify different patterns in the data, and ensemble methods can then combine these patterns to make more accurate and robust predictions, potentially outperforming individual models. Ensemble methods can be computationally expensive due to the need to train multiple base models. However, using sampling techniques, which can reduce the training data's size, the ensemble learning's computational requirements can be mitigated, making it more scalable and efficient for large datasets or resource-constrained environments.Sampling techniques can be easily combined with ensemble learning methods, such as bagging, boosting, or stacking, providing flexibility in designing ensemble models tailored to specific tasks or domains. Adaptability and customisation are possible, depending on the data's properties and the desired performance objectives.Combining sampling techniques with ensemble learning can improve model diversity, robustness, stability, prediction performance, scalability, and flexibility, making it a powerful approach.

Several standard learning algorithms suffer from poor classification performance due to class imbalance problems, which has led to development to address this issue. These methods include sampling, cost-sensitive learning, and bagging and boosting-based ensembles. However, these conventional approaches may lead to the loss of valuable information, unexpected errors, or increased overfitting risks as they may alter the original data distribution. A novel ensemble method that converts an imbalanced dataset into multiple balanced datasets and builds classifiers on these transformed data using a specific classification algorithm is proposed[4]. These classifiers' classification outputs are then blended based on a specific ensemble rule. The proposed method is compared with other conventional class imbalance handling methods, including sampling methods, cost-sensitive learning, bagging and boosting-based ensembles, fuzzy-rule-based classification methods, and a previous method called EM1vs1, through an empirical study on 46 imbalanced datasets. The results generally show that the proposed method outperforms conventional methods in handling highly imbalanced problems.

In order to manage unbalanced data, modern classification methods built on deep convolutional neural networks (CNNs) frequently use established methods like class resampling or cost-sensitive training. To balance the class distribution before training, the dataset is preprocessed using class resampling methods. To do this, one may oversample the minority class, undersample the majority class, or combine the two. Alternatively, during training, the model is trained using cost-sensitive learning, where the misclassification costs are assigned differently for different classes. Higher misclassification costs are given to the minority class to emphasize its importance during training. The CNN-based model is then trained using the modified dataset or cost-sensitive learning approach, which considers the rebalanced class distribution or the differential misclassification costs. By incorporating these classic strategies into contemporary CNN-based classification methods, the model can better handle imbalanced data and achieve improved performance[5].

Classifying imbalanced data in bioinformatics can be approached in various ways. Although data handling techniques and ensemble methods are suitable, incorporating data handling within the boosting is statistically insignificant. Deep learning methods, like CNNs and using Bagging, and stacking instead of boosting, can be the following approach.

References

[1] https://ieeexplore.ieee.org/document/7424370

[2]<https://ieeexplore.ieee.org/abstract/document/4761297>

[3]<https://ieeexplore.ieee.org/abstract/document/7372202>

[4]<https://www.sciencedirect.com/science/article/abs/pii/S0031320314004841>

[5]https://openaccess.thecvf.com/content\_cvpr\_2016/html/Huang\_Learning\_Deep\_Representation\_CVPR\_2016\_paper.html