2022 International Conference on Machine Learning, Computer Systems and Security (MLCSS) | 978-1-6654-5493-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/MLCSS57186.2022.00035

Progressive Heterogeneous Ensemble Learning for Cancer Gene Expression Classification

Vishal Kore
dept. of Computer Engineering
College of Engineering Pune
Pune, India
vishalkore20@gmail.com

Dr. Vijay Khadse
dept. of Computer Engineering
College of Engineering Pune
Pune, India
vmk.comp@coep.ac.in

***Abstract*—Ensemble learning is an extensively researched subject in machine learning due to its robust and reliable performance. Multiple machine learning models are combined in ensemble learning to improve performance and reliability. There are many algorithms and variations in ensemble learning, but most techniques focus on data space like Bagging, AdaBoost, etc., or feature space like Random Subspace, Attribute Bagging, etc. Traditionally an ensemble of the same learning algorithm is used, but to increase the diversity of base classifiers, using an ensemble of different learning algorithms will be more beneficial. Progressive Subspace Ensemble Learning (PSEL) is investigated in this paper, which combines both data sample and feature space at the same time and applies a sequential selection strategy to select the best classifiers. This work extends PSEL as a heterogeneous ensemble of classifiers to improve performance and reliability for cancer gene expression classification. The proposed work is tested for cancer gene expression classification and performs better than state-of-the-art ensemble techniques.**

***Index Terms*—supervised machine learning, ensemble learning, classification**

I. Introduction

Ensemble learning is a machine learning methodology that combines multiple machine learning models and outputs of these learned models using various combination methods like averaging and voting to achieve better performance than a single machine learning model [1]. The machine learning algorithms used to train these models are called Base Learners. When the identical base learners are used for all models in the ensemble, it is known as Homogeneous Ensemble Learning. When different base learners are used for training models in the ensemble, it is known as Heterogeneous Ensemble learning. Ensemble learning can be used for various tasks such as classification and clustering. Much research has been carried out on ensemble learning. There are many approaches proposed by researchers, including Bagging, Boosting, Random Forests, etc. Progressive Subspace Ensemble Learning (PSEL) is one such approach. Unlike other methods, it simultaneously considers data space and feature space. In PSEL, 'n' random subspaces are created, and 'n' models are trained on these subspaces. These are called the original ensemble models. Then an incremental selection process is applied to select a subset of models from the original ensemble. This subset forms the new ensemble. All the models in PSEL are decision tree classifiers. Thus, it is a homogeneous classifier ensemble [2]. This work extends PSEL by introducing a variety of base learners such as Support Vector Machine, Decision Tree and Naive Bayes, etc.

II. Related work

In this section, related work to the proposed work is discussed. Subsections include ensemble learning, data subsampling, feature subsampling, and heterogeneous ensemble learning.

A. Ensemble learning

Dong et al. [3] surveyed paper on ensemble learning. This study reviewed much research carried out in supervised classification ensembles. They also discuss issues of ensemble learning at different levels like issues at the classifier level (classifier selection, weight adjustment), at the feature level (feature selection, extraction, coverage), at the sample level (subset selection, noisy and imbalanced data), at ensemble level (model complexity, performance metric). After providing the framework for well-known ensemble learning techniques like Bagging, AdaBoost, Random subspace, etc., they discussed research carried out in ensemble learning topics like characteristics of features, feature selection, feature extraction, removing redundant features, reducing dimensionality, combining techniques, variety in base models, improving base models, model complexity, model stability, new performance metrics. The Paper also discussed applications of ensemble learning in various fields like medical science, pattern recognition, and social applications.

B. Data subsampling

Breiman [4] presented a method to combine multiple learning models trained using subsets of data samples. The author calls this method bagging (bootstrap aggregating). In this method, individual models are trained on a subset of data samples drawn randomly with replacement. If existing approaches are unstable, such as when a minor change in training data causes a considerable change in the predictor, bagging improves them. According to the author's previous work, regression, and classification trees, artificial neural networks are unstable, and the k-nearest neighbor is stable. Bagging may perform worse in the case of stable algorithms.

978-1-6654-5493-3/22/$31.00 ©2022 IEEE
DOI 10.1109/MLCSS57186.2022.00035

C. Feature subsampling

Ho [5] proposed a method for constructing an ensemble of decision tree classifiers referred to as decision forest. The ensemble of trees is created using the random subspace method in which random feature subsets are created each time to train individual classifiers. All samples have a constant value in unselected features in the subspace. A decision tree classifier is built on this subspace. For classification, the samples are projected to this subspace. Average conditional probabilities of each class at leaves by trees in the ensemble are used as the final result. Most machine learning techniques suffer from the curse of dimensionality. However, this method performs well on high-dimensional data because more features result in more subspaces, which leads to more trees. The method generalizes well with increasing complexity. Results show that this method outperforms data subsampling techniques like bootstrapping, boosting, and single tree classifiers like C4.5.

Bryll et al. [6] proposed attribute bagging, a wrapper method to improve the accuracy of classification ensemble. Wrapper models use classification algorithms to evaluate random feature subsets, whereas filter models use measures of information content for feature subset evaluation. The proposed method first finds optimal subset size by comparing classification accuracy of varying size subsets. After finding the optimal subset size, randomly selected subsets are evaluated using a classification algorithm. Only classifiers based on highest ranking subsets are used in voting. The proposed approaches are compared against bagging, and single-classifier algorithms on the hand pose recognition dataset.

Reid [7] reviewed heterogeneous ensemble methods. The Base Ensemble Method computes the average of predictions of the base models. The Generalized Ensemble Method discards correlated models and associates weights with each base model. In stacking, a meta-model is trained using input and output from other ensemble members. One paper used a greedy search to select the best N ensemble models from available models. Ensemble pruning using semidefinite programming is one of the approaches used for selecting the models in polynomial time. This can be extended for heterogeneous ensemble learning. Statistical techniques can be used to measure the performance of base models and select a subset of models. One paper proposed a significance index and model selection based on this index. In one paper, the authors proposed global and local competence to construct an ensemble based on model accuracies. Few other proposed Bayesian frameworks to create an ensemble of models.

Considering the requirement for a classifier system that can perform well on most real-world datasets Nanni et al. [8] proposed an approach that combines random subspace and AdaBoost to achieve better performance on most datasets with little to no parameter tuning. The proposed approach achieved comparable results with techniques like Support Vector Machine. The paper also discussed the shortcomings of AdaBoost and various methods used by researchers to tackle them.

Hybrid Adaptive Ensemble Learning (HAEL) is a framework that has been proposed by Yu et al. [9]. HAEL uses two improvement procedures to assign weights to classifiers in the ensemble: base classifier competition and classifier ensemble interaction. The base classifier competition adaptive process (BCCAP) is used for updating the classifier's weight. The classifier ensemble interaction adaptive process (CEIAP) considers the interaction between previously created classifiers and the current classifiers and searches for optimal random subspace. The proposed method uses a random exchange operator to avoid the local optima trap. Authors adopt HAEL to overcome the limitations of random subspace-based classifier ensembles. The proposed method was tested against state-of-art algorithms on the KEEL classification and real-world cancer gene expression datasets.

D. Heterogeneous Ensemble learning

Alshdaifat et al. [10] proposed a method to prune poor performance classifiers from a heterogeneous ensemble of classifiers. The various classification algorithms used in the ensemble were Naive Bayes, Decision Tree, Rule-Based, k-Nearest Neighbors, Artificial Neural Network, and Support Vector Machines. The proposed method creates a heterogeneous ensemble of classifiers using mentioned classifiers. Then poorly performing classifiers are trimmed based on effectiveness measures. The proposed method was a general purpose and was compared with the static best classifier selection method with average probability and majority voting schemes.

After expressing concern with fine-tuning machine learning methods for a specific task, Nanni et al. [11] proposed a general-purpose ensemble of heterogeneous classifiers. Some of the classification approaches are briefly presented. A combination of these techniques was used to compare heterogeneous classifier ensemble performance with other classification techniques. The heterogeneous classifier ensemble outperformed other approaches without any parameter tuning. The downside of such a heterogeneous ensemble is considerable computation time.

Gilpin and Dunlavy [12] conducted an extensive study of heterogeneous classifier ensemble performance using Heterogeneous Ensemble Machine Learning Open Classification Kit (HEMLOCK). HEMLOCK is a tool for evaluating the performance of a heterogeneous classifier ensemble. It uses Weka for the implementation of base classifiers. It currently focuses on the fusion-based heterogeneous ensemble. In fusion function, outputs of all base classifiers are associated with weights. The performance metric considered are confusion matrix, ROC curve, and AUC. Validation methods like holdout, stratified k-fold cross-validation, and bootstrapping are used.

III. Architecture

The proposed methodology consists of 3 phases which are discussed below.

a) ***Original Ensemble generation****:* The training dataset is used to create multiple random subspaces using random subsets of features and data samples. These subspaces are

used to train a set of base models using various classification algorithms such as Support Vector Machine, Decision tree, Naive Bayes, K-nearest neighbors, etc. An equal number of models of each classification algorithm are trained using these random subspaces. This set of trained models forms the original ensemble.

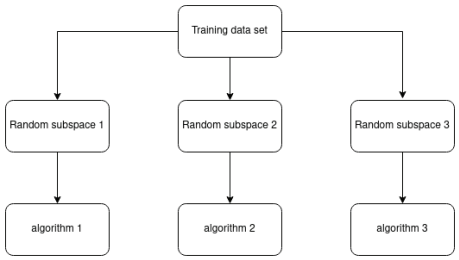

Fig. 1. Flow diagram of training original models

b) ***Progressive selection process****:* A progressive selection process is applied to the original ensemble to incrementally select base models from the original ensemble. Initially, uniform weights are assigned to all samples in training data. The most accurate model from the original ensemble is selected as the first model in the new ensemble. After that, two cost functions are used to incrementally select the models in the new ensemble, namely the current cost function (6) and the long-term cost function (8). After model selection, its weight is calculated based on the misclassifications error (1) and (3). Then sample weights are updated and normalized using (4) and (5). The model is removed from the orginal ensemble and included in the new ensemble with its weight value.

$$wse = \sum_{i=1}^{l} w_i \Theta(O_j, y_i) \tag{1}$$

where wse is the weighted sum error and w_i is the weight of sample x_i from the training set and $\Theta(O_j, y_i)$ is the loss function calculated as:

$$\Theta(O_j, y_i) = e^{-y_i * y\_pred} \tag{2}$$

where O_j is j^{th} model in the original ensemble and y_i is true label and y_pred is predicted value of sample x_i by j^{th} model in the original ensemble.

$$\delta = \frac{1}{2}\, ln\left(\frac{1 - wse}{wse}\right) \tag{3}$$

where δ is the model weight and wse is the weighted sum error.

$$w_i^{g+1} = w_i^g e^{-y_i * \delta * y\_pred_i} \tag{4}$$

where w_i^g is weight value in g^{th} iteration and y_i is true label and δ is the weigh t of selected model and y_pred is predicted value of label for sample x_i

$$\sum_{i=1}^{l} w_i^{g+1} = 1 \tag{5}$$

where w_i^{g+1} is updated weight.

After the first model selection, the current cost function (6) is calculated for all models in the original ensemble. The original ensemble is sorted according to the current cost function. Then each model from the original ensemble is added to the new ensemble to check if the performance of the new ensemble increases using a long-term cost function (8) and (9). If the performance of the new ensemble improves, then the model is added to the new ensemble; else, the next model from the original ensemble is considered. After a model is added to the new ensemble, its weight is calculated, and sample weights are adjusted. The model is removed from the orginal ensemble and included in the new ensemble with its weight value. This process is repeated until a required number of models are added to the new ensemble.

$$\Delta_C(O_j) = \beta_1 * ACC_j + \beta_2 * \Phi(O_j, N_k) \tag{6}$$

where ACC_j is accuaracy of j^{th} classifier on reweighted samples and $\Phi(O_j, N_k)$ gives dissimilarity bewteen j^{th} model in the original ensemble and last model selected in new ensemble which is calculated using Pearson's correlation coefficient (7). β_1 is set as 0.5 and β_2 is set as $1 - \beta_1$.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}} \tag{7}$$

where $\overline{x}$ and $\overline{y}$ are mean of x and y respectively.

$$\Delta_L(O_j) = \sum_{i=1}^{l} |y_i - combpred(x_i)| \tag{8}$$

where y_i is true value for sample x_i and $combpred(x_i)$ gives predicted value of sample x_i (9)

$$combpred(x_i) = argmax_c \sum_{k=1}^{n} \delta^k . 1\{pred(N_k, x_i) = C\} \tag{9}$$

where δ^k is weight associated with k^{th} model in new ensemble and $pred(N_k, x_i)$ gives predicated label for sample x_i by k^{th} model in new ensemble. $1\{.\}$ is defined as $1\{\text{true}\}$=1 and $1\{\text{false}\}$=0.

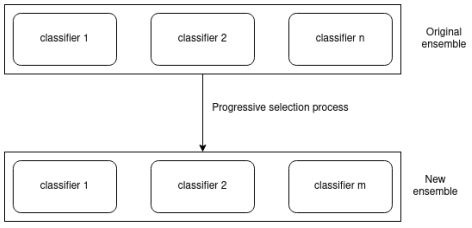

Fig. 2. Flow diagram of selection process

*c) **Combining output**:* For predicting sample labels, a weighted voting mechanism is utilised. The label for a particular sample is predicted by all of the models in the new ensemble. All predicted values are assigned weightage based on the model's weight. The associated weights are summed for each predicted label, and the label with max weightage is selected as the final prediction.

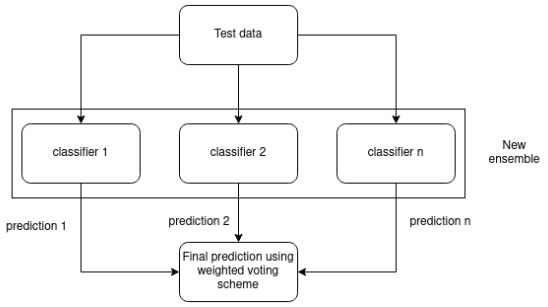

Fig. 3. Flow diagram of Final prediction

IV. Experiment and results

The proposed system is tested on cancer gene expression datasets to compare its performance with PSEL and other state-of-the-art classifiers. The cancer gene expression dataset contains many datasets related to blood, lung, breast, liver, and brain cancer tissues. Some datasets are collected using Affymetrix technology, and others are collected using double-channel technology. The datasets used are shown in Table I. Also, tissue types, the number of samples, features, and classes of each dataset are provided. The cancer gene expression datasets are challenging for classification as they have very few samples but many attributes.

TABLE I
Datasets

Datasets	tissue	samples	attributes	classes
armstrong-2002-v1	Blood	72	1081	2
armstrong-2002-v2	Blood	72	2194	3
chowdary-2006	Breast	104	182	2
golub-1999-v1	Bone marrow	72	1868	2
golub-1999-v2	Bone marrow	72	1868	3
gordon-2002	Lung	181	1626	2
pomeroy-2002-v2	Brain	42	1379	5
shipp-2002-v1	Blood	77	798	2
west-2001	Breast	49	1198	2
alizadeh-2000-v1	Blood	42	1095	2
khan-2001	Multi-tissue	83	1069	4
liang-2005	Brain	37	1411	3

The proposed model is run in a conda environment with the following package requirements:

- Python (3.x)
- scikit-learn
- pandas
- numpy
- tensorflow (cpu)

The proposed ensemble model is run on the datasets given in Table I using a 5-fold cross-validation technique. The training dataset is split into training set and testing set with ratio of 8:2. The 5-fold cross-validation was used to reduce the effect of randomness caused by training and testing splits. The accuracy of the proposed ensemble model is calculated by running a 5-fold cross-validation ten times and averaging those accuracy values. Table II displays the results. On all datasets, the proposed methodology outperforms other ensemble methods.

TABLE II
Results

Datasets	Proposed method	PSEL	Random forest	MultiboostAB	RTBoost
armstrong-2002-v1	0.9928	0.9705	0.9666	0.9667	0.8293
armstrong-2002-v2	0.9886	0.9611	0.9430	0.9085	0.7182
chowdary-2006	0.9744	0.9682	0.9634	0.9673	0.9156
golub-1999-v1	0.9830	0.9623	0.9165	0.9233	0.8267
golub-1999-v2	0.9303	0.9251	0.8874	0.9407	0.5276
gordon-2002	0.9966	0.9917	0.9917	0.9873	0.9193
pomeroy-2002-v2	0.8510	0.7603	0.7358	0.3842	0.4264
shipp-2002-v1	0.9280	0.8561	0.8227	0.8834	0.7346
west-2001	0.9237	0.8631	0.8629	0.8316	0.6596
alizadeh-2000-v1	0.9424	0.8808	0.8836	0.8786	0.6217
khan-2001	0.9911	0.9832	0.9797	0.5400	0.7579
liang-2005	0.9573	0.9232	0.8921	0.8750	0.8254

V. Conclusion and future work

A heterogeneous ensemble model was proposed to achieve better performance and reliability in cancer gene expression classification. The proposed model combined the benefits of the previous homogeneous ensemble model, which considered data and feature space simultaneously. By introducing a variety of base models, the proposed work achieves more diverse models and thus better regularization. The proposed model was tested against state-of-the-art ensemble models and outperformed state-of-the-art methods on all datasets. In the future, we plan to test the proposed work for other fields and will work on further optimizing the performance of the proposed work.

References

[1] Z. Z. H, *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, 2012.

[2] Z. Yu, D. Wang, J. You, H.-S. Wong, S. Wu, J. Zhang, and G. Han, "Progressive subspace ensemble learning," *Pattern Recognition*, vol. 60, pp. 692–705, 2016.

[3] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, no. 2, p. 241–258, 2019.

[4] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, p. 123–140, 1996.

[5] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[6] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291–1302, 2003.

[7] S. R. Reid, “A review of heterogeneous ensemble methods,” 2007.
[8] L. Nanni, S. Brahnam, and A. Lumini, “Double committee adaboost,” *Journal of King Saud University - Science*, vol. 25, p. 29–37, 01 2013.
[9] Z. Yu, L. Li, J. Liu, and G. Han, “Hybrid adaptive classifier ensemble,” *IEEE transactions on cybernetics*, vol. 45, no. 2, pp. 177–190, 2014.
[10] E. Alshdaifat, M. Al-hassan, and A. Aloqaily, “Effective heterogeneous ensemble classification: An alternative approach for selecting base classifiers,” *ICT Express*, vol. 7, no. 3, pp. 342–349, 2021.
[11] L. Nanni, S. Brahnam, S. Ghidoni, and A. Lumini, “Toward a general-purpose heterogeneous ensemble for pattern classification,” *Computational Intelligence and Neuroscience*, vol. 2015, p. 1–10, 2015.
[12] S. A. Gilpin and D. M. Dunlavy, “Heterogeneous ensemble classification,” *CSRI SUMMER PROCEEDINGS 2008*, vol. 90, 2008.