Proceedings of the Federated Conference on Computer Science and Information Systems pp. 149–152
DOI: 10.15439/2017F564
ISSN 2300-5963 ACSIS, Vol. 11

An ensemble model with hierarchical decomposition and aggregation for highly scalable and robust classification

Quang Hieu Vu
Zalora, Singapore
quanghieu.vu@zalora.com

Dymitr Ruta
EBTIC, Khalifa University, UAE
dymitr.ruta@kustar.ac.ae

Ling Cen
EBTIC, Khalifa University, UAE
cen.ling@kustar.ac.ae

***Abstract*—This paper introduces an ensemble model that solves the binary classification problem by incorporating the basic Logistic Regression with the two recent advanced paradigms: extreme gradient boosted decision trees (xgboost) and deep learning. To obtain the best result when integrating sub-models, we introduce a solution to split and select sets of features for the sub-model training. In addition to the ensemble model, we propose a flexible robust and highly scalable new scheme for building a composite classifier that tries to simultaneously implement multiple layers of model decomposition and outputs aggregation to maximally reduce both bias and variance (spread) components of classification errors. We demonstrate the power of our ensemble model to solve the problem of predicting the outcome of Hearthstone, a turn-based computer game, based on game state information. Excellent predictive performance of our model has been acknowledged by the second place scored in the final ranking among 188 competing teams.**

I. Introduction

RECENT Internet of Thing revolution coupled with the emergence of big data technologies present new opportunities to the process of automated data-driven decision making, especially in the presence of multiple sources and different types of data that normally require professional human skills and experience to process. As more and more data becomes available, further gains in classification performance are becoming possible but depend on the ability of the model algorithm to better reconstruct the relationship function between the inputs and outputs (targets) while dealing with typically noisier and more conflicting evidence and much larger computational overhead.

Many existing state-of-the-art Machine Learning models successfully take advantage of this extra evidence to reduce either bias (Deep Learning) or variance (Extreme Gradient Boosted Decision Trees) component of classification error, but fail to reduce both to the extend that would offer significant boost in predictive performance and its confidence. Besides, these models, typically implementing $\geq$O(n^2) learning algorithms simply lack scalability and often are intractable when faced with big data sizes that limit their utility down to small samples and typically exclude them from real-time apps.

The model we introduce in this work aims to address above-mentioned gaps and tries to significantly reduce both bias and variance at manageable and scalable computational footprint. We start our model introduction from a proposition of ensemble model along with rules that govern feature selection and model decomposition. Then, we introduce a simple classification training structure that uses robust but simple linear base classifier to leverage decomposed and ensemble based training to achieve the trade-off between bias and variance reduction with virtually no impact on the computational complexity of the original model. Both of our proposed methods can be used either independently or complementary to each other. To evaluate the performance of our proposed solutions, they were applied in a competition to predict the likelihood of winning a turn-based computer game: Heartstone [1], given intra-game states for both players of the game [2]. The second place our model scored in this competition has objectively proven its excellent design and predictive performance capabilities which surpassed other academic state-of-the-art solutions and off-the-shelf commercial tools proposed by 188 competing teams from all over the world. In summary, our paper brings the following two main contributions.

- An ensemble model that incorporates Logistic Regression, XGBoost and DL to solve the binary classification problem, along with the capability to decompose the model training along specially selected feature subsets.
- A hierarchical decomposition and aggregation scheme for highly scalable and robust classification and a discussion of how to use it in the case of logistic regression model.

The remainder of the paper is organized as follows. In Section II, we introduce related work. In Sections III and IV, we present our proposed ensemble model and the training scheme, respectively. In Section V, we demonstrate an application in a objectively evaluated competition setup as a case study for our proposed solutions. Finally, we draw some concluding remarks in Section VI.

II. Related Work

In this section, the machine learning approaches used in our model, specifically, Logistic Regression, XGBoost and Deep Learning, are briefly introduced.

A. Logistic Regression

Logistic regression is a statistical method for regression analysis to describe the relationship between one dichotomous

dependent variable (outcome) and one or more independent variables (predictors or features). Binary logistic model can be used for estimating the probability of a binary response based on predictors and gain insights on the factors that increase the probability of a given outcome. Logistic regression has been widely used in various areas, e.g., assessing injury mortality or severity for patients [4], predicting votes based on their characteristics such as age, income, sex, race, state of residence, previous votes, etc. [5], estimating probability of failure in various processes, systems or products [6], predicting customers' propensity to purchase a product or cease a subscription in marketing applications [7], etc..

B. Bagging and Boosting Methods

Bagging and Boosting are powerful meta-algorithms used in machine learning to improve prediction accuracy of classification models by combining a set of week classifiers with poor performance, unstable predictions, and high rate of misclassification error, into a strong and robust "wide margin" predictive model. Bagging can decrease the variance of unstable procedures and prediction outcomes, while boosting is an effective way to reduce prediction bias [8], [9]. Gradient boosting (GB) is a version of boosting method, which like in standard boosting uses an ensemble of weak prediction models, typically decision trees, yet manages to achieve deeper performance gains beating many other state-of-the-art predictors in a wide range of commercial and academic applications [10]. XGBoost, based on Extreme Gradient Boosting model [11], is an implementation of the gradient boosted decision trees algorithm with a goal to push the limit of computations resources for boosted tree algorithms [12], which recently has been used by many winning teams of a number of machine learning competitions, e.g. [13], due to its advantages of fast processing speed and high prediction accuracy.

C. Deep Learning

Deep Learning refers to a class of machine learning techniques and architectures, where many layers of non-linear information processing stages in hierarchical architectures are exploited for pattern classification and for feature or representation learning [14]. Unlike the conventional classification algorithms which heavily rely on feature extracting techniques, deep learning techniques could characterize the high-order correlation properties of the observed or visible data for pattern analysis or synthesis purposes, and/or characterize the joint statistical distributions of the visible data and their associated classes, which learn feature representations without the need of labeled data referring to unsupervised feature learning, and thus avoid substantial effort on hand-designing features [14]. In recent years, DL techniques have gain increasing attention and popularity due to drastically increased chip processing abilities (e.g. GPU units), significantly lowered cost of computing hardware and recent advances in research of machine learning and signal/information processing. They have been successfully applied in various areas, e.g. visual object recognition, image processing, speech recognition, handwriting recognition, natural language processing, information retrieval, etc. [14].

In the subsequent sections, we will introduce the proposed ensemble model that combines the advantages of Logistic Regression, XGBoost and Deep Learning. We will then demonstrate how it is able to reduce both variance and bias components of the classification error that enables to achieve improved and consistent prediction accuracy when solving binary classification problems.

III. Ensemble Model

Even though ensemble model is not a new concept since it has been extensively used and reported to win top prizes in many recent data mining competitions, there is no clear instruction of how to build a reliable ensemble model that would consistently outperform other predictors. In this part, we propose a general approach that tries to addresses this gap. However, before going into details of our proposal, we would like to emphasize a couple of important points for building an ensemble model as follows.

- Different sub-models in the ensemble model could be trained with different sets of features and examples to leverage the maximum benefits of the ensemble. Predictive performance improvements achieved by different models trained on the same features are possible although limited by the inability of the predictive model to match the evidence it is most compatible to work with.
- The sub-models can be aggregated in a number of ways, of which the most popular ones are averaging and stacking. By averaging, the final result is simply generated by getting the average from sub-model results. Stacking requires a more comprehensive train in the next layer using the results of sub-models.

Our proposed ensemble model in this paper focuses on how to split the feature set into sub-sets for training with different sub-models. As a result, it works with any method for combining results from sub-models. In our approach, we first perform feature selection to obtain a set of useful features for training models. Assuming that a total number of f features are selected and an ensemble model is built with n sub-models, two basic rules to select features for training the sub-model are described as follows.

- The set of f features are split into subsets, each of which contains f' number of features, defined as $f' = k \times \frac{f}{n} \pm t$ where k and t can be any value between 1 and n and decided on the course of cross-validation performance evaluation. Feature selection is applied to choose features for each subset.
- Each set of features should be used in at least two sub-models to increase the accuracy of the ensemble output.

It is interesting to note that our proposed approach for splitting feature set in training sub-models is actually similar to the cross-validation method when we leave a subset of data for validation. By splitting the feature set and training data in this way, we can leverage the maximum benefits of training

sub-model separately and obtain the best ensemble result from the combination of sub-models' results.

IV. Hierarchical decomposition and aggregation

The model introduces hierarchical training structure S involving decomposition and/or aggregation of the training data into exclusive sets of examples either along distinct values of one or more feature combinations or randomly along k-exclusive subsets of examples. The training structure upon the dataset X $S_D^P(X)$ is defined by two parameters: partitioning criterion P and the degree of partitioning D. Partitioning P can proceed either independently of the feature values ($P = 0$) or along all unique values of the feature F_P. For data independent partitioning ($P = 0$), the degree of partitioning D determines the number of exclusive equal-sized parts the training set will be split and trained following D-fold cross-validation for positive D or otherwise inverse $|D|$-fold cross-validation that we define simply by training on exclusive $|D|$ subsets of the training data. In case of feature value-based decomposition ($P > 0$), the partitioning degree D informs whether the unique sorted values of F_P feature should be grouped in exclusive set of subsequent $|D|$-sized groups. Then for each such grouped subset the training follows on either actual subset if D is negative or on the complement of such subset if D is positive. In both cases the degree of partition have similar effect of training on multiple overlapping subsets for positive D, or on exclusive subsets for negative D, thereby controlling the level of aggregation or decomposition in the training process.

Please note that such defined training structure operator S_D^P can be combined into sequential expressions defining open hierarchical training structure with virtually infinite number of variants left to be designed for skilled data scientist. Note also that the enumerated parameterized representation of the structure parameters allows for easy iteration procedure to traverse through the structure parameters in a search that maximizes the expected predictive performance that can be carried by an automated ML model designer.

Finally having defined the expression mechanism for creating training structures what is left to define a full classification model M is to pair it with the base classifier C such that the fully defined classification model becomes: $M = (C, S)$.

V. A case study

To demonstrate the performance of our proposed model as well as the training scheme, we apply it to build a model that predicts the result (the winner) of a computer game, Heartstone [1], given the input data of various intra-game states [2].

A. Feature Engineering

Before building the prediction model, it is important to analyze the data and perform feature engineering to extract maximum predictive power from the raw data. In this case study, we had over three million records that store data about different intra-game states of Hearthstone. The data is recorded at each game state represented by the turn number of the game and cover three sets of basic features with a total number of 40 features as follows.

- Opponent properties: hold information about different properties of the opponent at the current state as well as statistical data about played cards of the opponent.
- Player properties: keep similar information about different properties of the player at the current state and statistical data about played cards of the player.
- Player holding card information: this type of data is only available for the player.

These basic features are then complemented with an additional of 121 features generated in the following ways:

- Difference features: the different values of common numerical features between the opponent and the player.
- Player holding card features: the statistics of different types of cards in hand of the player. For these features, we simply count the number of holding cards in each type of the player.

All of the 161 features presented above were then exposed to various feature selection techniques. Different subset of features were selected for different sub-models that are vital unlock maximum predictive power from every model as well as inject diversity that is reported to be quite beneficial when combining multiple classifiers as discussed in Section III. Specifically, for each of the three sub-models we used the following different set of features:

- The set of 53 features - approximately equal to one third of the total number of features.
- The set of 107 features - approximately equal to two third of the total number of features.
- The set of all 161 features that include both basic features and extra features.

It is important to note that for the first two incomplete feature sets, the features were selected based on a combination (union) of feature selection for the top K_1 (KBest) and recursive feature elimination for the bottom K_2 (RFE). In particular, we selected 53 features for the first set from the top $K_1 = 50$ and the bottom $K_2 = 50$ in the KBest and RFE selection methods. On the other hand, the 107 features selected for the second set come from the top $K_1 = 100$ and the bottom $K_2 = 100$ in the KBest and RFE selection methods.

B. Evaluation of the ensemble model

Our ensemble model in this case study was built from 6 separate prediction models built on Logistic Regression, XGBoost and Deep Neural Network.

- Logistic regression: this approach is used for two prediction models. The first model is trained on a set of selected 53 features, decomposed along the $player.hero\_card\_id$ feature. The second model is trained on a set of selected 107 features, decomposed a long the $opponent.hero\_card\_id$ feature. These models respectively receive a score of 0.7963 and 0.7967 from the public leader board.

- XGBoost (eXtreme Gradient Boosting): this approach is used for the next two predictions models, which are trained respectively on a set of 53 original features and all 161 features with scores 0.7964 and 0.7956 in the public leader board.
- Deep neural network: this approach is use for the last two prediction models. The first model is trained on a set of 53 features with three layers: an input layer using relu activation, an hidden layer of 20 nodes using relu activation and an output layer using sigmoid activation. This model scores 0.7957 in the public leader board. The second model is trained on a set of selected 107 features. Since there are more features, this model has an extra hidden layer with 60 nodes using relu activation in between the input layer and the hidden layer of 20 nodes. This model scores 0.7968 from the public leader board.

C. Evaluation of the hierarchical decomposition and aggregation scheme

We have tested such hierarchical architecture for a classification model build with logistic regression as a base classifier and obtained the best results for the following design:

$$M = (LogReg, S^0_{-30}(S^4_1(X), S^{16}_1(X))) \quad (1)$$

Deciphering the structure expression of Logistic Regression $S^0_{-30}(S^4_1(X), S^{16}_1(X))$ in plan words means that the predictor is constructed by an aggregation of the two double-decomposed models: first along the unique values of feature $opponent.hero\_card\_id$ and feature $player.hero\_card\_id$ and then further into 30 unique random subsets trained exclusively in an inverse cross-validation fashion. These models generated trained classifiers that back-tested with the highest cross-validation accuracy and have been applied to classify the testing set yielding a score of 0.797. What is intriguing is that such deep decomposition as in the presented design leads to decomposition of over 3m data points into over 270 chunks of the size around 10000. Such large model fragmentation is perfect for extremely fast processing on the parallelized infrastructure and delivered very competitive prediction results literally in seconds. Note that it is interesting to have the following observations from the results

- There is no surprise that the decomposition along unique values of feature $player.hero\_card\_id$ and $opponent.hero\_card\_id$ improves the model performance. It simply means that playing as a different hero character with all its specific characteristic requires distinct set of model parameters that appear to improve the predictive performance of the game outcome if applied only to the same cases of games played with the same character. This type of decomposition is a clear proof of the bias classification error reduction through improved specificity of the models trained on significantly distinct subsets (clusters) of data.
- It appears surprising that further training set decomposition into 30 smaller subsets of around 10000 each leads to the improvement of predictive performance rather than training on most or all of the available training set. Indeed the experiments confirmed an optimal decomposition and aggregation level obtained for the training sets at around 10000 game states examples. Both, building and aggregating fewer models with larger training sets and more models trained on smaller training subsets results in apparent degradation of predictive performance.
- The identified structure parameters appear to achieve the optimal trade-off between the bias and variance error components reduction subject to logistic regression classifier abilities

VI. Conclusion

In this paper, we have introduced an ensemble model for binary classification with a clear solution of how to split and select features for sub-model training. In addition to the ensemble model, we present an approach for hierarchical decomposition and aggregation model to address the issue of slow and computationally intractable in model training. These proposed solutions have been proved to be good with the second prize in a recent competition, which predicts the likelihood of winning a game given intra-game states of players. Even though our proposed solutions were proved to be good, they are not fully automated. Thus, in our future work, we plan to extend this current work for the automated feature selection of the ensemble model as well as efficient search for the best possible training structure with respect to the hierarchical decomposition and aggregation model.

References

[1] Hearthstone, http://us.battle.net/hearthstone/en/
[2] AAIA'17 Data Mining Challenge: Helping AI to Play Hearthstone, https://knowledgepit.fedcsis.org/contest/view.php?id=120.
[3] D.R. Cox, "The regression analysis of binary sequences (with discussion)," *J Roy Stat Soc B.*, vol. 20, pp. 215–242, 1958.
[4] C.R. Boyd, M.A. Tolson, and W.S. Copes, "Evaluating trauma care: The TRISS method. Trauma Score and the Injury Severity Score," *The Journal of trauma*, vol. 27, no. 4, pp. 370–378, 1987.
[5] F.E. Harrell, Regression Modeling Strategies, Springer-Verlag, ISBN 0-387-95232-2, 2001.
[6] M. Strano, B.M. Colosimo "Logistic regression analysis for experimental determination of forming limit diagrams," *International Journal of Machine Tools and Manufacture*, vol. 46, no. 6, pp. 673–682, 2006.
[7] M.J.A. Berry, "Data Mining Techniques For Marketing, Sales and Customer Support," Wiley, pp 10, 1997.
[8] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.
[9] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting Algorithms as Gradient Descent," in S.A. Solla, T.K. Leen, and K.R. Muller, editors, Advances in Neural Information Processing Systems 12, pp. 512-518, MIT Press.
[10] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean, "Boosting Algorithms as Gradient Descent" In S.A. Solla and T.K. Leen and K. Müller, Advances in Neural Information Processing Systems 12. MIT Press. pp. 512–518, 1999.
[11] J.H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189-1232, 2001.
[12] XGBoost, https://github.com/dmlc/xgboost/.
[13] XGBoost:Machine Learning Challenge Winning Solutions, https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions, Retrieved 2016-08-01.
[14] L. Deng, "Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey," *APSIPA Transactions on Signal and Information Processing*, 2012.