2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

Stacking Approach for a Robust Prediction Model of Visualized Data Sets

Megat Norulazmi Megat Mohamed Noor [1], Shaidah Jusoh [2]

[1] College of Arts and Sciences,
Universiti Utara Malaysia
Sintok, Kedah, Malaysia
E-mail: megatnorulazmi@gmail.com

[2] College of Arts and Sciences,
Universiti Utara Malaysia
Sintok, Kedah, Malaysia
E-mail: shaidah@uum.edu.my

Abstract. In this paper, to produce better performance of F-score measurement, we applied the stacking concept by stacking the robust prediction model data sets instead of stacking multiple learner schemes with single data set. We are using K* Learning algorithm as a single base classifier to train multiple combine visualized data sets which were produced the best prediction outcome from respective area of expertise. The result shows that the proposed stacking approach improves the F-score measurement result compare to previous approach of individual prediction model datasets.

Keywords: Stacking concepts; multiple learner classifier schemes; visualized data sets; prediction model

1. Introduction

Our previous study shows that a prediction model of yield outcome in a manufacturing process can be achieved by generating visualized pattern data sets with our novel technique [8] from historical numerical data generated by the inspection machine. Further study on handling the imbalance nature of manufacturing data set [9] shows that the combination of random over-sampling and under-sampling technique were producing robust prediction model. The model was capable to predict different batches of test data, but one sided random over-sampling and under-sampling were failed to perform due to over-fitting issue. Our novel approach of Synthetic Majority Replacement Technique (SMaRT) and Synthetic Minority Over-sampling Technique (SMOTE) with the integration of K* based entropy similarity distance function and Tomek Links + CNN algorithm [10] improved the robustness of the classifiers prediction F-score. By combining the SMaRT and SMOTE with Tomek Links, CNN, random sampling and one-sided selection approach, they produced robust prediction models that performed their superiority at different area.

The goal of this paper is to propose a stacking approach for a robust prediction model of visualized data sets. This paper is organized as follows. Section 2 addressed the related work of stacking algorithms. Section 3 presents our proposed techniques and steps for stacking multiple data sets. Experiment procedures and results analysis are presented in Section 4. The conclusion of the work is presented in Section 5.

2. Related work

The ensembles of multiple learner classifier schemes with stacking approach recently were being intensively done by most of researchers to improve the prediction performance. The stacking generalization algorithm was successfully proposed by [4] and further improved by [1] with StackC algorithm and Ljupco [7] with Meta Decisions Tree Stacking algorithm.

The stacking algorithm [4] is to use the predictions of the base classifiers as attributes for a new training of the original class labels. Stacking combines output from a set of base classifiers with a single meta-classifier. [4] defined that the original data and the models built for the first step are referred to as level-0 data and level-0 models. While the set the second-stage learning algorithm is known to as level-1 data and the level-1 generalizer. The two issues that need to consider are; firstly is the type of attributes that should be used to form level-1 data, and secondly is the type of level-1 generalizer which is to improve the accuracy using the stacked generalization method. The approach extended by using class probability distributions of base classifiers which carries the prediction confidence for all classes. The approach was found to be done by [15] where they were using Multi-response Linear Regression (MLR) as meta-classifier. The use of class probabilities is to be crucial for the successful application of stacked generalization in classification tasks.

According to [3], the task of constructing ensembles of classifiers can be broken down into two sub-tasks. Firstly is to generate a diverse set of base-level classifiers and then, after the base-level classifiers have been generated, the matter of how to combine their predictions arises. There are many possible approaches to generating base-level classifiers. The approaches are; to generate classifiers by applying different learning algorithms to a single data set as performed by [12]. Next possible approach is to apply a single learning algorithm with different parameters settings to a single data set. Finally, the approaches such as bagging [2] and boosting [5] generate multiple classifies by applying a single learning algorithm to different versions of a given data set.

As previously mentioned, in stacking, a learning algorithm is used to learn how to combine the predictions of the base-level classifiers. The induced meta-level classifier is then used to obtain the final prediction from the predictions of the base-level classifiers. The work presented by [7] focuses on combining the predictions of base-level classifiers induced by applying different learning algorithms to a single data set.

3. Proposed approach

In this work, we are using the same data as in the previous study [10], TomekCnnUs, SmoteSmartTmkCnnUs300, SmoteCnnUs50, and SmoteSmartCnnUs50 were the best performer at their respective area. However, in this work, we ensemble them to generate multiple datasets by combining the data sets together. The proposed approach is based on the techniques in the following steps.

- Step1) Stacking Base Classifiers with a Dataset - Datasets TomekCnnUs, SmoteSmartTmkCnnUs300, SmoteCnnUs50, and SmoteSmartCnnUs50 from our previous study [10] were trained with multiple stacked base-classifiers.
- Step2) Stacking a Base Classifier with Stacking Datasets - Combined TomekCnnUs, SmoteSmartTmkCnnUs300, SmoteCnnUs50, and SmoteSmartCnnUs50 datasets were trained by stacking algorithm with a single base-classifier.
- Step3) Stacking a Base Classifier with CNN Stacking Datasets - Condensed Nearest Neighbor (CNN+ K*) [10] was used to clean the combined datasets SmoteSmartTmkCnnUs300, TomekCnnUs, SmoteCnnUs50, and SmoteSmartCnnUs50. The combined datasets were trained by stacking algorithm with a single base-classifier.
- Step 4) Stacking a Base Classifier with Tomek Stacking Data sets - Tomek link + K* [10] was used to clean the combined datasets SmoteSmartTmkCnnUs300, TomekCnnUs, SmoteCnnUs50, and SmoteSmartCnnUs50. The combined datasets were trained by stacking algorithm with a single base-classifier.
- Step 5) Stacking Base Classifiers with Stacking Datasets - Combination of TomekCnnUs, SmoteCnnUs50, SmoteSmartTmkCnnUs300, and SmoteSmartCnnUs50 datasets were trained with multiple stacked base-classifiers.
- Step 6) Stacking Base Classifiers with CNN Stacking - Combination of TomekCnnUs, SmoteCnnUs50, SmoteSmartTmkCnnUs300, and SmoteSmartCnnUs50 datasets were cleaned with CNN+ K* [10] and trained with multiple stacked base-classifiers.

- Step7) Stacking Base Classifiers with Tomek Stacking Datasets - Combination of TomekCnnUs, SmoteCnnUs50, SmoteSmartTmkCnnUs300, and SmoteSmartCnnUs50 datasets were cleaned with Tomek+ K* [10] and trained with multiple stacked base-classifiers.

4. Experiments and Results

4.1. Experiment Procedure

We used Wolpert, Stacking and Seewald, StackingC techniques with default meta-classifier scheme and K* meta-classifier scheme setting. For the stacking of multiple base-classifier, we applied K*, LWL and IBk which were the best performer in our previous study [8]. K* algorithm (non-stacking algorithm) was used for a single stacking as the base-classifier and also was used as a comparison to the stacking scheme.

Training datasets were trained as suggested in [8] for the learning process with confusion matrix and stratified 10-fold cross validation. Classifiers generated from the training data were then being used to be tested with test data [9] [10] from the same batch of the training data sets. The classifiers once again were being tested with two test data sets from different batches to test the robustness of the classifiers. The results of the training and prediction test of this paper were compared to our previous work [9] [10] to verify the improvement of the F-Score measure on the classifiers robustness.

4.2. Result Analysis

Table 1, shows that individual selected datasets from our previous study [10] which were trained with stacking multiple base-classifiers did not performed significantly better or inconsistently performed better compared to previous [10] result.

Table 2 to 7 show the training and test results for the approach techniques of Step 2 to Step 4 where a single base-classifier (K*) was used in the stacking algorithm with multiple stacked selected [10] data sets. The results show that the combined dataset produces improved performance especially with the training and test data. The combined dataset which was cleaned with CNN-K* indicates better performance. However, the one which was cleaned with Tomek did not show better performance but indicates an interesting behavior where all the results performed at a same level even with different algorithms. The results also show that the class recall performances are at the maximum level while the class precision are at lower level with varying result between different batches of testing data sets.

Table 7 to 13 indicates the training and test results for the approach techniques of Step 5 to Step 7 where multiple base-classifiers (K*) was used in stacking algorithm with multiple stacked selected [10] data sets. The results show that the combined data set produces improved performance, compare to the individual data set result while maintaining prediction robustness. However the results did not significantly improve compare to single base-classifier. The combined data set which was cleaned with CNN-K* [10] have shown better performance while the one which was cleaned with Tomek did not show better performance but indicates same behavior. The results performs same even with different algorithms, and class recall performances are at the maximum level while the class precision is at lower level with varying result between different batches of testing data sets.

Data Sets	Performance		
	R	P	F
TomekCnnUs			
Stacking-Wolpert	1.000	0.259	0.412
Stacking-Wolpert-Kstar	0.964	0.267	0.419
StackingC	1.000	0.259	0.412
StackingC-Kstar	1.000	0.259	0.412
SmoteSmartTmkCnnUs300			
Stacking-Wolpert	1.000	0.259	0.412
Stacking-Wolpert-Kstar	0.500	0.298	0.373
StackingC	0.500	0.298	0.373
StackingC-Kstar	0.500	0.298	0.000
SmoteCnnUs50			
Stacking-Wolpert	1.000	0.259	0.412
Stacking-Wolpert-Kstar	0.321	0.360	0.340
StackingC	0.036	0.125	0.056
StackingC-Kstar	0.036	0.100	0.167
SmoteSmartCnnUs50			
Stacking-Wolpert	0.000	0.000	0.000
Stacking-Wolpert-Kstar	0.107	0.177	0.133
StackingC	0.036	0.125	0.056
StackingC-Kstar	0.036	0.111	0.054

Table 1: Training result from stacking base-classifier with single data classifier.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	0.000	0.000	0.000
Stacking-Wolpert-Kstar	0.994	0.996	0.995
StackingC	0.998	0.931	0.963
StackingC-Kstar	0.997	0.932	0.964
Non Stacking-Kstar	0.998	0.927	0.961

Table 2: Training result of stacking a base-classifier with stacking data sets.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	0.000	0.000	0.000
Stacking-Wolpert-Kstar	0.840	0.875	0.857
StackingC	0.880	0.512	0.647
StackingC-Kstar	0.880	0.512	0.647
Non Stacking-Kstar	0.840	0.539	0.656

Table 3: Test data result from stacking a base-classifier and stacking datasets

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.757	0.862
Stacking-Wolpert-Kstar	1.000	0.999	0.999
StackingC	1.000	0.914	0.955
StackingC-Kstar	1.000	0.929	0.963
Non Stack-Kstar	1.000	0.805	0.892

Table 4: Training result of stacking a base-classifier and stacking data sets with CNN.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.122	0.217
Stacking-Wolpert-Kstar	1.000	1.000	1.000
StackingC	1.000	0.219	0.360
StackingC-Kstar	1.000	0.532	0.694
Non Stack-Kstar	1.000	0.192	0.323

Table 5: Test data result from stacking a base-classifier and stacking data sets with CNN.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.9998	1.000
Stacking-Wolpert-Kstar	1.000	0.9998	1.000
StackingC	1.000	0.9998	1.000
StackingC-Kstar	1.000	0.9998	1.000
Non Stacking-Kstar	1.000	0.9998	1.000

Table 6: Training result of stacking a base-classifier and stacking data sets with tomek.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.122	0.217
Stacking-Wolpert-Kstar	1.000	0.122	0.217
StackingC	1.000	0.122	0.217
StackingC-Kstar	1.000	0.122	0.217
Non Stacking-Kstar	1.000	0.122	0.217

Table 7: Test data result from stacking a base-classifier and stacking data sets with tomek.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	0.000	0.000	0.000
Stacking-Wolpert-Kstar	0.995	0.997	0.996
StackingC	0.999	0.940	0.968
StackingC-Kstar	0.999	0.980	0.990
Non Stack-Kstar	0.998	0.927	0.961

Table 8: Training result of stacking base-classifiers with stacking data sets.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	0.000	0.000	0.000
Stacking-Wolpert-Kstar	0.840	0.913	0.875
StackingC	0.880	0.524	0.657
StackingC-Kstar	0.880	0.688	0.772
Non Stack-Kstar	0.840	0.539	0.656

Table 9: Test Data result from stacking base-classifier and stacking data sets.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.757	0.862
Stacking-Wolpert-Kstar	1.000	0.998	0.999
StackingC	1.000	0.890	0.942
StackingC-Kstar	1.000	0.890	0.942
Non Stack-Kstar	1.000	0.805	0.892

Table 10: Training result of stacking base-classifiers and stacking data sets with CNNTABLE 10.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.122	0.217
Stacking-Wolpert-Kstar	1.000	0.893	0.943
StackingC	1.000	0.313	0.476
StackingC-Kstar	1.000	0.313	0.476
Non Stack-Kstar	1.000	0.192	0.323

Table 11: Test data result from stacking base-classifiers and stacking data sets with CNN.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.9998	1.000
Stacking-Wolpert-Kstar	1.000	0.9998	1.000
StackingC	1.000	0.9998	1.000
StackingC-Kstar	1.000	0.9998	1.000
Non Stack-Kstar	1.000	0.9998	1.000

Table 12: Training result of stacking base-classifiers and stacking data sets with tomek.

Data Sets	Performance		
	R	P	F
Stacking-Wolpert	1.000	0.122	0.217
Stacking-Wolpert-Kstar	1.000	0.122	0.217
StackingC	1.000	0.122	0.217
StackingC-Kstar	1.000	0.122	0.217
Non Stacking-Kstar	1.000	0.122	0.217

Table 13: Test data result from stacking base-classifiers and stacking data sets with tomek.

Conclusion

The experiment results conclude that, multiple base-classifiers stacking did not produced better prediction performance compared to the single dataset. However by stacking the multiple datasets, the prediction performance improved to a better result even with a single base-classifier.

Multiple stacking datasets performs much better by removing the redundant good yield (majority instances before been balanced) instances with CNN+K* algorithm. However by applying Tomek Link+K* algorithm to remove the noisy and borderline good yield instances, the class precision from the test result performed lower and fixed even with different algorithms.

From abovementioned results, it can be concluded that we might be able to get the best prediction performance by stacking multiple datasets and keep it up to date to the current and similar (same manufacturing area) batch datasets. It is seems that, removing noisy and borderline instances with Tomek links was not successful which caused different algorithms performed the same. We might be able to achieve better performance by correctly remove the noisy or borderline (removing both good and bad yield) instances or by using another technique and scheme.

Acknowledgment

This research is supported by the Yayasan Ilmuwan and Fundamental Research Grant Scheme, Malaysia Ministry of Higher Education.

References

[1] Alexander K. Seewald. *How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness*. Machine Learning, ICML 2002, 554-561, University of NSW, Sydney, Australia, July 8-12, 2002.

[2] J Breiman, L. *Bagging predictors.* Machine Learning, 24:2, 123–140, 1996.

[3] Dietterich, T. G. (1997). Machine-learning research: Four current directions. AI Magazine, 18:4, 97–136.

[4] David H. Wolpert: *Stacked generalization*. (1992) NeuralNetworks 5(2): 241-259

[5] Freund, Y., &Schapire, R. E. *Experiments with a new boosting algorithm*. In Proceedings of the Thirteenth International Conference on Machine Learning. San Mateo, CA: Morgan Kaufmann. 1996

[6] John G. Cleary and Leonard E. Trigg. *"{K}*: an instance-based learner using an entropic distance measure", Proc.12th International Conference on Machine Learning",* Morgan Kaufmann 108—114 1995.

[7] Ljupco Todorovski, Saso Dzeroski: *Combining Classifiers with Meta Decision Trees.* Machine Learning 50(3): 223-249 (2003)

[8] Megat Norulazmi Megat Mohamed Noor, Shaidah Jusoh,*Visualizing the Yield Pattern Outcome for Automatic Data Exploration, ams*, pp. 404-409, 2008 Second Asia International Conference on Modeling & Simulation.

[9] Megat Norulazmi Megat Mohamed Noor, Shaidah Jusoh, *Handling Imbalance Visualized Pattern Dataset for Yield Prediction"* 3rd International Symposium on Information Technology 2008, 26 -29 August

[10] Megat Norulazmi Megat Mohamed Noor, Shaidah Jusoh, *Improving F-Score of the Imbalance Visualized Pattern Dataset for Yield Prediction Robustness,* 21st International CODATA Conference, Kyiv, Ukraine, 2008.

[11] Merz, C. J. *Using correspondence analysis to combine classifiers.* Machine Learning, 36:1/2, 33–58. Dordrecht: Kluwer Academic Publishers, 1999.

[12] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* 16 321–357.

[13] Tomek, I.: Two Modifications of CNN, *IEEE Trans.Systems*, Man and Cybernetics, SMC.6, pp. 769-772 (1976).