Predicting Classifier Combinations

Matthias Reif[1], Annika Leveringhaus[2], Faisal Shafait[1], and Andreas Dengel[1]
[1]*German Research Center for Artificial Intelligence, Trippstadter Strasse 122, 67663 Kaiserslautern, Germany*
[2]*Department of Computer Science, University of Kaiserslautern, 67663 Kaiserslautern, Germany*
{*matthias.reif, faisal.shafait, andreas.dengel*}*@dfki.de, a_lever@informatik.uni-kl.de*

Keywords: classifier combination : meta-learning : meta-features : classification : classifier ensembles

Abstract: Combining classifiers is a common technique in order to improve the performance and robustness of classification systems. However, the set of classifiers that should be combined is not obvious and either expert knowledge or a time consuming evaluation phase is required in order to achieve high accuracy values. In this paper, we present an approach of automatically selecting the set of base classifiers for combination. The method uses experience about previous classifier combinations and characteristics of datasets in order to create a prediction model. We evaluate the method on over 80 datasets. The results show that the presented method is able to reasonably predict a suitable set of base classifiers for most of the datasets.

1 INTRODUCTION

According to Wolperts no-free-lunch theorem (Wolpert, 1996), no single learning scheme is able to generate the most accurate classifier for any domain. There are three reasons why a learning algorithm might fail for a given problem, that implies a true hypothesis (Dietterich, 2000): (1) If not sufficient training data is available, the learning algorithm can find several hypotheses that approximate the true hypothesis with the same accuracy. (2) Learning algorithms might get stuck in local optima because they often perform a local search to find the best hypothesis. (3) The true hypothesis can not be represented by any of the hypotheses that the learning algorithm is able to create.

Because of these reasons, a suitable classifier for a given domain is usually determined by either expert knowledge or an exhaustive evaluation of multiple algorithms. A different approach for avoiding the failure of a single algorithm is to join multiple algorithms. By combining the predictions of multiple classifiers, the weaknesses of a single classifier in one domain can be compensated by the strengths of a different classifier. Consequently, a combination of classifiers that are sufficiently accurate and diverse can outperform single classifiers (Dietterich, 2000). Additionally, by taking multiple classifiers into account, the variance of the predictions is reduced and the robustness of the classification system can be increased.

The critical point on combining classifiers is selecting the set of sufficiently accurate and diverse base-level classifiers. If all classifiers deliver correlated results, their combination would hardly provide any improvement. Diversity among the base classifiers can be introduced by using distinct algorithms, different parameter values of the same algorithm, different subsets of the samples, or different subsets of the features. The fusion strategy defines how the output of multiple classifiers are combined in order to get one result. An appropriate fusion strategy can further improve the performance of combined classifiers. The fusion strategy might be serial, parallel, or hierarchical. However, no type of combination has yet been found that works best for all cases (Kuncheva and Whitaker, 2003).

Also, the choice of classifiers that will be combined has an influence on the final performance of the classification system. An obvious approach would be evaluating different set of classifiers and, finally, select the one that achieved the best results. Although this probably will lead to good results, it is time consuming: Each considered classifier has to be trained, preferable including a parameter optimization. In this paper, we present an approach for automatically selecting a suitable set of distinct classifiers for a given dataset without the need of evaluating the classifiers.

The rest of the paper is structured as follows: In the next section, we describe previous work. In the following Section 3, the presented approach is explained in detail. Section 4 contains the evaluations. The final Section 5 comprises a conclusion.

2 RELATED WORK

Meta-learning is used to make selections or recommendations for new learning tasks. Knowledge about previous learning tasks is modeled in order to gain knowledge for the new learning task. A well known example is algorithm selection: Based on the knowledge about the best performing algorithm for multiple datasets, a suitable algorithm is automatically selected for a new dataset.

Typically, methods for algorithm or model selection are based on single algorithms, only, instead of combinations of them. The best algorithm might be predicted directly using classification (Bensusan and Giraud-Carrier, 2000a; Ali and Smith, 2006), a ranking approach creates a sorted list of all algorithms (Brazdil et al., 1994, 2003; Vilalta et al., 2004), or the actual accuracy of each considered algorithm is predicted using regression (Gama and Brazdil, 1995; Sohn, 1999; Reif et al., 2012).

Only less work has been done in automatically selecting suitable algorithm combinations based on the given problem. Cornelson et al. (2002) used meta-learning to combine families of information retrieval algorithms. Bennett et al. (2005) proposed a probabilistic method for combining classifiers taking context-sensitive reliabilities into account. Todorovski and Džeroski (2003) presented meta decision trees (MDT), that are used to decide which base classifier should be used to classify a sample. A MDT is trained on the class probability distributions created by the base classifiers for a given sample. However, the set of used base classifiers has to be fixed in advance. Kitoogo and Baryamureeba (2007) investigated the approach of selecting the best three out of five base classifiers based on three dataset properties (number of classes, number of attributes, and number of samples). However, the approach does not automatically select any classifiers but does a clustering on the dataset properties and the performance values of the different classifier combinations.

3 METHODOLOGY

In this paper, we investigate the approach of predicting the best combination of three out of five classifiers. The goal of the approach is to automatically get a set of three classifiers for a given dataset which combination achieves the highest possible accuracy. Therefore, we fix the fusion strategy and use plurality voting. We chose three classifiers because it is a good compromise between the run-time and the diversity of the classifiers. Additionally, using an odd number of voting classifiers reduces the probability of ties.

The five base-level classifiers were selected that their foundations make different assumptions. We included tree-based and instance-based classifiers as well as statistical classifiers and neural networks. Each classifier includes an optimization of its most important parameters using a grid-search and ten-fold cross-validation. This means, whenever a classifier is trained, its parameters are newly optimized. The selected classifiers and their optimized parameters are: k-Nearest Neighbor (k), MLP (learning rate), SVM (γ, C), Decision Tree (confidence, minimal gain), and Naive Bayes (laplace correction).

Like in most meta-learning approaches, datasets are represented by their characteristics and properties. Different measures are used to extract such properties, which are typically called meta-features. Obvious meta-features are the number of sample, the number of classes, and the number of attributes. Such simple meta-features are directly and easily extractable from the dataset (Michie et al., 1994).

Besides simple measures with only limited descriptive power, more sophisticated measures are used as meta-features. We used meta-features from five different groups: eight simple, five statistical (e.g., kurtosis, skewness, correlation) (Michie et al., 1994; Castiello et al., 2005; Engels and Theusinger, 1998), six information-theoretic (e.g., conditional entropy, mutual information, signal-noise ratio) (Michie et al., 1994; Segrera et al., 2008), 17 model-based (e.g., width and depth of a created decision tree) (Peng et al., 2002; Bensusan et al., 2000), and eight landmarkers (e.g., accuracy of Naive Bayes, Nearest Neighbor, and Decision Stumps) (Pfahringer et al., 2000; Bensusan and Giraud-Carrier, 2000b). The same 44 meta-features as used by Reif (2012) have been calculated for each dataset.

The presented approach uses supervised-learning for the prediction of a suitable set of classifiers. Therefore, the training of the meta-learner requires this information for each training dataset. First, the dataset is preprocessed by replacing missing values and converting nominal to numeric features because SVM as well as MLP do not support nominal features. Additionally, all features are normalized to the interval $[0;1]$. Then, all base classifiers are trained on the dataset using parameter optimization with a grid search and a ten-fold cross-validation. Afterwards, all ten possible combinations are evaluated by estimating their performance using ten-fold cross-validation and plurality voting. Finally, the combination maximizing the accuracy is selected as label.

Since the collected meta-data is structured like a traditional classification dataset, an arbitrary classi-

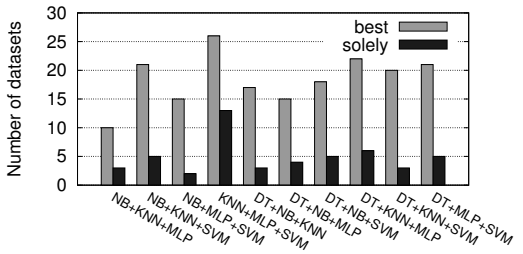

Figure 1: The number of datasets on which the different classifier combinations achieved the highest accuracy – possibly with other combinations or solely.

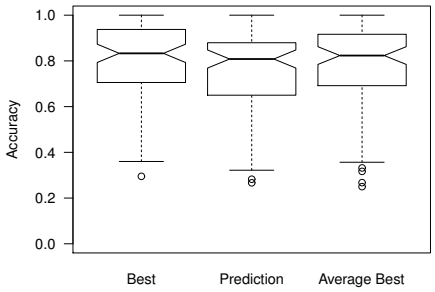

Figure 2: Box plot of the accuracies achieved by using the best combination, the averaged best combination, and the predicted combination.

fication algorithm can be applied on the meta-level. Based on the previously created meta-dataset, it delivers a classification model that is able to predict a suitable set of classifiers for a new dataset. We selected a SVM as the meta-level learning scheme since it has been successfully used in the past on a variety of domains. However, we also tried different algorithms, but we did not observe significant improvements compared to SVM. Since the set of meta-features is relatively big and the usefulness of each meta-feature is not guaranteed, we applied forward selection (Kohavi and John, 1997) of the features.

4 EVALUATION

We evaluated the approach on 84 datasets that were randomly selected from UCI (Asuncion and Newman, 2007), StatLib (Vlachos, 1998), and (Simonoff, 2003). They contain 2 to 24 classes, 1 to 359 features, and 10 to 435 samples. The resulting meta-dataset contains 84 samples, 44 features, and 10 classes.

As a preceding analysis, we counted how often each combination is the best for a certain dataset because it achieves the highest accuracy. Additionally, we determined how often a combination achieves the highest accuracy for a dataset solely. The results are shown in Figure 1. Two things are notable from this plot: (1) The combination `KNN+MLP+SVM` seems to be a good choice in general because it achieves the highest accuracy most frequent, together with other combinations but also solely. (2) Each combination achieves the highest accuracy solely for at least two datasets. This strengthens the necessity of selecting the set of used base classifiers depending on the dataset.

Since the training data only consists of 84 samples, we applied a leave-one-out cross-validation for evaluating the presented approach: For the prediction of a classifier combination for a particular dataset, a classification model based on the remaining 83 samples is trained. Afterwards, the predicted combination can be compared to the ground-truth information. Since our meta-learning approach is a classification task, typically classification measures such as classification accuracy might be used to evaluate the performance of the prediction model. However, this would lead to the following issues: If multiple combinations achieve the highest accuracy, the label includes only one of them and predicting any other combination with the same accuracy will lead to an error. Also, predicting a sub-optimal combination with only a slightly decreased accuracy as compared to the highest accuracy would receive the same error as predicting the worst combination with a very low accuracy. Therefore, we compared the accuracy achieved by the predicted combination and the accuracy achieved by the best combination.

Figure 2 shows a box plot of the accuracies achieved by three strategies for selecting the classifier combination: (1) the optimal combination achieving the highest possible accuracy, (2) the combination that achieved the highest average accuracy over all datasets (`KNN+MLP+SVM`), and (3) the combination predicted by the presented approach. Unfortunately, just using the combination that worked best in average during the past seems to be a better strategy then the presented approach.

For a deeper investigation of the results we also looked at each dataset individually. Figure 3 shows the accuracy achieved by the three strategies for each dataset.

A first result is that the presented method achieves the accuracy of the baseline or even the best accuracy for many datasets. For more than the half of the datasets, the accuracy of the predicted combination is less than 2.5% smaller than the highest accuracy. However, a second result from Figure 3 are the small differences between the different selection strategies for most of the datasets. Many datasets have a very low variance within the different combinations. While this fact is an indication of the robustness of combining multiple classifiers, it also counteracts the

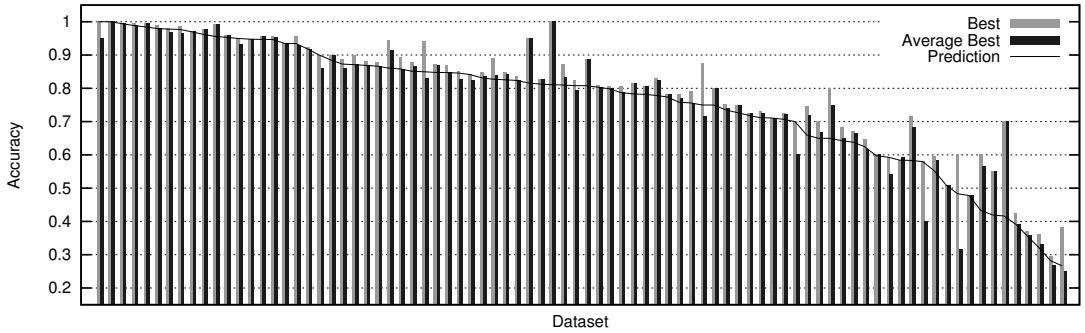

Figure 3: The accuracy values achieved by the three methods for each dataset individually (sorted according to the accuracy of the prediction for a better visualization).

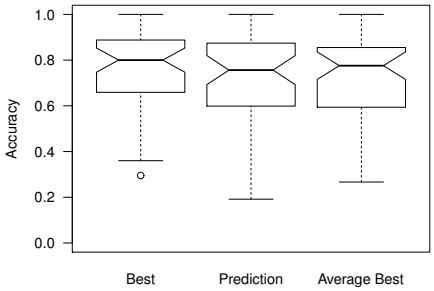

Figure 4: Box plot of the accuracies achieved by using the best combination, the averaged best combination, and the predicted combination for the reduced set of datasets.

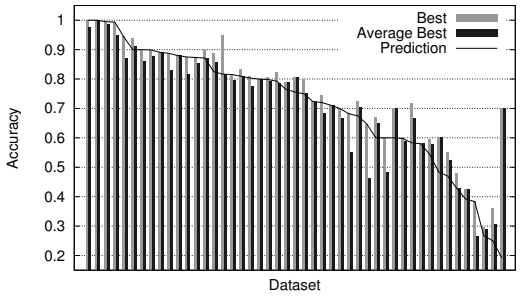

Figure 5: The accuracy values achieved by the three methods for the reduced set of datasets (sorted according to the accuracy of the prediction for a better visualization).

meta-learning approach.

Learning to predict a good classifier combination based on datasets with a very low variance over the candidate combinations is problematic. It is hard for the learning algorithm to create a discriminative model if the training data is not discriminative itself. Therefore, we investigated if using more discriminative combinations will improve the results. We created a second meta-dataset that includes only knowledge about base datasets with at least 5% accuracy difference between the best and the worst classifier combination. This was the case for 47 out of the 84 datasets. The reduced meta-dataset was used for both training and testing. While removing particular samples from the training set is obviously valid, testing a method on a reduced set might make the evaluation less convincing. However, since we test our method on datasets where the selection of the used base classifiers actually matters, we think that the evaluation is still valid and convincing.

Figure 4 shows the box plot of the accuracies achieved by the three strategies based on the reduced dataset. It is visible that the performance of the presented method was improved compared to the baseline method ("Average Best"). Unfortunately, a clear benefit of the presented method is not noticeable.

Again, we plotted the accuracies achieved for each dataset individually, as shown in Figure 5. For most of the datasets, the presented method was able to predict a classifier combination that is at least as good as the baseline method. For some datasets, the prediction is still worse than the baseline, especially for the "parity5" dataset (rightmost in Figure 5). It is notable that the presented method achieves even the highest accuracy on over 20 of the 47 datasets.

5 CONCLUSION

In this paper we presented a novel approach for predicting the best classifier combination for a given dataset. Based on dataset characteristics, the approach automatically selects three out of five base classifiers that should be combined in order to achieve high accuracy values on the dataset. Therefore, a meta-learning approach was developed. A classification model is trained based on the meta-features and the knowledge about the optimal classifier combination for multiple datasets. The presented approach was evaluated on 87 datasets. The results show the overall suitability of the approach while its performance could be increased if only datasets with diverse combination accuracies were used for training.

REFERENCES

Ali, S. and Smith, K. A. (2006). On learning algorithm selection for classification. *Applied Soft Computing*, 6:119–138.

Asuncion, A. and Newman, D. (2007). UCI machine learning repository. `http://www.ics.uci.edu/~mlearn/MLRepository.html` University of California, Irvine, School of Information and Computer Sciences.

Bennett, P. N., Dumais, S. T., and Horvitz, E. (2005). The combination of text classifiers using reliability indicators. *Information Retrieval*, 8(1):67–100.

Bensusan, H. and Giraud-Carrier, C. (2000a). Casa batl is in passeig de grcia or how landmark performances can describe tasks. In *Proc. of the ECML-00 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pages 29–46.

Bensusan, H. and Giraud-Carrier, C. (2000b). Discovering task neighbourhoods through landmark learning performances. In *Proc. of the 4th Europ. Conf. on Principles of Data Mining and Knowledge Discovery*, pages 325–330.

Bensusan, H., Giraud-Carrier, C., and Kennedy, C. (2000). A higher-order approach to meta-learning. In *Proc. of the ECML'2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pages 109–117.

Brazdil, P., Gama, J., and Henery, B. (1994). Characterizing the applicability of classification algorithms using meta-level learning. In *Machine Learning: ECML-94*, volume 784 of *Lecture Notes in Computer Science*, pages 83–102.

Brazdil, P. B., Soares, C., and da Costa, J. P. (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277.

Castiello, C., Castellano, G., and Fanelli, A. M. (2005). Meta-data: Characterization of input features for meta-learning. In *Modeling Decisions for Artificial Intelligence*, volume 3558, pages 295–304.

Cornelson, M., Grossmann, R. L., Karidi, G. R., and Shnidman, D. (2002). *Survey of Text Mining: Clustering, Classification, and Retrieval*, chapter Combining Families of Information Retrieval Algorithms using Meta-Learning, pages 159–169.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proc. of the First Int. Workshop on Multiple Classifier Systems*, pages 1–15.

Engels, R. and Theusinger, C. (1998). Using a data metric for preprocessing advice for data mining applications. In *Proc. of the Europ. Conf. on Artificial Intelligence*, pages 430–434.

Gama, J. and Brazdil, P. (1995). Characterization of classification algorithms. In *Progress in Artificial Intelligence*, volume 990 of *Lecture Notes in Computer Science*, pages 189–200.

Kitoogo, F. E. and Baryamureeba, V. (2007). Meta-knowledge as an engine in classifier combination. *International Journal of Computing and ICT Research*, 1(2):74–86.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence – Special issue on relevance*, 97:273–324.

Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*.

Peng, Y., Flach, P., Soares, C., and Brazdil, P. (2002). Improved dataset characterisation for meta-learning. In *Discovery Science*, volume 2534 of *Lecture Notes in Computer Science*, pages 193–208.

Pfahringer, B., Bensusan, H., and Giraud-Carrier, C. (2000). Meta-learning by landmarking various learning algorithms. In *Proc. of the 17th Int. Conf. on Machine Learning*, pages 743–750.

Reif, M. (2012). A comprehensive dataset for evaluating approaches of various meta-learning tasks. In *First International Conference on Pattern Recognition and Methods*.

Reif, M., Shafait, F., Goldstein, M., Breuel, T., and Dengel, A. (2012). Automatic classifier selection for non-experts. *Pattern Analysis and Applications*. 10.1007/s10044-012-0280-z.

Segrera, S., Pinho, J., and Moreno, M. (2008). Information-theoretic measures for meta-learning. In *Hybrid Artificial Intelligence Systems*, volume 5271 of *Lecture Notes in Computer Science*, pages 458–465.

Simonoff, J. S. (2003). *Analyzing Categorical Data*. Springer Texts in Statistics. `http://people.stern.nyu.edu/jsimonof/AnalCatData/`.

Sohn, S. Y. (1999). Meta analysis of classification algorithms for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1137–1144.

Todorovski, L. and Džeroski, S. (2003). Combining classifiers with meta decision trees. *Machine Learning*, 50:223–249.

Vilalta, R., Giraud-Carrier, C., Brazdil, P., and Soares, C. (2004). Using meta-learning to support data mining. *International Journal of Computer Science and Applications*, 1(1):31–45.

Vlachos, P. (1998). StatLib datasets archive. `http://lib.stat.cmu.edu` Department of Statistics, Carnegie Mellon University.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computing*, 8(7):1341–1390.