Predicting Classification Decisions with Data Point Based Meta-learning

Irene Cramer[1], Barbara Rauch[2], Hagen Fürstenau[2], Dan Shen[2], and Maria Staudte[2]

[1] Dortmund University, 44227 Dortmund, Germany
irene.cramer@uni-dortmund.de,
[2] Saarland University, 66125 Saarbrücken, Germany
Contact : barbara.rauch@LSV.uni-saarland.de

Abstract. Meta-learning involves the construction of a classifier that predicts the performance of another classifier. Previously proposed approaches do this by making a single prediction (such as the expected accuracy) for a complete data set. We suggest modifying this framework so that the meta-classifier predicts for each data point in the data set whether a particular base-classifier will classify it correctly or not. While this information can be converted into a standard meta-learning output such as an overall accuracy estimate for the complete data set, the approach has the added advantage of providing more fine-grained information which promises to be useful in Multiple Classifier Selection and Semi-Supervised Learning. This paper describes the new framework and reports the results of an initial evaluation on a medium-sized database of classification data sets.

1 Introduction

When faced with a classification task that we want to solve with a machine learning (ML) method, we have a choice between many different algorithms and implementations. While factors such as run time and memory restrictions may influence the selection, expected classification accuracy will be of major importance. Unfortunately, predicting the latter is not trivial. An empirical approach that has been tried in the past is meta-learning: here the algorithm selection or accuracy prediction problem is phrased as a classification or regression task, i.e. a meta-level learner is trained to predict the performance of a base ML algorithm.

For example, in the StatLog project [1], a decision tree was trained to assess whether an algorithm is *applicable* to a data set, meaning it would achieve low error rates or costs. In later work, Fürnkranz and Petrak [2] predict which of a pair of algorithms is *more accurate* on a given task. In contrast, Bensusan and Kalousis [3] predict classification accuracy with one regression model per base-classifier and evaluate algorithm rankings generated from the accuracies. More recently, the MetaL project led to the development of an online advisory system which takes both accuracy and run time into account [4]. It constructs a ranking of several algorithms based on their predicted accuracies and the loss in accuracy the user is willing to trade in for a 10-times speed up.

2 Data Point Based Predictions

The input unit in standard meta-learning is the data set: one meta-feature vector is extracted per data set, and the output is a prediction for the complete set. We propose creating a meta-learner which makes predictions not for the data set as a whole, but rather for each individual data point in it. More specifically, our meta-classifier (MC) should predict for a given data point whether a particular base-classifier (BC) will classify it correctly or not. Figure 1 illustrates how such a system can be trained, tested and evaluated.

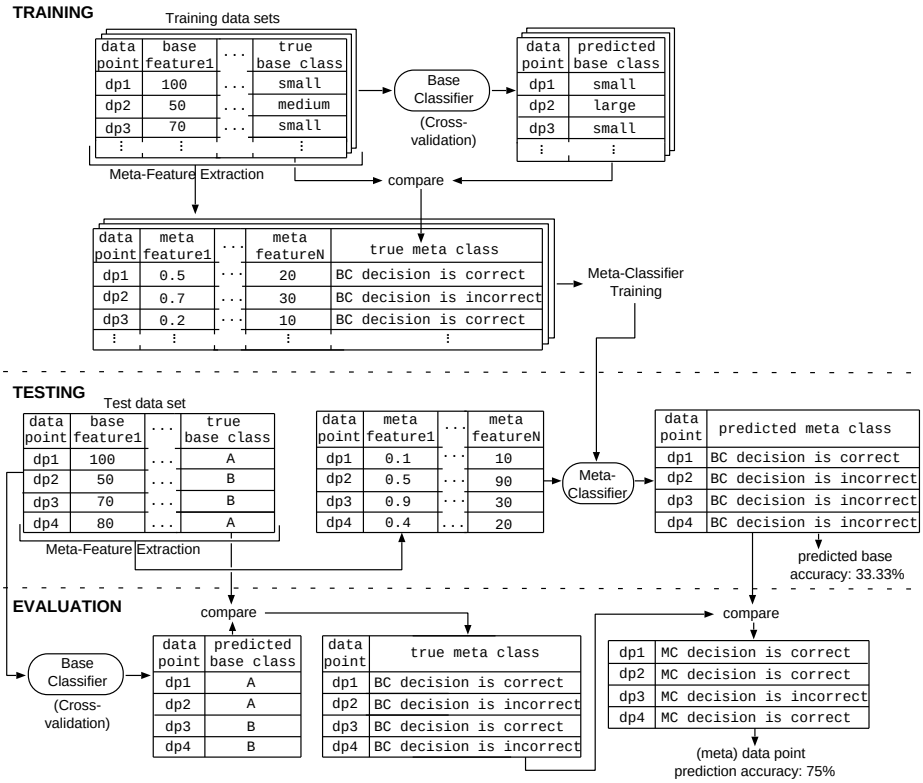

Fig. 1. Data point based meta-learning for one combination of meta- and base-classifier

The **meta-level training stage** requires a number of labelled data sets, which include the real base-class for each data point. Note that these sets represent different classification problems and therefore may have different classes, numbers of base-features and numbers of data points. The *predicted* base-class for all points in each set can be obtained by training and testing the BC in a cross-validation setup. By comparing the predicted with the real base-level class, we compute the true meta-level class, that is: was the BC right or wrong? This information, together with the meta-features extracted for each data point, is then used to train the meta-classifier. Figure 1 implies that the true base-level class

is used in the meta-feature extraction, which is reasonable in all use cases where the meta-classifier would be applied to training or validation data. However, our framework is equally applicable without this information. The **meta-level testing stage** assumes a single test data set. We extract the meta-features for each data point in this set and feed them into the trained meta-level classifier, which predicts for each data point whether the BC will classify it correctly or not.

The main advantage of data point level decisions over those at the data set level is that they are more versatile. They can provide the same information as set based methods: individual data point predictions can be averaged to give an accuracy estimate for the complete data set, and this can be used to construct a ranking for a set of classifiers. However, knowing whether a BC will classify a point correctly can also be useful in other processes. In **Multiple Classifier Systems**, it has been shown that the best combination of classifiers is not necessarily the combination of the best single classifiers, and therefore knowing which BC achieves the highest accuracy is not always of help. Even if some classifiers achieve higher accuracies than others, they might make different mistakes, so it is really the complementarity of correct classifier decisions that is of interest. A successful data point based meta-learning system could provide this information. In fact, estimates of classifier accuracy have already been applied to Dynamic Classifier Selection, where the BC performance on the training data has been used to select BCs for test data points of the same data set (e.g. [5]). However, we are not aware of any work that uses a *classifier* trained on *other* data sets to do the selection. Whether generalising in this way leads to better selection strategies still needs to be investigated, but the differences are also practical: in our approach the meta-classifiers need to be trained once before they can be applied to a number of unknown data sets; in the other case all BCs need to be trained and evaluated (on the training data) for each unknown data set.

Another potential application of data point based meta-learning is in the field of **Semi-Supervised Learning**, where the goal is to train a classifier with very little annotated data to reliably classify a much larger set of unannotated data. In methods such as bootstrapping we iteratively train and test the classifier, each time augmenting the training set with new, previously unlabelled, data. At each cycle the new training data may have been labelled incorrectly, which can completely mislead the classifier. Meta-learning can help to avoid this situation by providing an estimate of the classification quality. Data set based meta-learning could give an overall estimate, but data point based meta-learning could additionally help to identify problematic data points (which may then be omitted in the retraining step or recommended for manual annotation).

3 Experimental Evaluation

This section describes a simple implementation designed to test the feasibility of the proposed approach and the results obtained with it on a medium-sized data set. We used the YALE machine learning toolkit [6] for our experiments

and selected five **classifiers** implemented by it for both the base- and meta-level: the decision tree algorithm C4.5 (DT), a Naive Bayes classifier (NB), a k-Nearest-Neighbour implementation (KNN), the rule learner Ripper (RL), and a Support Vector Machine with a radial basis function kernel (SVM). In all experiments reported below the default settings in YALE were used. To train and evaluate the classifiers we used 127 **data sets** from the Weka toolkit archive [7], a collection of machine learning data from various other archives such as the UCI KDD archive and the StatLib data archive. Table 1 describes the data.

	min.	max.	mean	median	std. dev.
#classes	2	48	6	3	7.5
#instances	8	20,000	1,131.2	345	2,549.6
#attributes	1	7129	102.5	13	654.2

Table 1: Data set statistics

	all sets	train. sets	test sets
DT	69,97%	69.79%	70.69%
KNN	68,33%	69.08%	68.53%
NB	66,22%	66.55%	64.07%
RL	68,63%	68.81%	67.89%
SVM	55,63%	55.84%	54.77%

Table 2: Mean accuracy of base-learners

Table 2 shows the average performance of the BCs obtained by 5-fold cross-validation. The accuracies for the test sets serve as a majority baseline result for our MC experiments reported below.[3] For the meta-level experiments the sets were randomly divided into 80% training data (102 sets) and 20% test data (25 sets). We developed a small list of **meta-features** intended to capture aspects of a test data point that make it hard for classifiers to classify a point correctly, given the training data the classifier had seen. The meta-features for each data point were therefore extracted with respect to the training set used in its corresponding cross-validation run.

For reasons of space we do not give a complete list here, but summarise that the 19 meta-features are either related to the problem (e.g. the number of features), the test point (e.g. the proportion of features with undefined values), or the training set (e.g. the proportion of training data with the same class as the test point). Since all our meta-features require the base-level features to be categorical, we discretised numerical low-level features before the extraction of the meta-features[4]. The 25 MCs were then trained as described in section 2 and evaluated on the 25 test data sets, resulting in 625 classification runs. Table 3 shows the mean **data point prediction accuracy** for each combination of MC and BC algorithms, averaged over all test sets.

MC	**BC**	**Acc.**	**MC**	**BC**	**Acc.**	**MC**	**BC**	**Acc.**	**MC**	**BC**	**Acc.**	**MC**	**BC**	**Acc.**	**Avg.**
DT	DT	76.08	DT	KNN	**77.56**	DT	NB	68.93	DT	RL	79.52	DT	SVM	66.86	73.79
KNN	DT	76.79	KNN	KNN	73.25	KNN	NB	**75.21**	KNN	RL	60.38	KNN	SVM	69.12	70.95
NB	DT	70.51	NB	KNN	60.60	NB	NB	64.48	NB	RL	62.66	NB	SVM	81.44	67.94
RL	DT	**79.28**	RL	KNN	69.03	RL	NB	69.87	RL	RL	74.82	RL	SVM	71.12	72.83
SVM	DT	62.66	SVM	KNN	69.55	SVM	NB	59.00	SVM	RL	**93.54**	SVM	SVM	**91.11**	75.17
Average		73.06	Average		70.00	Average		67.50	Average		74.18	Average		75.93	72.14
Baseline		70.69	Baseline		68.53	Baseline		64.07	Baseline		67.89	Baseline		54.77	65.19

Table 3. Mean data point prediction accuracy in % for the 25 MC/BC combinations

[3] Since all five BCs are more often correct than incorrect on training data, the majority baseline at the meta-level is to predict that the base-learner is always correct.

[4] Based on the training data, the range of attribute values was divided into a maximum of 200 equally large intervals so that each interval contained at most 5% of the values.

We observe that in most cases (for DT and RL meta-learners in all cases), the MCs beat the majority baseline. When taking the best MC to predict the decisions of each BC (the bold entry in each column), the accuracy is always higher than the baseline, and the average accuracy is as high as 83.34%. The results show that we can predict the behaviour of each BC to a relatively high degree even with the simple setup of this initial study.

4 Conclusions

We have presented and motivated a new meta-learning framework which predicts the correctness of classification decisions for each test pattern. Initial experiments with five base- and meta-classifiers show that even a simple implementation of meta-classifiers can predict this information with a relatively high degree of accuracy (between 75.21% and 93.54% when using the best meta-classifiers).

We now plan to continue this work on three levels. Firstly, we hope to improve on these results with more sophisticated meta-features and the incorporation of parameter tuning and feature selection at the meta-level. Secondly, we intend to conduct a more thorough evaluation. The use of a greater number of classifiers and data sets, as well as tuning and feature selection at the base-level, should lead to results that are more representative of real classification problems. Finally, we plan to investigate the contribution that the proposed framework can make to tasks such as algorithm selection or ranking, and in particular to Multiple Classifier Systems and Semi-Supervised Learning.

Acknowledgements

We are grateful to Katharina Morik, Donald Michie, Stefan Evert, Dietrich Klakow, and Miles Osborne for their helpful comments. This research was partially funded by DFG studentships in the International Post-Graduate College "Language Technology and Cognitive Systems".

References

1. Michie, D., Spiegelhalter, D.J., Taylor, C.C., eds.: Machine Learning, Neural and Statistical Classification. Ellis Horwood (1994)
2. Fürnkranz, J., Petrak, J.: An evaluation of landmarking variants. In: Proc. of the IDDM Workshop. (2001)
3. Bensusan, H., Kalousis, A.: Estimating the predictive accuracy of a classifier. In: Proc. of the European Conference on Machine Learning. (2001)
4. Brazdil, P., Soares, C., da Costa, J.P.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning **50**(3) (2003) 251–277
5. Woods, K., Kegelmeyer, W.P., Bowyer, K.W.: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(4) (1997) 405–410
6. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Proc. of the ACM SIGKDD. (2006)
7. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann (2005) For the data sets, see `http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html`.