Knowledge-based Selection of Data Characteristics for Algorithm Recommendation Using Ranking Methods

Carlos Soares and Pavel Brazdil

LIACC/Faculty of Economics, University of Porto, R. Campo Alegre 823, 4150-180 Porto, Portugal, {csoares,pbrazdil}@liacc.up.pt

Abstract. We show that information about the past performance of algorithms can be used for algorithm recommendation with small loss in accuracy and significant savings in experimentation time, when compared to cross-validation. This result is obtained with a meta-learning approach that uses a set of data characteristics that were manually selected using our knowledge of the algorithms. We also demonstrate the advantage of providing recommendation in the form of a ranking.

1 Introduction

Ideally, we would like to be able to identify or design the single best algorithm to be used in all situations. However, both experimental results and theoretical work indicate that this is not possible. Therefore, the choice of which algorithm(s) to use depends on the data set at hand and systems that can provide such recommendations would be very useful. We could reduce the problem of algorithm recommendation to the problem of performance comparison by estimating the performance of all the algorithms on the data currently available, assuming that it is representative of future data. Cross-validation (CV) is the most accurate method available for that purpose. However, it is not usually feasible in practice because there are too many algorithms to try out, some of which may be quite slow. Another approach to algorithm recommendation involves the use of meta-knowledge, that is, knowledge about the performance of algorithms, which is followed here.

The performance and the usefulness of meta-learning for algorithm recommendation depends on several issues, namely the measures used to characterize data sets, the type of recommendation provided and the meta-learning method used. Here we focus on the former and, so, we must make appropriate choices for the others. Another important issue for algorithm recommendation is the criteria used to evaluate the algorithms (e.g. accuracy, interpretability of models). Here we will concentrate on accuracy. One example of a multicriteria meta-learning approach can be found in [1].

Concerning the type of recommendation provided, we opted for a ranking of the algorithms. This approach is more flexible than the recommendation of a single algorithm or of a small set of algorithms which is expected to perform not

significantly worse than the best one, which are commonly used in meta-learning [2, 3]. Flexibility is important in the algorithm recommendation setting because it is not known beforehand how many alternatives the user will actually take into account. Furthermore, suppose that a single algorithm is recommended and that this algorithm fails, e.g. because it has a bug or uses too much memory. Given that no information regarding the expected performance of the other algorithms is provided, the user is left without guidance. Such a situation will not occur with a ranking because the user may simply try the next algorithm. Algorithm recommendation using meta-learning was first handled as a ranking task by [4]. Recently, there has been a growing interest in this approach (e.g. [5]).

The choice of ranking for the type of recommendation has limited our choice of meta-learners. We adopted an IBL framework that uses the k-Nearest Neighbor algorithm combined with a method to aggregate and rank performance information is selected. A few alternative ranking methods have been described in [5]. Here we have opted by the average ranks method (AR), which is simple and competitive [5]. This method consists of calculating the average rank of each algorithm on the neighbor datasets and ranking the algorithms accordingly. More details can be found in [1].

We start by describing the data characteristics selected (Section 2), then we present a comparative evaluation of this subset against the original set of measures (Section 2.1). Finally, we present some conclusions.

2 Selection of Data Characteristics

The most important issue in meta-learning is probably data characterization. We need to extract measures from the data that characterize relative performance of the candidate algorithms and that can be computed significantly faster than running those algorithms. It is known that the performance of different algorithms is affected by different data characteristics. For instance, the performance of k-Nearest Neighbor will suffer if there are many irrelevant attributes. Most work on meta-learning uses general, statistical and information theoretic (GSI) measures or *meta-attributes* [4]. Recently, other approaches to data characterization have been proposed, namely *landmarkers* [3] and model-based characterization [6], which we will not address here.

Many measures have been proposed for data characterization in the GSI approach. However, some of them may be irrelevant, others may not be adequately represented (e.g., the proportion of numeric attributes is probably more informative than the number of numeric attributes), while some important ones may be missing. Furthermore, given that the performance information available typically includes relatively few examples (data sets), a large number of meta-features creates the danger of overfitting. However, in most previous meta-learning work, not enough effort has been dedicated to selecting an appropriate subset of data characteristics. An exception is the work of [7], who applied a wrapper-based feature selection method to a large number of meta-features (and combinations of meta-features). The drawback of this approach is that many hypotheses are

tested when compared to the number of examples. This increases the probability of finding a subset of the data characteristics that obtains good performance merely by chance.

Here, we follow a knowledge engineering approach. Based on our expertise on the learning algorithms used and on the properties of data that affect their performance, we select and combine existing GSI measures to define *a priori* a small set of meta-features that are expected to provide information about those properties. The measures and the properties which they are expected to represent are presented in Table 1. All three proportional features proposed (2nd to 4th features) represent combinations of previously defined data characteristics. The number of examples represents one aspect of scalability, which is also affected by other data characteristics, like number of attributes and number of values in symbolic attributes. The need for a measure that combines all these characteristics remains. A numeric attribute is considered to have outliers, possibly due to noise, if the ratio of the variances of mean value and the α-trimmed mean is smaller than 0.7. We have used $\alpha = 0.05$.

Table 1. Properties of algorithms which affect relative performance and data characteristics that affect those properties. More details about the basic features used here can be found in [8].

Property	Measure
Scalability	Number of examples
Preference for symbolic or numeric attributes	Proportion of symbolic attributes
Robustness to missing values	Proportion of missing values
Robustness to outliers	Proportion of numeric attributes with outliers
Number of classes	Class entropy
Class frequency	Class entropy
Useful information in symbolic attributes	Average mutual information of class and symbolic attributes
Useful information in numeric attributes	Canonical correlation of the most discriminating single linear combination of numerical attributes and the class distribution

2.1 Empirical Evaluation and Comparison

Our meta-data consists of 53 data sets mostly from the UCI repository [9] but including a few others from the METAL project[1] (SwissLife's Sisyphus data and a few applications provided by DaimlerChrysler). Ten algorithms were executed on those data sets[2]: two decision tree classifiers, C5.0 and Ltree, which is a decision

[1] Esprit Long-Term Research Project (#26357) *A Meta-Learning Assistant for Providing User Support in Data Mining and Machine Learning* (www.metal-kdd.org).

[2] References for these algorithms can be found in [3].

tree that can introduce oblique decision surfaces; the IB1 instance-based and the naive Bayes classifiers from the MLC++ library; a local implementation of the multivariate linear discriminant; two neural networks from the SPSS Clementine package (Multilayer Perceptron and Radial Basis Function Network); two rule-based systems, C5.0 rules and RIPPER; and an ensemble method, boosted C5.0. Results were obtained with 10-fold cross-validation using default parameters on all algorithms.

To assess whether the subset of meta-features yields better rankings, we use a methodology for ranking evaluation and comparison that has been proposed earlier for meta-learning [5]. The rankings recommended by the ranking methods are compared against the true observed rankings using Spearman's rank correlation coefficient. We note that the performance of two or more algorithms may be different but not with statistical significance. To address this issue, we exploit the fact that in such situations the tied algorithms often swap positions in different folds of the N-fold cross-validation procedure which is used to estimate their performance. Therefore, we use N orderings to represent the true ideal ordering, instead of just one. The correlation between the recommended ranking and each of those orderings is calculated and its score is the corresponding average. Leave-one-out was used to estimate meta-level performance.

The mean average correlation for increasing number of neighbors obtained by the AR ranking method using the original set of 25 measures and the manually selected subset (Section 2) is shown on the right-hand side of Figure 1. We observe that the results are better with the reduced set than with the extended set. In fact, the combination of Friedman's test and Dunn's Multiple Comparison Procedure (significance levels of 5% and 25%, respectively), which is appropriate for this kind of comparison, shows that the AR with the reduced set of measures is significantly better than the same ranking method with the full set of measures and the baseline method of aggregating all performance information. More details about this comparison methodology can be found in [5]. We also observe that the quality of the rankings obtained with the reduced set decreases as the number of neighbors increases. This is not true when the extended set is used. This indicates that the space of meta-features is an approximation of the space of relative performance of the algorithms. This means that we were the measures selected are indeed representative of properties that affect relative algorithm performance. The shape of the curves also indicates that the extended set probably contains many irrelevant features, which, as is well known, affects the performance of the k-NN algorithm used at the meta-level.

One may ask how do these results reflect in terms of the quality of the advice provided, in the perspective of the user. Figure 1 shows a comparison of our meta-learning method with the cross-validation strategy (left-hand side), which is the most accurate algorithm selection method (an average accuracy of 89.93% in our setting) but it is very time consuming (approximately four hours on average, in our setting) and boosted C5.0, which is the best algorithm on average (87.94%) and also very fast (less than two min.). One could argue that, with such a small margin for improvement (2%), it is not worthwhile to

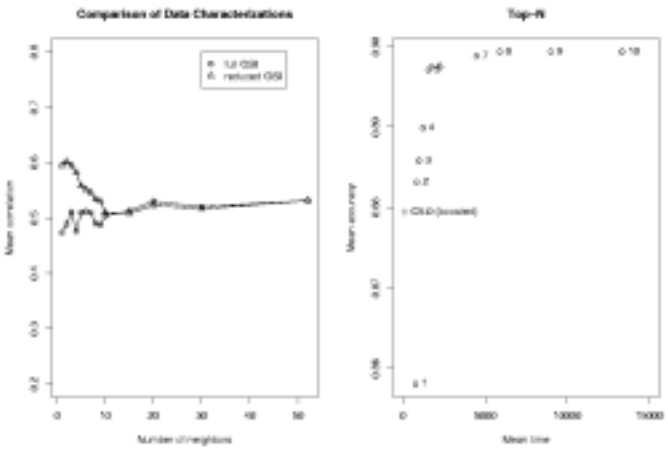

Fig. 1. Mean correlation obtained by AR ranking method for increasing number of neighbors using two sets of GSI data characteristics: full and reduced (on the right). Average accuracy versus average execution time for the strategy of executing the top-N algorithms in the recommended ranking, for all possible values of N, and for the strategy of always executing boosted C5.0 (on the left).

do algorithm selection: choosing boosted C5.0 will provide quite good results on average. However, in some applications (e.g. cross-selling in a website that sells thousands of items daily), an improvement of 2% or even less may be significant from a business point-of-view. The strategy of executing the algorithm ranked in the first position is worse than always executing boosted C5.0. However, if we use the full potentially of a ranking method, and execute the Top-2 algorithms in the ranking, the time required is larger than boosted C5.0's, although still acceptable in many applications (less than 15 min.) but the loss in accuracy would be only 1.62%. Running two more algorithms another algorithm would provide further improvement in accuracy (1.35% and 0.95% losses) while taking only a little longer (16 and 20 min.).

3 Conclusions

We have investigated the effect of careful selection of meta-features on the quality of rankings generated by an IBL meta-learning approach for ranking. We considered a large set of general, statistical and information-theoretic meta-features, commonly used in meta-learning, and selected a subset, containing measures that represent properties of the data that affect algorithm performance. This selection has significantly improved the results. Although the average difference in accuracy between cross-validation and the best algorithm is only 2%, which makes the goal of improving the result of the latter very hard, our meta-learning

approach is able to reduce this difference to less than 1%. Although, this is a positive result, we plan to investigate how much improvement is achieved in the worst case, where the difference between CV and the best algorithm is 36%. We also plan to compare the subset of selected measures with new data characterization approaches, like landmarking.

Acknowledgments We thank the anonymous reviewers for useful comments. Thanks also to all the METAL partners for a fruitful working atmosphere, in particular to Johann Petrak for providing the scripts to obtain the meta-data. We also thank DaimlerChrysler and Guido Lindner for providing us the data characterization tool. Finally, we thank Rui Pereira for implementing part of the methods. The financial support from ESPRIT project METAL, project ECO under PRAXIS XXI, FEDER, Programa de Financiamento Plurianual de Unidades de I&D and from the Faculty of Economics is gratefully acknowledged.

References

1. Soares, C., Brazdil, P.: Zoomed ranking: Selection of classification algorithms based on relevant performance information. In Zighed, D., Komorowski, J., Zytkow, J., eds.: Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2000), Springer (2000) 126–135
2. Todorovski, L., Dzeroski, S.: Experiments in meta-level learning with ILP. In Rauch, J., Zytkow, J., eds.: Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD99), Springer (1999) 98–106
3. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms. In Langley, P., ed.: Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000), Morgan Kaufmann (2000) 743–750
4. Brazdil, P., Gama, J., Henery, B.: Characterizing the applicability of classification algorithms using meta-level learning. In Bergadano, F., de Raedt, L., eds.: Proceedings of the European Conference on Machine Learning (ECML-94), Springer-Verlag (1994) 83–102
5. Brazdil, P., Soares, C.: A comparison of ranking methods for classification algorithm selection. In de Mántaras, R., Plaza, E., eds.: Machine Learning: Proceedings of the 11th European Conference on Machine Learning ECML2000, Springer (2000) 63–74
6. Bensusan, H., Giraud-Carrier, C., Kennedy, C.: A higher-order approach to meta-learning. In: Proceedings of the ILP'2000 (Work in Progress Track). (2000)
7. Todorovski, L., Brazdil, P., Soares, C.: Report on the experiments with feature selection in meta-level learning. In Brazdil, P., Jorge, A., eds.: Proceedings of the Data Mining, Decision Support, Meta-Learning and ILP Workshop at PKDD2000. (2000) 27–39
8. Henery, R.: Methods for comparison. In Michie, D., Spiegelhalter, D., Taylor, C., eds.: Machine Learning, Neural and Statistical Classification. Ellis Horwood (1994) 107–124
9. Blake, C., Keogh, E., Merz, C.: Repository of machine learning databases (1998) http:/www.ics.uci.edu/~mlearn/MLRepository.html.