# Algorithm Selection Support for Classification

Guido Lindner[1], Rudi Studer[2]

[1] DaimlerChrysler AG, Research & Technology FT3/KL,
PO: DaimlerChrysler AG, T-402, D-70456 Stuttgart, Germany
guido.lindner@daimlerchrysler.com *

[2] Institute AIFB
University of Karlsruhe,
D-76128 Karlsruhe, Germany
studer@aifb.uni-karlsruhe.de

**Abstract**

Providing user support for the application of Data Mining algorithms in the field of Knowledge Discovery in Databases (KDD) is an important issue. Based on ideas from the fields of statistics, machine learning and knowledge engineering we provided a general framework for defining user support. The general framework contains a combined top-down and bottom-up strategy to tackle this problem. In the current paper we describe the Algorithm Selection Tool (AST) that is one component in our framework.

AST is designed to support algorithm selection in the knowledge discovery process with a case-based reasoning approach. We discuss the architecture of AST and explain the basic components. We present the evaluation of our approach in a systematic analysis of the case retrieval behaviour and thus of the selection support offered by our system.

## 1 Introduction

It is well known that there is no best algorithm for all classification problems [Schaffer, 1994]. However, what exactly is defined as *best* strongly depends on application specific goals and the characteristics of the available data. Where application specific goals should be requested from the user, meta data on the data can be calculated automatically. An approach integrating

*currently glindner@wuerttag.de

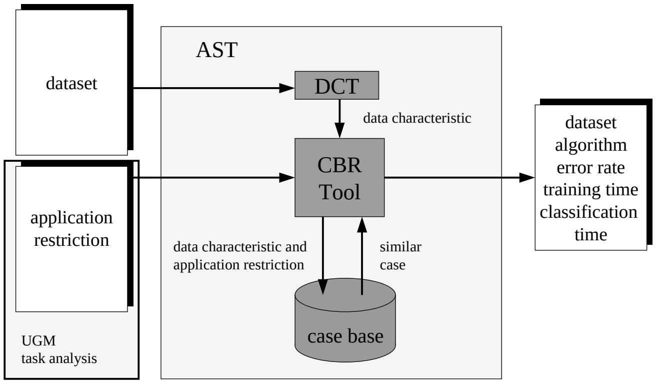


Figure 1: Architecture of AST

this user interaction and the calculation of domain characteristics as a top-down and bottom-up strategy is described in [Engels et al., 1997]. It is our firm opinion that user interaction with the goal of getting user's restrictions on the functionality of a data mining application has to form an integral part of every approach of algorithm selection.

Both, Consultant [Krodratoff et al., 1992] and Statlog [Michie et al., 1994] have different disadvantages when considering the application of these approaches in real-life scenarios.

Consultant uses a static rule set which discriminates between a set of possibly applicable algorithms [Sleeman et al., 1995]. Such an approach is very difficult to maintain: each time a new algorithm has to be included one has to recompute all the rules. The Statlog project tried to describe data sets for a meta learning step to generate rules that specify in which case which algorithm is (possibly) applicable. The generated rules use hard boundaries within their condition part. However, instead of hard boundaries one would like to have more fuzzy conditions. A CBR approach enables a smooth similarity calculation for similar application problems. The main idea is to recommend to the user an algorithm or a set of algorithms based on the most similar cases that are found in the case base. Such a case is defined by application restrictions, a description of the data and experience gained in former applications. The basic architecture of our AST (Algorithm Selection Tool) system is described in section 2. The description of the data, called data characteristics, is outlined in section 3. Another advantage for CBR is the possibility to extend the model with a characterization of algorithms. In this case also queries about similar algorithms are possible. Examples of algorithm descriptions are presented in section 4. Finally, we discuss first evaluations of our system AST in section 5 and give an outlook about future work in the last section 6.

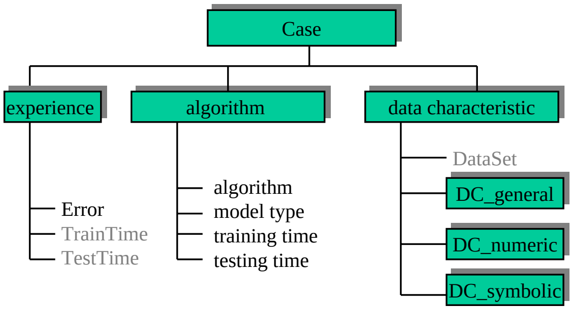


Figure 2: Case structure in AST

## 2 Architecture of AST

The top-level architecture of our AST system is shown in Figure 1. As outlined in the introduction, the problem of algorithm selection is a decision based on three aspects: application restrictions, given data and existing experience.

Embedded in our UGM [Engels et al., 1997](User Guidance Module) approach the application restrictions are analyzed by the task analysis component. In addition, the user can also feed his/her restrictions directly into the system. Application restrictions address aspects like the interpretability of the generated model or the amount of training time that may be used (see section 4 for more details). From the given data, the data characterization tool (DCT) [1] computes data characteristics with focus to our algorithm selection scenario (see section 3 for more details). The existing experience contains knowledge about the application of a specific algorithm to a given dataset, e.g. the error rate or the used training time.

From the three aspects just discussed we derived the structure of the cases in our case base (see figure 2). In general, a CBR-approach distinguishes between a problem description and a solution description. The problem description contains all information that is known about the current problem. In our algorithm selection scenario the problem description is defined by the data characteristics and the application restrictions, i.e. the algorithm description. These description may be partial. The solution description is the completion to the problem description. In our scenario, the solution description consistis of the experience part of the case.

The general work flow is that the user specifies his application requirements and that DCT computes the data characteristics for the given dataset. These two aspects constitute the problem description. In our AST system we compute the most similar cases by comparing this problem description with the problem descriptions that are found in the cases of our case base.

[1] DCT is developed by Guido Lindner and Robert Engels in collaboration with the master thesis of U. Zintz and C. Theusinger. A first presentation can be found in [Engels and Theusinger, 1998], which focuses on supporting data mining pre-processing.

Today, the case base contains more than 1600 cases as the result of 21 classification algorithms and more than 80 datasets. At the moment, our system is realized for supervised classification tasks which are an important task type in machine learning and KDD applications. The collected datasets are taken from the UCI repository [Merz and Murphy, 1996] and from real world applications from DaimlerChrysler.

# 3 Data Characteristics with DCT for AST

The data characterization tool DCT computes various meta data about a given data set. Subsequently, we just briefly characterize the relevant data characteristics. The data characteristics can be separated into three different parts:

1. simple measurements or general data characteristics

2. measurements of discriminant analysis and other measurements, which can only be computed on numerical attributes.

3. information theoretical measurements and other measurements, which can only be computed on symbolic attributes.

The first group contains measurements which can be simply calculated for the whole dataset like the number of attributes or default error rate. The other groups can only be computed for a subset of attributes in the dataset. The measurements of discriminant analysis are calculated only for numerical attributes whereas the information theoretical measurements are calculated for symbolic ones. All these measurements are calculated by our data characteristic tool (DCT). The used measurements are described in [Lindner and Studer, 1999] and are available from the web site
$www.aifb.uni-karlsruhe.de/publications$.

# 4 Algorithm Characteristics

Normally, the user can define some characteristics regarding the algorithms that should be used for his/her data mining application. For AST we started with a set of simple and easily understandable characteristics. This set of characteristics for algorithms is not complete, but can be specified by every user, independent of his or her skill in data mining or machine learning. The following characteristics which have to be provided by the user of the AST system, are used in our approach today (compare figure 2): algorithm or algorithm class, interpretability of the model (model type), training time and testing time.

The algorithms which the system handles are modeled in a taxonomy . Such a taxonomy makes it possible to assign algorithms to algorithm class. Currently, we use the following algorithm classes: rule learner, decision trees, neuronal nets, bayes nets and instance based learner.

To characterize the model that is generated by an algorithm from an application point of view, we only use the interpretability of the model and the specific value *no* for algorithms which compute no operational model. For the moment, we do not consider the different kinds of learning result representations. Training and testing time contain symbolic values like fast or slow.

These values describe properties of the algorithms, i .e . here we make only statements about the algorithm in general and not about the examined application [2]. In order to achieve understandability and simple usage, we classify the learning (TrainTime) time in five classes and the classification time (Test Time) in three classes. To build these clusters we use KMEANS [Clementine$^{\text{TM}}$, 1998] to get compact clusters. Furthermore we have to add the selected parameter values to the algorithm descriptions. Today, all algorithms of the case base are tested with their default parameters values. One special property, which is currently not supported is the cost of misclassification. This aspect will be added in the near future.

# 5 Experiments on Recommendation Quality

At first we have to define applicability for the algorithms on the datasets. In [Gama and Brazdil, 1995] three different methods are presented to define applicability. We use method 1 of that proposal: Based on the error rate ($ER$) of the best algorithm and the number of records ($NT$) we compute an error margin ($EM$):

$$EM = \sqrt{\frac{ER \cdot (100 - ER)}{NT}} \tag{1}$$

An algorithm is applicable to a dataset if its error rate is smaller than $ER + k \cdot EM \quad (k \in N)$. In our evaluation we use $k = 4$. This definition of applicable is equal to the definition used in the Statlog project, however we use a small constant $k$ for all datasets to get small ranges of intervals, which define the set of applicable algorithms for a dataset.

In the following we describe the procedure of our evaluation:

1. For the selected dataset each associated case is extracted from the case base. This means that 21 cases are removed from the case base (currently, we handle 21 algorithms in our case base)

2. For the selected dataset we compute the most similar dataset by comparing the data characteristics [3].

[2] The training and testing time for a special application is part of the solution description of a case (TrainTime and TestTime).

[3] Since the UCI repository does not provide application restrictions the problem description, of the cases are reduced to the data characteristics of the datasets.

| case | most similar |
|---|---|
| algorithm | best ∈ applicable algo. |
| mixed | 85.71% |
| numeric | 86.21% |
| symbolic | 67.74% |
| all | 79.01% |

Table 1: Applicability of the recommendation

3. If the best algorithm of the most similar dataset is applicable to the selected dataset, we count this test as a positive recommendation. This means that the recommendated algorithm must be element of the applicable algorithms for the selected dataset.

This comparison was done for all datasets. Table 1 shows the results of this evaluation. Over all datasets the best algorithm of the most similar dataset is applicable in 79%. For applications with only numeric attributes or with numeric and symbolic (mixed) ones the rate is higher than 85%. These are rather good results. It can be seen that the result for datasets with only symbolic attributes is not so good. This is an indicator that the data characteristics for the symbolic attributes are still insufficient and that some additional measurements are needed.

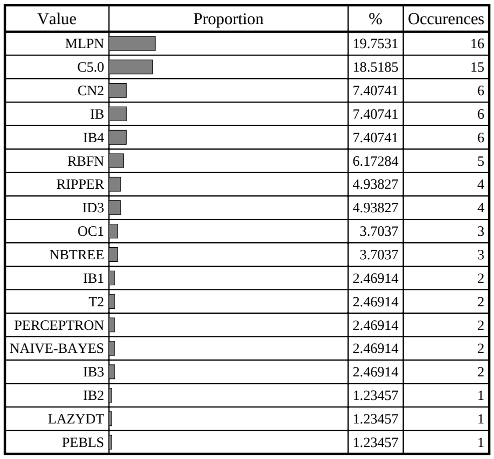
Several other aspects are of interest in the context of our experiments:

- How is the distribution of the *best* algorithm in our experiments? Figure 3 shows that there is no best algorithm in general.

- What is the distribution of numerical, symbolic and mixture datasets? Our collection of datasets contains 31 symbolic, 30 numeric and 22 mixed datasets. This show that the weakness of recommendation for symbolic applications is not cause on few cases in the case base.

- How many algorithms are applicable? The number of applicable algorithms depends on the complexity of the application. For some applications are only a few algorithms are applicable (satimage or nursery), for other rather simple applications like iris nearly all algorithms are applicable.

As mentioned, our experiments did not use application restrictions for selecting cases since the UCI repository does not provide such application restrictions. However, in a real-life project within DaimlerChrysler we made use of both aspects of problem descriptions: application restrictions and data characteristics. Actually, we had two types of learning algorithms at hand: a decision tree learner (C5.0) and a neural network (a multi-layer perceptron). When applied to the given problem, the neural network resulted in lower error rates than C5.0. Therefore, we selected the neural network



| Value | Proportion | % | Occurences |
|---|---|---|---|
| MLPN | | 19.7531 | 16 |
| C5.0 | | 18.5185 | 15 |
| CN2 | | 7.40741 | 6 |
| IB | | 7.40741 | 6 |
| IB4 | | 7.40741 | 6 |
| RBFN | | 6.17284 | 5 |
| RIPPER | | 4.93827 | 4 |
| ID3 | | 4.93827 | 4 |
| OC1 | | 3.7037 | 3 |
| NBTREE | | 3.7037 | 3 |
| IB1 | | 2.46914 | 2 |
| T2 | | 2.46914 | 2 |
| PERCEPTRON | | 2.46914 | 2 |
| NAIVE-BAYES | | 2.46914 | 2 |
| IB3 | | 2.46914 | 2 |
| IB2 | | 1.23457 | 1 |
| LAZYDT | | 1.23457 | 1 |
| PEBLS | | 1.23457 | 1 |

Figure 3: Best Algorithm Distribution

as the learning algorithm for our project. This result coicides with the recommendation that was given when applying our AST system in the project context: based on the data characteristics and the application restrictions of the given problem, AST determined those cases as the most similar cases which used the neural network for learning. Besides providing a recommendation for selecting the right algorithm, our AST system also computes the degree of similarity of the selected cases with the problem at hand. In that way, the user of our system receives valuable information about the quality of the recommendation.

# 6 Conclusion and Future Work

In this paper, we introduced our CBR approach for algorithm selection and described first evaluations of our protoype system AST. Our approach contains several advantages for algorithm selection. The user does not only get a recommendation which algorithm should be applied, he/she gets also an explanation for the recommendation in the form of past experiences available in the case base. Another strong point is the maintenance of such a system. In contrast to other approaches, a new algorithm can be added to the case base without having to test this algorithm on all datasets that have been considered so far. Furthermore, with an extension of the algorithm description it will also be possible to determine similar algorithms and to compare their model generation results on similar datasets. Finally, with a CBR approach we can use similarity operators instead of the strong, hard-coded rules which are used in approaches like Statlog [Michie et al., 1994]

In the future we have to refine our case description of algorithms and datasets. A main point is to include the parameter settings of the algorithms into the case structure.

We also plan to integrate our approach into an internet service for algorithm selection. Such an internet service will offer an algorithm recommendation for a specific application problem that was defined by the user of this service.

# References

[Clementine$^{\text{TM}}$, 1998] Clementine$^{\text{TM}}$(1998). *Clementine$^{TM}$ Data Mining System, Reference Manual.* Integral Solutions Limited.

[Engels et al., 1997] Engels, R., Lindner, G., and Studer, R. (1997). A guided tour through the data mining jungle. In D. Heckerman, H.Manilla and D.Pregibon, editor, *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Newport Beach, CA, August 14 -17. AAAI Press, Menlo Park, CA.

[Engels and Theusinger, 1998] Engels, R. and Theusinger, C. (1998). Using a data metric for offering preprocessing advice in data mining applications. In Prade, H., editor, *Proceedings of the 13th biennial European Conference on Artificial Intelligence (ECAI-98)*, pages 430–434. John Wileys & Sons.

[Gama and Brazdil, 1995] Gama, J. and Brazdil, P. (1995). Characterization of classification algorithms. In Mamede, N. and Ferreira, C., editors, *Advances on Artificial Intelligence - EPIA95.* Springer Verlag.

[Krodratoff et al., 1992] Krodratoff, Y., Sleeman, D., Uszynski, M., Causse, K., and Craw, S. (1992). *Enhancing the Knowledge Engineering Process*, chapter Building a Machine Learning Toolbox. Elsevier Science Publishers, North Holland.

[Lindner and Studer, 1999] Lindner, G. and Studer, R. (1999). AST: Support for Algorithm Selection with a CBR Approach. In *Recent Advances in Meta Learning and Future Work*, Workshop Proceedings of the ICML 1999, Bled, Slowenien.

[Merz and Murphy, 1996] Merz, C. J. and Murphy, P. M. (1996). Uci repository of machine learning databases. [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[Michie et al., 1994] Michie, D., Taylor, C., and Spiegelhalter, D. (1994). *Machine Learning, Neural and Statistical Classification.* Ellis Hoorwood.

[Schaffer, 1994] Schaffer, C. (1994). A conservation law for generalization performance. In *Proc. Eleventh Intern. Conf. on Machine Learning*, pages 259 – 265, Palo Alto, CA. Morgan Kaufmann.

[Sleeman et al., 1995] Sleeman, D., Rissakis, M., Craw, S., and N.Graner (1995). Consult 2 Pre- and Post of Machine Learning Applications. *International Journal of Human Computer Studies*, (43):43 – 63.