AST: Support for Algorithm Selection with a CBR Approach

Guido Lindner
DaimlerChrysler AG
Research and Technology FT3/KL
po: DaimlerChrysler AG, T-402
D-70456 Stuttgart, Germany
Email: guido.lindner@daimlerchrysler.com

Rudi Studer
Institute AIFB
University of Karlsruhe
D-76128 Karlsruhe, Germany
Email: 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. The problem of algorithm selection is a decision based on application restrictions (top-down), a given dataset with its meta-data characteristics (bottom-up) and on knowledge about the available algorithms. These aspects also define a case in our approach. We discuss the architecture of AST and explain the basic components. We present the evaluation of our approach in a real world application and provide in addition a systematic analysis of the case retrieval behaviour and thus of the selection support offered by our system.

1 Introduction

One main aspect of developing a data mining application is the selection of a most suitable algorithm for the model generation phase by the user [Brachman and Anand, 1996]. This is a crucial step since the applied algorithm has a strong impact on the overall behavior of the developed data mining application. This selection problem may be found in other disciplines addressing data analysis aspects, as well. Indeed, the problem of algorithm selection can be found in statistics cf. [Hand, 1994a] [Hand, 1994b]) as well as in machine learning.

In the field of machine learning various approaches address the problem of algorithm selection, for example the MLT project with its Consultant system [Consortium, 1993] as well as the Statlog project [Michie et al., 1994]), aiming at comparing the performance of a fixed set of algorithms on several data sets. In the Statlog project 23 algorithms were evaluated on 21 data sets. A similar perspective on model selection can be found in [Kohavi et al., 1997], where these ideas form the background motivation for the MLC++ library. In accordance with the Statlog approach, the MLC++ approach also advises to apply all available algorithms in order to select the *best* model generating algorithm for the current application.

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 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 and Statlog 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

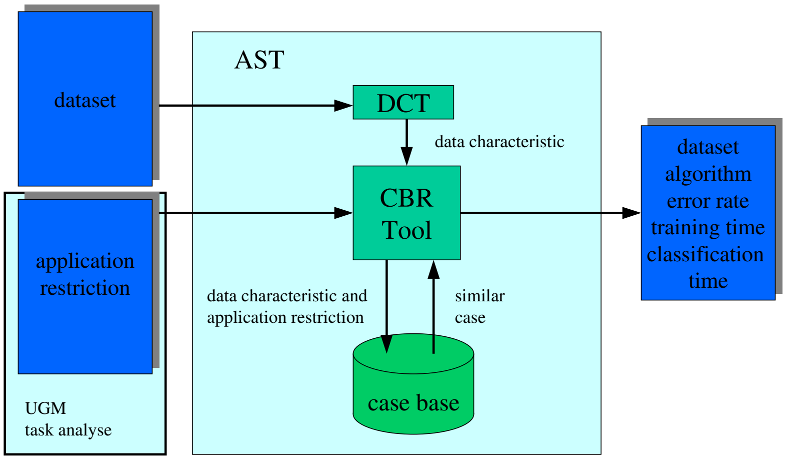

Figure 1: Architecture of AST

applicable algorithms [Consortium, 1993]. 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 general problems with rules as learned in this meta learning step are as follows:

1. How representative are the examples of the meta learning step?
2. Instead of hard boundaries one would like to have more fuzzy rules, so that surfaces of decisions are smoother. It is not acceptable in a real-world application to exclude a particular algorithm from further consideration on the fact that 35.040 records are present instead of a (fictive) maximum of 35.000.

Having these disadvantages in mind, we decided to use a case-based reasoning approach (CBR) for providing support in selecting most suitable algorithms. 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 given in detail in section 3. Another point 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 about algorithm descriptions in such a approach 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.

2 Architecture of AST

The top-level architecture of our AST system is shown in Figure 1. [1]. As outlined in the introduction, the problem of algorithm selection is a decision based on three factors:

- application restrictions
- given data
- existing experience

Application restrictions and the given data define the problem description for our CBR approach and the existing experience with the applied algorithm is the solution description to the known applications.

Embedded in our UGM approach [Engels et al., 1997] the application restrictions are analyzed by the task analysis component. In addition, the user can also feed his/her restrictions directly into the system. From the given data, the data characterization tool (DCT) [2]

[1] To realize our CBR approach we use the CBR-Tool CBR-Works (trademark from tecinno)

[2] DCT is developed by Robert Engels and Guido Lindner in collaboration with the master thesis of U. Zintz

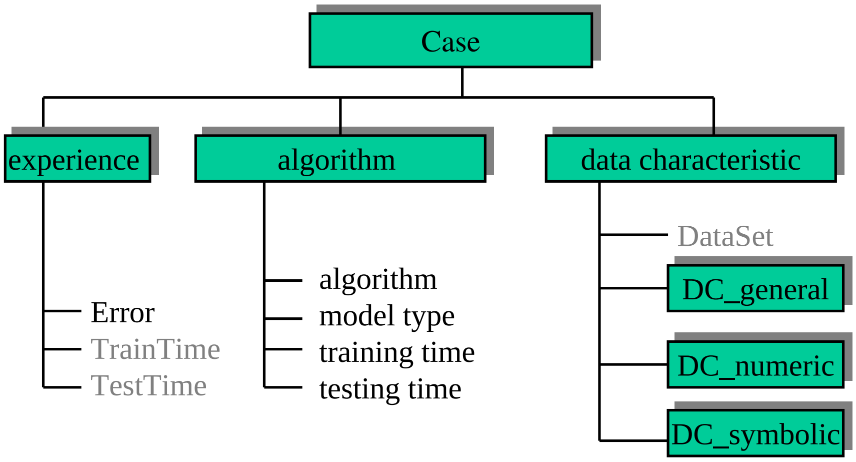

Figure 2: Case structure in AST

computes data characteristics with focus to algorithm selection. Furthermore a case contains the experience on the known applications as well, i.e. a case represents the knowledge about the execution of a special algorithm on a specific dataset. This knowledge includes the training and test time and the error rate on this known dataset. Furthermore, a case a contains description of the algorithm and the data characteristic. Details are given in the next sections.

The general work flow is that the user specifies his requirements and that the data characteristic for the given dataset is computed by DCT. These two groups of information define the problem description. Each case is defined by this problem description and a solution part, which specifies the applied algorithms as well as the experience gained from applying the algorithm. In our system we compute the most similar applications problem description and offer the user also the results of the applied algorithms. Important is also to remark that the problem description may be incomplete.

In figure 2 we illustrate the case structure. The grayed objects (TrainTime, ...) are part of the solution part. 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 classification tasks which are an important task type in machine learning and KDD applications. The collected datasets are from the UCI repository [Merz and Murphy, 1996] and real world applications from DaimlerChrysler.

and C. Theusinger. A first presentation can be found in [Engels and Theusinger, 1998], which focuses on supporting data mining pre-processing.

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

The first group contains measurements which can be calculated for the whole dataset. 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). Figure 3 lists our definition of data characteristics in AST.

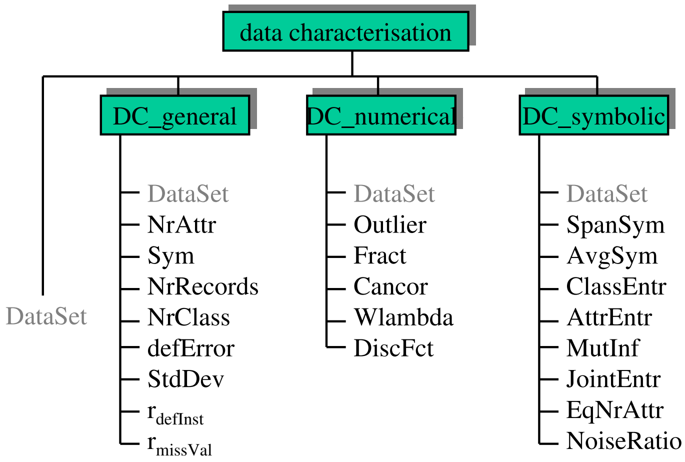

Figure 3: Data Characteristic in AST

3.1 Simple Measurements or General Data Characteristic

In the context of our algorithm selection problem the DCT tool determines the following general characteristics:

- NrRecords: number of records (n)
- NrAttr: number of attributes (m)
- Sym:ratio of symbolic attributes (m_{sym}/m)
- NrClass: number of classes (q)
- defError: default error rate $defError = 1 - Acc_{def}$,
  (Acc_{def}) probability of the largest class or default accuracy
- StdDev: standard derivation of the class distribution(σ_{class})
- $r_{defInst}$: relative probability of defective records $r_{defInst} = \frac{n_{defTuple}}{n}$
  $n_{defTuple}$: number of recordswith missing values
- $r_{missVal}$: relative probability of missing values : $r_{missVal} = \frac{h_{missVal}}{n*m}$
  $h_{missVal}$: number of missing values

Beside the *normal* simple measurements we selected relative measurements like the two last measurements. Such a ratio measurement contains more information and is more interpretable.

3.2 Discriminant Measurements

Beside these general measurements we use some measurements for algorithm selection which can only be calculated for numerical attributes. In essence, DCT computes a discriminant analysis leading to the following measurements:

- **Fract**: Describes the relative importance of the largest eigenvalue as an indication for the importance of the 1st discriminant function.
- **Cancor**: Canonical correlation, which is an indicator for the degree of correlation between the most significant discriminant function and the class distribution. There is a strong correlation between the classes and the 1st discriminant function if this measurement is close to unity.
- **DiscFct:** Number of discriminant functions
- **Wlambda:** Wilks Lambda or U-statistic, describes the significance of the r discriminant functions and is defined as follows:

  $$\Lambda = \prod_{j=1}^{DiscFct} \frac{1}{1+\lambda_j} \tag{1}$$

  If Wilks Lambda is near zero, in principle this indicates there is a good possibility for making good discriminations.

3.3 Information Theoretical Measurements

Besides continuous (numerical) attributes, it is likely that symbolical attributes are used for describing a data space. Therefore, measures are needed to cover these (symbolical) dimensions as well. Again, the goal is primarily to investigate and deploy measures that are useful for the algorithm selection process. All these measurements are well known and based on the entropy of the attributes. Entropy measures have the common property that they deliver information on the information content of attributes.

- class entropy (ClassEntr)
- join entropy (JoinEntr)
- average attribute entropy (AttrEntr)
- average mutual information (MutInf)
- relevance-measure (EqNrAttr)
- Signal Noise Ratio (NoiseRatio)

Additionally, we also use a measurement of range of occurrence defined by $SpanSym = SymMax - SymMin$ [3] and $AvgSym$ which is the average number of symbolic values. Such measurements are indicators of the complexity and the size of the hypotheses space for the problem.

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/class
- interpretability of the model (model type)
- training time
- testing time

The algorithms which the system handles are modeled in a taxonomy (cf. figure 4). Such a taxonomy makes it possible to model the relation of algorithm and algorithm class.

[3]SymMax is the maximal number of distinct values for one symbolic attribute and SymMin the minimum.

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 (see also table 1). 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 [4]. 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^{TM}, 1998] to get compact clusters. Table 1 shows the included algorithms with their properties. 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.

Additionally to these requirements the user can define an acceptable error rate for the application.

Additionally, several other algorithm characterizations are possible. A good overview about the vocabulary to describe classification algorithms is given by [Hilario and Kalousis, 1999]. One special property which is not supported, is the question of cost handling for misclassification. This aspect will be added in the near future.

5 Experiments on Recommendation Quality

In this section, we discuss the recommendation quality of our system. First, we evaluate the recommendation quality for the datasets of our case base. In a second step we present the results for a real world application from DaimlerChrysler AG. This application is presented in detail in [Lindner and Studer, 1999].

5.1 Evaluation of the Approach

To evaluate our approach, we constructed a large amount of tests as follows. For each dataset we compute the three most similar datasets and compare the

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

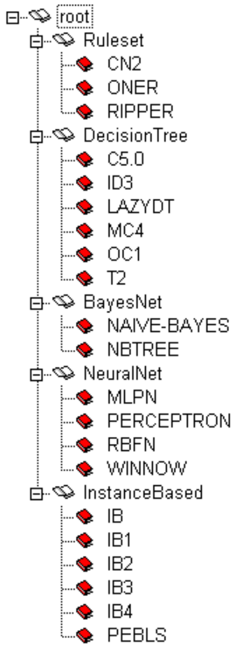

Figure 4: Algorithm taxonomy

applicable algorithms. So at first we have to define applicability for this test. 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):

$$EM = \sqrt{\frac{ER \cdot (100 - ER)}{NT}} \tag{2}$$

Pre-condition for this method is that each dataset is tested with every algorithm. 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. In our approach it is not nessesary to get several algorithms that are applicable on a dataset. Additionally, this measure is only used for the evaluation and is not part of the solution description.

Each case of a dataset was extracted from the case base and the recommendation of our system compared with the results of the further tests on this application. This means that one dataset (21 cases) is removed from the case base. For this dataset we compute the similar applications. The best is compared with the test results on the removed dataset. If the best algorithm on the similar dataset element of the applicable algorithm for the requested dataset, we count it as a positive recommendation. This comparison was done for all datasets. Table 2 shows the results of this evaluation. Over all applications the best algorithm of the 1^{st} similar case is applicable in 79%. For applications with only numeric attributes or with numeric and symbolic (mixed) ones the rate is over 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.

5.2 Model Selection for a Real World Application

In section 5.1 we evaluated our approach without considering the application restrictions. With the following example of a real world application we demonstrate that the application restrictions are also used in the algorithm selection process.

When developing a data mining application for DaimlerChrysler AG that forecasts the complaint behavior

algorithm	model	training time	testing time
C5.0	interpretable	very fast	fast
CN2	interpretable	fast	fast
IB	no	very fast	slow
IB1	no	moderate	slow
IB2	no	moderate	slow
IB3	no	fast	moderate
IB4	no	fast	moderate
ID3	interpretable	very fast	fast
LAZYDT	interpretable	very fast	slow
MC4	interpretable	very fast	fast
MLPN	non interpretable	very slow	moderate
NAIVE-BAYES	no	very fast	fast
NBTREE	no	slow	fast
OC1	interpretable	slow	fast
ONER	interpretable	moderate	fast
PEBLS	no	very fast	moderate
PERCEPTRON	non interpretable	very fast	fast
RBFN	non interpretable	very slow	slow
RIPPER	interpretable	moderate	fast
T2	interpretable	very slow	fast
WINNOW	non interpretable	very fast	fast

Table 1: Properties of classification algorithm in AST

case	1^{st} similar
algorithm	best $\in$ applicable algo.
mixed	85.71%
numeric	86.21%
symbolic	67.74%
all	79.01%

Table 2: Applicability of the recommendation

of cars, we also had to deal with the algorithm selection problem within the modeling phase of the knowledge discovery process [Chapman et al., 1998]. In this project, application restrictions and requirements for the algorithm to be selected were defined as follows:

- No restriction for the model type.
- Learning time and test time are not critical in this application.
- As part of a larger project a special KDD-tool was selected, which contains C5.0 [Quinlan, 1997] as a decision tree learner and a multi layer perceptron as a neural network.
- Besides the predictive accuracy the robustness of the model was of interest. However, this aspect is not part of our application restrictions yet.

We also computed the data characteristic for the given data. The problem description composed of application restrictions and data characteristics are used by AST to compute the most similar cases. In figure 5 we show the selected most similar data sets with the results of C5.0 on these datasets which are glass2 and liver from the UCI repository.

Besides the results of applying the different algorithms on these applications the user gets a degree of similarity of the selected cases. This degree serves as an indicator of the quality of the recommendation given by AST. In our example the value for liver is 0,795 and for glass2 is 0,799.

When comparing C5.0 with the neuronal net for these two most similar applications the neural net reached a better error rate than C5.0. This result coincides with our experience during the model selection phase of the project: the neural net delivered better results than C5.0 in this forecasting application.

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]

Another approach to support classifier selection showing some similarities to our work is presented in [Kalousis and Theoharis, 1999]. In this approach the main idea to support model selection by data characteristics. They define different data characteristics and use a nearest neighbor algorithm. However, they get only an estimated accuracy of 58% for the proposed algorithms. Additionally, they use only data characteristics. We also consider application restrictions for the algorithm selection process.

Another approach to support the knowledge discovery process was presented by [Zhong et al., 1997]. They support the KDD–process with a planning component, but they do not show a technical solution for the selection of an appropriate data mining algorithm.

Also work on ranking of algorithms, like the work of [Nakhaeizadeh and Schnabl, 1997] have a strong influence on our approach with respect to the evaluation of the applicability of algorithms. [Nakhaeizadeh and Schnabl, 1997] show an approach which also considers other requirements of the application to compute a ranking. Some aspects of such an approach are already integrated in our CBR-approach in the way that we also consider application restrictions.

In the future work 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. Each algorithm together with its parameter specification then defines one method. At the moment it was not necessary to include that aspect because we started our work with the default parameter settings for the different algorithms. Furthermore we have to extend our case base with other public datasets and to test the applied algorithms, including different parameter settings.

But we also plan to integrate our approach into a internet service for algorithm selection. In such an internet service we will offer an algorithm recommendation for a specific application problem. For such a service, we need the application requirements and the data characteristic of the data. For real world applications, where the data are very sensitive, we have to offer DCT to the clients of our service. Obviously, the success of such a project depends on the clients, because we also need a feedback of the results of the application of the algorithms. However, we are rather optimistic that the clients will provide such a kind of feedback since there is a clear need in the community for such a service.

Acknowledgments
We thank Andreas Hotho, Robert Engels, Melanie Hilario and Alexandros Kalousis for fruitful discussions and reading of previous versions.

References

[Brachman and Anand, 1996] Brachman, R. and Anand, T. (1996). Advances in Knowledge Discovery and Data Mining. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, chapter The Process of Knowledge Discovery in Databases: A Human–Centered Approach, pages 33–52. AAAI/MIT Press.

[Chapman et al., 1998] Chapman, P., Clinton, J., Hejlesen, J. H., Kerber, R., Khabaza, T., Reinartz, T., and Wirth, R. (1998). The Current CRISP-DM Process Model for Data Mining. http://www.ncr.dk/CRISP/index.html.

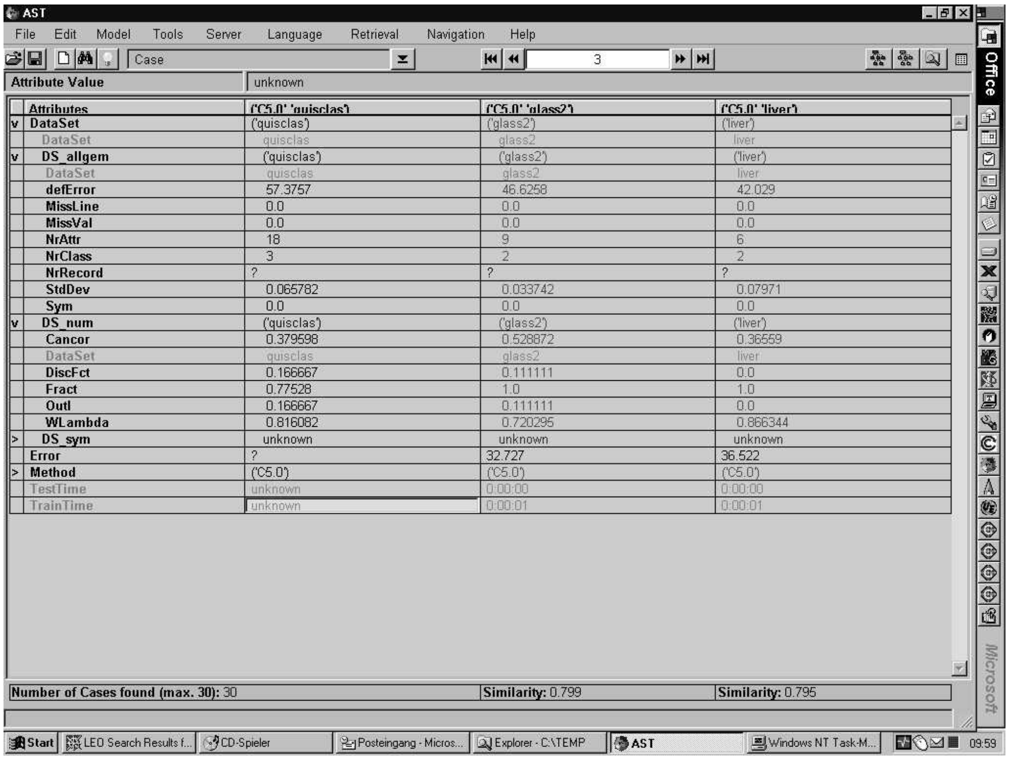

Figure 5: Simalarity Result of AST

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

[Consortium, 1993] Consortium, M. (1993). Final public report. Technical report, Esprit II Project 2154.

[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.

[Hand, 1994a] Hand, D. (1994a). Deconstructing statistical questions. *Journal of the Royal Statistical Society*, pages 317 – 356.

[Hand, 1994b] Hand, D. (1994b). Statistical strategy: step1. In Cheeseman, P. and Oldford, R., editors, *Selecting Models from Data: AI and Statistics IV*, volume 89. Lecture Notes in Statistics.

[Hilario and Kalousis, 1999] Hilario, M. and Kalousis, A. (1999). Characterizing Learning Models and Algorithms for Classification. Technical Report UNIGE-AI-99-01, CUI - University of Geneva.

[Kalousis and Theoharis, 1999] Kalousis, A. and Theoharis, T. (1999). NOEMON: An Intelligent Assistant for Classifier Selection. In *Workshop proceedings of the ICML 1999, Workshop 1.*

[Kohavi et al., 1997] Kohavi, R., Sommerfield, D., and Dougherty, J. (1997). Data Mining using

MLC++, A Machine Learning Library in C++. *Int. Journal on Artificial Intelligence Tools*, 6(4):537–566. [http://www.sgi.com/Technology/mlc].

[Lindner and Studer, 1999] Lindner, G. and Studer, R. (1999). Forecasting the Fault Rate Behaivior of Cars. In *Proc. of the workshop from the ICML 1999, Workshop 5,*. also submitted to the KDD'99.

[Merz and Murphy, 1996] Merz, C. J. and Murphy, P. M. (1996). Uci repository of machine learning databases. [http://www.ics.uci.edu]. 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.

[Nakhaeizadeh and Schnabl, 1997] Nakhaeizadeh, G. and Schnabl, A. (1997). Development of a Multi–Criteria Metrics for Evaluation of Data Mining Algorithms. In D.Heckerman , H.Manilla and D. Pregibon, editor, *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 36 – 42, Newport Beach, CA. AAAI, Press, Menlo Park, CA.

[Quinlan, 1997] Quinlan, J. R. (1997). An Informal Tutorial. http://www.rulequest.com.

[Zhong et al., 1997] Zhong, N., Liu, C., Kakemoto, Y., and Ohsuga, S. (1997). Kdd process planning. In D. Heckerman, H. Manilla, D.Pregibon, editor, *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 291 – 294, Newport Beach. AAAI Press, Menlo Park, CA.