Representational Issues in Meta-Learning

Alexandros Kalousis KALOUSIS@CUI.UNIGE.CH
Melanie Hilario HILARIO@CUI.UNIGE.CH
University of Geneva, Computer Science Department, 24-rue du General Dufour, Geneva, Switzerland.

Abstract

To address the problem of algorithm selection for the classification task, we equip a relational case base with new similarity measures that are able to cope with multi-relational representations. The proposed approach builds on notions from clustering and is closely related to ideas developed in similarity-based relational learning. The results provide evidence that the relational representation coupled with the appropriate similarity measure can improve performance. The ideas presented are pertinent not only for meta-learning representational issues, but for all domains with similar representation requirements.

1. Introduction

Classification algorithm selection can be seen as one of the 'holy grails' of the machine learning field, it is common knowledge that there is no algorithm uniformly superior over all possible problems.

The most common and widely accepted methods for performing algorithm selection require substantial expertise on machine learning and involve systematic experimentation and evaluation. In recent years there have been some efforts to automate the selection and lift the experimental burden by relying on some form of learning, most often called meta-learning. The whole idea dates back to the work of (Rendell et al., 1987), it became more concrete within the European project STATLOG, (Michie et al., 1994), to finally find its full expression in another European project, METAL, which produced a considerable amount of publications (Brazdil et al., 2003; Pfahringer et al., 2000).

The main concept is very simple and views classification algorithm selection as just another learning problem. The training examples consist of descriptions of complete classification datasets and the target; the latter is usually defined from a performance based preference order over the set of available algorithms for the given dataset. While there has been significant work on the definition of preference order over learning algorithms and the construction of the target, little attention has been given to representational issues that arise when one tries to describe datasets.

The heart of the representational issues can be traced to the one-to-many relationships that appear in the descriptions of classification datasets. A training instance in a meta-learning setup is a dataset description, which comprises characterizations of each of the attributes that constitute the dataset[1]. In the propositional framework it is not possible to retain the complete information about the individual attributes, since this would clearly result in meta-instances of variable length, depending on the number of the attributes of a given dataset.

The bulk of the work in meta-learning naively addressed the representational problem by resorting to descriptions of properties that are computed for each one of the attributes of a dataset using averages or at most the min and max statistics. Two exceptions are, the work of (Todorovski & Dzeroski, 1999) where the problem was treated as a multi-relational problem, as it actually is, and handled via first-order learners; and the work of (Kalousis & Theoharis, 1999) where the distributions of the properties are described using histograms. The former suffers from a well known problem of first-order learning; the high number of degrees of freedom of the search space, which is determined by the number of properties used to describe datasets and the actual number of attributes of the

[1]This is true when statistical and information based properties are used to described the datasets. But it is not true when datasets are described via landmarkers (Pfahringer et al., 2000), or via model based characteristics as in (Peng et al., 2002)

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

dataset. This results in an extremely time consuming learning process and at the same time increases the chances of overfitting by accidental discovery of invalid patterns, especially when the number of training instances is small as is typical in applications of meta-learning. The latter attacks the high dimensionality of the search space adequately, as its dimensions are only determined by the examined properties. However the semantic power of the distribution based representation is not fully exploited in the framework of classical propositional learners.

In a previous paper, (Hilario & Kalousis, 2001), we described the case representation of a relational case-based system which served as a repository of classification experiments. Thorough records were kept on every aspect of these experiments. The goal was to use the case-base as a predictive tool, that would provide support in classification algorithm selection, but also as an explanatory tool determining the areas of expertise of classification algorithms.

Although case-based systems do not induce first-order theories they can make use of multi-relational representations addressing thus the representational requirements of the meta-learning problem. The collective treatment of dataset attributes and their properties in the similarity measures can make case based systems less susceptible to overfitting phenomena. In a first order meta-learning scenario we would be looking for rules of the form: $\exists$ attribute whose property I satisfies condition X. As the number of attributes increases chances that we would, accidentally, discover invalid rules of that form which hold on the training set also increase. The appropriate selection of similarity measures, which would be defined over all attributes and their properties in contrast to the existential approach, can alleviate that problem.

In this paper we continue the work started in (Hilario & Kalousis, 2001). We equip the relational case-base with precise similarity measures, that can cope with the multi-relational structure of the information describing the classification datasets, Section 3, and undertake a systematic evaluation of the predictive power of the relational case-based classification, Section 5. The evaluation takes place within a specific meta-learning framework described in Section 4. The next section gives a brief overview of the cases.

2. Overview of the Cases

In Figure 1 we give the overall description of a case in the relational case-based system. We now give a brief description of the main components of a case.

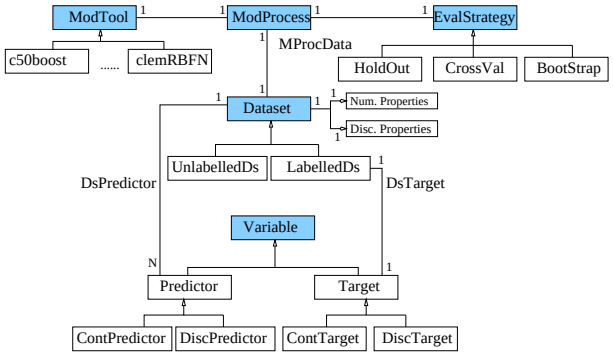

Figure 1. Case representation

- ModTool : A general class that provides the profile of a modeling tool. In principle it can be any learning algorithm, for example a supervised learner, a regression or a clustering algorithm, although here we focus only on classification algorithms. The profiles of the learning algorithms have been constructed via extensive experimentation. The subclasses of ModTool represent specific classification algorithms along with specific instantiations of their training and testing parameters.

- EvalStrategy : A general class that defines the concept of an evaluation strategy. Its subclasses are specific evaluation strategies, like HoldOut or CrossValidation. Each subclass contains characteristics that are specific to the corresponding evaluation method (e.g. the number of folds and the number of times that a Cross Validation is repeated).

- Dataset : The class of all datasets that have been used in the different learning experiments. It contains a collection of N attributes (attributes are instances of the Variable object). It can be a Labeled Dataset, i.e. a dataset with a target variable, continuous (regression task), or discrete (classification task), or an UnLabelled one where no target variable is defined. For each dataset a number of high level characteristics are recorded, like the number of classes, number of attributes, and summary statistics of statistical and information based properties of the individual attributes of the dataset.

- Variable : The class of variables or attributes of a dataset, whether predictors or targets. A number of statistical and information based properties are recorded for every attribute depending on its nature (discrete or continuous) and its role (target or predictor).

A Modeling Tool can be applied and evaluated on a Dataset using a specific Evaluation Strategy. The complete results of the evaluation are stored in a ModProcess object, which links together the Dataset, ModTool and EvalStrategy objects. A case in the case base is a learning episode, i.e. the application and evaluation of a learning algorithm to a specific dataset.

3. Defining the Similarity Metrics

We have defined the representation of cases in what is essentially a relational case base. In order to fully exploit that representation we have to define similarity measures that are able to cope with it.

The most important component of the relational case base is the Dataset object. Each instance of the Dataset class is associated with a set of instances of the Variable class. Any similarity measure defined over a Dataset object should also take into account the elements of the Variable class associated with that Dataset object. Before addressing the problem of similarity definition between sets of variables we have to define a similarity measure among the individual elements of the sets, i.e. among the variables.

A dataset I includes a set of variables S_I. Each variable $v_i \in S_I$ is described by a vector u of n dimensions describing various properties of the variable. The similarity between any two variables v_i, v_j, whether from the same or from different datasets, will be given by :

$$sim(v_i, v_j) = \frac{1}{n}\sum_n (sim(u_{v_{i_n}}, u_{v_{j_n}})) \quad (1)$$

$$= \frac{1}{n}\sum_n (1 - \frac{|u_{v_{i_n}} - u_{v_{j_n}}|}{u_{n_{max}} - u_{n_{min}}}) \quad (2)$$

The given similarity measure is based on the normalized Manhattan distance between the two vectors that describe the corresponding variables. Any measure of similarity between two sets of variables will make use of the similarity measure between individual variables defined by Formula 1.

The definition of similarity measures between sets of objects, in our case the sets of variables that constitute the datasets, is not trivial. There is no unique-best similarity measure and the appropriate selection always depends on the semantics of the problem at hand. For example in the scope of similarity-based multi-relational learning (Kirsten et al., 2001), the similarity between two sets is defined as the sum of the maximum similarities of the elements of the set with the lower cardinality with the elements of the set with the greater cardinality, normalized by the cardinality of the greater set. The normalization with the cardinality of the larger set results in very low similarities when the two sets have very different cardinalities. More formally for two sets S_I and S_J with possible different cardinalities n_i and n_j, where the similarity between elements $v_i \in S_I, v_j \in S_J$, is given by Formula 1, the similarity according to Kirsten et al. (2001) is:

$$sim_K(S_I, S_J) = \begin{cases} \frac{1}{n_j}\sum_i (\max_i sim(v_i, v_j)), & n_i < n_j \\ \frac{1}{n_i}\sum_j (\max_j sim(v_i, v_j)), & n_i \geq n_j \end{cases}$$

Very similar ideas are quite developed in clustering algorithms and more precisely for defining the similarities between sets in agglomerative hierarchical clustering (Duda et al., 2001). Exploiting the expertise developed there, we have chosen to implement and evaluate two different similarity measures based on measures commonly used by the clustering community. The first one is based on the similarity used in the *single linkage* clustering algorithm, where the similarity between two sets is defined as the maximum similarity observed between all pairs of elements of the two sets, while the second one is based on the similarity used in the *average linkage* clustering algorithm, where the similarity between two sets is defined as the average similarity between all pairs of elements from the two sets. More formally we have:

- *Single Linkage Based Similarity*

$$sim_{SL}(S_I, S_J) = max_{ij}(sim(v_i, v_j)),$$

- *Average Linkage Based Similarity*

$$sim_{AL}(S_I, S_J) = \frac{1}{n_i n_j}\sum_{ij}(sim(v_i, v_j)),$$

For all three measures similarity is determined after the computation of *all* the pair-based similarities, $sim(v_i, v_j)$. The difference comes from the way it is computed from them. In sim_{SL} the similarity between two sets is determined by the most similar elements of the two sets. As a result this method is more sensitive to outliers. On the other hand sim_{AL} reduces the effect of outliers by averaging over all element pairs. Sim_K emphasizes the best matches between the elements of the two sets, and penalizes the similarity when there is a large difference in cardinalities. Of course there are many possible refinements of the above similarity measures, some of which may be

worthy of investigation in a more detailed study. For example, in computing the final similarity with sim_{AL} one could remove similarity outliers, i.e. extremely high or low similarities that do not conform with the general similarity distribution. However, this should be done with caution because for the algorithm selection application it might be exactly these cases that determine the relative performance of classification algorithms. In the general case the appropriate measure depends heavily on the semantics of the problem.

Apart from these detailed descriptions of individual variables there are also higher level characteristics that are part of the description of a Dataset object. These characteristics do not make sense for all datasets. For example summary statistics of the properties of continuous variables do not make sense on datasets that contain only discrete variables and vice versa. In order to be able to handle such cases the description of a dataset is divided into two groups of characteristics; the first one contains summary statistics for the discrete variables (Disc. Properties object), while the second summary statistics for the continuous (Cont. Properties object). Each group is treated as a single variable of the dataset object when computing the similarity of datasets. When the computation of the characteristics of one group does not make sense then the corresponding variable takes the value *non-applicable, na*. The problem is known in the CBR community as the *heterogeneity* problem. Aha et al. (2001) dealt with the problem in the automatic construction of tree-structured representations of case libraries in a similar way.

Integrating the notion of *non-applicability* into the definition of similarity, the similarity of values A_i, A_j of variable A, $A \in [A_{min}, A_{max}] \cup na$ becomes:

$$sim(A_i, A_j) = \begin{array}{l} 1 - \frac{|A_i - A_j|}{A_{max} - A_{min}}, \text{ if } A_i, A_j \neq na \\ 0, \text{ if } A_i = na \oplus A_j = na \\ 1, \text{ if } A_i, A_j = na \end{array}$$

The similarity of two datasets A, B is given by:

$$sim(A, B) = \frac{\sum_i (sim(A_i, B_i))}{N}$$

where N is the number of variables that constitute the description of a Dataset object. The variables A_i, B_i are :

- any of the high level characteristics that are part of a Dataset object description,
- the set of variables associated with the datasets A and B,
- and the grouped descriptions of continuous and discrete variables.

4. Meta-learning Framework

One of the goals of the system is to act as an assistant for algorithm selection. The analyst wishes to identify the most suitable classification algorithm for a new dataset based on past learning episodes and the description of the dataset. Given a new classification task, the analyst uses a Data Characterization Tool (Lindner & Studer, 1999) to extract dataset meta-attributes which are stored in a new instance of the Dataset class. This dataset is then posed as a query to the case-based system. Using the similarity measures presented in section 3, the case-base returns the k most similar datasets. For each j of these k datasets there are j_i associated cases; these are instances of the ModProcess class which describe past learning experiments performed on this dataset (i.e. past applications of ModTool instances to the dataset using a specific EvalStrategy. The number of instances j_i of the ModProcess object which are associated with each dataset j of the k most similar datasets depends on the number of the classification algorithms that have been evaluated on the j dataset and the evaluation strategies that have been used; it is not necessary to have complete results for every dataset registered in the case base, so j_i can be different for different j.

For each retrieved dataset the "best" classification algorithm is identified using the error rates recorded in the j_i ModProcess instances associated with it. The algorithm that most frequently obtained the lowest error on the k most similar datasets is recommended by the system for the new dataset. A more systematic approach for defining the "best" algorithm should be based on the notions of pairwise comparisons and significant wins between classification algorithms given in (Kalousis & Theoharis, 1999). We have chosen to work with the more simplistic approach simply because our main goal was the exploration of the representational and predictive power of the relational case-base system and it was more straightforward to use the simplistic scenario.

5. Evaluation of relational case-based classification

The meta-learning problem is a typical classification problem where each instance corresponds to the description of a dataset; the class label that we are trying to predict is simply the algorithm that achieves the lowest error on the specific dataset. We use a simple 0/1 loss function to compute what we will call the *Strict Error*.

Algorithm selection was performed amongst the ten

following: c50boost, c50rules, c50tree, mlcib1 (1-nearest neighbor), mlcnb (naive bayes), ripper (rule inducer (Cohen, 1995)), ltree (multivariate decision tree (Gama & Brazdil, 1999)), lindiscr (linear discriminants), clemMLP (multi-layer perceptron) and clemRBFN (radial basis function network). Table 1 gives the distribution of the class labels; the majority class corresponds to c50boost which is the best performing algorithm in 35.92% of the 103 datasets used in the study. The default error of the meta-learning problem —that of the default learner which simply predicts the majority class—is 64.08%. To be at all useful, the case-based system's algorithm recommendation should exhibit an error rate lower than this default error.

Table 1. Distribution of class labels for the meta-learning problem

Class Label	Frequency	Percentage
c50boost	37	35.92%
c50rules	10	9.70%
c50tree	4	3.88%
clemMLP	12	11.65%
clemRBFN	6	5.82%
lindiscr	12	11.65%
ltree	9	8.73%
mlcib1	10	9.70%
mlcnb	3	2.91%
ripper	0	0.00
Total	103	100%

We used 10-fold stratified cross validation in order to estimate the classification errors of the algorithms. All algorithms were applied using their default parameter setting. Normally this should give rise to 10×103 cases in the case base, but some learning algorithms could not be successfully applied on all the datasets.

In addition based on the results of the evaluation of the algorithms we determined the group of the best algorithms for each one of the 103 datasets using McNemar's test of significance. This consists of learning algorithms whose performance did not vary significantly within the group but was significantly better than that of any algorithm outside the group. The significance level was set to 0.05. Table 2 gives the most frequent groups of significantly better performing algorithms. Each column corresponds to a given group whose members are indicated via $\surd$.

Table 2. Distribution of groups of the significantly better performing algorithms

Algorithms	*Group Membership*					
c50boost	$\surd$	$\surd$				
c50rules		$\surd$				
c50tree		$\surd$				
clemMLP			$\surd$			
clemRBFN						
lindiscr				$\surd$		
ltree					$\surd$	
mlcib1						$\surd$
mlcnb						
ripper						
Freq.	24	4	7	7	7	7
Percent	23.3	3.88	6.8	6.8	6.8	6.8

The establishment of the significantly better group of algorithms will provide the basis of one more evaluation scenario. Here the suggestion of the system will be considered successful if the recommended algorithm belongs to the set of best algorithms for the given dataset. This method reflects the idea that when dealing with a classification task, for which the main goal is low classification error, we will be satisfied if the algorithm that we select belongs to the group of the significantly better algorithms for the given dataset. We will call the error estimated via this method *Loose Error*. In this case the default *Loose Error* is the error that we get when we predict the algorithm that most often appears in the top groups. In the datasets examined here this is c50boost, which in 52.43% of the datasets is part of the group of significantly better algorithms. This default strategy corresponds thus to an error of 100%-52.43%=47.57%.

For evaluation we used leave-one-out cross validation. We experimented with three different values of k, $k = 1, 3, 10$ and the two different similarity measures defined over the set of variables, i.e. sim_{AL}, sim_{SL}. We also report results for sim_K, and for a simple flat attribute-value version of the case-base[2], (AV), where the information on the individual variables of each dataset was simply ignored, (remember though that the high level description of a dataset contains summary statistics of the information on the individual variables mostly in the form of averages). The later is done in order to examine whether the synergy of the relational case-based representation and of the similarity measures can bring an improvement over the flat AV representation. We report the estimated error and the results of a Wilcoxon test on the significance of the difference of the estimated error with the default error.

5.1. Results

In Table 3 we present the estimated *Strict Error* for all the experimental setups. Both methods that use the

[2]This is in essence a simple k-nearest neighbor classification algorithm.

Table 3. Strict Error, (SE), estimation via leave-one-out cross validation. *p-values, (p),* of the Wilcoxon test of significance with the default error, (default = 64.08%). In bold the cases where we have a significant win over the default.

	sim_{SL}		sim_{AL}		sim_K		AV	
k	SE	p	SE	p	SE	p	SE	p
1	64.1	1.00	66.0	0.75	69.9	0.39	67.0	0.63
3	56.3	0.07	**55.3**	**0.03**	66.0	0.63	63.1	0.80
10	61.7	0.36	61.2	0.36	63.1	0.76	64.1	1.00

Table 4. Loose Error, (LE), estimation via leave-one-out cross validation. *p-values, (p),* of the Wilcoxon test of significance with the default error (default = 47.57%).

	sim_{SL}		sim_{AL}		sim_K		AV	
k	LE	p	LE	p	LE	p	LE	p
1	51.4	0.52	52.4	0.43	52.4	0.48	53.4	0.34
3	41.7	0.17	42.7	0.25	49.5	0.63	49.5	0.61
10	44.6	0.31	43.7	0.20	46.6	0.73	48.5	0.79

clustering based similarity measures perform consistently better than the AV version over all the values of k, though not always at a statistically significant level. The only two experimental setups in which a significant improvement over the default error is observed are for $k = 3, sim_{AL}, sim_{SL}$. For sim_{AL} that difference is statistically significant. Sim_K exhibits an error which is worst than the error of AV. One explanation for the low performance of sim_K could be the penalization of similarity for datasets pairs with very different number of attributes, in which case sim_K becomes very small. In the algorithm selection problem it is probable that the most important factor in determining relative performance is the highest similarity observed between any pair of attributes independently of any differences in the number of attributes, a fact that is supported by the good performance of the sim_{SL} which takes into account only the most similar pair of attributes.

The results with respect to the *Loose Error* are fairly similar. Again sim_{AL} and sim_{SL} perform better than AV, which in this case is worse than the default for all the values of k. Nevertheless the difference between sim_{AL}, sim_{SL}, and the default *loose error* is not statistically significant. Like before sim_K and AV have similar levels of performance.

The results provide evidence that the exploitation of the representational power of the relational case base can indeed improve the predictive performance over the flat attribute-value version. Nevertheless this by itself is not sufficient, it should be coupled by the selection of the appropriate similarity measure, otherwise we can even have performance deterioration.

5.2. Recommendations

The recommendations of the system are mainly guided by the nature of the dataset that is given as a query, Table 5 presents some examples of the recommendations of the case-base. For datasets composed of numeric variables, (`typewriter fonts`), the most similar datasets are also composed of numeric variables, the same holds for datasets with discrete (*slanted fonts*) or a mixture of discrete and continuous variables (sans serif fonts). However this intuitively appealing behavior does not guarantee the best performance.

For example for dataset *allrep* the algorithm exhibiting the lowest error in its three most similar datasets is c50boost, but for allrep itself the algorithm with the lowest error is c50tree, a fact that results in an erroneous recommendation. Taking a closer look at the performance of the algorithms for allrep we can see that c50tree, c50rules and c50boost have very similar errors, 0.9%, 0.8% and 0.7% respectively. Moreover the differences between the three algorithms are not statistically significant. All these make the prediction of the algorithm that achieves the lowest error on the specific dataset a very hard task. The problem could have been alleviated by adopting the notion of significant wins as in (Kalousis & Theoharis, 1999).

Another interesting observation is that the case-based system seems to group together datasets that come from similar application domains. For example for dataset *byzantine*, a dataset related to the recognition of byzantine printed notes, almost all its similar datasets come from the pattern recognition domain and even more specifically from character recognition, with the exceptions of vowel (acoustic vowel recognition), *lrs* (low resolution spectrometer dataset) and abalone (predicting the age of gastropod mollusks from some of its physical characteristics). The same holds for the *allrep* and *allbp* datasets where their six most similar datasets are the well known thyroid based datasets. The seventh dataset is the hepatitis dataset another medical diagnosis dataset; only the last three most similar datasets do not involve medical applications. In a sense the relational case-based system seems to group datasets into clusters that correspond to specific application regions.

6. Discussion and Future Work

The use of CBR systems to support the exploration, comparison, categorization and application of tools

Table 5. Examples of recommendations.

QUERY	EIGHT MOST SIMILAR DATASETS	
allrep	allbp	allhyper
C50TREE	C50BOOST	C50BOOST
	sick	dis
	C50BOOST	C50RULES
	allhypo	ann–thyroid
	C50BOOST	C50BOOST
	hepatitis	*tic*
	CLEMMLP	C50RULES
allbp	allrep	sick
C0BOOST	C50TREE	C50BOOST
	allhyper	dis
	C50BOOST	C50RULES
	allhypo	ann–thyroid
	C50BOOST	C50BOOST
	hepatitis	*tic*
	CLEMMLP	C50RULES
byzantine	byzantine32	vowel
MLCIB1	MLCIB1	MLCIB1
	lrs	segmentation
	C50BOOST	C50BOOST
	abalone	pendigits
	CLEMMLP	MLCIB1
	char	letter
	C50BOOST	C50BOOST
dna-splice	*monk1*	*parity5_5*
C50BOOST	C50BOOST	C50BOOST
	splice	*monk3*
	MLCNB	C50BOOST
	tic-tac-toe	*connect–4*
	MLCIB1	C50BOOST
	monk2	*balance-scale*
	CLEMMLP	C50BOOST

from a given domain is not new. Althoff et al. (2000) use a CBR system to build an experience base which does exactly that for Knowledge Managment tools.

One of the major learning paradigms in machine learning is lazy learning. Central to this paradigm is the retrieval, based on some notion of similarity, of past instances and their reuse in order to determine the class of an unseen instance. On the other hand the standard Case Based Reasoning cycle consists of four phases: retrieval, reuse, revision and retention (Aamodt & Plaza, 1994). Past cases are retrieved, using similarity measures, reused and possibly revised in order to solve a new unseen problem, with the final solution being retained as part of a new case. Lazy learning can profit from the experience developed in CBR systems, and more specifically relational CBR, in order to adapt its instance representation, and the retrieval and reuse phases to tackle classification problems that cannot be adequately addressed within the propositional framework. One example of this synergy is the work of Armengol and Plaza (2001), where they used a relational case-based representation and defined similarity measures to perform classification. Their approach is not based on pairwise similarity comparisons, they view similarity as a symbolic description of what the cases present in the case base and the case to be classified have in common. Of direct utility is the work on similarities over complex structures, for example Bergmann and Stahl (1998) discuss the definition of similarities over object oriented structures.

The meta-learning task serves as an opportunity to set forth a number of representational issues that cannot be tackled adequately via propositional learners. The main issue that arises is the ability to handle one to many relationships that appear when one tries to describe properties of datasets which consist of many variables. Relational case based systems offer a natural solution. In the current paper we continue previous work on meta-learning using such a system. We define precise similarity measures between sets exploiting well established results from clustering and present the results of a thorough evaluation of the system. The relational case base allows us to overcome naturally the representational limitations inherent in propositional learning algorithms since it makes use of the relational representations of the training instances.

The evaluation of the system with respect to *Strict Error* showed that it provides suggestions which are statistically significant better than the default strategy for $sim_{AL}, k = 3$. Comparing the relational versions with the attribute-value version showed a consistent advantage of the former, albeit not statistically significant, for the two cluster inspired similarity measures. The sim_K measure had a performance very similar to the attribute-value version, thus providing support for the hypothesis that the two first are more appropriate for the algorithm selection problem. The *Loose Error* results, i.e. how often the recommendation is a part of the truly best set of algorithms for a given dataset, were similar though the differences observed this time were not statistically significant.

Another interesting dimension of the present work are the groups of datasets that the case-base was forming whenever it was presented with a query. These seem to comprise datasets that come from very similar application domains. So we observed groups of datasets from pattern recognition problems, or groups of datasets from medical diagnosis problems. An interesting further research direction would be the application of clustering algorithms making use of the multi-relational representation capabilities of the case-base.

The proposed relational case-based representation of datasets can be used together with any of the various

meta-learning frameworks for algorithm selection, e.g. algorithms ranking, direct error prediction etc.

Overall the performance results are encouraging but there is still space for significant improvement. We plan to focus our efforts on the representational issues that are set forth here by providing more elaborate representation schemas for the description of various properties of the set of attributes of a dataset, but also more general of sets of objects. More precisely we want to use histogram representations, as they were introduced in (Kalousis & Theoharis, 1999), to describe the distributions of properties of sets of objects in a compact form and use them to extend the classical propositional learning schema. This will give rise to a new representational schema that will lie between the propositional and the multi-relational paradigms.

Acknowledgements

The authors would like to thank all reviewers for their most helpful comments, that considerably improved the paper.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, *7*, 39–59.

Aha, D., Breslow, L., & Munoz-Avila, H. (2001). Conversational case-based reasoning. *Applied Intelligence*, *14*, 9–32.

Althoff, K.-D., Muller, W., Nick, M., & Snoek, B. (2000). KM-PEB: An online experience base on knowledge management technology. *Advances in Case-Based Reasoning, 5th European Workshop* (pp. 335–347). Springer.

Armengol, E., & Plaza, E. (2001). Lazy induction of descriptions for relational case-based learning. *Proceedings of the 12th European Conference on Machine Learning* (pp. 13–24). Springer.

Bergmann, R., & Stahl, A. (1998). Similarity measures for object-oriented case representations. *Advances in Case-Based Reasoning, 4th European Workshop* (pp. 25–36). Springer.

Brazdil, P., Carlos, S., & Costa, J. (2003). Ranking learning algorithms. *Machine Learning*.

Cohen, W. (1995). Fast effective rule induction. *Proceedings of the 12th International Conference on Machine Learning* (pp. 115–123). Morgan Kaufman.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification and scene analysis*, chapter Unsupervised Learning and Clustering. John Willey and Sons.

Gama, J., & Brazdil, P. (1999). Linear tree. *Intelligent Data Analysis*, *3*, 1–22.

Hilario, M., & Kalousis, A. (2001). Fusion of meta-knowledge and meta-data for case-based model selection. *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer.

Kalousis, A., & Theoharis, T. (1999). Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis*, *3*, 319–337.

Kirsten, M., Wrobel, S., & Horvath, T. (2001). *Relational data mining*, chapter Distance Based Approaches to Relational Learning and Clustering, 212–232. Springer.

Lindner, C., & Studer, R. (1999). AST: Support for algorithm selection with a CBR approach. *Proceedings of the 16th International Conference on Machine Learning, Workshop on Recent Advances in Meta-Learning and Future Work*.

Michie, D., Spiegelhalter, D., & Taylor, C. (1994). *Machine learning, neural and statistical classification*. Ellis Horwood Series in Artificial Intelligence.

Peng, Y., Flach, P., Soares, C., & Brazdil, P. (2002). Improved dataset characterisation for meta-learning. *Proceedings of the 5th International Conference on Discover Science 2002*. Springer-Verlag.

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Tell me who can learn you and I can tell you who you are: Landmarking various learning algorithms. *Proceedings of the 17th International Conference on Machine Learning* (pp. 743–750). Morgan Kaufman.

Rendell, L., Seshu, R., & Tcheng, D. (1987). Layered concept learning and dynamically variable bias management. *Proceedings of the 10th International Joint Conference on AI* (pp. 366–372). Morgan Kaufman.

Todorovski, L., & Dzeroski, S. (1999). Experiments in meta-level learning with ILP. *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 98–106). Springer.