

Combination of Task Description Strategies and Case Base Properties for Meta-Learning

Christian Köpf and Ioannis Iglezakis

DaimlerChrysler AG, Research & Technology, RIC/AM, P.O.-Box 2360,
D-89013 Ulm, Germany,
{christian.koepf,ioannis.iglezakis}@daimlerchrysler.com

Abstract. Describing a learning task is crucial, not only for meta-learning but also to gain insight in this learning task. The paper evaluates the performance of a recent method for assessing quality standards for case bases when used for a supervised meta-learning. Empirical results on real-world data show this approach in combination with others as a promising one.

1 Introduction

The problem of selecting an appropriate model for a given learning task is a crucial one. Often, there is neither enough time nor space to select learning algorithms from a given pool by simply trying them out. Thus, as users, we want to relate to past experiences of and with learners in the pool to predict which one is most suitable for a given task. This might be in terms of measures such as predictive accuracy, time, or comprehensibility. In this work, we limit ourselves to predictive accuracy.

How can we relate to a past experience with a learner, how can we describe a new learning task adequately? We will concentrate on two already known strategies for task description here, namely landmarking ([15], [2]) and data characterization (which we will refer to as DCT due to the name of a software used to compute the characteristics) ([7], [12], [11]). Additionally, we will experiment with a new approach which has recently been in use in the field of case base reasoning to assess the quality of case bases. This approach will be used for meta-learning and will also be combined with other already used measures.

We begin by introducing the already mentioned strategies for task description, landmarking and DCT. Afterwards, we will dwell on the case properties extracted from case bases to evaluate their quality by possible conflicts between items within a case base. This is followed by empirical results with real-world data. Eventually, the last section summarizes the paper and points at future work.

2 Task Description Strategies

Probably the most common way is to use data characteristics to describe a given learning task for either classification or prediction. At first, basic information is

computed such as number of classes, number of attributes, both symbolic and numeric, number of observations and number of missing values. These measurements are supposed to give a first estimation of the learning problem.

They are extended by statistical measures which are supposed to inform about the distribution of the numeric attributes. For instance, several measures might be computed to check if the given learning task meets the assumption of a discriminant analysis. Measures such as eigenvalues and discriminant functions are computed from the data where the relative proportion of the first discriminant function is given by

$$Fract1 = \frac{\lambda_{max}}{d} \quad (1)$$

where λ_{max} denotes the largest eigenvalue. In addition, the canonical correlation of the best linear combination of attributes is given by [10]

$$CanCor1 = \sqrt{\frac{\lambda_{max}}{1 + \lambda_{max}}}. \quad (2)$$

Eventually, information theoretic measurements are computed to test how much the symbolic attributes contribute to correctly classifying the labeled objects. The entropy of an attribute A as a realization of a discrete random variable \mathbf{X} with k characteristics is given by [19]

$$H_A = H(A) = - \sum_{i=1}^k p_i \log_2(p_i) \quad (3)$$

where p_i ($1 \leq i \leq k$) is the probability for A taking the i^{th} value ($\sum_{i=1}^k p_i = 1$). The entropy of a symbolic target variable with q characteristics, referred to as the class entropy, is given by

$$H_C = H(C) = - \sum_{i=1}^q p_i \log_2(p_i). \quad (4)$$

If p_{ij} denotes the joint probability of observing class C_i and the j^{th} value of attribute A , the joint entropy defined as

$$H_{CA} = H(C, A) = - \sum_{i,j} p_{ij} \log_2(p_{ij}) \quad (5)$$

is a measure of the total entropy of the joint system of these attributes, i.e. of all combinations (C, A) . Then, the information gain or mutual information is given by [16][17]

$$I_{gain}(C, A) = H_C + H_A - H_{CA}. \quad (6)$$

These measurements have been explored for a meta-learning approach within the StatLog project (1991–1994) and a detailed description can be found in [14].

A completely different approach was chosen by [15] and [2]. Instead of using measurements for describing a given learning problem statistically which is not

directly related to the performance of algorithms, fast learning algorithms are used to describe the problem adequately. There is little point in measuring data characteristics to predict the performance of a classifier if this measurement process takes longer to execute than running the classifier(s) in question. Rooted in the StatLog project under the term "yard stick methods", this approach is known as landmarking. The above mentioned authors proposed the use and motivation of the following landmarkers.

1. *Decision Node*: Using *C5.0*'s information gain-ratio [17] a single decision node is chosen which is then to be used for classifying test observations. The goal of this landmark learner is to establish closeness to linear separability.
2. *Randomly Chosen Node*: An attribute is chosen randomly and then used for splitting the training set and classifying new observations. The goal of this landmark learner is to inform about irrelevant attributes.
3. *Worst Node*: By using the information gain ratio again, the least informative attribute is used to make the single split. Together with the first landmark learner, this landmarker is supposed to inform about linear separability.
4. *Naïve Bayes*: The necessary probabilities for using Bayes' theorem [6] are computed on the training set in order to classify observations in the test set. The goal of this landmark learner is in measuring the extent to which attributes are conditionally independent given the class.
5. *1-Nearest Neighbor*: According to the closest observation in the training set, a new observation in the test set is classified [6]. The goal of this landmark learner is in determining how close instances belonging to the same class are.
6. *Elite 1-Nearest Neighbor*: This landmarker works like the previous one, although it is computed on a subset of all attributes. This subset is determined by the most informative attributes.¹
7. *Linear Discriminant*: Using the training set, a linear target function is computed which is then used to classify observations from the test set [10].

Landmarking proved to be a competitive method for task description since the results of the landmarkers are directly related to more "sophisticated" algorithms instead of the indirect data characteristics. However, we might also want to consider the meaning and interpretation of possible outcomes. At the end of a meta-learning experiment, we might like to discover some useful insights into when algorithms perform well. Thus, data characteristics are still of importance for meta-learning.

Finally, a meta-data set comprises a number of meta-observations each of which represents an actual data set. The above described data characteristics (basic, statistical, and information theoretic measures) and landmarkers are used as meta-attributes trying to adequately describe the original data sets with the final aim of model selection. This can be either done by a classification approach trying to predict the algorithm out of a given learners pool that will yield the

¹ Here, only attributes are taken into account for which the information gain ratio was smaller than 1. This threshold is due to results obtained by [1]. This algorithm is part of a set of algorithms called *Edited 1-Nearest Neighbor*.

lowest misclassification rate or by a regression approach where the error rate of each learner from the pool is to be predicted [3] [11]. In the latter case, it is up to the user's experience which of the learners is eventually chosen. In the section following the next one, we will use different sets of meta-attributes for meta-learning. Their formation will then be explained in more detail.

3 Case Base Properties

One major drawback of the data characterization scenario is that information theoretic and statistical measures take either numeric or symbolic attributes into account. Often though, the measures describing the basic properties of a learning task, say, contain nearly equally as much information. A possible way of taking information contained in all attributes into account is to compare observations with each other. This might be helpful in various ways. A data set may contain two observations with similar or equal attribute values, but with different labels which might cause a classifier to get "confused". Analogously, there might be two or more observations which are identical. In such a case, the observation might be given more weight, however, the information contained in it might be redundant for the classifier. Also, in this very case, attribute values might be missing, so that one observation would actually be a subset of another observation. Such an approach is described in detail in [9]. There, case base properties are used to assess the quality of given case bases in terms of measures such as redundancy or incoherency. Following and using the notation given in [9], we will briefly introduce some necessary requirements for the implementation of the case based properties which is followed by an example demonstrating the approach. To begin with, we have to settle on the notation. For a more thorough description, however, see [9] and [18].

Definition 1 (Cases and Case Base).

1. An attribute a_j is a name accompanied by a set $V_j := \{v_{j1}, \dots, v_{jk}, \dots, v_{jN_j}\}$ of values. We denote the set of attributes as $A := \{a_1, \dots, a_j, \dots, a_N\}$.
2. A problem is a set $p_i := \{p_{i1}, \dots, p_{ij'}, \dots, p_{iN_i}\}$ with $\forall j' \in [1; N_i] \exists a_j \in A$ and $\exists v_{jk} \in V_j : p_{ij'} = v_{jk}$, and $\forall j \in [1; N] : |(p_i \cap V_j)| \leq 1$. We denote the set of problems as $P := \{p_1, \dots, p_i, \dots, p_M\}$.
3. A solution s_i is any item.
4. A case is a tuple $c_i := (p_i, s_i)$ with a problem p_i and a solution s_i . A case base is a set of cases $C := \{c_1, \dots, c_i, \dots, c_M\}$.
5. We further assume a separation of C into a training set T and a test set (or query set) Q with $C = T \cup Q$ and $T \cap Q = \emptyset$.

Additionally, we have to define functions to be able to determine the similarity between two given cases, that is to say two observations.

Definition 2 (Auxiliary Functions). Assume a local similarity measure $sim_j : V_j \times V_j \mapsto [0; 1]$.

1. $S_{\leftrightarrow} : P \times P \mapsto \{1..N\}$,
 $S_{\leftrightarrow}(p_i, p_{i'}) := |\{j \in \{1..N\} : |p_i \cap V_j| = |p_{i'} \cap V_j| = 1 \wedge sim_j(p_{ij}, p_{i'j}) = 1\}|$
2. $S_{\rightsquigarrow} : P \times P \mapsto \{1..N\}$,
 $S_{\rightsquigarrow}(p_i, p_{i'}) := |\{j \in \{1..N\} : |p_i \cap V_j| = |p_{i'} \cap V_j| = 1 \wedge sim_j(p_{ij}, p_{i'j}) \neq 1\}|$
3. $S_{\leftarrow} : P \times P \mapsto \{1..N\}$,
 $S_{\leftarrow}(p_i, p_{i'}) := |\{j \in \{1..N\} : |p_i \cap V_j| > |p_{i'} \cap V_j|\}|$
4. $S_{\rightarrow} : P \times P \mapsto \{1..N\}$,
 $S_{\rightarrow}(p_i, p_{i'}) := |\{j \in \{1..N\} : |p_i \cap V_j| < |p_{i'} \cap V_j|\}|$

The overall similarity in the following definition is the normalized weighted sum of the above introduced and computed auxiliary values. Values coinciding for the same attribute as positive are considered. Different values, however, do not contribute positive to local similarity values. Note also that for all other values ($S_{\leftarrow}(p_i, p_{i'})$, $S_{\rightarrow}(p_i, p_{i'})$, and $S_{\rightarrow}(p_i, p_{i'})$), weights w_{\leftarrow} , w_{\rightarrow} , and w_{\rightarrow} decide whether we consider their relations as positive ($w = 1$) or negative ($w = 0$).

Definition 3 (Similarity Measure). Assume $w_{\leftarrow}, w_{\rightarrow}, w_{\rightarrow} \in \{0, 1\}$.

$$sim : P \times P \mapsto [0; 1],$$

$$sim(p_i, p_{i'}) := N^{-1} \cdot \left(S_{\leftrightarrow}(p_i, p_{i'}) + w_{\leftarrow} \cdot S_{\leftarrow}(p_i, p_{i'}) \right. \\ \left. + w_{\rightarrow} \cdot S_{\rightarrow}(p_i, p_{i'}) + w_{\rightarrow} \cdot S_{\rightarrow}(p_i, p_{i'}) \right).$$

Eventually, the case base properties are defined as follows.

Definition 4. Assume $G \subseteq C$, $c_i \in G$, and $1 \leq \Delta \in \mathbb{N}$.

1. c_i consistent within $G : \iff \nexists c_{i'} \in G : s_i \neq s_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) + S_{\leftarrow}(p_i, p_{i'}) = N_i \geq N_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) > 0 \wedge S_{\leftarrow}(p_i, p_{i'}) \geq 0 \wedge S_{\rightarrow}(p_i, p_{i'}) = 0$.
2. c_i unique within $G : \iff \nexists c_{i'} \in G, c_{i'} \neq c_i : s_i = s_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) = N_i = N_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) > 0$.
3. c_i minimal within $G : \iff \nexists c_{i'} \in G : s_i = s_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) + S_{\leftarrow}(p_i, p_{i'}) = N_i > N_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) > 0 \wedge S_{\leftarrow}(p_i, p_{i'}) > 0 \wedge S_{\rightarrow}(p_i, p_{i'}) = 0$.
4. c_i incoherent $_{\Delta}$ within $G : \iff \nexists c_{i'} \in G : s_i = s_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) + S_{\rightsquigarrow}(p_i, p_{i'}) + S_{\leftarrow}(p_i, p_{i'}) = N_i = N_{i'} \wedge S_{\leftrightarrow}(p_i, p_{i'}) > 0 \wedge S_{\rightsquigarrow}(p_i, p_{i'}) \geq 0 \wedge S_{\leftarrow}(p_i, p_{i'}) \geq 0 \wedge S_{\rightarrow}(p_i, p_{i'}) \geq 0 \wedge S_{\leftarrow}(p_i, p_{i'}) = S_{\rightarrow}(p_i, p_{i'}) \wedge S_{\rightsquigarrow}(p_i, p_{i'}) + S_{\leftarrow}(p_i, p_{i'}) = \Delta$.

To illustrate the given definitions, the examples in Table 1 will be helpful. Pairs of cases, their conflict to each other and the resulting values for the auxiliary functions in Definition 2 are shown. Note that the symbol \neg denotes the negation of a proposition. Note as well that by using these case base properties, suspicious observations which might impair the results of learning algorithms can be removed which was the original intention behind this approach. This, however, is more of a preprocessing task which is beyond the scope of our work. Instead, we use the computed measurements as meta-attributes to add more information to the meta-learners.

	p_i	s_i	$p_{i'}$	$s_{i'}$	Proposition	S_{\leftrightarrow}	S_{\rightsquigarrow}	S_{\leftarrow}	S_{\rightarrow}	S_{-}	Δ						
1	v_{11}	v_{21}	v_{31}	s_1	v_{11}	v_{21}	s_2	\neg consistent	2	0	1	0	2	-			
2	v_{11}	v_{21}	v_{31}	s_1	v_{11}	v_{21}	v_{31}	s_1	\neg unique	3	0	0	0	2	-		
3	v_{11}	v_{21}	v_{31}	s_1	v_{11}	v_{21}	s_1	\neg minimal	2	0	1	0	2	-			
4	v_{11}	v_{21}	v_{31}	v_{41}	s_1	v_{11}	v_{21}	v_{42}	v_{51}	s_1	\neg incoherent ₂	2	1	1	1	0	2

Table 1. Examples for Pairs of Cases and the corresponding propositions with respect to general case properties

4 Results

A meta-data set was constructed using 78 data sets from the UCI repository [4]. The number of observations did not exceed 1066, and the number of attributes ranged from 4 to 69. 32 data sets contained only symbolic attributes, 20 data sets contained only numeric attributes. The remaining sets were mixed. The data contained up to 25% missing values. Error rates for ten different classification algorithms from the Metal project [13] were determined for different subsets of data characteristics by a ten-fold cross validation, viz. c50boost, tree, and rules [17], the neural networks clemMLP, clemRBFN, both implemented in Clementine, the discriminant tree learner Ltree [8], the rule learner RIPPER [5], a linear discriminant learner, a naive Bayes learner and an instance-based learner. In all cases, the default settings were used.

To begin with, we tried to evaluate various ways on how to represent the data adequately. By adequately, we mean a representation that would give an error rate as small as possible for each algorithm. As a basic set of data characteristics to be used for meta-learning, denoted by DCT_b , we computed the number of attributes, both symbolic and numeric, the number of observations and the number of classes. Additionally, this basic set was either amended by the accuracy and standard deviation of the default class, denoted by DCT_{bd} , or the number of missing values and tuples containing missing values, denoted by DCT_{bm} . Consequently, DCT_{bmd} represents the combination of all measurements. Additionally, an often proposed strategy is to use seven features given by the proportion of both symbolic attributes, attributes with outliers and missing values, the number of observations, the class entropy and mutual information as well as CanCor1, denoted by DCT_{com} . Note that we restricted ourselves here to three base learners, namely Ltree, Naive Bayes, and c50rules. In case, the learners performed equally, the meta-observation was labelled as "TIE". Our goal was to predict the algorithm with the lowest error rate. The corresponding error rates for a ten-fold cross-validation are given in Table 2. Obviously, DCT_{bd} performs best, being significantly better than most other approaches. The information contained within the missing values contributes poorly to predicting the correct class labels whereas the default accuracy seems much more appropriate.

As previously mentioned, we followed various ways to describe a given learning task. First, we computed data characteristics for a given data set. This was

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bdm}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	43.42	25.00	44.73	34.21	35.52
c50tree	50.00	43.42	53.94	44.73	40.79
IB	42.11	36.84	43.42	36.84	46.05
Ripper	52.63	52.63	59.21	51.32	39.47
Average	47.04	39.47	50.32	41.78	40.46

Table 2. Percentage error rates for DCT strategies and different meta-learners

followed by computing error rates using the landmarking algorithms as meta-attributes. Ext-Land is based on the seven landmarkers given in [2] whereas Landmarking goes without those learners being both in the learners and landmarkers pool, viz. LinDiscr, NB, and IB. Eventually, we computed the case base properties for each of the data sets for different values of ε , $\varepsilon = 0.01, 0.05, 0.1$, which indicates the possible distance between observations. As can be seen from table 3, both Landmarking and Ext-Land perform on average significantly better than the approach using case base properties, in particular than $CBR_{0.1}$.

Meta-learner	Landmarking	Ext-Land	$CBP_{0.01}$	$CBP_{0.05}$	$CBP_{0.1}$
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	56.57	52.68	69.74	60.52	63.16
c50tree	56.58	53.94	59.21	67.11	71.05
IB	55.26	47.36	50.00	57.89	64.47
Ripper	57.89	52.63	59.00	69.73	67.11
Average	56.25	52.30	59.48	63.81	66.45

Table 3. Error rates for different meta-learners and task description strategies

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bmd}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	38.15	36.82	46.05	42.11	34.12
c50tree	47.36	44.73	51.31	47.37	40.29
IB	47.36	39.47	51.31	44.73	44.39
Ripper	57.89	55.26	55.26	48.68	42.32
Average	47.69	44.07	50.98	45.72	40.28

Table 4. Error rates for different meta-learners combining case base properties with $\varepsilon = 0.01$ and various DCT strategies

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bmd}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	40.78	35.52	42.11	34.21	36.21
c50tree	38.15	39.47	40.79	40.79	42.11
IB	51.31	43.32	55.26	47.36	42.11
Ripper	61.84	56.57	53.94	47.36	44.32
Average	48.02	43.72	48.03	42.43	41.19

Table 5. Error rates for different meta-learners combining case base properties with $\varepsilon = 0.05$ and various DCT strategies

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bmd}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	48.68	39.47	43.42	38.15	37.24
c50tree	40.79	43.42	44.73	43.42	43.16
IB	57.89	46.05	59.21	48.68	44.73
Ripper	56.57	59.21	57.89	55.26	43.42
Average	50.98	47.03	51.31	46.38	42.13

Table 6. Error rates for different meta-learners combining case base properties with $\varepsilon = 0.1$ and various DCT strategies

Tables 4 through 6 show error rates of meta-learners on combined measures from DCT and the case base approach. Although results on the average deteriorate, they are still quite similar when compared to table 2. This seems in particular interesting, since we added a total of ten variables from the case base approach to the different DCT approaches and the meta-data set consists only of 78 observations. It is our believe that by choosing the right mixture of DCT and case base measures, we might improve meta-learning, although maybe not significantly. Encouraged by our results, we tried to evaluate them using all learners as base and as meta-learners. The results for DCT strategies are given in table 7. On average, all methods perform better than the default. The missing values could not be computed. Table 8 shows the results for the landmarking and case-based reasoning approaches. Again, methods perform better than the default on average, though sometimes close to it. Tables 9 through 11 show various combinations of DCT strategies and case base measures. Again, the results of the learners on average are not much different from the case when using only DCT. This is particularly true for $\varepsilon = 0.1$.

5 Conclusions and Future Work

We have presented a new approach for task description as a means of model selection in meta-learning. Tasks are described by their similarity, consistency, incoherency, uniqueness and minimality. While this method does not outperform any of the existing approaches on its own, combinations of methods seem very

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bdm}	DCT_{com}
Default Class	77.63	77.63	77.63	77.63	77.63
C5.0boost	67.11	61.84	64.47	63.16	63.16
C5.0rules	65.79	67.11	61.84	64.47	64.48
C5.0tree	67.11	69.74	64.48	65.79	64.48
ClemMLP	77.61	73.68	80.26	77.63	77.63
ClemRBFN	68.08	?	62.91	?	78.19
LinDiscr	68.42	76.32	75.00	78.95	78.95
Ltree	67.11	68.42	67.11	67.11	72.37
IB	64.47	68.42	65.79	69.74	68.42
NB	73.68	69.74	77.63	76.32	86.84
Ripper	64.47	69.74	69.73	68.43	68.42
Average	68.38	69.44	68.92	70.18	72.29

Table 7. Percentage error rates for DCT strategies using all learners as base and meta-learners

Meta-learner	Landmarking	Ext-Land	$CBP_{0.01}$	$CBP_{0.05}$	$CBP_{0.1}$
Default Class	77.63	77.63	77.63	77.63	77.63
C5.0boost	61.84	78.95	68.42	75.00	71.05
C5.0rules	65.79	78.95	68.42	75.00	71.05
C5.0trees	64.47	77.63	69.73	73.68	71.05
MLP	80.26	77.63	80.26	80.26	78.94
RBFN	70.58	80.26	?	84.21	75.00
LinDiscr	84.21	75.00	72.37	73.68	72.37
Ltree	64.47	76.31	73.68	77.68	71.05
IB	64.47	68.42	71.05	80.26	77.63
NB	73.68	77.63	80.26	71.05	76.31
Ripper	71.05	75.00	76.31	78.95	73.68
Average	70.08	76.58	73.39	76.97	73.82

Table 8. Error rates for different meta-learners and task description strategies using all learners as base and meta-learners

Meta-learner	DCT_b	DCT_{bd}	DCT_{bmd}
Default Class	77.63	77.63	77.63
C5.0boost	61.84	61.84	65.79
C5.0rules	68.42	67.11	68.42
C5.0trees	68.42	67.11	67.11
MLP	80.26	81.58	78.95
RBFN	81.58	81.58	78.95
LinDiscr	72.37	73.68	69.74
Ltree	71.05	71.05	69.74
IB	68.42	71.05	75.00
NB	73.68	73.68	75.00
Ripper	68.42	71.05	72.37
Average	71.44	71.97	72.11

Table 9. Case base properties using $\varepsilon = 0.01$ and various DCT strategies using all learners as base and meta-learners

Meta-learner	DCT_b	DCT_{bd}	DCT_{bmd}
Default Class	77.63	77.63	77.63
C5.0boost	60.52	61.84	60.53
C5.0rules	63.16	64.47	65.79
C5.0trees	67.11	67.11	67.11
MLP	80.26	77.63	81.58
RBFN	76.31	76.31	76.32
LinDiscr	73.68	75.00	73.61
Ltree	68.42	69.73	69.74
IB	67.11	68.42	73.68
NB	68.42	67.11	67.11
Ripper	73.68	71.05	73.68
Average	69.87	69.87	70.92

Table 10. Case base properties using $\varepsilon = 0.05$ and various DCT strategies using all learners as base and meta-learners

Meta-learner	DCT_b	DCT_{bd}	DCT_{bmd}
Default Class	77.63	77.63	77.63
C5.0boost	59.21	63.16	63.16
C5.0rules	63.16	63.16	63.16
C5.0trees	63.16	64.47	63.16
MLP	78.95	78.95	77.63
RBFN	85.52	82.83	84.21
LinDiscr	73.68	77.63	77.63
Ltree	60.53	61.84	61.82
IB	69.73	69.73	69.73
NB	71.05	69.73	67.11
Ripper	71.05	69.73	76.31
Average	69.61	70.13	70.39

Table 11. Case base properties using $\varepsilon = 0.1$ and various DCT strategies using all learners as base and meta-learners

promising, in particular for real-world data. Using case base properties might also help in understanding why methods perform differently. However, this serves as an outlook for future work. We also intend to use larger data sets for creating our meta-data set and to eventually use a larger meta-data set itself. Additionally, we want to evaluate useful combinations including landmarks as well as testing which distance measure is most appropriate for meta-learning. One problem to overcome is the computational complexity of the case base properties. Since the complexity is quadratic, we think about drawing samples of smaller sizes, as the size of data sets increases. In general, this seems to be an interesting field of research.

Acknowledgements

The authors would like to thank Thomas Reinartz from DaimlerChrysler and the members of the METAL consortium for fruitful discussions. The research was supported financially by EC METAL project (ESPRIT # 26.357) and DaimlerChrysler.

References

1. Hilan Bensusan. *Automatic Bias Learning: An Inquiry Into The Inductive Basis of Induction*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, UK, 1999.
2. Hilan Bensusan and Christophe Giraud-Carrier. Casa Batló is in Passeign de Gràcia or landmarking the expertise space. In *Proceedings of the Meta-Learning Workshop at the ECML-2000*, 2000.

3. Hilan Bensusan and Alexandros Kalousis. Estimating the predictive accuracy of a classifier. In Luc De Raedt and Peter Flach, editors, *Proceedings of the Twelfth European Conference on Machine Learning ECML-2001*. Springer, New York, NY, 2001.
4. C.L. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, 1998. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
5. William W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufman, San Mateo, CA, 1995.
6. Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.
7. Robert Engels and Christiane Theusinger. Using a data metric for offering pre-processing advice in data mining applications. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence*, pages 430–434, 1998.
8. João Gama. Discriminant trees. In *Proceedings of the Sixteenth International Conference on Machine Learning – ICML’99*, 1999.
9. Ioannis Iglezakis and Thomas Reinartz. Relation between customer requirements, performance measures, and general case properties for case base maintenance. In *Proceedings of the Sixth European Workshop on Case-Base Maintenance*, 2002.
10. William R. Klecka. *Discriminant Analysis*. Sage Publications, Newbury Park, London, UK, 1980.
11. Christian R. Köpf, Charles C. Taylor, and Jörg Keller. Meta-learning: From data characterisation for meta-learning to meta-regression. In *Proceedings of the Workshop on "Data Mining, Decision Support, Meta-Learning and ILP: Forum for Practical Problems" at the PKDD-2000*, 2000.
12. Guido Lindner and Rudi Studer. AST: Support for algorithm selection with a cBR approach. In *Proceedings of the Third International Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 418–423, 1999.
13. MetaL. EC ESPRIT MetaL Project #26.357. <http://www.cs.bris.ac.uk/cgc/METAL>, 1998–2001.
14. Donald Michie, David J. Spiegelhalter, and Charles C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, NY, 1994.
15. Bernhard Pfahringer, Hilan Bensusan, and Christophe Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.
16. J. Ross Quinlan. Induction of decision trees. In *Proceedings of the First International Conference on Machine Learning*, pages 81–106. Morgan Kaufman, San Mateo, CA, 1986.
17. J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA, 1992.
18. Thomas Reinartz, Ioannis Iglezakis, and Thomas Roth-Berghofer. On quality measures for case base maintenance. In *Proceedings of the Fifth European Workshop on Case-Base Maintenance*, pages 247–259. Springer, New York, NY, 2001.
19. Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana and Chicago, IL, 1963.