Machine Learning, 54, 275–312, 2004

On Data and Algorithms: Understanding Inductive Performance

ALEXANDROS KALOUSIS kalousis@cui.unige.ch
University of Geneva, Computer Science Department, 24, rue du General Dufour, CH-1211 Geneva 4, Switzerland

JOÃO GAMA jgama@liacc.up.pt
LIACC, FEP—University of Porto, Rua Campo Alegre 823, 4150 Porto, Portugal

MELANIE HILARIO hilario@cui.unige.ch
University of Geneva, Computer Science Department, 24, rue du General Dufour, CH-1211 Geneva 4, Switzerland

Editors: Christophe Giraud-Carrier, Ricardo Vilalta and Pavel Brazdil

Abstract. In this paper we address two symmetrical issues, the discovery of similarities among classification algorithms, and among datasets. Both on the basis of error measures, which we use to define the *error correlation* between two algorithms, and determine the *relative performance* of a list of algorithms. We use the first to discover similarities between learners, and both of them to discover similarities between datasets. The latter sketch maps on the dataset space. Regions within each map exhibit specific patterns of error correlation or relative performance. To acquire an understanding of the factors determining these regions we describe them using simple characteristics of the datasets. Descriptions of each region are given in terms of the distributions of dataset characteristics within it.

Keywords: classification, meta-learning, error correlation, classifier ranking, clustering datasets, clustering classifiers

1. Introduction

The task of classification in Machine Learning is profoundly characterized by a dual relation, that between algorithms and datasets. This duality is established on the performance results of the former when applied to the latter. One can use the performance results of algorithms to characterize a dataset, but also in the opposite direction to characterize algorithms according to their performance on specific datasets. One of the most challenging problems in Machine Learning, is the determination of the properties of a dataset that make one classification algorithm more appropriate than another; an effort to describe the performance link between algorithms and datasets via the use of the dataset characteristics. This *selective superiority problem* has been identified and studied both empirically (Aha, 1992; Brodley, 1994; Michie, Spiegelhalter, & Taylor, 1994; Gama & Brazdil, 1995; Kalousis & Theoharis, 1999; Soares & Brazdil, 2000; Todorovski, Blockeel, & Dzeroski, 2002) and analytically (Langley & Sage, 1999; Scheffer, 2000).

In this paper we make a stop one step ahead, before trying to tackle that issue. We seek relations between algorithms by means of their performance on the datasets, and in

the opposite direction relations between datasets based on performance. In both cases we will be working with the same data, but in the first trying to discover similarities between algorithms and in the second between datasets. The results of this two-way analysis will define regions on the dataset space, which we will then try to chart using simple descriptions of the datasets.

Overall we propose a methodology for tackling metalearning problems, and try to exhibit that it results in understandable and interesting insights. Our goal is to demonstrate the validity of the proposed approach on the basis of the produced knowledge and its agreement with common knowledge in the machine learning domain. If that proves to be the case the method could serve as a tool used by machine learning researchers in order to further explore the relations of datasets and algorithms leading hopefully to really new insights.

The rest of the paper is organized as follows: Section 2 gives a brief presentation of the most relevant related work on meta-learning issues. In Section 3 we present the different approaches to the characterization of the relative performance of learning algorithms that we will be using throughout the paper. In Section 4 we will be using the performance characterizations in order to discover relations and similarities between learning algorithms while in Section 5 the same performance measures will be used to discover similarities this time between datasets. On the basis of the results of Sections 4 and 5 a number of meta-learning issues will be defined, Section 6, and further explored using a descriptive approach relying on a simple but carefully chosen set of dataset characteristics. In an effort to quantify the validity of the observations of Section 6 we will treat the same meta-learning problems under a predictive scenario, Section 7, using the same set of dataset characteristics. In Section 8 we try to combine the insights gained from Section 6 together with the meta-learning approach of Section 7 into a single coherent view. In the last section, Section 9, we summarize the conclusions of this work and provide a list of open issues for further exploration.

2. Related work

A lot of work, empirical and analytical, exists on the study of the performance of learning algorithms. The PAC related work gives general worst-case bounds on the errors of specific hypothesis classes in relation with the sample size, the error observed on the sample, and the capacity of the family of models under consideration. In the Vapnik-Chernovenkis theory the capacity of a family of functions is expressed in terms of their VC dimension. The VC-based error bounds are used to perform model selection from a series of nested model classes resulting in a model selection strategy known as Structural Risk Minimization. In both PAC and VC theories the error bounds are distribution free i.e. they hold no matter what the distribution generating the examples is. However the characteristics of the data are taken into account via the number of examples of the sample and the empirical error computed on the sample.

Support Vector Machines which constitute a specific application of the structural risk minimization principle exploit the fact that some data distributions might be more benign for learning, i.e. they are not worst case distributions. They exploit the margin distribution, a data dependent quantity, measured on the sample set which provides a measure of the synergy between the data distribution and the family of linear models used by SVMs, to

bound the generalization error and then choose the model with the lowest bound according to the structural risk minimization principle. In this case the bound on the generalization error can be seen as a joint property of the family of models and the distribution that generated the data, as this is depicted in the data sample. A more than complete presentation of the SVMs field along with issues of generalization theory can be found in (Cristianini & Shawe-Taylor, 2002).

Average-case analysis focuses on given algorithms and specific learning problems whose generating distribution is known by construction and derives analytical formulas predicting performance of algorithms given characteristics of the problems, like number of irrelevant and relevant attributes, sample size, etc, (Langley & Sage, 1999; Langley & Iba, 1993; Hirschberg, Pazzani, & Ali, 1994). It is thus able to make much more precise predictions for the behavior of a learning algorithm on a given data sample than worst case analysis; unfortunately in real world problems these characteristics are not known a priori. Scheffer (2000) provides an average-case analysis for families of models with finite capacity which is based on the histogram of the error estimates of models of a given family on a given learning problem, thus avoiding the problem of unknown properties of the data distribution.

In an effort to overcome the difficulties of average-case analysis (Aha, 1992) proposed a methodology for generalizing results of specific case studies. The goal was to derive rules that describe the conditions under which specific performance differences between learning algorithms are observed. The rules describe *when* differences occur rather than *why* but they help focus subsequent analyses on explaining the whys. One of the limitations of the approach is that it demands the construction of artificial datasets which are similar to the one on which the case study was performed, in order to examine how the learning algorithms perform in the neighborhood of the initial dataset. The quality of the results therefore depends on the existence of a good characterization of the original dataset. For example properties like the number of prototypes per class and the number of relevant and irrelevant attributes should be known, which is not always the case in practice.

A number of empirical approaches have been proposed which tried to associate measurable properties of datasets with algorithm performance, mainly for predictive purposes, with the interpretation part severely neglected. The properties mostly used were statistical and information based measures of the data sample in an effort to capture unknown and difficult to measure traits of the data distribution like the number of irrelevant or redundant attributes. This approach was followed in STATLOG (Michie, Spiegelhalter, & Taylor, 1994) and later in (Brazdil, Carlos, & Costa, 2003; Kalousis & Theoharis, 1999; Todorovski & Dzeroski, 1999). Finding the right set of properties is not evident and depends on the class of algorithms examined (Kalousis & Hilario, 2001a; Kalousis & Hilario, 2001b). In another empirical approach called landmarking (Pfahringer, Bensusan, & Giraud-Carrier, 2000), the data sample is described in terms of the empirical errors of very simple learners, which are then associated with the empirical error of full blown learners, hoping to establish performance relations between the former and the latter. Finally in an approach which shares some similarities with the idea of margins in SVMs the model produced by a decision tree algorithm on the data sample is analyzed morphologically and its characteristics are used to describe the data sample (Bensusan, 1999; Peng et al., 2002). This is actually an idea whose origins could be traced back to the work of Brodley (1994) who used what she defined as

hypothesis pathology, 'recognizable symptoms of when a hypothesis fits poorly the data', in the form of hard coded rules derived from experts knowledge, to guide the selection of the appropriate class of models within a hybrid top down decision tree induction algorithm. What we see again here is an exploration of the level of synergy between different model classes and the data distribution, although the empirical error was not used in the selection but only the morphological characteristics of the decision surfaces.

In an orthogonal dimension Bay & Pazzani (2000) characterize the areas of expertise of a given model or the areas of agreement of pairs of models, for a given classification problem, within the feature space defined by that classification problem. A work that is more similar to meta-learning as this was defined by Wolpert (1992) where the goal was to learn how to combine predictions of a number of models on a specific classification problem.

The work presented in this paper shares elements with the work done in the STATLOG project and later in the METAL project (2002), but it is closer to the spirit of the generalization from case studies of Aha (1992), a work with a strong descriptive flavor. But at end the core of both approaches is the same, the description/prediction of performance patterns of learning algorithms using dataset characteristics. Our work extends that core since it views performance as a bidirectional link between datasets and algorithms that can be used by itself to discover relations both between datasets or between algorithms.

3. Characterizing performance

The link between datasets and algorithms is performance; there are many aspects of it, but we focus solely on classification error. Based on the estimated errors of the learners on a given dataset we characterize performance in two different dimensions, *relative performance* and *diversity*; it should be noted here that both of them are relative and not absolute ways of viewing performance. Diversity is a measure of the correlation of the errors that the learning algorithms commit.

3.1. Determining relative performance

The relative performance of a set of algorithms on a given dataset can be depicted by their ranking based on the statistical significance of the performance differences observed from all pairwise comparisons. In a pair of algorithms, x, y, if x is significantly better than y, then x is credited with one point and y with zero, if there is no difference each one gets half a point. The overall ranking is given by the number of points that each algorithm scored. The top ranked algorithm is the one with the most points. In the case of no significant difference everyone is credited with $(N-1)/2$ points, N the number of algorithms.

3.2. Measuring diversity

Diversity is a desirable property of an ensemble of classifiers, (Ali & Pazzani, 1996; Tumer & Ghosh, 1995). One metric of diversity in an ensemble of classifiers is the *error correlation*.

The error correlation between two classifiers, i, j, is defined as the conditional probability of the two classifiers making the same error given that one of them makes an error. Formally:

$$\phi_{ij} = p(\hat{f}_i(x) = \hat{f}_j(x) \mid \hat{f}_i(x) \neq f(x) \vee \hat{f}_j(x) \neq f(x)),$$
$$\phi_{ij} \in [0, 1],\ \phi_{ii} = 1$$

This definition provides a measure of similarity between classifiers, from which we can derive a *dissimilarity* measure using $\Phi_{ij} = 1 - \phi_{ij}$, a semi-metric since: $\Phi_{ii} = 0$ and $\Phi_{ij} = \Phi_{ji}$.

3.3. Experimental setup

We used 80 classification datasets, mainly from the UCI repository; the full list is given at Table 12 of the appendix. We worked with ten classification algorithms—*c50boost, c50tree, c50rules*: boosting, decision tree and rule inducers respectively, *clemMLP*: a multilayer perceptron, *clemRBFN*: a radial basis function network, *mlcNB*: a simple version of Naive-Bayes which assumes a nornal unimodal class conditional distribution for the continuous attributes, *mlcIB1*: 1-nearest neighbor (Duda, Hart, & Stork, 2001); *ltree*: a multivariate decision tree, *lindiscr*: a linear discriminants algorithm (Gama & Brazdil, 1999) and *ripper*: a sequential covering rule inducer (Cohen, 1995). All algorithms were used with their default settings.

We estimate error using 10-fold stratified cross validation, and control the statistical significance with McNemar's test. The overall significance level is 0.05, adjusted to 0.001 via the Bonferroni adjustment to account for the multiple pairwise comparisons.

Computing rankings and error correlations gives rise to work-matrices, R and EC. Rows of R correspond to datasets and columns to learning algorithms, $dim(R) = 80 \times 10$. Cell (i, j) gives the ranking of the j algorithm for the i dataset. Similarly the rows of EC correspond to datasets, but now columns correspond to pairs of algorithms, $dim(EC) = 80 \times 45$. The content of a cell is now the error correlation between a given algorithm pair for a given dataset. These tables represent explicitly the dual relation that characterizes classification; datasets in one dimension, algorithms in the other, linked by the performance results contained in the cells. They will constitute the basis of all our further analysis; both of them can be used in two ways: to characterize relations between datasets or algorithms.

4. Discovering similarities between algorithms

To discover relations between classification algorithms we studied the patterns of error correlation of pairs of algorithms among the different datasets. We can obtain the distribution of error correlation of a given pair by just retrieving the column of the *EC* matrix associated with the given pair. We describe that distribution by means of its histogram.

In order to summarize the different patterns of error correlation observed among the 45 pairs of algorithms we relied on clustering, where each clustering instance

corresponds to the histogram of the error correlation of a given pair of algorithms over a set of learning problems. We used an agglomerative hierarchical clustering algorithm (Duda, Hart, & Stork, 2001), together with *Ward*'s minimum-variance method to determine which clusters should be merged at each agglomeration step. The method merges these two clusters from a set of $c + 1$ that in the new configuration of c clusters minimize the following criterion:

$$J_e = \frac{1}{2} \sum_{i=1}^{c} n_i s_i$$

$$s_i = \frac{1}{n_i^2} \sum_{x \in c_i} \sum_{y \in c_i} s(x, y)$$

where n_i is the number of instances belonging to cluster c_i and $s(x, y)$ a measure of similarity between instances x and y. Since instances in our case are histograms we had to use a similarity measure which can exploit the semantics of histograms. We relied on the *affinity coefficient* (Bock & Diday, 2000). The affinity coefficient between two histograms P, Q, constructed over the same k bins, where $P(i)$ and $Q(i)$ are the corresponding frequencies of the ith bin, is defined as:

$$s(P, Q) = \sum_{i=1}^{k} \sqrt{P(i)Q(i)}$$

it takes values in [0, 1]; 1 when the distributions are exactly the same and 0 when they are orthogonal. Obviously the distance measure finally used by the agglomerative clustering algorithm was simply:

$$d(P, Q) = 1 - s(P, Q)$$

Four clearly distinct patterns of error correlation arose. The distribution associated with the center of each cluster along with the corresponding pairs of algorithms are given in figure 1.

Not surprisingly the algorithms that exhibited the highest error correlations were of the c50 family. Cluster-1 and cluster-4 contain only c50 algorithms and they are the only ones indicating strong error correlation. In the case of *c50rules* and *c50tree* the error correlation is higher than 80% for around 65% of the datasets. Cluster-3 contains the algorithm pairs that exhibit the lowest error correlation. It consists mainly of pairs of the form (a, b) where $a \in A = \{$*c50rules, c50tree, clemMLP, ripper, c50boost, ltree, clemRBFN, mlcIB1, mlcNB, lindiscr*$\}$ and $b \in B = \{$*mlcIB1, mlcNB, lindiscr*$\}$. Learning algorithms of group B are the most uncorrelated. Increasing the number of clusters to six, the cluster breaks up to three subclusters each one mainly associated with one of B's elements. Their visual inspection reveals that *mlcNB* and *lindiscr* are the most uncorrelated with algorithms of group A; *mlcIB1* exhibits slightly higher levels of correlation. Cluster-2 is associated with the pairs

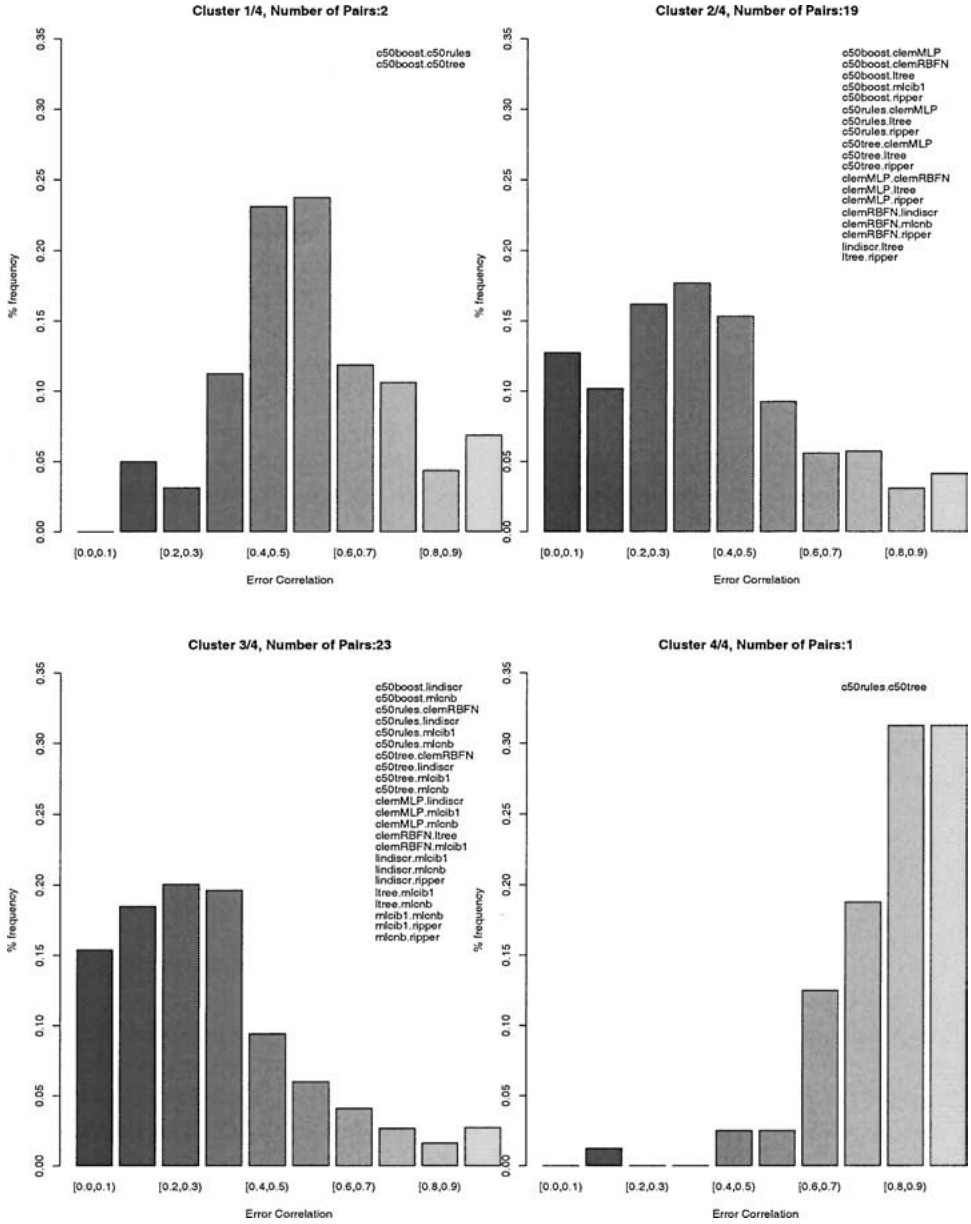

Figure 1. Clusters of algorithm pairs based on the distribution of error correlation.

of algorithms formed by the elements of set $A - B$, with low to average levels of error correlation.

The most relevant implication of these results is in the design of heterogeneous multiple models like *stacking generalization* (Wolpert, 1992). Usually the choice of base-level

algorithms is a *black-art* and done *ad-hoc*. The method described in this section allows us to identify clusters of algorithms with similar behaviors and to quantify the degree of diversity between algorithms.

It should be noted that these results hold on average and describe general trends. What exactly happens on specific datasets mainly depends on the given dataset; nevertheless taking into account these results provides a more informative basis of performing model combination, for example one could choose to combine only algorithms that are part of the uncorrelated group. In later sections, 6.1, 7.1, we will try to identify directly the datasets for which learning algorithms are expected to exhibit low error correlations. These results together with a selection limited to the group of the most uncorrelated algorithms can provide a strategy for efficient model combination.

5. Discovering similarities between datasets

So far we used performance data to discover similarities between learning algorithms. Now we will work in the opposite direction and use the same data to discover similarities between datasets. The result will be a rough charting of the dataset space with regions of this chart being characterized by the specific patterns of relative performance and error correlation. For example a given region of the dataset space could be described by the fact that algorithm X is systematically ranked at the top.

5.1. Clustering datasets using error correlations

We will use directly the *EC* matrix introduced in Section 3.3 to discover clusters of datasets with specific patterns of error correlations between algorithms, like for example clusters where the error correlations are low or clusters with high error correlation. Figure 2 gives the centers of the clusters discovered when applying hierarchical clustering. The height of each bar is the average error correlation of the corresponding pair of algorithms in the given cluster of datasets. For each cluster the pairs of algorithms are sorted with decreasing error correlation.

There are four clearly distinguished clusters. The high correlations between the members of the c50 family is constant among all of them. Cluster-2 and and cluster-4 correspond to low and high error correlation for all the pairs of algorithms, with medians of 16% and 64%. Between these two extremes we have cluster-1 and cluster-3, with medians of 30% and 48%.

In cluster-4 we see clearly the separation in groups A and B. Members of group B have lower error correlation with members of group A, around 50%, the pairs involving them appear on the right side of the sorted error correlation values. *mlcNB* is the most uncorrelated algorithm for that cluster of datasets. Members of group A exhibit a quite high correlation, around 80%. In cluster-3 we get an average level of error correlation, this time it is *mlcIB1* which is the most uncorrelated with any of the other algorithms. In cluster-1 we observe higher error correlations between the *c50boost*, *c50rules*, *c50tree*, *ltree*, *ripper* algorithms, again the pairs with the lowest error correlation systematically involve inducers

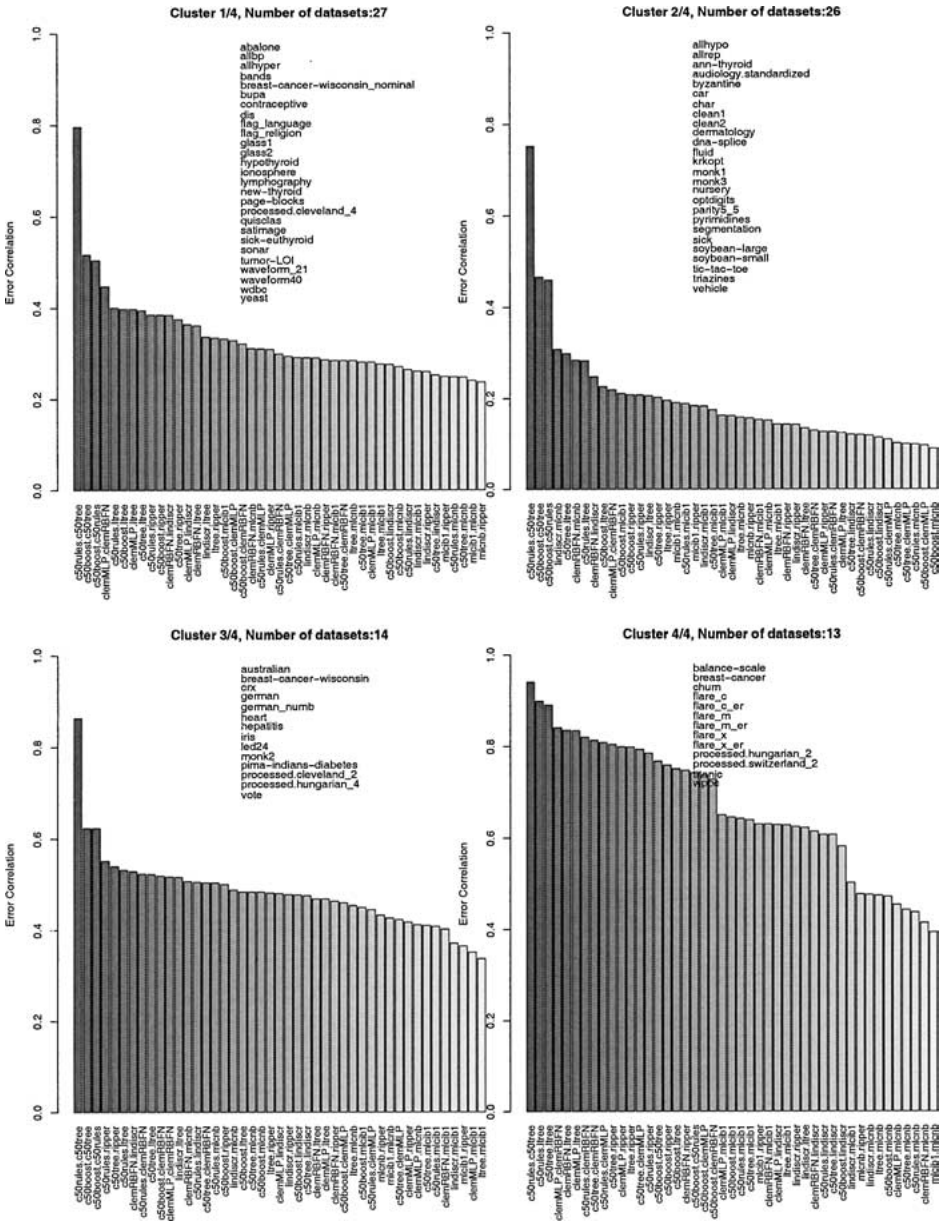

Figure 2. Clustering the datasets, based on the error correlations of pairs of algorithms.

from the *B* set. Finally in cluster-2, which contains more than 30% of the datasets, we get the lowest levels of average error correlation. For most of the algorithm pairs it is lower than 20%. For datasets falling within that cluster it is promising to perform model combination.

5.2. Clustering datasets using ranks

We will now try to chart the dataset space based on the relative performance of learning algorithms. To this end we will perform the same procedure as in the previous section but this time on the rankings matrix R.

Applying the hierarchical clustering algorithm we discover four main clusters of datasets, figure 3. Cluster-4 contains all datasets for which the large majority of the learning algorithms have a performance which is not significantly different. In fact it is the largest one and contains 28 out of the 80 datasets used in the study. Moreover in 14 datasets out of the 28 all the algorithms had exactly the same score. The second largest cluster is cluster-2, marked by the poor relative performance of *lindiscr, mlcNB* and *clemRBFN*. The c50 family takes the top positions, with *c50boost* achieving a considerable advantage relative to *c50tree*. In cluster-1 we have datasets where *mlcIB1* and *mlcNB* exhibit quite low relative performance, the other algorithms have similar performance with a slight advantage given to *ltree* and *clemMLP*. Cluster-3 is marked by the low position of the two neural networks algorithms, especially *clemMLP* which is by far the worst, and the significant improvement that *c50boost* achieves over *c50tree*.

In the next section we will seek a description of areas defined by the clustering processes in terms of dataset characteristics. Our primary goal is attaining an understanding of these areas by means of the dataset characteristics. This could be used to identify the regions of the dataset space to which a dataset belongs providing an informative way to select or combine appropriate algorithm(s).

6. Gaining Insights: descriptive metalearning

We will focus on specific regions of the dataset space that correspond to interesting and well defined *meta-learning* questions as these emerged from the analysis that took place in the two previous sections, 4, 5. The results of error correlation clustering, figure 2, set forth the following question:

– Could we characterize datasets on which the learning algorithms exhibit very low or very high error correlation?

From the patterns that emerged from the ranking clustering, figure 3, we can identify the following questions:

– Could we characterize datasets on which we expect all algorithms to perform the same?
– When *c50boost* improves strongly over *c50tree*?
– What are the datasets for which the multi-layer perceptrons perform really poorly?

To answer these questions we need a dataset description. We tried to keep that description as simple as possible and we came up with a list of eight characteristics given in Table 1. *LgE* is simply the logarithm of the number of examples that a dataset contains, a raw indication of the available amount of training data. The Logarithm of the Ratio of Examples to Attributes,

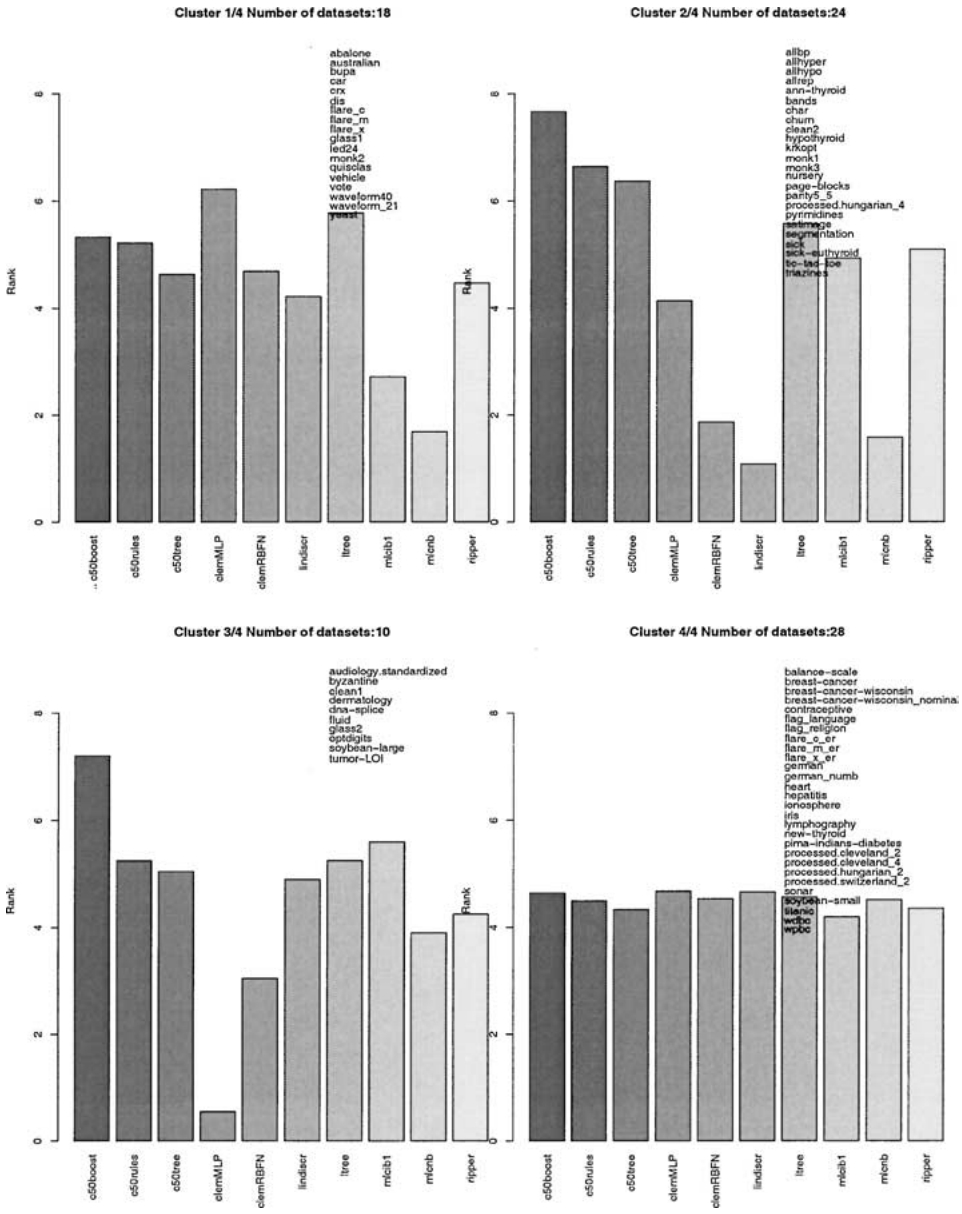

Figure 3. Clustering the datasets, based on the ranks of algorithms.

LgRAE, is a rough indicator of the number of examples available relative to the number of attributes, an indication of the dimensionality of the problem. The Logarithm of the Ratio of Examples to Classes, *LgREC*, is a rough approximation of the average number of examples available per class, under the simplistic assumption that examples are uniformly distributed

Table 1. Dataset characteristics.

Characteristic	Symbol
$\lg \#Examples$	*LgE*
$\lg \frac{\#Examples}{\#Attributes}$	*LgREA*
$\lg \frac{\#Examples}{\#Classes}$	*LgREC*
% of Symbolic Attributes	*PSA*
% of Missing Values	*PMV*
Class Entropy	*CE*
Normalized Class Entropy	*NCE*
Median of the Uncertainty Coefficients	*MedUC*

among classes. *PSA* gives the proportion of symbolic attributes; and *PMV* the percentage of missing values, the latter can be seen as an indicator of the quality of the data. Class Entropy, *CE*, gives an indication of the number of classes and of the distribution of the examples in these classes. For datasets with high number of classes and uniform distribution of examples among classes we get high absolute values of *CE*; as the class distribution becomes more and more uneven, *CE* approaches to zero. The problem with *CE* is that it is not bounded above by a fixed bound; its upper bound, $lg_2(C)$, depends on the number of classes, C. As a result of that a high value of *CE* can mean both a large number of classes and/or a uniform class distribution, while a low value indicates a small number of classes and/or an uneven distribution. To pinpoint the exact cause of a high value of *CE* we use the Normalized Class Entropy, *NCE*, which is simply $CE/\lg_2(C)$. *NCE* is bounded above by one, values close to one indicate uniform class distribution, while values close to zero indicate uneven distributions. The Median of the Uncertainty Coefficients, *MedUC*, provides an indication of the amount of information that each individual attribute contains about the class variable. The Uncertainty Coefficient, $UC(X, Y)$, between a variable X and a target variable Y, is the mutual information between the two variables divided by the entropy of the target variable Y. $UC(X, Y)$ measures the proportional reduction in the variation of target variable Y when X is known, (Agresti, 1990). Notice here the close relation between UC and the information gain ratio commonly, used in decision tree algorithms, (Quinlan, 1992), where the mutual information is divided by the entropy of variable X, in essence information gain ratio is the $UC(Y, X)$ thus it measures the proportional reduction in the variation of X when the value of the target variable Y is known. The computation of $UC(X, Y)$ when variable X is categorical is straightforward. When X is numeric we rely on the same technique used to compute information gain in decision tree algorithms, i.e. we compute $UC(X, Y)$ for all the binary splits of variable X and keep the maximum value. Finally since in a classification dataset we have a number of predictive attributes that give rise to an equal number of $UC(X, Y)$ values we report the median of these values, i.e. the already mentioned *MedUC*.

Each of the questions stated in the beginning of the section will be associated with one or more clusters. We will try to get a rough description of the formed groups of datasets on the basis of the characteristics given above. Since the amount of data at our disposal is limited

we will try to get a statistical description of the groups in terms of the distributions of the values of the dataset characteristics within each group. Moreover to spot those features, c_j, whose group-conditional distributions, $P(c_j \mid \omega_i)$, differ considerably among the different groups of datasets, ω_i, we will once more use the *affinity coefficient*. Characteristics with very different distributions among groups will serve as their basic descriptors.[1] In a similar study (Todorovski, Blockeel, & Dzeroski, 2002), focused on ranking clustering and used predictive clustering trees to predict rankings of algorithms.

The list of characteristics is by no means exhaustive. The definition of an appropriate set of characteristics is a topic which has attracted a lot of attention and depends on the nature of the metalearning problem examined each time. There will be probably other characteristics more appropriate in answering the meta-learning questions presented above, nevertheless this does not invalidate the study of how *these specific* characteristics relate to the given questions. One can easily imagine different sets of characteristics that represent other properties of the datasets and apply the same path of analysis in order to determine whether and how they are related to the given questions.

6.1. Low versus high error correlation

Using as guide the clustering results on error correlations, figure 2, we identify three clusters directly related with the question. Cluster-2 contains datasets for which almost all pairs of algorithms exhibit very low error correlation. Cluster-3 and cluster-4 contain datasets for which we have high to very high error correlation. We form two groups of datasets; ω_1, with the 26 datasets of cluster-2, and ω_2, with the 27 datasets of clusters 3 and 4. The first group corresponds to datasets with very low error correlation, while the second to datasets with high or very high error correlation.

The affinity distance, $d(P(c_j \mid \omega_1), P(c_j \mid \omega_2))$, for each characteristic c_j of Table 1 is given in column *ER.C.* of Table 2. In figure 4 we give the group-conditional distributions

Table 2. Ranking of dataset characteristics for the meta-learning questions using the affinity distance. See text for explanation; ER.C (§6.1), Equi (§6.2), MLP (§6.3), c50 (§6.4).

	ER.C.		EQUI		MLP		C50	
Char.	Rank	Affinity	Rank	Affinity	Rank	Affinity	Rank	Affinity
LgE	1	0.7351	1	0.6510	4	0.7471	1	0.5424
LgREA	3	0.7759	2	0.7271	1	0.2781	3	0.7737
LgREC	2	0.7756	3	0.8016	2	0.4607	2	0.6001
PSA	4	0.7759	5	0.8686	8	0.9098	4	0.8192
PMV	7	0.8672	7	0.8899	7	0.8876	8	0.9462
CE	5	0.8144	6	0.8754	5	0.7705	6	0.8821
NCE	6	0.8345	8	0.8946	6	0.7957	5	0.8268
MedUC	8	0.9160	4	0.8628	3	0.6568	7	0.9370

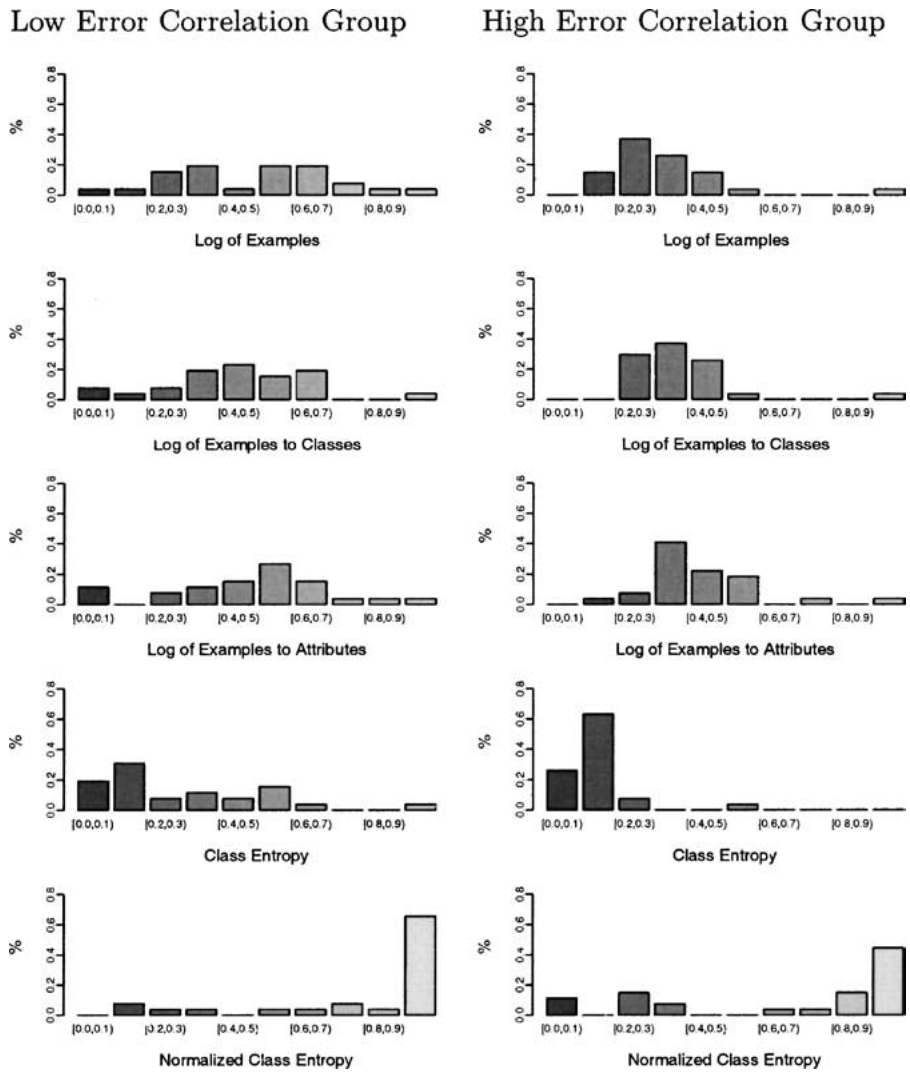

Figure 4. Group-conditional distributions of data characteristics.

in the form of histograms for each dataset characteristic, the presentation of characteristics is ordered by the affinity distance. In general we will limit comments to the ones that give the highest affinity distance, i.e. the ones that are most different between the two groups.

The three log-scaled characteristics exhibit similar distributions for the Low Error Correlation group. Their values are relatively uniformly distributed among the whole interval of possible values with slightly higher peaks appearing in the middle of the log-scale interval. On the other hand the corresponding distributions in the High Error Correlation group are more strongly peaked, more concentrated and appear closer to the lower side of the

scale intervals which means very low to low values for the three characteristics given in logarithmic scale and low values for the *CE*. The high rank of the Percentage of Symbolic Attributes seems to be a random effect; the two group conditional distributions are similar but with non-overlapping bins which explains the low value of the affinity coefficient.

For the High Error Correlation group the Class Entropy is strongly peaked and concentrated to the low values of the scale compared to a more uniform distribution within the Low Error Correlation group. The low values of *CE* can be due to two factors, large number of classes, or uneven distributions of examples to classes. To precisely determine their cause we have to take into account the distributions of the Normalized Class Entropy, NCE, the high(low) values of which indicate balanced(unbalanced) class distributions. For both groups the values of NCE are mainly concentrated to the higher end indicating quite balanced class distributions and in association with the high values of *CE* a large number of classes. However for the High Error Correlation group there is also a considerable concentration at the very low end of the interval, which is not observed in the Low Error Correlation group, indicating the presence of datasets with strongly unbalanced class distributions and thus explaining a portion of the observed low *CE*. For the Low Error Correlation group the high values of *CE* are mainly explained from the large number of classes.

Altogether what differentiates the High Error Correlation group from the Low Error Correlation group, is datasets with:

– low number of classes and/or unbalanced class distributions,
– limited number of examples,
– limited number of examples in relation to the number of attributes, i.e. high problem dimensionality,
– and a limited, on average, number of examples available per class.

The overall picture arising indicates that for datasets exhibiting these types of data pathology, i.e. as in the High Error Correlation group, one could expect learning algorithms to perform similarly, falling into the same type of errors on the same regions of the data space. It is the nature of the dataset that does not allow for a differentiation of algorithms performance; otherwise said with that quality of data nobody can do really better than its competitors. One plausible hypothesis is that errors are strongly correlated to these done by the Bayesian optimal classifier.

6.2. Equiperformance

Based on the results of dataset clustering using the rankings of the algorithms, figure 3, we can divide the datasets in two groups. The first group, ω_1, will contain all the datasets of cluster four, 28 datasets, the Equiperformance group, while the second, ω_2, the remaining 52 datasets, the Non-Equiperformance group.

By construction the Equiperformance and Non-Equiperformance groups have a strong association with the High and Low Error correlation groups defined in the previous section. In Table 3 we can see that almost all the datasets that belong to the Low Error Correlation group are part of the Non-Equiperformance group but not vice-versa; the Non-Equiperformance

Table 3. Cross classification in low, high, average error correlation and equiperformance and non-equiperformance groups

	Low-EC	High-EC	Avg-EC
Equi	1	17	10
Non-Equi	24	10	17

group contains datasets that belong either to the Low or to the High Error Correlation group, more over it contains some datasets that were not included in the High or Low Error Correlation groups. The Equiperformance group is strongly exclusive with the Low Error Correlation group. Summarizing:

$$\textit{Equiperformance} \rightarrow \neg\, \textit{Low-Error Correlation}$$

This association is an indication that we should not expect to increase the accuracy by model combination in problems were the existing models are equal because most probably they will have an average to high error correlation, i.e. not enough diversity.

In what concerns the group characterizations we expect them to be similar between the Non-Equiperformance and the Low Error Correlation groups, and between the Equiperformance and High Error Correlation. The affinity distances for the dataset characteristics are given in column *EQUI* of Table 2, and the corresponding group-conditional distributions in figure 5, ordered by the affinity distance.

Indeed there is a great degree of similarity between the two different ways to partition the datasets. The log-scaled characteristics have relative uniform distributions with slightly higher peaks towards the center for the Non-Equi-performance group, as it was the case for the Low-Error Correlation. For the Equiperformance group the distributions are more concentrated and shifted towards the low values similar to the ones of the High-Error Correlation group. The Class Entropy is strongly concentrated to low values for the Equiperformance group and less strongly for the Non-Equiperformance group. However this strong concentration to the lower scale for the Equiperformance group is now mainly attributed to a limited number of classes and not to an unbalanced class distribution; as we can see from the graph of the distribution of *NCE* most of the datasets have relatively balanced class distributions. In the Non-Equiperformance group we have many more datasets with unbalanced class distributions which explain a part of the low values of *CE*. Another characteristic whose group-conditional distributions vary among the two groups is the Median Uncertainty Coefficient. For the Non-Equiperformance group the values are strongly concentrated at the lower end of the scale indicating datasets where the attributes, at least when considered individually, contain a low amount of information about the class variable. For the Equiperformance group the concentration to the left side of the scale is less strong, there are cases that have a very high *MedUC* indicating datasets where the individual attributes contain significant information about the class, obviously this kind of datasets correspond to very easy classification problems where all algorithms should be expected to achieve high performance, thus not leaving space for significant differences between the algorithms.

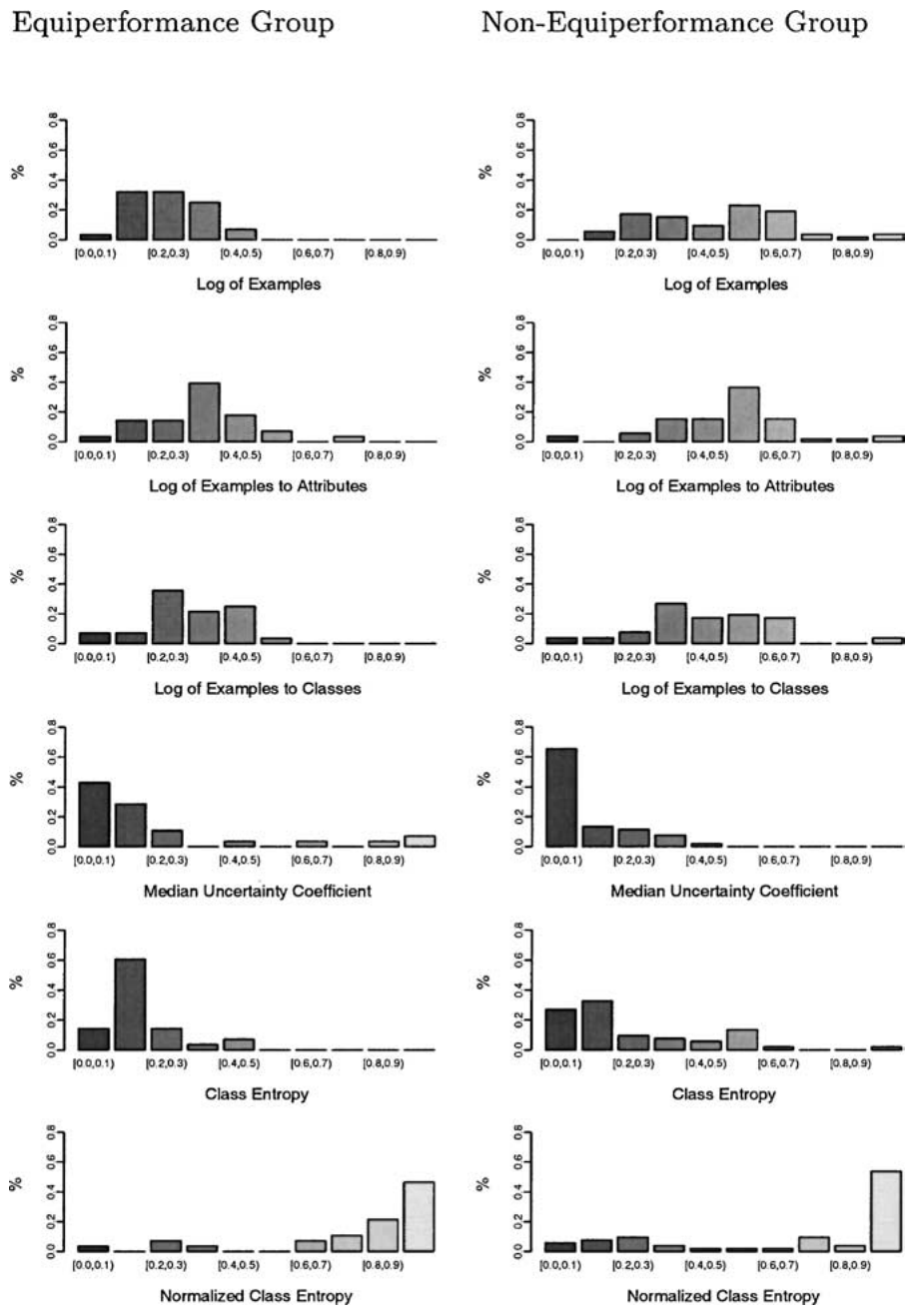

Figure 5. Group-conditional distributions of data characteristics.

On the whole the Equiperformance group is dominated by datasets with:

– low number of classes,
– limited number of examples,
– small number of examples relative to attributes, i.e. high dimensionality of the problem,
– and limited number of examples per class.

Another interesting property of the Equiperformance group, at least for the given experimental setup, is that the examined learning algorithms exhibit average to high error correlation on the datasets that belong to that group. Along the same lines as with the High Error Correlation group we could hypothesize that for datasets with the characteristics of the Equiperformance group the algorithms exhibit error levels that are very close to that of the Bayesian classifier.

Overall the Non-Equiperformance group does not have a specific pattern for the aforementioned characteristics, its datasets cover the whole spectrum of possible values, by contrast with the characteristics of the Equiperformance group that exhibit a more concentrated range of values. Unfortunately the Equiperformance group overlaps with a significant part of the Non-Equiperformance group, possibly hardening the discrimination task.

6.3. Multilayer perceptron

We will slightly reformulate the question that is related to MLP as follows: are there any characteristics that describe the group of datasets for which MLP performs very poorly, and how do they differ from the group of datasets where MLP performs well? From the clusters discovered by clustering the datasets using the ranks of the learning algorithms, figure 3, we select number one and three, and create two groups of datasets, ω_1, ω_2, with 18 and 10 member respectively. The first corresponds to datasets for which MLP is among the top performing learning algorithms and the second to datasets where it is by far the worst. We will not include cluster four, since as we saw it corresponds to datasets where it is very difficult to obtain significant differences and all the learning algorithms exhibit a high error correlation, and cluster two because of the average performance of MLP on the datasets that belong to that cluster. One important aspect in the definition of the two groups is that the high or low MLP performance is defined relatively to the other learning algorithms. The total sample size is quite small so the results should be cautiously interpreted.

The affinity distances between the two group conditional distributions are given in column *MLP* of Table 2. This time the list of the top four characteristics whose distributions differ most among the two groups of datasets consists of: the Logarithm of the Ratio of Examples to Attributes, *LgREA*, the Logarithm of the Ratio of Examples to Classes, *LgREC*, the Median Uncertainty Coefficient, *MedUC*, and the Logarithm of Examples, *LgE*. The group-conditional distributions of all characteristics are given in figure 6, in order of importance according to the affinity distance.

The datasets in which MLP exhibits very low performance are characterized by very low to low ratios of Examples to Attributes, *LgREA*, and a limited number of training Examples, *LgE*, i.e. problems with high dimensionality and not enough training data. On the other hand

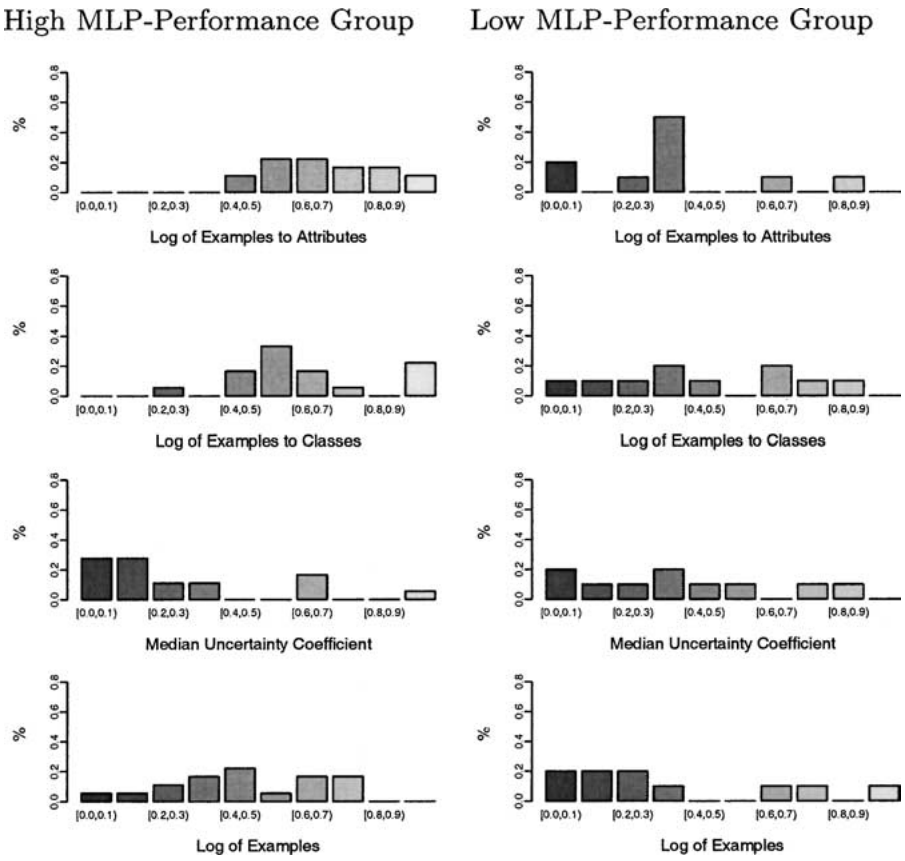

Figure 6. Group-conditional distributions of data characteristics.

for the high MLP-Performance group we have medium and higher values of the ratio of Examples to Attributes, and a larger number of training Examples, i.e. lower dimensionality and more training data. In fact we can very well separate the two groups just by examining the Ratio of Examples to Attributes, for which the two group conditional distributions are quite disjoint. The distribution of the Ratios of Examples to Classes, *LgREC*, for the low MLP-Performance group is relatively uniform, while for the high MLP-performance group this distribution is concentrated mainly to the center and to higher values, i.e. on average a high number of training instances per class-more training data. The overall picture so far is that MLP performs well in problems with low dimensionality and ample training data.

The third most important characteristic, i.e. the Median Uncertainty Coefficient, *MedUC*, provides us with some more unexpected insights. MLP achieves high performance for datasets that have a low *MedUC*, while it performs poorly for datasets with high *MedUC*, this is the opposite of what someone would expect[2]. In order to explain this paradox remember that performance, whether high or low, is defined relatively to that of the other learning

algorithms. In other words, MLP's performance is worse relative to the other algorithms for datasets that have high *MedUC*. If we examine the other learning algorithms we see that most of them incorporate explicitly a search bias based on the mutual information, information gain, so they capitalize better on the cases that we have high mutual information. At the same time within the group for which MLP performs well we have low *MedUC* in other words low mutual information. Since MLP is not dependent on the mutual information and it handles all attributes in parallel it can achieve better performance. What arises here is the distinction between parallel and sequential learning problems.

Summarizing, the high MLP performance group contains datasets with:

– average to high Ratios of Examples to Attributes, i.e. low problem dimensionality,
– an average to high number of Examples per Class,
– average to high number of Examples, and
– lower levels of mutual information,

However we should recall the low sample size and be careful in the level of confidence we place on these observations.

6.4. C50boost or Tie?

In order to characterize datasets for which *c50boost* significantly outperforms *c50tree* we will create two groups of datasets. The first, ω_1, will contain all the datasets for which *c50boost* is significantly better than *c50tree*, (18 datasets), while the second, ω_2, contain the datasets for which there is no significant difference between the two learners, (60 datasets), there are only two datasets for which *c50tree* is better than *c50boost* and we will not consider them here.

The affinity distances are given once again in Table 2, column *c50*, while the group-conditional distributions in figure 7. In the top position we find the Number of Examples, *LgE*, the Ratio of Examples to Classes, *LgREC*, the Ratio of Examples to Attributes, *LgREA*, and the Percentage of Symbolic Attributes, *PSA*. The datasets for which *c50boost* has a performance which is significantly better than *c50tree* are characterized by high numbers of training examples, and high ratios of examples to classes and examples to attributes. On the other side the datasets for which the two algorithms do not have a significant difference are described by low numbers of examples, and low values of the ratios. In short abundance of data is connected with a significantly better performance of *c50boost* over *c50tree*. Again we chose to ignore the Percentage of Symbolic Attributes, although it was highly ranked, because it does not provide a consistent pattern.

The Normalized Class Entropy, *NCE*, provides us with another interesting insight. The group of datasets for which boosting does not improve over c50tree is partially characterized by low values of *NCE*, i.e. quite uneven class distributions. In fact we could even argue just by observing the group conditional distributions of *NCE* that if the value of *NCE* is lower than 0.5 then we could be almost certain that the two algorithms will have a similar performance,[3] for datasets with *NCE* higher than 0.5 we cannot make any kind of statement since for that range of values the two distributions overlap considerably. This

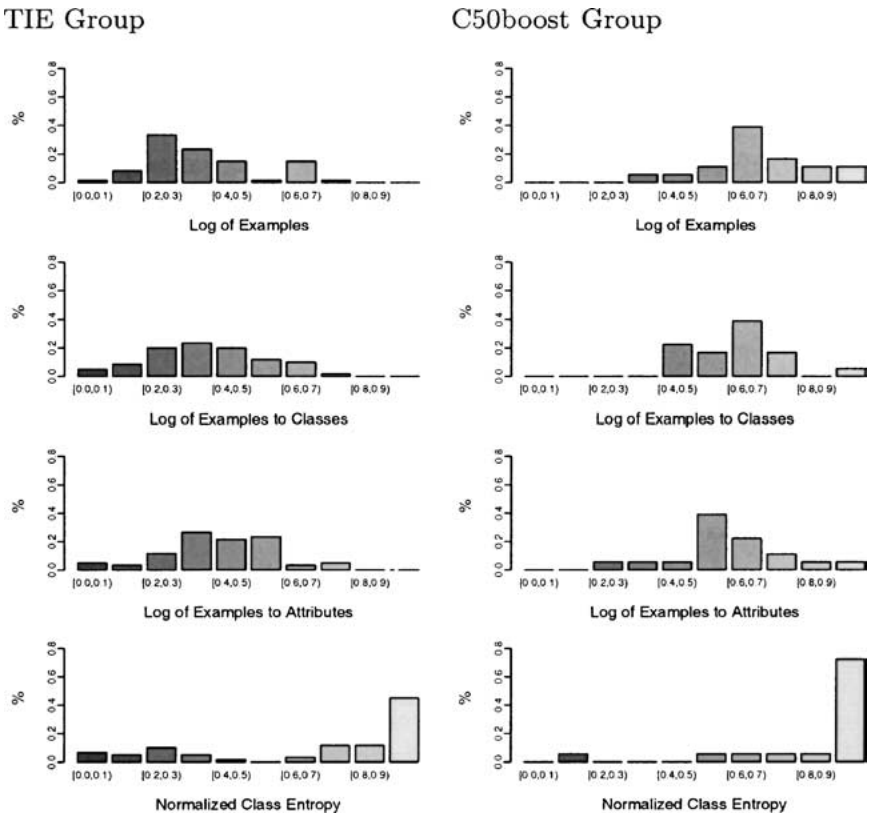

Figure 7. Group-conditional distributions of data characteristics.

does not necessarily mean that *c50boost* cannot improve over *c50tree* on problems with uneven class distributions. The absence of a significant difference could be attributed to the inherent difficulty that these problems pose due to class imbalance. In order to address this issue we need a way to assess the difficulty of a given classification problem. We will use a rather naive way based on the default error. A problem will be considered 'easy' if at least one of the two algorithms manages to achieve an error lower than the default error at a statistically significant level, and 'difficult' otherwise. Taking a closer look at the datasets that have $NCE < 0.5$ we found 17. For 16 of them, *c50boost* and *c50tree* did not have a significant difference. For half of these 16 datasets, the two algorithms were significantly better than the default, 'easy' classification problems, while for the other half they were not significantly different than the default or they were even significantly worse, (one case). Altogether a low value of *NCE* is not linked with a 'difficult' learning problem for which the two algorithms would have been expected to perform similarly. It is thus an effect of the uneven class distribution, rather than the inherent difficulty of the problems, that boosting does not improve over *c50tree*.

The group of datasets for which boosting does not improve *c50tree* is characterized by:

– lower number of examples,
– lower number of examples relative to classes,
– lower number of examples relative to attributes, and
– a strong presence of datasets with uneven class distribution

The first three dimensions are rather good descriptors of the general tendency within the group of datasets for which we get no difference, while the last characterizes a part of these datasets. Their discriminating and predictive power of the characteristics will be examined further in Section 7.4.

6.5. Summary of descriptive metalearning

Based on the results of sections 4,5, we identified a number of metalearning questions. We established a number of dataset characteristics and undertook an exploratory analysis in order to identify characteristics that are useful in tackling the metalearning questions.

The analysis revealed two types of metalearning questions. In the first one, the group conditional distributions of the most important characteristics exhibited a low level of overlap among the groups of datasets associated with each question. Usually one of the groups had a much more compact and concentrated distribution for some of the characteristics, while the distributions of the other group were more uniform. In some cases the limited level of overlap seem to indicate that we could even hope for a fair level of discrimination among the groups.

In the second type of metalearning questions, although we still observe the same pattern of more concentrated and compact distributions of characteristics, the overlap is now considerable. While the characteristics are good descriptors of the groups of datasets they seem to lack sufficient discriminatory power to distinguish between groups.

Nevertheless since the analysis performed so far was limited in examining only one attribute at the time we can still hope for a fair level of discrimination between groups using tools that take into account more than one attributes and model their interactions.

Among the characteristics that were deemed to be good descriptors we find the number of examples, the ratios of examples to classes and examples to attributes, all of them indicators of data availability and problem dimensionality. Moreover in some of the problems the class distribution, measured by the class entropy, and the amount of information delivered by individual attributes for the class are also good data descriptors.

A last interesting observation was the strong empirical evidence that datasets for which the classification algorithms tend to have equal performance they are datasets in which the error correlation of the algorithms is average to high making model combination less promising.

7. Predictive metalearning

The analysis so far has been exploratory and rather descriptive in nature. The main emphasis was placed on the discovery and description of the factors which were relevant in answering the given meta-learning questions. We now move one step ahead and set prediction as the goal, i.e. we cast the problems as typical classification problems. Viewing the problems as meta-learning problems has two benefits. First and most important it provides a natural way to validate the followed approach, an indirect way to determine the relevance of the selected characteristics for the given meta-learning questions, based on the estimated errors of the classification algorithms on the meta-level. Second, it acts complementary to the descriptive analysis of Section 6, the classification algorithms used are able to model interactions between attributes thus capturing more complex relations that could not be detected by the descriptive analysis.

We examined three different classification algorithms on the metalearning level that correspond to well defined and clearly separated learning biases. An orthogonal, univariate decision tree, *c50tree* (Quinlan, 2000), an oblique, multivariate decision tree, *ltree* (Gama & Brazdil, 1999) and a simple linear discriminant algorithm, *lindiscr* (Duda, Hart, & Stork, 2001). Performance differences of the three learners can reveal further information about the type of relations between the dataset characteristics and how these determine the class. All three algorithms produce simple and easily understandable models.

The error estimation is done using 10-fold stratified cross-validation. Estimated errors are compared to the default error,[4] via the McNemar test of significance. The significance level is set to 0.05 and adjusted via the Bonferroni adjustment to 0.016 to take into account the three comparisons, i.e. each algorithm against the default. ++ indicates a difference which is significant at the 0.016 level and + a difference which is significant only at the 0.05 level. The results for all the meta-learning questions examined, using the full set of characteristics given in Table 1, are gathered in Table 11 under the column Full Set.

In what concerns the dataset characteristics we should mention here that for every problem they were all rescaled at the interval [0, 1].

7.1. Low versus high error correlation

In Section 6.1 we divided datasets into two groups based on whether algorithms applied to them produced weakly or higly correlated errors. In this section the predictive task is: given the characteristics of a dataset, determine whether the ten base learning algorithms used will exhibit a high/low error correlation. The number of available instances is 53, the majority class is the High Error Correlation class with 27 datasets, and the corresponding default error estimated at 0.4906.

All three meta-learning algorithms give quite good results and all of them are significantly better than the default error (Table 11.i, column Full Set); the lowest error is achieved by *c50tree*, 0.1509. Thus there is strong indication that the set of characteristics established is really relevant in describing and discriminating datasets according to the level of error correlation that learning algorithms exhibit on them.

Table 4. Model built by *c50tree* on the Error Correlation problem.

```
(Errors on Training Set: 6 / 53)
[]ClassEntropy > 0.155 (0.995)
|       Low Error Correlation (22.0/19.0)
[]ClassEntropy <= 0.155 (0.995)
|       []logExamples <= 0.501 (7.745)
|       |       High Error Correlation (25.0/23.0)
|       []logExamples > 0.501 (7.745): Low Error Correlation (6.0/5.0)
```

Table 4 gives the model derived by the algorithm with the lowest estimated error, in this case *c50tree*, when it was applied to the *full* training set. The models constructed by the other algorithms are given in Table 13 of the appendix. In the following sections we will also limit the presentation to the model constructed by the algorithm with the lowest estimated error and give the other models in the appendix. *C50tree* constructed a very simple model[5] which uses only two dataset characteristics, Class Entropy and log of Examples, and missclassifies six of the 53 instances of the training set. In essence what it did was to define two orthogonal splits on the plane defined by the two aforementioned characteristics, splits that are clearly identifiable if one would examine the corresponding two–dimensional plot of the two characteristics. The constructed decision tree quantifies in a precise way some of the conclusions already given in Section 6.1. It simply says that low values of class entropy, i.e. limited number of classes and/or uneven class distributions, and limited number of training examples lead to high error correlation between the learning algorithms.

What about the other characteristics which are not present in the model? Are they useless? Remember that the presented model was constructed on the full training set, when examining the models constructed in each of the folds of the cross-validation we see that there are more characteristics used than just these two. In Table 5 we give the frequency with which every characteristic is selected as a decision node for a given level within a decision tree (*No. of presences* column) and establish a score that quantifies the 'importance' of the

Table 5. Frequency and level of selection of dataset characteristics by c50tree among the 10-fold of the cross-validation.

	No. of presences				
Tree Level	1	2	3	4	Score
LgE	1	3			1.263
LgREA			7		1.535
LgREC		8			2.988
PSA				1	0.021
PMV		1	1		0.138
CE	9	1			8.053
NCE					0
MedUC		1	1		0.155

given characteristic. This is inversely proportional to the tree level in which a characteristic appeared and proportional to the accuracy of that decision tree on the test set. Furthermore we weight the importance of a characteristic by the percentage of times, P, it was present in the ten models produced in the ten fold cross validation. The exact formula is:

$$Score = P \times \left(\sum_{i=1}^{\#presences} \frac{1}{level_i} \times accuracy_{DecTree_i} \right).$$

A characteristic is more important the more often it appears at high levels of decision trees that achieve good accuracy.

Class Entropy is by far the most important characteristic, according to the score, being selected as the first decision node in nine out of the ten cross validation folds and once at the second decision level. Other characteristics achieving relatively high score are the log of the Ratio of Examples to Classes, the log of the Ratio of Examples to Attributes and the log of Examples. Although the log of Examples is one of the two characteristics appearing in the model built on the whole training set its 'importance' measured by its score, is lower than those of the log of the Ratio of Examples to Classes and log of the Ratio of Examples to Attributes which are selected more frequently. The importance of the various characteristics as given by their selection rates among the different folds tends to agree with the affinity coefficient given in Section 6.1. Furthermore each of the various models produced from the folds quantifies in a precise way some of the observations already done in Section 6.1.

7.2. Equiperformance

The classification problem is derived directly from Section 6.2 and involves two classes, Equiperformance and Non-Equiperformance with 28 and 52 datasets respectively, resulting in a default error of 0.35.

This time results are not as good; none of the algorithms manages to beat the default error at a statistically significant level (Table 11.ii, column Full Set). Their averaged error is around 0.26 with the best performance achieved by *lindiscr*, 0.225. The low performance in terms of predictive error implies that the chosen characteristics do not contain enough information to discriminate between the two classes of the Equiperformance problem. Initial evidence of that was already apparent from the analysis in Section 6.2 where datasets of the Equiperformance group exhibited a more constrained range of values of their characteristics than the Non-Equiperformance group, with nevertheless an important overlap between the two groups. In a few words there is a difference in the description of the two groups but this difference is not sufficient for discrimination purposes.

Since the lowest estimated error was achieved by the *lindiscr* in Table 6 we give the model that it constructs when applied on the full training set. Because all dataset characteristics were rescaled on the same interval, the coefficients of the linear discriminant equation provide an indication of their discriminatory power. According to the model the most important characteristic is the number of examples, *LgE* which is positively correlated with the Non-Equiperformance class, followed by the Ratio of Examples to Classes, *LgREC* and the Class Entropy, *CE*, which are positively correlated with the Equiperformance group.

Table 6. Model built by *lindiscr* on the Equiperformance problem.

```
(Errors on Training Set: 15 / 80)
IF +1.363-41.63*LgE +3.081*LgREA +33.1*LgREC -0.06203*PSA
       -1.875*PMV+14.46*CE-5.155*NCE+3.088*MedUC > 0
THEN Equiperformance
ELSE Non-Equiperformance
```

There is a partial agreement between the descriptive power of the characteristics as determined by the affinity coefficient, and their predictive power as determined by the coefficients of the linear discriminant. In both cases the most important characteristic seems to be the number of examples, *LgE*. The Ratio of Examples to Classes, *LgREC*, is also assigned high relative importance by both types of analysis. The degree of agreement is much higher when we examine the models produced by *c50tree* and *ltree*, Table 14 of appendix. For example in the case of the *c50tree* model the characteristics selected as decision nodes are the four most important descriptive characteristics.

The poor predictive performance of all meta-learning algorithms indicates that the current set of characteristics is not adequate to safely determine whether a dataset is an Equiperformance dataset or not.

7.3. Multilayer Perceptron

The predictive problem of this section is directly derived from the descriptive task of Section 6.3, with two classes, one that corresponds to datasets for which MLP exhibits high performance, and one where it exhibits low performance. Remember that the performance is defined relative to that of the other learning algorithms on the same dataset. The goal of the predictive task will be simply to determine whether for a given dataset MLP is expected to have high or low relative performance. The low MLP performance class contains 10 datasets and the high MLP performance class contains 18, resulting in a default error of 0.35.

None of the algorithms beats the default error at the significance level set with the Bonferroni adjustment (Table 11.iii, column Full Set). However *ltree* manages to beat the default error with a significance level of 0.0455. Taking into account the fact that the sample size is quite small, only 28 instances, the level of performance achieved by *ltree* can be considered satisfactory.

The model constructed by *ltree* on the complete set is given in Table 7. It is a very simple model with just two characteristics selected as decision nodes, the Ratio of Examples to Attributes and the number of Examples. This result agrees with the observations of Section 6.3 where the affinity coefficient revealed that the Ratio of Examples to Attributes is the most important characteristic for the description of the two groups of datasets. Furthermore when we examined the models build by *ltree* on each one of the ten folds of the cross validation we saw that in seven out of the ten folds the model produced is exactly the same as the one produced from the full training set, indicating both the strong discriminatory power of the two chosen characteristics, but also the consistency of the model with respect to different training sets. The produced model concretely specifies the surfaces that provide the best discrimination within the space defined by these two characteristics.

Table 7. Model built by *ltree* on the MLP problem.

```
(Errors on Training Set: 1 / 28)
[]log.examplesToAttr <= 0.389 (3.080)
|       Low MLP Performance (8.00/8.00)
[]log.examplesToAttr > 0.389 (3.080)
|       []logExamples <= 0.729  (8.517)
|       |       High MLP Performance (17.00/17.00)
|       []logExamples > 0.729  (8.517)
|       |       Low MLP Performance (3.00/2.00)
```

Table 8. Frequency and level of selection of dataset characteristics by *ltree* among the 10-fold of the cross-validation, MLP.

	No. of presences		
Tree Level	1	2	Score
LgE		7	2.45
LgREA	10		8.98
LgREC			
PSA			
PMV			
CE		1	0.033
NCE			
MedUC			

Most of the remaining characteristics are never selected in any of the fold models (score of zero in Table 8). However this should not lead us to fast conclusions, e.g. consider the remaining characteristics useless in what concerns their predictive or descriptive properties. We can only say that for the selection bias of the specific learner, (i.e. information gain ratio), the specific characteristics are of low quality. To get a more complete picture of the predictive power of the unchosen characteristics we simply repeated the specific meta-learning experiment removing the LgE characteristic, (the second most informative characteristic for *ltree*). This resulted in a deterioration of the errors of *c50tree* and *ltree* but in an improvement of the error of *lindiscr* that fall to 0.1071 thus equaling the error of *ltree*. Moreover linear discriminant assigns high discriminating power to a number of characteristics as indicated by their corresponding coefficients (Table 9), thus supporting the utility of a larger set of characteristics than just the two selected by *ltree*.

7.4. C50boost or Tie?

In this section we will explore the predictive and discriminative power of the dataset characteristics for the problem examined in Section 6.4, i.e. what characteristics of a dataset determine whether boosting improves over *c50tree*. The corresponding predictive task will consist of two classes of datasets. The first class consists of the 18 datasets for which

Table 9. Model built by *lindiscr* on the MLP problem after removing *LgE*.

```
(Errors on Training Set: 2 / 28)
IF -3.282  +26.91*LgREA  -12.34*LgREC +0.7066*PSA
          -1.2930*PMV -24.91*CE +6.831*NCE -2.0560*MedUC > 0
THEN High MLP Performance
ELSE Low MLP Performance
```

Table 10. Model built by *lindiscr* on the C50boost or Tie problem.

```
(Errors on Training Set: 5 / 78)
IF +12.68-17.410*LgE +5.428*LgREA -3.136*LgREC +0.346*PSA
          -0.404*PMV -3.475*CE    -3.849*NCE   +0.708*MedUC > 0
THEN TIE
ELSE c50boost
```

c50boost is significantly better from *c50tree*. The second class consists of the 58 datasets for which the two algorithms do not have a significant difference. The default error is 0.2307.

Lindiscr exhibits an estimated error significantly lower than the default error, an error which is the result of the missclassification of only four instances in the 10-fold stratified cross-validation (Table 11.iv, column Full Set). The very good performance of *lindiscr* implies that we have a classification problem for which the decision surface is a simple hyperplane in the space defined on the dataset characteristics, i.e. the two classes are linearly separable.

Examining the discriminating importance of the dataset characteristics, as determined by the coefficients of the linear discriminant constructed on the full training set (Table 10), we see that the four most important characteristics are: the number of Examples, LgE, the Ratio of Examples to Attributes, *LgREA*, the Normalized Class Entropy, NCE and the Class Entropy, *CE*. The discriminating power of the characteristics is in close agreement with their descriptive power as discussed in Section 6.4. If we were to inspect the two dimensional plots of the data characteristics we would see that a very clear separation between the two classes is provided by the pair *NCE-LgE*. Two characteristics that according to the linear discriminant are ranked first and third in terms of discriminating power. On the other hand the least discriminating characteristics are the Percentage of Missing Values, *PMV*, the Percentage of Symbolic Values, *PSA*, and the Median Uncertainty Coefficient, *MedUC*, characteristics which were also considered of poor descriptive power. The models produced from *ltree* and *c50tree* on the full training set (Table 16), also support the above conclusions indicating as the most discriminating characteristics the: *LgE*, *NCE*. The superior performance of *lindiscr* could be also explained from the fact that it assigned high importance to five out of the eight dataset characteristics while the decision tree algorithms used just two of them.

7.5. Summary of predictive metalearning

Section 7 was meant to complement Section 6 in two ways. First by exploiting stronger analytical tools which overcome the limitations of the analysis given in Section 6. The algorithms used here are able to model interactions between many characteristics. Second

Table 11. Estimated errors and p-values on different feature sets for the four metalearning problems.

	Error Correlation					
	Full Set		*LgE*, *LgREA*, *LgREC*		*LgE*, *LgREA*, *LgREC*, *CE*	
	Error	p-value	Error	p-value	Error	p-value
i						
default	0.4906					
c50tree	0.1509	*0.00083* ++	0.3396	0.0664	0.1509	*0.00083* ++
lindiscr	0.2264	*0.00351* ++	0.2264	*0.00511* ++	0.1886	*0.00085* ++
ltree	0.2075	*0.00326* ++	0.2264	*0.00511* ++	0.1886	*0.00208* ++
	Equiperformance					
ii	Full Set		*LgE*, *LgREA*, *LgREC*		*LgE*, *LgREA*, *LgREC*, *medUC*	
default	0.35					
c50tree	0.2750	0.3268	0.25	0.18588	0.25	0.20124
lindiscr	0.2250	0.1003	0.25	0.22995	0.2375	0.13739
ltree	0.2875	0.4414	0.225	0.13361	0.25	0.24335
	MLP					
iii	Full Set		*LgE*, *LgREA*, *LgREC*		*LgE*, *LgREA*, *LgREC*, *medUC*	
default	0.35					
c50tree	0.1428	0.0771	0.1071	0.02334 +	0.1071	0.0233 +
lindiscr	0.1428	0.1489	0.0357	*0.00003* ++	0.0714	0.0269 +
ltree	0.1071	0.0455 +	0.0357	*0.00003* ++	0.0714	*0.0133* ++
	c50boost or TIE					
iv	Full Set		*LgE*, *LgREA*, *LgREC*		*LgE*, *LgREA*, *LgREC*, *NCE*	
default	0.2307					
c50tree	0.1410	0.1686	0.16666	0.3588	0.12820	0.080
lindiscr	0.0641	*0.0019* ++	0.17948	0.5224	0.06410	*0.0019* ++
ltree	0.1153	0.0388 +	0.12820	0.0801	0.06410	*0.0019* ++

the estimated classification error provides a mean to assess the utility and relevance of the selected set of characteristics.

For two of the problems, *error correlation, c50boost*, we got error estimations which were significantly lower than the default error. For the *MLP* problem the difference was significant only at the 0.05 level, while for the *equiperformance* problem there was no significant difference. We can thus argue that the set of characteristics chosen is really relevant at least in addressing the first three problems. The fact that for the *MLP* problem we could not obtain a difference which would be significant at the 0.016 level could be attributed to the low number of training instances, only 28.

Analysing the meta-models produced no surprises; they support the results obtained in Section 6 at least for the *error correlation, c50boost* and *MLP* problems. The characteristics identified as important descriptors of the different groups of datasets were also important in the induced models. Moreover some of the produced models quantify precisely observations that were already apparent. The obvious advantage of the produced meta-models over the analysis of Section 6 is that they go beyond pairs of characteristics modeling many feature interactions.

8. Using descriptions for predictions

In an effort to coherently combine the two analysis paths, i.e. descriptive and predictive, giving a final global picture we undertook two more series of experiments. Here the idea was to directly exploit observations from the descriptive analysis in order to select a better subset of features for each one of the meta-learning problems. The first series is based on the observation that a number of characteristics related to data availablity and problem dimensionality were always pointed out by the descriptive analysis as quite informative; these were the number of examples, the ratio of examples to attributes and the ratio of examples to classes. We reduced the training sets to only these three characteristics and examined their predictive power. In the second one we simply selected the four most descriptive characteristics, for each metalearning problem, according to the results of the descriptive analysis. The complete results of these experiments are given in the second and third columns of Table 11. The set constructed from the four top characteristics gave the best results for the Error Correlation and the *c50boost* or Tie problem, it also reduced the errors for the MLP problem to a significantly lower level than the default. For the last one the feature set composed of the three descriptors was the one achieving the best results. Overall it seems that the most important characteristics are the three indicators of data availability, *LgE*, *LgREC*, *LgREA*, accompanied depending on the problem by class entropy, normalized class entropy or the uncertainty coefficient. As before the only problem for which we did not manage to get a significant improvement was the Equiperformance problem.

9. Conclusions and future work

This paper follows the empirical tradition, but unlike most previous work of its kind the main emphasis is placed on gaining understandable insights and not on predictive performance. Furthermore it spans two orthogonal dimensions, datasets and learning algorithms. The link between theese is performance which can be seen as a function of their characteristics. We looked for similarities between algorithms by means of error correlation, and similarities between datasets based on patterns of error correlation and relative performance of algorithms. These chart the dataset space where each region is governed by specific patterns of relative performance or error correlation of the learning algorithms. We described parts of that space by means of the distributions of simple properties of the datasets; in some of the cases the results can also be exploited for discriminatory purposes, in others they serve as qualitative descriptions. The most frequent finding was that the described areas were

characterized by rather concentrated distributions of descriptors of:

- *data availability, problem dimensionality*: number of examples, ratio of number of examples to number of classes, ratio of number of examples to number of attributes
- *class distribution*: class entropy and normalized class entropy
- *information content*: uncertainty coefficient of attributes and class

Overall the provided insights were consistent with common knowledge in the machine learning field. Moreover some of them receive quite strong support from evaluation under a prediction scenario. The results support the validity of the methodology as a tool for the exploration of metalearning issues.

We should stress once more that the chosen set of characteristics is in no way complete. For sure there are other factors that are also relevant, even more relevant, in answering the metalearning issues that were raised here. The conclusions drawn are not global, in the sense that they do not take into account all the possible important properties. Nevertheless they are valid to the extend that they shed light on to how the specific characteristics are implicated on the examined metalearning problems. The presented methodology is not bound to the set of characteristics used. We could undertake the same type of analysis with a different set of characteristics exploring different dimensions of the meta-learning problem. For example we could focus on characteristics that measure the redundancy of information contained in a dataset, e.g. correlations of attributes, and examine the patterns of relative performance with respect to these dimensions.

An interesting, but yet unexplored, direction on dataset characterization would be the combination of landmarking and the model based description of the data sample. Simple learners would be applied on a given sample and the morphologies of the produced models would be analyzed trying to look for synergies between the data sample and the form of the functions that the simple learners assume. Synergies that can be then used to guide the selection among a number of full fledged learning algorithms, but also to provide further insight to these data factors that determine algorithms relative performance.

Another possible direction is the systematic evaluation of the proposed strategy for model combination, i.e. identify whether a dataset is a low or high error correlation dataset by examining its characteristics and in the case that it is, combine algorithms that were discovered to be mostly uncorrelated. Compare this approach to a more classical one that will be based on the direct estimation of error correlation.

Further future work includes the description of more meta-learning issues that remained unaddressed. One important dimension that was not considered here was the effect of parameter tunning. This study was performed using the default strategies provided by each learning algorithm. We would like for example to examine how the results presented here fare when we perform careful parameter tunning and select for each algorithm the best set of parameters. In an alternative scenario on parameter tunning we could perform the same type of analysis but limited to a given learning algorithm whose parameters vary, e.g. the behavior of the prunning parameter for decision trees. Further extensions could examine other performance dimensions than simply error based, for example incorporate the dimension of training time in the definition of the metalearning problems.

Appendix

Table 12. Datasets characteristics.

Dataset	No. of classes	No. of continuous	No. of symbolic	No. of examples	% Missing	Class entropy	Median UC
abalone	29	7	1	4177	0.00	3.6020	0.0905
allbp	3	6	22	3772	0.02	0.2751	0.0061
allhyper	5	6	21	3772	0.02	0.2075	0.0147
allhypo	5	6	22	3772	0.02	0.4666	0.0039
allrep	4	6	22	3772	0.02	0.2599	0.0067
ann-thyroid	3	6	15	7200	0.00	0.4476	0.0027
audiology.standardized	24	0	69	200	0.02	3.4462	0.0154
australian	2	6	8	690	0.00	0.9912	0.0460
balance-scale	3	0	4	625	0.00	1.3181	0.1027
bands	2	19	17	540	0.05	0.9825	0.0168
breast-cancer	2	0	9	286	0.00	0.8778	0.0294
breast-cancer-wisconsin	2	9	0	699	0.00	0.9293	0.5118
breast-cancer-wisconsin-nominal	2	0	9	699	0.00	0.9293	0.5532
bupa	2	6	0	345	0.00	0.9816	0.0235
byzantine	71	60	0	17750	0.00	6.1497	0.1126
car	4	0	6	1728	0.00	1.2057	0.0706
char	10	6	0	5109	0.00	3.2848	0.0399
churn	2	7	8	110414	0.00	0.2970	0.0056
clean1	2	166	0	476	0.00	0.9877	0.0366
clean2	2	166	0	6598	0.00	0.6201	0.0378
contraceptive	3	0	9	1473	0.00	1.5390	0.0211
crx	2	6	9	690	0.01	0.9912	0.0415
dermatology	6	1	33	366	0.00	2.4325	0.1755
dis	2	6	22	3772	0.02	0.1146	0.0029
dna-splice	3	0	180	3186	0.00	1.4798	0.0035
flag-language	10	10	18	194	0.00	2.8753	0.0373
flag-religion	8	10	18	194	0.00	2.5463	0.0369
flare-c	9	0	10	1066	0.00	0.9500	0.0259
flare-c-er	4	0	10	323	0.00	0.5835	0.0358
flare-m	9	0	10	1066	0.00	0.2491	0.0378
flare-m-er	4	0	10	323	0.00	0.5665	0.0642
flare-x	9	0	10	1066	0.00	0.0464	0.1510
flare-x-er	4	0	10	323	0.00	0.1507	0.0490
fluid	9	26	4	537	0.00	2.9376	0.2083
german	2	7	13	1000	0.00	0.8813	0.0114
german-numb	2	7	17	1000	0.00	0.8813	0.0074
glass1	7	9	0	214	0.00	2.1765	0.1482
glass2	2	9	0	163	0.00	0.9967	0.0894

(*Continued on next page*.)

Table 12. (*Continued*).

Dataset	No. of classes	No. of continuous	No. of symbolic	No. of examples	% Missing	Class entropy	Median UC
heart	2	13	0	270	0.00	0.9911	0.1109
hepatitis	2	6	13	155	0.05	0.7346	0.0808
hypothyroid	2	7	18	3163	0.06	0.2767	0.0071
ionosphere	2	34	0	351	0.00	0.9418	0.1060
iris	3	4	0	150	0.00	1.5850	0.4655
krkopt	18	0	6	8056	0.00	3.5042	0.0500
led24	10	0	24	3200	0.00	3.3207	0.0007
lymphography	4	0	18	148	0.00	1.2277	0.1199
monk1	2	0	6	432	0.00	1.0000	0.0000
monk2	2	0	6	432	0.00	0.9136	0.0047
monk3	2	0	6	432	0.00	0.9978	0.0022
new-thyroid	3	5	0	215	0.00	1.1851	0.3409
nursery	5	0	8	12960	0.00	1.7165	0.0122
optdigits	10	64	0	5620	0.00	3.3218	0.0751
page-blocks	5	10	0	5473	0.00	0.6355	0.1525
parity5-5	2	0	10	1024	0.00	1.0000	0.0000
pima-ind-diabetes	2	8	0	768	0.00	0.9331	0.0354
processed.cleveland-2	2	6	7	303	0.00	0.9951	0.1130
processed.cleveland-4	5	6	7	303	0.00	1.8459	0.0773
processed.hungarian-2	2	6	7	294	0.19	0.9431	0.0627
processed.hungarian-4	5	6	7	294	0.19	0.9431	0.0627
processed.switzerland-2	2	6	7	123	0.16	0.3471	0.0680
pyrimidines	2	54	0	6996	0.00	1.0000	0.0664
quisclas	3	18	0	5891	0.00	1.5579	0.0479
satimage	6	36	0	6435	0.00	2.4833	0.1792
segmentation	7	19	0	2310	0.00	2.8074	0.2047
sick	2	7	22	3772	0.05	0.3324	0.0054
sick-euthyroid	2	7	18	3163	0.06	0.4452	0.0149
sonar	2	60	0	208	0.00	0.9967	0.0429
soybean-large	19	0	35	683	0.10	3.8355	0.1201
soybean-small	4	35	0	47	0.00	1.9558	0.0535
tic-tac-toe	2	0	9	958	0.00	0.9310	0.0146
titanic	2	0	3	2201	0.00	0.9077	0.0628
triazines	2	120	0	52264	0.00	1.0000	0.0065
tumor-LOI	22	0	17	339	0.04	3.6437	0.0506
vehicle	4	18	0	846	0.00	1.9991	0.0667
vote	2	0	16	435	0.00	0.9623	0.2329
waveform40	3	40	0	5000	0.00	1.5849	0.0014
waveform-21	3	21	0	5000	0.00	1.5848	0.1442
wdbc	2	30	0	569	0.00	0.9526	0.2281
wpbc	2	32	0	198	0.00	0.6730	0.0378
yeast	10	8	0	1484	0.00	2.4904	0.0451

Table 13. Error correlation: Models constructed on the full training set.

Linear Discriminant:

```
IF -2.083 +11.100*LgE -9.234*LgREA +0.625*LgREC +0.334*PSA
          -1.166*PMV +5.388*CE    +0.556*NCE   -2.091*MedUC >0
THEN Low Error Correlation
ELSE High Error Correlation
(Errors on Training Set: 11 / 53)
```

Ltree:

```
[]L(x) <= 0.225
|       High Error Correlation (17.00/17.00)
[]L(x) > 0.225
|       []ClassEntropy <= 0.155 (0.996)
|       |       []MissingValues.Relative <= 0.078 (0.015)
|       |       |       []L(x) <= 0.794
|       |       |       |       High Error Correlation (11.00/9.00)
|       |       |       []L(x) > 0.794
|       |       |       |       Low Error Correlation (2.00/2.00)
|       |       []MissingValues.Relative > 0.078 (0.015)
|       |       |       Low Error Correlation (3.00/3.00)
|       []ClassEntropy > 0.155 (0.996)
|       |       Low Error Correlation (20.00/19.00)

L(x) = -2.083 +11.100*LgE -9.234*LgREA +0.625*LgREC +0.334*PSA
               -1.166*PMV +5.388*CE    +0.556*NCE   -2.091*MedUC
(Errors on Training Set: 3 / 53)
```

C50tree:

```
[]ClassEntropy > 0.155 (0.995)
|       Low Error Correlation (22.0/19.0)
[]ClassEntropy <= 0.155 (0.995)
|       []logExamples <= 0.501 (7.745)
|       |       High Error Correlation (25.0/23.0)
|       []logExamples > 0.501 (7.745): Low Error Correlation (6.0/5.0)
(Errors on Training Set: 6 / 53)
```

Table 14. Equiperformance: Models constructed on the full training set.

Linear Discriminant:

```
IF +1.363-41.63*LgE +3.081*LgREA +33.1*LgREC -0.06203*PSA
       -1.875*PMV+14.46*CE-5.155*NCE+3.088*MedUC > 0
THEN Equiperformance
ELSE Non-Equiperformance
(Errors on Training Set: 15 / 80)
```

(*Continued on next page.*)

Table 14. (*Continued*).

Ltree:

```
[]L(x) <= 0.181
|       Non-Equiperformance (33.00/33.00)
[]L(x) > 0.181
|       []medianUC <= 0.0131 (0.007)
|       |       Non-Equiperformance (6.00/6.00)
|       []medianUC > 0.0131 (0.007)
|       |       []ClassEntropy <= 0.144 (0.930)
|       |       |       Equiperformance (12.00/12.00)
|       |       []ClassEntropy > 0.144 (0.930)
|       |       |       []logExamples <= 0.248 (5.778)
|       |       |       |       Equiperformance (13.00/11.00)
|       |       |       []logExamples > 0.248 (5.778)
|       |       |       |       Non-Equiperformance (16.00/11.00)
L(x) = +1.363-41.63*LgE +3.081*LgREA +33.1*LgREC -0.06203*PSA
       -1.875*PMV+14.46*CE-5.155*NCE+3.088*MedUC
(Errors on Training Set: 7 / 80)
```

c50tree:

```
[]logExamples > 0.495 (7.696)
|       Non-Equiperformance (27.0/27.0)
[]logExamples <= 0.495 (7.696):
|       []logExamples <= 0.248 (5.777)
|       |       Equiperformance (22.0/18.0)
|       []logExamples > 0.248 (5.777)
|       |       []logexamplesToClasses <= 0.423 (5.843)
|       |       |       []medianUC <= 0.146 (0.080)
|       |       |       |       Non-Equiperformance (13.0/13.0)
|       |       |       []medianUC > 0.146 (0.080)
|       |       |       |       []symAttrProp <= 0
|       |       |       |       |       Equiperformance (2.0/2.0)
|       |       |       |       []symAttrProp > 0
|       |       |       |       |       Non-Equiperformance (6.0/5.0)
|       |       []logexamplesToClasses > 0.423 (5.843)
|       |       |       []logexamplesToAttr <= 0.495 (4.564)
|       |       |       |       Equiperformance (5.0)
|       |       |       []logexamplesToAttr > 0.495 (4.564)
|       |       |       |       []logExamples <= 0.402 (6.971)
|       |       |       |       |       Non-Equiperformance (2.0/2.0)
|       |       |       |       []logExamples > 0.402 (6.971)
|       |       |       |       |       Equiperformance (3.0/2.0)
(Errors on Training Set: 6 / 80)
```

Table 15. MLP: Models constructed on the full training set.

Ltree:

```
[]log.examplesToAttr <= 0.389 (3.080)
|       Low MLP Performance (8.00/8.00)
[]log.examplesToAttr > 0.389 (3.080)
|       []logExamples <= 0.729  (8.517)
|       |       High MLP Performance (17.00/17.00)
```

(*Continued on next page.*)

Table 15. (*Continued*).

```
|       []logExamples > 0.729  (8.517)
|       |       Low MLP Performance3 (3.00/2.00)
(Errors on Training Set: 1 / 28)
```

Linear discriminant:

```
IF -5.164 -9.334*LgE +27.930*LgREA -3.286*LgREC +0.746*PSA
          -1.802*PMV -16.830*CE    +3.628*NCE   -2.137*MedUC > 0
THEN High MLP Performance
ELSE Low MLP Performance
(Errors  on Training Set: 1/ 28)
```

c50tree:

```
[]log.examplesToAttr <= 0.372 (2.992)
|       Low MLP Performance (8.0/8.0)
[]log.examplesToAttr > 0.372 (2.992)
|       []logExamples <= 0.729 (8.517)
|       |       High MLP Performance (17.0/17.0)
|       []logExamples > 0.729 (8.517)
|       |       Low MLP Performance (3.0/2.0)
(Errors  on Training Set: 1 / 28)
```

Table 16. C50boost: Models constructed on the full training set.

Linear Discriminant:

```
IF +12.68-17.410*LgE +5.428*LgREA -3.136*LgREC +0.346*PSA
          -0.404*PMV    -3.475*CE    -3.849*NCE   +0.708*MedUC > 0
THEN TIE
ELSE c50boost
(Errors on Training Set: 5 / 78)
```

Ltree:

```
[]logExamples <= 0.652 (8.427)
|       TIE (64.00/58.00)
[]logExamples > 0.652 (8.427)
|       []normalizedEntropy <= 0.443 (0.451)
|       |       TIE (2.00/2.00)
|       []normalizedEntropy > 0.443 (0.451)
|       |       c50boost (12.00/12.00)
(Errors on Training Set: 6 / 78 )
```

C50tree:

```
[]logExamples <= 0.639 (8.337)
|       TIE (64.0/58.0)
[]logExamples > 0.639 (8.337):
|       []normalizedEntropy <= 0.436 (0.445)
|       |       TIE (2.0/2.0)
|       []normalizedEntropy > 0.436 (0.445)
|       |       c50boost (12.0/12.0)
(Errors on Training Set: 6 / 78)
```

Acknowledgments

João Gama acknowledges the financial support given by the FEDER, and the project Adaptive Learning Systems—ALES (POSI / SRI / 39770 / 2001).

Notes

1. Remember that high values of the coefficient correspond to similar distributions, low values to orthogonal.
2. A high *MedUC* corresponds to a relatively easy learning problem since the individual attributes would carry substantial information about the class. On the other hand a low *MedUC* would mean a more difficult learning problem.
3. There is only one dataset that does not obey this observation.
4. The default error is the error produced by systematically predicting the majority class.
5. The splits in all the decision tree models are given in the normalized scale of the variables, on which the algorithms are applied; in the parentheses we give the actual value associated with the split.

References

Agresti, A. (1990). *Categorical Data Analysis*. John Wiley and Sons: Series in Probability and Mathematical Statistics.

Aha, D. (1992). Generalizing from case studies: A case study. In D. Sleeman & P. Edwards (Eds.), *Proceedings of the 9th International Machine Learning Conference* (pp. 1–10). Morgan Kaufman.

Ali, K., & Pazzani, M. (1996). Error reduction through learning multiple descriptions. *Machine Learning, 24*, 1996.

Bay, S., & Pazzani, M. (2000). Characterizing model errors and differences. In P. Langley (Ed.), *Proceedings of the 17th International Conference on Machine Learning* (pp. 49–56). Morgan Kaufman.

Bensusan, H. (1999). *Automatic Bias Learning: An Inquiry Into the Inductive Basis of Induction*. Doctoral dissertation, University Of Sussex, Cognitive Science.

Bock, H., & Diday, E. (2000). *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag.

Brazdil, P., Carlos, S., & Costa, J. (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning, 50*, 251–277.

Brodley, C. (1994). *Recursive Automatic Algorithm Selection for Inductive Learning*. Doctoral dissertation, University of Massachusetts.

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

Cristianini, N., & Shawe-Taylor, J. (2002). *An Introduction to Support Vector Machines*. Cambridge University Press.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern Classification and Scene Analysis*. John Willey and Sons.

Gama, J., & Brazdil, P. (1995). Characterization of classification algorithms. In C. Pinto-Ferreira & N. Mamede (Eds.), *Proceedings of the 7th Portugese Conference in AI, EPIA 95* (pp. 83–102). Springer-Verlag.

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

Hirschberg, D. S., Pazzani, M. J., & Ali, K. M. (1994). *Average Case Analysis of k-CNF and k-DNF Learning Algorithms*, vol. II: Intersections Between Theory and Experiment, 15–28. MIT Press.

Kalousis, A., & Hilario, M. (2001a). Feature selection for meta-learning. In D. Cheung, G. Williams & Q. Li (Eds.), *Proceedings of the 5th Pasific Asia Conference on Knowledge Discovery and Data Mining* (pp. 222–233). Springer.

Kalousis, A., & Hilario, M. (2001b). Model selection via meta-learning: A comparative study. *International Journal On Artificial Intelligence Tools*, *10*.

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

Langley, P., & Sage, S. (1999). Tractable average-case analysis of naive Bayesian classifiers. In I. Bratko & S. Dzeroski (Eds.), *Proceedings of the 16th International Conference on Machine Learning* (pp. 220–228). Morgan Kaufmann.

Langley, P., & Iba, W. (1993). Average-case analysis of a nearest neighbor algorithm. In R. Bajcsy (Ed.), *Proceedings of the 13th International Joint Conference on AI* (pp. 889–894). Morgan Kaufmann.

METAL project (2002). http://www.metal-kdd.org.

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. In S. Lange & K. Satoh (Eds.), *Proceedings of the 5th International Conference on Discovery Science 2002* (pp. 141–152). 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. In P. Langley (Ed.), *Proceedings of the 17th International Conference on Machine Learning* (pp. 743–750). Morgan Kaufman.

Quinlan, J. (1992). *C4.5: Programs for Machine Learning*. Morgan Kaufman Publishers.

Quinlan, R. (2000). http://www.rulequest.com/see5-info.html.

Scheffer, T. (2000). Average-case analysis of classification algorithms for boolean functions and decision trees. In H. Arimura, S. Jain & A. Sharma (Eds.), *Proceedings of the 11th International Conference Algorithmic Learning Theory* (pp. 194–208). Springer Verlag, LNCS 1968.

Soares, C., & Brazdil, P. (2000). Zoomed ranking: Selection of classification algrorithms based on relevant performance information. In D. Zighed, J. Komorowski & J. Zytkow (Eds.), *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 126–135). Springer.

Todorovski, L., Blockeel, H., & Dzeroski, S. (2002). Ranking with predictive clustering trees. In T. Elomaa, H. Mannila and H. Toivonen (Eds.), *Proceedings of the 13th European Conference on Machine Learning* (pp. 444–455). Springer.

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

Tumer, K., & Ghosh, J. (1995). Classifier combining: Analytical results and implications. In *In AAAI-95 - Workshop in Induction of Multiple Learning Models*.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*, 241–259.

Received April 23, 2003
Revised October 8, 2003
Accepted October 8, 2003
Final manuscript November 20, 2003