Meta-data: Characterization of Input Features for Meta-learning

Ciro Castiello

Lecture Notes in Computer Science

Meta-data: Characterization of Input Features for Meta-learning

Ciro Castiello, Giovanna Castellano, and Anna Maria Fanelli

CILab - Computational Intelligence Laboratory
Computer Science Department, University of Bari
Via E. Orabona, 4 - 70126 Bari, Italy
{castellano,castiello,fanelli}@di.uniba.it
http://www.di.uniba.it/~cilab/

Abstract. Common inductive learning strategies offer the tools for knowledge acquisition, but possess some inherent limitations due to the use of fixed bias during the learning process. To overcome limitations of such *base-learning* approaches, a novel research trend is oriented to explore the potentialities of *meta-learning*, which is oriented to the development of mechanisms based on a dynamical search of bias. This could lead to an improvement of the base-learner performance on specific learning tasks, by profiting of the accumulated past experience. As a significant set of I/O data is needed for efficient base-learning, appropriate meta-data characterization is of crucial importance for useful meta-learning. In order to characterize meta-data, firstly a collection of meta-features discriminating among different base-level tasks should be identified. This paper focuses on the characterization of meta-data, through an analysis of meta-features that can capture the properties of specific tasks to be solved at base level. This kind of approach represents a first step toward the development of a meta-learning system, capable of suggesting the proper bias for base-learning different specific task domains.

1 Introduction

Common learning procedures proposed in the realm of machine learning are characterized by a fixed form of bias (base-learning paradigm), that forces the learner to specialize in a limited domain. This is a common circumstance in the base-learning paradigm, where different learning models perform well in some context, but appear to be inadequate in others. Meta-learning is a novel field of investigation intended for overcoming the limitations of traditional base-learning approaches by performing a dynamical search of bias [1], [2]. In other words, meta-learning is aimed at improving, by means of learning, the performance of a base-learner.

Different strategies can be cast within the meta-learning paradigm and several approaches have been proposed in the literature (see [2] for a comprehensive survey). A large amount of the research efforts have been addressed to investigate methods for adaptively selecting a particular learning model (among a pool of

V. Torra et al. (Eds.): MDAI 2005, LNAI 3558, pp. 457–468, 2005.

candidates) that could prove to be best suited for a given task or an application domain [3], [4]. This kind of *model selection* approach moves from the assumption that every single learning algorithm can be considered as a form of bias. In this sense, meta-learning is achieved by studying the performance of several models when applied to different problems. A number of research projects have produced results in this area, prominent examples include STATLOG [5] and METAL [6] projects.

Our peculiar conception of meta-learning is aimed to dynamically adjust the bias characterising a specific learning model. This kind of approach, distinct from selection and combination of models, belongs to a potential avenue of research claiming that a learning algorithm should be able to change its internal mechanisms according to the task under analysis [2], [7]. In this way, a meta-learning strategy is centred on the extensive analysis of the special capabilities of a single base-learner, favouring the continuous accumulation of meta-knowledge useful for indicating the most appropriate form of bias for each problem. This means that a meta-learner should improve the learning performance of the base-learner by exploiting experience accumulated on previous tasks, which has to be retained as a form of meta-knowledge.

Of course, as for any standard base-learner, the success of a meta-learner is greatly dependent upon the quality of the (meta-)data chosen; especially it depends heavily on the input features used to describe the problem. Thus, a fundamental problem to be addressed in meta-learning is how to find appropriate meta-features that capture the properties of specific tasks to be solved through base-learning. Various strategies for defining these meta-features have been proposed [5], [8], [9]. Most of them are oriented to describe base-level tasks by means of a set of measures, including general measures (e.g. number of attributes, number of classes), statistical measures (e.g. mean and variance of numerical attributes) and information theory based measures (e.g. average joint entropy of classes and attributes). The description of a dataset in terms of its statistical/information properties appeared for the first time within the framework of the STATLOG project [5]. However, to date, there is no consensus on how good meta-features should be chosen.

In this paper, we proceed with a careful analysis of the most commonly employed meta-features, discussing their intrinsic properties in a systematic way. We suggest guidelines to select the most informative measures and we introduce new features that are transformations of existing ones. As a result of our analysis, a set of selected measures is suggested, that can be conveniently used as meta-features to compile a meta-dataset.

2 Meta-data Characterization

The idea underlying meta-learning is to identify a meta-knowledge apparatus that could be useful in supporting the search for a suitable bias when applying a base-learner to solve a specific task. Practically, the ordinary base-learning tasks are tackled by a learner $\mathcal{L}_{base}$ on the basis of a dataset $\mathcal{T}_{base}$, containing observational data related to the problem at hand. We assume that $\mathcal{T}_{base}$ can be expressed as a set of K samples, each of them can be decomposed in a couple of input-output vector, respectively indicated by $\mathbf{x}_k = (x_{k1}, \ldots, x_{km})$ and $\mathbf{y}_k = (y_{k1}, \ldots, y_{kn})$, namely:

$$\mathcal{T}_{base} = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{K} \tag{1}$$

The knowledge derived after base-learning concerns the particular task under study. The meta-learning process, instead, is performed by a learner $\mathcal{L}_{meta}$ which starts from the analysis of meta-data, or equivalently a meta-dataset $\mathcal{T}_{meta}$, and ends up by formulating a meta-knowledge that describes the correlation between the specific task (described by a set of meta-features) and the proper bias to be used during base-learning of the task. $\mathcal{T}_{base}$ and $\mathcal{T}_{meta}$ have a different nature, that leads to the main distinction between base-learning and meta-learning. Whilst the dataset $\mathcal{T}_{base}$ is simply drawn up on the basis of the available data, representing the task tackled by the base-learner, the meta-dataset $\mathcal{T}_{meta}$ has to be carefully built up by analyzing the characteristics of each single task and exploiting the accumulated past experience.

The goal of meta-learning is to let the meta-learner discover the relationships between categories of tasks and base-learner bias configurations. Hence, in order to arrange the meta-data, particular attention must be paid in defining the distinguishing meta-features of every task. Successively, the identified meta-features should be correlated with the most appropriate bias employed by $\mathcal{L}_{base}$ to efficaciously solve a particular class of tasks.

The definition of a proper bias involves an analysis of the free parameters engaged in the particular base-learning algorithm adopted, and it is out of the scope of this paper. This paper focuses essentially on the characterization of meta-features, which is independent on the particular base-learner adopted. In the following section we provide a characterization of the meta-features in terms of the most common measures found in the literature.

3 Characterization of Meta-features

In order to collect meta-features about a given learning task, it is necessary to perform an analysis of the information embedded inside a dataset associated to it. Obviously, this means that a dataset, besides banally providing information concerning the data (related to mean values, standard deviations, and so on), could be able to furnish a somewhat precise knowledge of the particular task underlying the dataset at hand. Therefore, meta-features can be conceived as specific collections of morphological characteristics of a dataset, that jointly affect the performance of a learning algorithm when it is applied to the particular task represented by the dataset. The main assumption is that the information codified in the meta-features should exhibit some kind of general hints (related, for instance, to the task complexity), other than the self-evident information content embedded in the particular dataset at hand (simply related to the data configuration).

Generally speaking, for the purpose of meta-learning, an adequate set of meta-features must satisfy mainly two basic conditions. Firstly, it should prove to be useful in determining the relative performance of individual learning algorithms. Secondly, it should not be too difficult to calculate. A number of measures have been proposed to characterize data for meta-learning and a common practice has been established in focusing over general, statistical and information-theoretic measures (see for example the meta-features employed in the STATLOG [5] and the METAL [6] projects). In the following we describe in more details the most frequently adopted meta-features.

3.1 General Meta-features

General meta-features include general information related to the dataset at hand and, to a certain extent, they are conceived to measure the complexity and/or the size of the underlying problem. The following general meta-features can be distinguished:

Number of observations: it represents the total number K of samples in the dataset (i.e. the cardinality of the dataset $\mathcal{T}_{base}$ defined in (1)).

Number of attributes: it represents the total number m of attributes in the dataset (i.e. the dimensionality of the input vector $\mathbf{x}_k$ in (1)).

Number of output values: it represents the total number n of output values in the dataset (i.e. the dimensionality of the output vector $\mathbf{y}_k$ in (1)).

Dataset dimensionality: it represents the ratio between the number of attributes and the number of observations constituting the dataset, i.e. $dim_{data} = \frac{m}{K}$.

3.2 Statistical Meta-features

It is straightforward to make use of standard statistical measures to describe the numerical properties of a distribution of data. By means of particular statistical meta-features it could be possible also to define the properties of the numerical subspace where a task domain evolves. Therefore, statistical meta-features can be employed to take into account the number of properties which enable a learner to discriminate the degree of correlation of numerical attributes and estimate their distribution. The following statistical meta-features can be distinguished:

Standard deviation: this quantity estimates the dispersion of a random variable $X = x_1, x_2, \ldots, x_K$ with respect to its mean $\overline{X} = 1/K \sum_{k=1}^{K} x_k$. It is computed as $std_X = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (x_k - \overline{X})^2}$. In (1), the components of each sample represent particular instances of random variables. Therefore, from being $\mathbf{x}_k = (x_{k1}, \ldots, x_{ki}, \ldots, x_{km})$, a number of m random variables in the dataset assume the values $(x_{1i}, \ldots, x_{Ki})$, for $i = 1, \ldots, m$. As observed before, since the standard deviation depends on the size of the dataset and on the range of the data, it should be convenient to include the minimum and the maximum value of std_X as measures of a single dataset, or equivalently the mean standard deviation evaluated over all the attributes of the dataset, that is:

$$\overline{std_X} = \frac{1}{m}\sum_{i=1}^{m}(std_{X_i}), \tag{2}$$

Coefficient of variation: it evaluates the normalization of the standard deviation of a random variable X with respect to its mean value: $VarCoeff_X = \frac{std_X}{\overline{X}}$. As a measure of the coefficient of variation of an entire dataset, the average of $VarCoeff_{X_i}$ over all the m numerical attributes could be considered:

$$\overline{VarCoeff_X} = \frac{1}{m}\sum_{i=1}^{m}(VarCoeff_{X_i}). \tag{3}$$

It should be pointed out that the standard deviation is independent from the scaling of an attribute, while the coefficient of variation is not influenced by translations. Indeed, random variables X and αX possess the same coefficient of variation, while X and $X + \alpha$ have the same standard deviation. These properties should be taken into account whenever the meta-features should help a learner to discriminate for similar translation and/or scaling.

Covariance: the covariance extends the variance concept to the bidimensional case. In fact, it expresses the linear relationship between two random variables $X = x_1, \ldots, x_K$ and $Y = y_1, \ldots, y_K$, defined as:

$$Cov(X,Y) = \sum_{k=1}^{K}\frac{(x_k - \overline{X})(y_k - \overline{Y})}{K-1}. \tag{4}$$

In our dataset formalization, the covariance could be evaluated for each pair of input variables, $\mathbf{x}_k$, $\mathbf{x}_l$, for $k \neq l$. As a measure of the covariance of an entire dataset, the average of $Cov(X,Y)$ over all the possible distinct pairs of numerical attributes could be considered.

Linear correlation coefficient: correlation analysis attempts to measure the strength of a relationship between two random variables X and Y. The linear correlation coefficient shows the linear association strength between X and Y by means of a single value. The coefficient of linear correlation can be estimated by the following formula:

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sqrt{std_X std_Y}}. \tag{5}$$

The values of $\rho_{X,Y}$ range in the interval $[-1,1]$. The linear correlation coefficient could be evaluated for each pair of input variables. As a coefficient of the entire dataset linear correlation, the mean of the absolute values of the correlations between all distinct pairs of attributes could be evaluated.

Skewness: it measures the lack of symmetry in the distribution of a random variable X. Negative skewness values indicate data that are skewed left, while positive skewness values denote data that are skewed right. By skewed left (respectively, right) we mean that the left tail is heavier than the right tail (respectively, the right tail is heavier than the left one). An estimator of the skewness is represented by the third moment of the distribution of a random variable X, divided by the third power of standard deviation, i.e. $Skew_X = \frac{1}{std_X^3} \frac{\sum_{k=1}^{K}(x_k - \overline{X})^3}{K}$. As a measure of the skewness of an entire dataset, the average of $Skew_{X_i}$ over all the m numerical attributes could be considered:

$$\overline{Skew_X} = \frac{1}{m} \sum_{i=1}^{m} (Skew_{X_i}). \tag{6}$$

Kurtosis: it measures the peakedness in the distribution of a random variable X. The kurtosis can be estimated by the ratio of the fourth moment of the distribution of X to the fourth power of the standard deviation, i.e. $Kurt_X = \frac{1}{std_X^4} \frac{\sum_{k=1}^{K}(x_k - \overline{X})^4}{K}$. As a measure of the kurtosis of an entire dataset, the average of $Kurt_{X_i}$ over all the m numerical attributes could be considered:

$$\overline{Kurt_X} = \frac{1}{m} \sum_{i=1}^{m} (Kurt_{X_i}). \tag{7}$$

1-D variance fraction coefficient: this coefficient indicates the relative importance of the largest eigenvalue of the attribute covariance matrix and it measures the representation quality of the first principal component. In principal component analysis (PCA), methods for transforming a given dataset into a new dataset of reduced dimensions are studied, to concentrate the information about the differences between samples into a small number of dimensions [10]. In the PCA context, the first principal component is an axis in the direction of maximum variance; the variance of this principal axis is given by the largest eigenvalue of the covariance matrix of the original data. The fraction of the total variance retained in the 1-dimensional space can be computed as the ratio between the largest eigenvalue λ_1 of the covariance matrix S (whose elements are the covariances of the considered random variables) and the sum of all its eigenvalues:

$$Frac1 = \frac{\lambda_1}{trace(S)} \tag{8}$$

where $trace(S)$ is the trace of the covariance matrix S. In order to evaluate this meta-feature for a dataset formalized as in (1), the covariance matrix has to be extracted from the attribute matrix, whose elements are x_{ki}, for $k = 1, \ldots, K$ and $i = 1, \ldots, m$.

3.3 Information-Theoretic Meta-features

Information-theoretic meta-features are particularly appropriate to describe discrete (categorical) attributes, but they also fit continuous (numerical) ones.

Firstly, we define the entropy of a discrete random variable X as a measure of the randomness in the variable, evaluated by:

$$H(X) = -\sum_{i=1}^{n} q_i \log_2(q_i) \tag{9}$$

where $q_i = p(X = x_i)$ denotes the probability that X assumes the ith value x_i, for $i = 1, \ldots, n$. Conventionally, logarithms are to base 2 and entropy is then said to be measured in units called "bits" (binary information units). The highest entropy value is $\log_2(n)$, when n distinct values of X could appear with an equal probability. Therefore, the entropy values range in the interval $[0, \log_2(n)]$ and they are strictly connected with the degree of uniformity of the distribution of a variable.

The entropy definition given in (9) assumes that the random variable X is characterized by discrete values $(x_1, \ldots, x_n)$. In order to deal with continuous real-valued random variables, we should replace the summation term with an integral and the probabilities q_i with the probability density function $p_{\mathcal{D}}(\mathbf{x})$, being $\mathcal{D}$ the distribution of probability employed to draw the instances of X from an instance space. Whenever the distribution probability is unknown, in order to apply the formula (9), it is necessary to discretize the numerical data. A number of n equally spaced intervals have to be determined and then the relative frequency histogram for the i-th interval can be adopted as an estimate of q_i. It can be observed that, when dealing with classification tasks, the dataset expressed in (1) can be considered as a collection of values for m continuous random variables (corresponding to the m input attributes), together with the n values of a discrete random variable (corresponding to the output class, which can assume n distinct and discrete values). For the sake of simplicity, in the following we are going to describe the information-theoretic meta-features by referring to a classification problem, assuming the discretization process performed over the input attribute values.

Normalized class entropy: the entropy value $H(C)$ of a class variable C indicates how much information is necessary to specify one class. The value $H(C)$ can be evaluated using (9), where $q_i \equiv \pi_i$ defines the prior probability for a class (being π_i the relative frequency of occurrence of a particular class value). When we suppose that each class value in a dataset has the same probability to appear, then the theoretical maximum value for the class entropy is $\log_2(n)$. Therefore the normalized entropy can be computed as:

$$H(C)_{norm} = \frac{H(C)}{\log_2(n)} = -\frac{\sum_{i=1}^{n} \pi_i \log_2(\pi_i)}{\log_2(n)}. \tag{10}$$

Normalized attribute entropy: the attribute entropy value $H(X)$ of a random variable (which could be represented by one of the m discretized attribute input) measures the information content related to the values that X may assume. Since $\log_2(n)$ represents the maximum value for the attribute entropy, the normalized entropy can be computed as $H(X)_{norm} = \frac{H(X)}{\log_2(n)} =$

$-\frac{\sum_{i=1}^{n} q_i \log_2(q_i)}{\log_2(n)}$. As a measure of the attribute entropy of an entire dataset, the average of $H(X_i)_{norm}$ over all the m input attributes could be considered:

$$\overline{H(X)_{norm}} = \frac{1}{m}\sum_{i=1}^{m}(H(X_i)_{norm}). \tag{11}$$

Joint entropy of class and attribute: it measures the total entropy of the combined system of variables, i.e. the pair of variables (C, X), which could be represented by a class variable and one of the m discretized input attributes, respectively. If p_{ij} denotes the joint probability of observing the i-th value of attribute X and the j-th class value, the joint entropy is defined as $H(C, X) = -\sum_{ij} p_{ij} \log_2(p_{ij})$. As a measure of the joint entropy of an entire dataset, the average of $H(C, X_i)$ over all the m input attributes could be considered:

$$\overline{H(C,X)} = \frac{1}{m}\sum_{i=1}^{m}(H(C, X_i)). \tag{12}$$

Mutual information of class and attribute: it measures the common information shared between two random variables C and X. If C and X respectively represent a class variable and one of the m discretized input attributes, then the meta-feature measures the information conveyed by the attribute X about a class value and describes the reduction of uncertainty for C due to the knowledge of X. The mutual information of class and attribute can be evaluated by $MI(C, X) = H(C) + H(X) - H(C, X) = \sum_{ij} p_{ij} \log_2(\frac{p_{ij}}{\pi_i q_j})$. As a measure of the joint entropy of an entire dataset, the average of $MI(C, X_i)$ over all the m input attributes could be considered:

$$\overline{MI(C,X)} = \sum_{i=1}^{m} MI(C, X_i). \tag{13}$$

Equivalent number of attributes: when we refer to classification tasks, the information required to specify the class is $H(C)$ and no classification scheme can be completely successful unless it provides at least $H(C)$ bits of useful information. This information has to come from the attributes taken together, and in the simplest (even if often unrealistic) case that all attributes are independent, we would have:

$$MI(C, X) = MI(C, X_1) + \ldots + MI(C, X_m).$$

In this case, we could count up how many attributes would be required, on average, by taking the ratio between the class entropy $H(C)$ and the average mutual information $\overline{MI(C,X)}$. The meta-feature expressing the equivalent number of attributes, therefore, can be evaluated by:

$$EN_{attr} = \frac{H(C)}{\overline{MI(X,C)}}. \tag{14}$$

This meta-feature furnishes a rough information about the complexity of a problem, specifically it indicates if the number of attributes in a given dataset is suitable to optimally solve the classification task (under the assumption of independence among attributes).

Noise-signal ratio: it measures the amount of irrelevant information contained in a dataset. If we consider $\overline{MI(C,X)}$ as a measure of useful information about class, and $\overline{H(X)} - \overline{MI(C,X)}$ as a measure of non-useful information (where $\overline{H(X)}$ represents the mean of the attribute entropy), the meta-feature can be evaluated by:

$$NS.ratio = \frac{\overline{H(X)} - \overline{MI(C,X)}}{\overline{MI(C,X)}}. \tag{15}$$

The above listed ensemble of measures represents a set of candidates which could be employed as meta-features in a meta-learning task.

4 Selection of Meta-features

Despite many works in literature explicitly admit that the choice for particular input meta-features does not necessarily relies upon preventive analyses (see, for instance, [8]), it is our opinion that, in order to choose the meta-features to be employed for discriminating across different tasks, a deeper analysis is necessary. Moving from these considerations, in this section we revisit the above listed meta-features on the basis of an analysis having a twofold goal: (i) deepening the understanding of the employed meta-data and (ii) taking into account the particular application domain.

Since we are interested in discriminating among different learning problems, we resolved to extract numerical meta-features from the available observational data related to each task. Obviously, these kinds of information could provide both knowledge connected to the data alone and the different ways they have been collected, and knowledge which more directly concerns the underlying task and its intrinsic “difficulty”. Even if there is no doubt that the process of learning a particular task could become much more complex in dependence of the configuration of the adopted dataset (which could reveal itself to be inadequate for the presence of excessive noise, missing values, and so on), we consider that a meta-learning system should be able to discriminate among different problems in terms of their inherent characterization, renouncing to taking account also of data arrangement. This represents, therefore, a further assumption for our analysis: the meta-features employed to characterize a learning problem should furnish pieces of information as unrelated as possible to the dataset configuration, and should attempt to capture, instead, the actual complexity of the task under consideration. In this sense, we could undertake a brief revisitation of the ensemble of meta-features described in section 3.

Among the general meta-features, we suggest the only use of *number of attributes* and *number of output values*, since the *number of observations* and the *dataset dimensionality* provide information related to the dataset configuration.

Moreover, the *number of output values* could be discarded if the application domain involves only concept learning tasks, each one characterized by two output values.

As concerns the statistical meta-features, we observe that, while the *mean standard deviation* and the *mean coefficient of variation* appear to be strongly related to low-level information pertaining to dataset configuration, on the other hand *covariance*, *linear correlation coefficient*, *skewness* and *kurtosis*, even if they evaluate purely statistical information, could be computed in a slightly different way, in order to furnish some kind of high-order task information. In fact, the computation of statistical meta-features is generally performed by employing only the input attribute variables of a dataset; it could be possible to obtain some more information if we could exploit also the knowledge related to the class membership[1]. Therefore, we suggest to discard the *mean standard deviation* and the *mean coefficient of variation*, and to proceed in evaluating the remaining statistical meta-features on a "*per class*" basis, that is by separately computing the meta-features values for each group of samples belonging to different output classes, and then retaining the average value (over the number of classes) as the final meta-feature value. A further consideration regards the *mean covariance*, which determines a value strongly dependent on the range of the data. For the sake of generality, we decided to discard such a feature, considering that a similar information can be obtained by the *mean linear correlation coefficient*, with the additional advantage that this results to be independent on the ranges of input data (in fact its range is [-1,1]). Finally, the *skewness* and the *kurtosis* lose their discriminating character when evaluated on uniformly distributed data, thus such features can be discarded if the application domain involves data with uniform distribution.

In order to analyze the information-theoretic meta-features, it is important to recall that, from (9), the entropy values of any random variable X range in the interval $[0, \log_2(n)]$, being n the number of the distinct values that X may assume. Higher and lower values of $H(X)$ depend on the more or less uniformity in the distribution of the values of X. Therefore, even if the information provided by the *normalized class entropy* and the *mean normalized attribute entropy* could potentially be useful for deriving some kind of high-order task knowledge, it is unsuitable for application domains involving dataset characterized by the uniform distribution. As concerning the *mean mutual information of class and attribute*, we considered that the average computed in (13) could waste the greater part of the information content embedded in this meta-feature. Therefore, we propose to refer to the maximum value of the c/a mutual information, instead of considering the average over all the m input attributes:

$$maxMI(C, X) = \max(MI(C, X_1), \ldots, MI(C, X_m)). \tag{16}$$

[1] Obviously, this kind of approach is applicable only for classification tasks.

Table 1. The set of selected meta-features

Meta-feature	Notation
General meta-features	
Number of attributes	m
Number of output values	n
Statistical meta-features	
Mean linear corr. coeff. (on class basis)	$\overline{\rho^{class}_{X,Y}}$
Mean skewness (on class basis)	$\overline{Skew^{class}_{X}}$
Mean kurtosis (on class basis)	$\overline{Kurt^{class}_{X}}$
1-D var. fraction coeff. (on class basis)	$Frac1^{class}$
Information-theoretic meta-features	
Normalized class entropy	$H(C)_{norm}$
Mean normalized attribute entropy	$\overline{H(X)_{norm}}$
Max. norm. mutual info. c/a	$maxMI_{norm}(C, X)$
Equivalent number of attributes	$EN.attr$
Noise-signal ratio	$NS.ratio$

The obtained value could be normalized for the sake of generality by the following formula:

$$maxMI_{norm}(C, X) = \frac{maxMI(C, X)}{\min(H(C), H(X_l))}, \tag{17}$$

where $H(C)$ is the class entropy, $H(X_l)$ is the attribute entropy evaluated for the l-th input attribute and $l = \arg\max_i(MI(C, X_i))$. In this way, the new value computed for such meta-feature expresses the portion of information that would be retained if the task under consideration were analyzed only on the basis of the most relevant (in sense of information content) input attributes.

Table 1 summarizes the set of selected meta-features that we considered suitable for meta-learning applications.

5 Conclusions and Future Works

Similarly to what happens with the employment of any standard machine learning algorithm, the success of a meta-learning strategy greatly depends on the quality of the (meta-)data to be used during learning. In order to characterize meta-data, firstly a collection of significant meta-features, discriminating among different learning tasks, is to be identified. In this paper we have presented a characterization of meta-features, through a systematic analysis of the most frequently used measures. As a result of such analysis, new features that are transformations of existing ones have been proposed, and some guidelines have been delineated to select the most informative ones. The selected set of meta-features represents a first step toward the design of a meta-learner, capable of suggesting the proper bias for base-learning different specific task domains. In order to develop such a meta-learning system, our current research activity is addressed to evaluating the selected meta-features appropriateness in characterizing different learning domains, each one including groups of related tasks. The aim is to relate each set of meta-feature values (properly derived from a dataset representing a base-learning task) to a bias configuration. Particulary, the bias configuration should be identified as the base-learner parameter setting yielding the best performance results for the considered task. Such pieces of information, gathered from different base-learning experiences, will permit the compilation of a proper set of meta-data, useful to derive meta-knowledge through a meta-learning process.

References

1. Giraud-Carrier, C., Vilalta, R. and Brazdil, P.: Introduction to the special issue on meta-learning. Machine Learning, **54** (2004) 187–193.
2. Vilalta, R., and Drissi, Y.: A perspective view and survey of meta-learning. Journal of Artificial Intelligence Review **18**(2) (2002) 77–95.
3. Merz, C.J.: Dynamical selection of learning algorithms. In: Fisher, D., Lenz, H.J. (eds.): Learning from data: Artificial Intelligence and Statistics, Springer-Verlag, Berlin Heidelberg New York (1995).
4. Brazdil, P.B.: Data transformation and model selection by experimentation and meta-learning. Proceedings of the ECML98 Workshop (1998) 11–17.
5. Michie, D., Spiegelhalter, D., and Taylor, C.: Machine learning, neural and statistical classification. Ellis Horwood, New York (1994).
6. Kalousis, A. and Hilario, M.: Model selection via meta-learning: a comparative study. Proceedings of the 12th International IEEE Conference on Tools with AI. IEEE Press (2000).
7. Soares, C., Brazdil, P.B. and Kuba, P.: A meta-learning method to select the kernel width in support vector regression. Machine Learning, **54** (2004) 195–209.
8. Brazdil, P., Soares, C., and Costa, J.: Ranking learning algorithms: Using IBL and meta-learning on accuracy & time results. Machine Learning **50**(3) (2003) 251–277.
9. Linder, C. and Studer, R.: AST: Support for Algorithm Selection with a CBR Approach. Proceedings of the 16th International Conference on Machine Learning, Workshop on Recent Advances in Meta-Learning and Future Work. (1999).
10. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995).