A Perspective View And Survey Of Meta-Learning

Ricardo Vilalta and **Youssef Drissi**
IBM T.J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY., 10532 U.S.A.
Email: vilalta@us.ibm.com, youseffd@us.ibm.com

Abstract. The first part of this paper provides a perspective view of meta-learning in which the goal is to build self-adaptive learning algorithms. The idea is to improve the learning bias dynamically through experience by the continuous accumulation of meta-knowledge. The second part of this paper provides a survey of meta-learning as reported by the machine-learning literature. We find that different researchers hold different views of what the term meta-learning exactly means. Despite different views and research lines, a question remains constant: how can we exploit knowledge about learning (i.e., meta-knowledge) to improve the performance of learning algorithms? Clearly the answer to this question is key to the advancement of the field and continues being the subject of intensive research.

Keywords: inductive learning, classification, meta-knowledge.

1. Introduction

Meta-learning studies how learning systems can increase in efficiency through experience; the goal is to understand how learning itself can become flexible according to the domain or task under study. All learning systems work by adapting to a specific environment, which reduces to imposing a partial ordering or bias on the set of possible hypotheses explaining a concept (Mitchell, 1980). Meta-learning differs from *base-learning* in the scope of the level of adaptation: meta-learning studies how to choose the right bias dynamically, as opposed to base-learning where the bias is *fixed* a priori, or user parameterized. In a typical inductive-learning scenario, applying a base-learner (e.g., decision tree, neural network, or support vector machine) over some data produces a hypothesis that depends on the *fixed* bias embedded in the learner. Learning takes place at the base-level because the quality of the hypothesis normally improves with an increasing number of examples. Nevertheless, successive applications of the learner over the same data always produces the same hypothesis, independently of performance; no knowledge is extracted across domains or tasks (Pratt and Thrun, 1997).

Meta-learning aims at discovering ways to dynamically search for the best learning strategy as the number of tasks increases (Thrun, 1998; Rendell et al., 1987B). A computer program qualifies as a learning machine if its performance improves with experience (Mitchell, 1997; Cohen and Feigenbaum, 1989). Experience is best understood as the knowledge gained from the analysis of several tasks; the definition is not limited to the ability to refine a hypothesis after presenting examples that belong to one task. Hence, meta-learning advocates the need for continuous adaptation of the learner at different levels of abstraction. If a base-learner fails to perform efficiently, one would expect the learning mechanism itself to adapt in case the same task is presented again. Thus, learning can take place not only at the example (i.e., base) level, but also at the across-task (i.e., meta) level.

After describing our own perspective view of meta-learning and posing some interesting challenges for future research, this paper provides a survey of the field as reported in the machine-learning literature. Some areas of study that bear close relationship to meta-learning include building a meta-learner of base-learners (Section 4.1), selecting an inductive bias dynamically (Section 4.2), building meta-rules matching task properties with algorithm performance (Section 4.3), inductive transfer and learning to learn (Section 4.4), learning classifier systems (Section 4.5), and others (Section 4.6). Our survey shows how the term meta-learning means differently to different research groups; we find that each of the areas mentioned above covers only a few pieces in the big puzzle conformed by the field of meta-learning. Our ultimate goal is to see the field progressing towards a uniform and coherent view.

This paper is organized as follows. Section 2 gives definitions and background information in classification. Section 3 provides our own perspective view of the nature and potential avenues of research in meta-learning. Section 4 is a survey of meta-learning as reported in the machine-learning literature. Section 5 ends with discussion and conclusions.

2. Preliminaries

Our study is centered on the classification problem exclusively. The problem is to learn how to assign the correct class to each of a set of different objects (i.e., events, situations). A **learning algorithm** L is first trained on a set of pre-classified examples $T_{\text{train}} : \{(\tilde{\mathbf{X}}_{\mathbf{i}}, c_i)\}_{i=1}^{m}$. Each object $\tilde{\mathbf{X}}$ is characterized by features, and can be represented as a vector in an n-dimensional **feature space**, $\tilde{\mathbf{X}} = (X_1, X_2, \cdots, X_n)$. Each feature X_k can take on a different number of values. $\tilde{\mathbf{X}}_{\mathbf{i}}$ is labeled with class c_i according to an unknown target function F, $F(\tilde{\mathbf{X}}_{\mathbf{i}}) = c_i$ (we assume a deterministic target function, i.e., zero-bayes risk). In classification, each c_i takes one of a fixed number of categorical values. T_{train} will consist of independently and identically distributed (i.i.d.) examples obtained according to a fixed but unknown joint probability distribution Φ in the space of possible feature-vectors $\mathcal{X}$. The goal in classification is to produce a hypothesis h that best approximates F, i.e., that minimizes a loss function (e.g., zero-one loss) in the space of feature vectors and classes, $\mathcal{X} \times \mathcal{C}$, according to distribution Φ.

Classification begins when learning algorithm L receives as input a training set T_{train} and conducts a search over a **hypothesis space** $\mathcal{H}_{\mathcal{L}}$ until it finds a **hypothesis** h, $h \in \mathcal{H}_{\mathcal{L}}$, that approximates the true function F. Thus a learning algorithm L maps a training set into a hypothesis, $L : \mathcal{T} \rightarrow \mathcal{H}_{\mathcal{L}}$, where $\mathcal{T}$ is the space of all training sets of size m. The selected hypothesis h can then be used to guess the class of unseen examples.

Learning algorithm L embeds a set of assumptions or **bias** that affects the learning process in two ways: it restricts the size of the hypothesis space $\mathcal{H}_{\mathcal{L}}$, and it imposes an ordering or ranking over all hypotheses in $\mathcal{H}_{\mathcal{L}}$. The bias of a learning algorithm L_A is **stronger** than the bias of another learning algorithm L_B if the size of the hypothesis space[1] considered by L_A is smaller than the size of the hypothesis space considered by L_B (i.e., if $|\mathcal{H}_{\mathcal{L}_\mathcal{A}}| \leq |\mathcal{H}_{\mathcal{L}_\mathcal{B}}|$). In this case the bias embedded by L_A conveys more extra-evidential information (Watanabe, 1969) than the bias in L_B, which enables us to narrow down the number of candidate hypotheses estimating the true target concept F. We say the bias of a learning algorithm is **correct** if the target concept is contained in the hypothesis space (i.e., if $F \in \mathcal{H}_{\mathcal{L}}$). An incorrect bias precludes finding a perfect estimate to target concept F.

3. A Perspective View of Meta-Learning

In this section we lay down some definitions and concepts that will be helpful to compare some of the current research directions adopted in meta-learning. Our own view of the field advocates the construction of self-adaptive learners.

In base-learning, the hypothesis space $\mathcal{H}_{\mathcal{L}}$ of a learning algorithm L is fixed. Applying a learning algorithm (e.g., decision tree, neural network, or support vector machine) over some data produces a hypothesis that depends on the *fixed* bias embedded by the learner. This implies a learning algorithm can only learn efficiently over a limited number of tasks. More formally, let a learning task s be defined as

[1] We limit our study to hypothesis spaces that are finite.

a 3-tuple, $s = (F, m, \Phi)$, comprising a target concept F, a training-set size m, and a sample distribution Φ from which the examples in a training set are drawn[2]. If we represent the space of all possible learning tasks (i.e., the cross product of all target concepts, training-set sizes, and distributions) as $\mathcal{S}$, then algorithm L can learn efficiently over a limited region R_L in $\mathcal{S}$ that favors the bias embedded in L; algorithm L can never be made to learn efficiently over all tasks in $\mathcal{S}$ as long as its bias remains fixed (Schaffer, 1994; Watanabe, 1985; Wolpert, 1996; Rao et al., 1995).

3.1. Random vs Structured Tasks

One may rightly argue that the space of all tasks contains many *random* regions; failing to learn over those regions carries in fact no negative consequences. For this reason, we will assume R_L comprises a subset of structured tasks, $S_{\text{struct}} \subset S$, where each task is non-random and can be ascribed a low degree of complexity. Here we simply want to distinguish between two sets of tasks: structured and random. We attempt to give a more formal definition of both types of tasks.

One dimension along which we can differentiate between structured and random tasks lies in the expected amount of data compression that can be obtained over the training sets. Structured tasks S_{struct} denote regular patterns over the training sets that commonly lead to the discovery of concise representations. Random tasks, on the other hand, are characterized by many irregularities; long representations are then necessary to reproduce the original body of data. But how can we determine the degree of structure (conversely the degree of randomness) of a task? Let us assume we have a measure of complexity K applicable to any task. We can classify all possible tasks according to its complexity in the following way. For a fixed concept F, training-set size m, and distribution Φ, we denote the complexity of a tasks s as

$$K(s) = E_{\Phi}(K_{F,m}(T_i)) = \sum_{\forall\ \mathrm{T_i\ of\ size\ m}} P(T_i)\ K_{F,\Phi,m}(T_i) \qquad (1)$$

where $P(T_i)$ is the probability of generating training set T_i according to Φ, and $K_{F,m}(T_i)$ is the value of the complexity (i.e., degree of randomness) of training set T_i (conditioned on F), for a fixed size m, and a fixed distribution Φ. This definition can serve to classify the universe of all possible tasks according to K.

[2] A task can be seen as a random variable where each possible outcome is a training set of size m drawn according to distribution Φ; examples are labeled according to concept F.

The nature of K is left unspecified. Ideally one would use a measure such as *Kolmogorov Complexity* (Vitanyi, 1996; Li and Vitanyi, 1992; Li and Vitanyi, 1997; Vitanyi, 1997). Given a training set T, the Kolmogorov complexity of T, $K(T)$, is defined as the length of the shortest effective description of T. More rigorously, $K(T)$ is the length of the shortest binary program from which the data can be reconstructed (Vitanyi, 1997). Unlike other measures (e.g., classical information theory) Kolmogorov complexity considers the maximal degree of compressibility over the data under analysis.

3.2. Goals in Meta-Learning

One goal in meta-learning is to learn what causes L to dominate in region R_L. The problem can be decomposed in two parts: 1) determine the properties of the tasks in R_L that make L suitable for such region, and 2) determine the properties of L (i.e., what are the components contained by algorithm L and how they interact with each other) that contribute to the domination of R_L. A solution to the problem above would provide guidelines for choosing the right learning algorithm on a particular task. As illustrated in Figure 1, each task s_i may lie inside or outside the region that favors the bias embedded by a learning algorithm L. In Figure 1, task s_1 is best learned by algorithm L_A because it lies within the region R_{L_A}. Similarly, s_2 is best learned by algorithm L_B, whereas s_3 is best learned by both L_A and L_B. A solution to the meta-learning problem can indicate how to match learning algorithms with task properties, in this way yielding a principled approach to the dynamic selection of learning algorithms.

In addition, meta-learning can solve the problem of learning tasks lying outside the scope of available learning algorithms. As shown in Figure 1, task s_4 lies outside the regions of both L_A and L_B. If L_A and L_B are the only available algorithms at hand, task s_4 is prone to receiving a poor concept estimation. One approach to solve the problem above is to use a meta-learner to combine the predictions of base-learners in order to shift the dominant region over the task under study. In Figure 1, the goal would be to embed the meta-learner with a bias favoring a region of tasks that includes s_4. Section 4 describes current research heading in this direction.

3.3. Self-Adaptive Learning Algorithms

The combination of base-learners by a meta-learner offers no guarantee of covering every possible (structured) task of interest. We claim a potential avenue of research in meta-learning is to provide the foundations to construct self-adaptive learning algorithms, i.e., learning algorithms

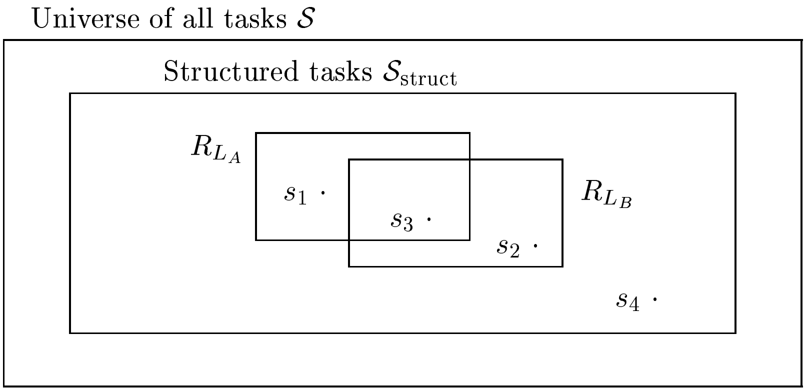
Universe of all tasks $\mathcal{S}$
Structured tasks $\mathcal{S}_{\text{struct}}$
R_{L_A}
R_{L_B}
s_1
s_3
s_2
s_4

Figure 1. Each learning algorithm covers a region of (structured) tasks favored by its bias. Task s_1 is best learned by algorithm L_A, s_2 is best learned by algorithm L_B, whereas s_3 is best learned by both L_A and L_B. Task s_4 lies outside the scope of L_A and L_B.

that change their internal mechanism according to the task under analysis. In Figure 1, this would mean enabling a learning algorithm to move along the space of structured concepts S_{struct} until the algorithm learns to cover the task under study. We assume this can be achieved through the continuous accumulation of meta-knowledge indicating the most appropriate form of bias for each different task. Beginning with no experience, the learning algorithm would initially use a fixed form of bias to approximate the target concept. As more tasks are observed, however, the algorithm would be able to use the accumulated meta-knowledge to change its own bias according to the characteristics of each task. This is one kind of life-long learning (Thrun, 1998).

Figure 2 is a (hypothetical) flow diagram of a self-adaptive learner. The input and output components to the system are a training set and a hypothesis respectively. Each time a hypothesis is produced, a performance assessment component evaluates its quality. The resulting information becomes a new entry in a performance table; an entry contains a vector of meta-features characterizing the training set, and the bias employed by the algorithm if the quality of the hypothesis exceeds some acceptable threshold. We assume the self-adaptive learner contains a meta-learner that takes as input the performance table and generates a set of rules of experience (i.e., meta-hypothesis) mapping any training set into a form of bias. The lack of rules of experience at the beginning of the learner's life would force the mechanism to use a fixed form of bias. But as more training sets are observed, we expect

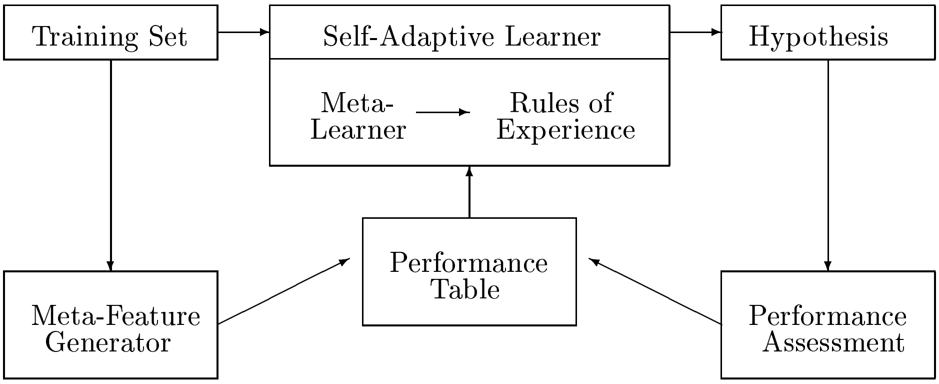

Figure 2. A flow diagram of a self-adaptive learner.

the expertise of the meta-learner to dominate in deciding which form of bias best suits the characteristics of the training set.

The self-adaptive learner described in Figure 2 poses major challenges to the meta-learning community (a more detailed analysis of these challenges is provided by Vilalta (2001)). Briefly, we need to define how can we assess the quality of a hypothesis, or how can we assess the quality of the bias employed by the the learning algorithm. Also we need to define how can we characterize a task in terms of relevant meta-features. Finally, one must be aware of a problem related to the flexibility of the self-adaptive learner in Figure 2: whereas the bias is now selected dynamically, the meta-learner is not self-adaptive and employs a fixed form of bias. Clearly the meta-learner can be seen as a learning algorithm too, but lacking the adaptability ascribed to the base learner. Ideally we would like the meta-learner to be self-adaptive, i.e., to improve through experience. One solution could be to continue with the same logical fashion as in Figure 2, and define a meta-meta-learner helping the meta-learner improve through experience. The problem, however, does not disappear because the meta-meta learner would exhibit a fixed form of bias. The challenge lies on how to stop the apparently infinite chain of meta-learners needed to achieve complete flexibility in the selection of bias.

The problems just described provide interesting goals that we hope will stimulate the research community to contribute to the field of meta-learning.

4. A Survey of Meta-Learning

This section provides a survey of meta-learning as reported in the machine-learning literature. Any survey of this kind is prone to omit relevant work and adopt a single-minded view for which we offer our apologies. Our goal is simply to unify current views and definitions of what is meant by the term meta-learning.

4.1. A Meta-Learner of Base-Learners

Originally proposed be Wolpert (1992), one common approach to meta-learning is known as *stacked generalization.* Here, a set of q base-learners are applied to a training set $T_{\text{train}} : \{(\tilde{\mathbf{X}}_{\mathbf{i}}, c_i)\}_{i=1}^{m}$ to produce q hypotheses, $\{h_j\}_{j=1}^{q}$, also called level-0 generalizers. Meta-learning takes place when training set T_{train} is redefined into a new set T'_{train}. The redefinition replaces each vector $\tilde{\mathbf{X}}_{\mathbf{i}}$ with a new vector $\tilde{\mathbf{X}}'_{\mathbf{i}}$ that contains the class predicted by each of the q hypothesis on $\tilde{\mathbf{X}}_{\mathbf{i}}$:

$$T'_{\text{train}} = \{(\tilde{\mathbf{X}}'_{\mathbf{i}}, c_i)\} = \{((h_1(\tilde{\mathbf{X}}_{\mathbf{i}}), h_2(\tilde{\mathbf{X}}_{\mathbf{i}}), \cdots, h_q(\tilde{\mathbf{X}}_{\mathbf{i}})), c_i)\} \tag{2}$$

The new training set T'_{train} serves as input to a set of meta-learners, which produce a new set of hypotheses or level-1 generalizers. The redefinition of T_{train} into T'_{train} is done via k-fold cross validation (Kohavi, 1995).

Stacked generalization is considered a form of meta-learning because the transformation of the training set conveys information about the predictions of the base-learners (i.e., conveys meta-knowledge). We do not consider as part of meta-learning other model-combination techniques where the idea is to produce variations of the data (e.g., bagging (Breiman, 1996) and boosting (Freund and Schapire, 1996)), although definitions of relevant meta-features have been obtained from this work. Stacked generalization has a severe limitation in that both base-learners and meta-learners have a fixed form of bias, i.e., no dynamic selection of bias takes place. The dominant (task) region for the meta-learner may be different from the base-learners, but ultimately fixed (Section 3 and Figure 1).

Research in the stacked-generalization paradigm investigates what base-learners and meta-learners produce best empirical results (Chan and Stolfo, 1998; Chan and Stolfo, 1993; Chan, 1996). After transforming the original training set, each example contains the predictions of the base-learners, but it may also contain the original features. Results show how certain combinations of learners and meta-learners can yield significant improvements in accuracy.

Several variations to stacked generalization have been explored. For example, Chan and Stolfo (1998) experiment with a modified approach where each base-learner is trained with a fraction of the total data. While running each learning algorithm in parallel, a hierarchical tree structure is built where each leaf is a level-0 generalizer and each internal node is a high-level generalizer (see reference for details). This strategy outperformed a majority-voting approach. In a similar vein, Prodromidis and Stolfo (1999B) build a tree of generalizers from meta-data and then prune the tree to eliminate redundant generalizers. The same idea is studied in a distributed system (Prodromidis et al., 1999; Prodromidis and Stolfo, 1999A). Todorovski and Dzeroski (2000) introduce meta-decision-trees, where each leaf of the meta-tree comprises a hypothesis for prediction. Domingos (1997,1998) shows empirical evidence supporting the claim that a meta-learner can improve the accuracy of base-learners while retaining comprehensibility.

The study of how to combine the predictions of base-learners has produced novel meta-features; these meta-features are useful to understand and predict the accuracy of the meta-learner. For example, Fan et al. (1999) introduce a conflict-based measure that indicates the proportion of examples in the training set accurately classified by the base-learners. Other meta-features include coverage (Brodley and Lane, 1996) –fraction of examples for which at least one of the base classifiers produces correct predictions; diversity (Brodley and Lane, 1996; Ali and Pazzani, 1996)) –degree of difference in the predictions of the base-learners; and correlated error (Ali and Pazzani, 1996) –fraction of examples for which two base-learners make the same incorrect prediction.

4.2. Dynamic Selection of Bias

Dynamic selection of bias enables a learning algorithm to shift its region of expertise along the set of (structured) tasks (Figure 1). The goal is to modify the hypothesis space to have a better coverage of the task under analysis. Meta-learning is a necessary component during dynamic-bias selection, often acting as a guideline in the search over the bias space.

An introduction to the field of dynamic-bias selection is given by DesJardins and Gordon (1995A). The authors develop a framework for the study of dynamic bias as a search in three different tiers (DesJardins and Gordon, 1995B). The first tier refers to a search over a hypothesis space $\mathcal{H}_\mathcal{L}$ where a learning algorithm L looks for the best hypothesis approximating the target concept; most learning algorithms assume this space fixed. For dynamic bias selection to take place, a learning algorithm L must search in a second tier, where the strength and size of

$\mathcal{H}_{\mathcal{L}}$ can be modified separately (Section 2). A third tier helps to modify the meta-spaces defined at the second tier. Although no more tiers are introduced in this framework, the problem of where to stop building more tiers (i.e., more meta-meta-spaces) is evident (Section 3).

One approach to the dynamic selection of bias is to change the representation of the feature space by adding or removing features. One of the earliest systems using a form of dynamic bias is STABB (Utgoff, 1986). With the goal of generating a hypothesis space that is strong and correct (Section 2), STABB continually exerts a form of change of representation. As an example, the system is able to construct a new feature as the disjunction of two original features; additional features increase the size of the hypothesis space and thus help to alleviate the problem of a strong hypothesis space (i.e., of having few available hypotheses). In contrast, Gordon (1992) shows how to weaken the hypothesis space by eliminating features when bias is deemed inappropriate. Hence, bias can be made stronger by eliminating features or weaker by restoring features (Gordon, 1990). In addition one can filter out hypotheses using meta-rules as a form of explicit bias selection (Gordon and Perlis, 1989). Baltes (1992) describes a framework for dynamic selection of bias as a case-based meta-learning system; concepts displaying some similarity to the target concept are retrieved from memory and used to define the hypothesis space.

Dynamic-bias selection applies to the algorithm-selection problem. Rendell et al. (1987A) describe the VBMS system that learns how to select a learning algorithm depending on the properties of the task. VBMS uses a dynamic similarity measure that evolves with experience; as more tasks are attempted VBMS learns relationships between task characteristics and biases embedded in the learning algorithms (Rendell et al., 1987B). The task characteristics used by VBMS are simple (e.g., number of features), and bias is not modified but rather depends on the available learning algorithms. A related approach is described by Bruha (2000) in a rule-based system. Here, predicting the class of new examples depends on the quality of each rule; such quality is updated during the testing phase: a dynamic process that changes bias in the rule-selection policy.

4.3. Meta-Rules Matching Domains With Algorithm Performance

One important facet of meta-learning is to provide guidelines of how to relate a learning algorithm with those domains in which the algorithm performs well. Most often the main performance criterion is predictive accuracy, but in reality other criteria may be equally important, e.g., computational complexity, expressiveness, compactness, comprehensibility, etc. (Giraud, 1998).

The general approach consists of defining a set of domain characteristics or meta-features that are relevant to the performance of a learning algorithm (Michie, 1994); those meta-features enable us to build a meta-domain T_{meta} relating domain characteristics with algorithm performance (once a sufficient number of domains has been analyzed). Finally, we can induce a set of rules using a meta-learner over T_{meta} to discover the conditions under which a learning algorithm outperforms others.

Under this framework, Aha (1992) aims at obtaining rules specifying when a learning algorithm outperforms others significantly. Examples of domain characteristics are the degree of correlation between features and the target concept, the distribution of examples within concepts disjuncts, the distribution of examples among concepts, etc. The rules reveal the conditions under which significant differences in performance hold. Gama and Brazdil (1995) extract domain characteristics such as the number of examples, number of attributes, number of classes, standard deviation ratio for each feature, skewness, kurtosis, noise-signal ratio, etc., to generate meta-rule models. Similar work is reported by Brazdil (1998) who proposes using meta-learning as a pre-processing step to model selection; experimentation on accuracy performance is then used to select the best algorithm. Meta-rules matching domain characteristics with inductive bias have also been crafted manually (Brodley, 1993; Brodley, 1994). In addition, a domain may be represented by properties of the final hypothesis rather than the data itself. For example, Bensusan et. al (2000) measure properties of a decision-tree, e.g., nodes per feature, maximum tree depth, shape, tree imbalance, etc., and convert them into meta-features.

4.3.1. *Finding Regions In The Feature Space And Meta-Feature Space*

Instead of using meta-learning to select a learning algorithm for a particular domain, a –more granular– approach consists of selecting a learning algorithm for each individual test example. The idea is to choose that learning algorithm displaying best performance around the neighborhood of the test example (Merz, 1995A; Merz, 1995B). Algorithm selection is done according to best performance, using cross-validatory history.

A slight variation to the approach above is to look at the neighborhood of a test example in the space of meta-features. Specifically, after learning from several domains, one can construct a meta-domain T_{meta}, where each element pair is a description of a domain (meta-feature vector) and a class label corresponding to the best-performance learning algorithm on that domain. When a new domain arrives, we can gather the k-nearest neighbor examples in T_{meta} to select the learning algorithm with best average performance (Keller et. al, 2000). Meta-features like accuracy, storage space, and running time can be used for performance evaluation (Keller et. al, 2000). A similar approach is defined by Brazdil and Soares (2000), in which the learning algorithms corresponding to the k-nearest neighbor domains are ranked as a function of accuracy and running time.

4.3.2. *Landmarking*

A recent piece of work in meta-learning is called *landmarking.* The idea is to have two sets of learning algorithms Q_{landmark} and Q_{pool}. The first set, Q_{landmark}, is composed of simple learners that exhibit significant differences in their mechanism. We will use their accuracy (or error rate) to characterize a domain, and refer to them as *landmarkers.* The second set, Q_{pool}, contains advanced learners, one of which must be selected for our current domain. A meta-domain is constructed as follows. Each example (i.e., each domain) is characterized by the error rates of the landmarkers in Q_{landmark}. The label or class of each example is the algorithm in Q_{pool} with best cross-validatory accuracy. A meta-learner can then associate the performance of the landmarkers with the best algorithm in Q_{pool}. From this point of view, meta-learning is the process of finding areas of expertise of learners called landmarkers, and of correlating those areas with the performance of other –more advanced– learners (Bensusan and Giraud-Carrier, 2000; Pfahringer et. al, 2000).

4.4. Inductive Transfer and Learning to Learn

Learning is not an isolated task that starts from scratch every time a new problem domain appears. As experience accumulates, a learning mechanism is expected to perform increasingly better (Section 3). For learning to improve through time, knowledge about learning, or meta-knowledge, must be transferred across domains or tasks. The process is known as inductive transfer (Pratt and Thrun, 1997).

An interesting study in inductive transfer falls in the realm of neural networks. A review of how neural networks can learn from related tasks is provided by Pratt and Jennings (1998). Caruana (1997) shows why multitask learning works well in the context of neural networks using backpropagation. The claim is that training with many domains in parallel on a single neural network induces information that accumulates in the training signals; a new domain can then benefit from such past experience. Thrun and Sullivan (1998) propose a learning algorithm where domains are clustered when mutually related. A new domain is assigned to the most related cluster; inductive transfer takes place when generalization exploits information about the selected cluster. Multitask learning can also be used in other learning paradigms such as kernel regression and decision trees.

An introduction to the benefits of learning from multiple tasks to improve generalization is provided by Thrun and Pratt (1998). The authors propose a general framework that shows the distinction between learning at the base-level and at the meta-level. In the base-level one simply tries to find the correct hypothesis h on a fixed hypothesis space $\mathcal{H}_{\mathcal{L}}$. In the meta-level one needs to find properties of target functions to characterize entire hypothesis spaces $\{\mathcal{H}\}$. It must be clear that both levels require some form of bias, i.e., no-free lunch theorems (Schaffer, 1994; Watanabe, 1985; Wolpert, 1996; Rao et al., 1995) apply at both levels.

4.4.1. *Learning to Learn*

Learning-to-learn relies on the main assumption that learning is simplified when it continues working in a life-long context (Thrun, 1998). The assumption is supported by the existence of patterns on each domain, *and* across domains. The general understanding of the nature of patterns across domains is that of *invariant transformations*. For example, image recognition of a target object is simplified if the object is invariant under rotation, translation, scaling, etc. Hence, learning-to-learn studies how to improve learning by detecting, extracting, and exploiting invariant transformation across domains. As an example, Thrun and Mitchell (1995) describe how to search for certain forms of invariance in life-long learning using a neural network. These kinds of invariance are used to bias the learner as it selects a hypothesis on a new domain.

4.4.2. *Theoretical Studies*

A theoretical analysis of the learning-to-learn paradigm is found within an empirical and Bayesian view (Baxter, 1998), and within a Probably Approximately Correct (Valiant, 1984) or PAC view. We focus on the PAC view (Baxter, 2000). In this case, the learner is assumed embedded in an environment of related learning domains. Meta-learning takes place because the learner is not only looking for the right hypothesis h in a hypothesis space $\mathcal{H}_{\mathcal{L}}$, but in addition is searching for the right hypothesis space in a family of hypothesis spaces $\{\mathcal{H}\}$. The right hypothesis space $\mathcal{H}_{\mathcal{L}} \in \{\mathcal{H}\}$ must be large enough to embed a solution to the problem domain, but small enough to make any form of generalization possible. The study draws an analogy between the role of the VC dimension (Blumer et. al, 1989) and the size of the family of hypothesis spaces $|\{\mathcal{H}\}|$. It turns out both measures can be used to derive bounds on the number of domains, and the number of examples on each domain, required to ensure with high probability that we will find a solution having low error on new training domains. Hence, under certain assumptions, the number of examples required for each domain decreases as the number of observed domains increases.

4.5. Learning Classifier Systems

Learning classifier systems originated from the pioneer work of Holland (Holland, 1992; Holland and Reitman, 1978). An excellent review of the subject is given by Lanzi et. al (2000). A classifier system is a parallel, message-passing, rule-based system. Each message or rule –referred in this context as a classifier– is a condition-action pair; if a message matches the condition part, the rule is candidate to activate and execute the action part. The system assumes an input interface or set of detectors that translates signals from an external environment into messages. Similarly an output interface translates messages through effectors into external actions (Booker et. al, 1989).

A classifier system is a learning mechanism working at two different levels. At the first level, the system *learns* to identify rules that can return high profit from the environment. The problem of how to assign credit to the right rules, also known as the *credit-assignment problem*, is solved through some form of reinforcement learning (Sutton and Barto, 1998), e.g., bucket-brigade, profit-sharing plan, Q-learning. The mechanism assigns a credit or value of strength to each rule based on its contribution. At the second level, the system *learns* how to construct new rules that have the potential of further increasing the reward from the environment. Normally a set of genetic operators come into play for evolving the rule set. Rules with high strength have a higher probability of being selected to produce new offspring rules (Holland et. al, 2000).

Classifier systems may appear at first glance disconnected from meta-learning. A closer examination, however, reveals the opposite. For example, a learning algorithm can be dissected into a set of components (Vilalta, 1998), each with a specific function during classification, e.g., select the most relevant features, partition the training set following a separate-and-conquer approach or a divide-and-conquer approach, hypothesis pruning, etc. For each domain, activating some components may give a higher reward (e.g., higher predictive accuracy) than others. The framework adopted by learning classifier systems can be used in meta-learning by mapping classifiers or rules with learning components. A form of reinforcement learning can decide what learning strategy, i.e., combination of learning components, maximizes the learner's performance. In addition, a discovery system may also try to find new components that can produce more efficient learning algorithms. The idea above sheds light on a research direction to build self-adaptive learners (Figure 2), where the assessment of a hypothesis is based on the successful performance of a combination of learning components, and on a meta-learner using that meta-knowledge to build new learning algorithms.

4.6. Other Approaches

Outside the scope of classification, meta-learning has been applied to areas like case-based reasoning, constraint satisfaction, learning agents, etc. We end our survey by briefly mentioning some work related to these areas.

In the context of cased-based reasoning, Goel (1996) describes a case-based interactive system for problem solving. The system displays the ability to reason about its own performance by keeping track of how a problem is solved, i.e., by keeping track of meta-cases. As a result, the system is able to provide explanations of its reasoning and justifications of its solutions.

Meta-learning has been used in analytic learning for constraint-satisfaction problems (Minton, 1993). Analytic learning (e.g., explanation based learning, derivational analogy), exploits problem-solving experience (Minton, 1989). When applied at a meta-learning level, the idea is to use meta-level theories to help the system reason about the problem solver's base-level theory. A meta-level analysis is appropriate when the base-level theory is intractable (Minton, 1993).

Meta-learning can also be applied to areas like learning agents. Baum (1998) provides an extensive study and discussion on how to make agents collaborate (using a kind of reinforcement learning). The system embeds learning agents that can generate other agents. Other approaches include the use of meta-level information of problem-solving knowledge for cooperation in a multi-agent system (Prasad and Lesser, 1997).

5. Discussion and Conclusions

Our survey shows how the term meta-learning has been ascribed different meanings by different research groups. From building meta-learners of base classifiers (Section 4.1), to looking for dynamic forms of bias (Section 4.1), to studying how learning can continue in a life-long environment (Section 4.1), meta-learning continues to enrich the field of machine learning with a constant question: how can we exploit knowledge about learning (i.e., meta-knowledge) to improve the performance of learning algorithms? In spite of the many research directions, no clear answer has emerged.

Perhaps broadening our view of the scope of meta-learning can provide better insights on how meta-knowledge can be used. For example, the approach adopted by stacked generalization (Section 4.1) assumes no fundamental distinction between learning at the base-level and at the meta-level. Transforming the training set by including the predictions of base learners is a form of re-using our learning tools at different levels of abstraction. The idea of making no fundamental differences between learning and meta-learning is shared by several researchers (Schmidhuber, 1995).

But meta-learning may be radically different from learning at the base level. For example, we could define meta-learning as the problem of taking the right action (i.e., the right bias) according to a specific world state (e.g., the type of input-output distribution). This definition allows us to equate meta-learning with some form of reinforcement learning (Ring, 1998). The definition also points to the mechanism behind learning classifier systems (Section 4.5).

Whether we consider meta-learning to have the same fundamental structure as base-learning or not, an important goal in machine learning is to combine the ability of a learning algorithm to improve performance when the number of examples increases, with the ability of the same learning algorithm to improve its learning bias when the number of tasks increases. To achieve this goal, we believe the field of meta-learning would benefit greatly from a study of how learning algorithms can improve their performance through experience, i.e., through meta-knowledge (Section 3).

Acknowledgements

This work was supported by IBM T.J. Watson Research Center.

References

Aha David W. (1992). Generalizing from Case Studies: A Case Study. *Proceedings of the Ninth International Workshop on Machine Learning*, 1–10, Morgan Kaufman.

Ali Kamal and Pazzani Michael J. (1996). Error Reduction Through Learning Model Descriptions. *Machine Learning*, 24, 173–202.

Baltes Jacky (1992). Case-Based Meta Learning: Sustained Learning Supported by a Dynamically Biased Version Space. *Proceedings of the Machine Learning Workshop on Biases in Inductive Learning.*

Baum Eric B. (1998). Manifesto for an Evolutionary Economics of Intelligence. *Neural Networks and Machine Learning*, 285–344, Editor C.M. Bishop, Springer-Verlag.

Baxter Jonathan (1998). Theoretical Models of Learning to Learn. *Learning to Learn*, Chapter 4, 71–94, Kluwer Academic Publishers, MA.

Baxter Jonathan (2000). A Model of Inductive Learning Bias. *Journal of Artificial Intelligence Research*, 12, 149–198.

Bensusan Hilan and Giraud-Carrier Christophe (2000). Casa Batlo in Passeig or landmarking the expertise space. *Eleventh European Conference on Machine Learning, Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, Barcelona, Spain.

Bensusan Hilan, Giraud-Carrier Christophe, and Kennedy C. J. (2000). A High-Order Approach to Meta-Learning. *Eleventh European Conference on Machine Learning, Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, Barcelona, Spain.

Blumer, A., Ehrenfeucht, A., Hausler, D., and Warmuth, M.K. (1989). Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the ACM, 36, 929–965.*

Booker L., Goldberg D., and Holland J. (1989). Classifier Systems and Genetic Algorithms. *Artificial Intelligence, 40, 235–282.*

Brazdil Pavel B. (1998). Data Transformation and model selection by experimentation and meta-learning. *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, 11–17, Technical University of Chemnitz.

Brazdil Pavel B. and Soares Carlos (2000). Ranking Classification Algorithms Based on Relevant Performance Information. *Eleventh European Conference on Machine Learning, Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, Barcelona, Spain.

Breiman (1996). Bagging Predictors. *Machine Learning*, 24, 123–140.

Brodley Carla (1993). Addressing the Selective Superiority Problem: Automatic Algorithm/Model Class Selection. *Proceedings of the Tenth International Conference on Machine Learning*, 17-24, San Mateo, CA, Morgan Kaufman.

Brodley Carla (1994). Recursive Automatic Bias Selection for Classifier Construction. *Machine Learning*, 20.

Brodley Carla and Lane T. (1996). Creating and Exploiting Coverage and Diversity. *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models*, 8–14, Portland, Oregon.

Bruha Ivan (2000). A feedback loop for refining rule qualities in a classifier: a reward-penalty strategy. *Eleventh European Conference on Machine Learning, Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, Barcelona, Spain.

Caruana Rich (1997). Multitask Learning. *Second Special Issue on Inductive Transfer. Machine Learning*, 28, 41–75.

Chan Philip K. and Stolfo S. (1998). On the Accuracy of Meta-Learning for Scalable Data Mining. *Journal of Intelligent Integration of Information*, Ed. L. Kerschberg.

Chan Philip K. and Stolfo S. (1993). Experiments on Multistrategy Learning by Meta-Learning. *Proceedings of the International Conference on Information Knowledge Management*, 314–323.

Chan Philip K. (1996). An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. *PhD Thesis, Graduate School of arts and Sciences at Columbia University.*

Cohen Paul and Feigenbaum Edward (1989). Learning and Inductive Inference. *The Handbook of Artificial Intelligence, Volume III.* 326-334. Addison-Wesley.

DesJardins Marie and Gordon Diana F. (1995A). Special issue on bias evaluation and selection. *Machine Learning*, 20 (1/2).

DesJardins Marie and Gordon Diana F. (1995B). Evaluation and Selection of Biases in Machine Learning. *Machine Learning*, 20, 5–22.

Domingos Pedro (1997). Knowledge Acquisition from Examples Via Multiple Models. *Proceedings of the Fourteenth International Conference on Machine Learning*, 98–106. Morgan Kaufmann, Nashville TN.

Domingos Pedro (1998). Knowledge Discovery Via Multiple Models. *Intelligent Data Analysis*, 2, 187–202.

Fan Wei, Stolfo S., and Chan Philip K. (1999). Using Conflicts Among Multiple Base Classifiers to Measure the Performance of Stacking. *Proceedings of the ICML-99 Workshop on Recent Advances in Meta-Learning and Future Work*, 10–15, Giraud-Carrier Christophe and Pfahringer Bernhard (eds.), Stefan Institute Publisher, Ljubljana.

Freund Y. and Schapire R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on Machine Learning*, 148–156. Morgan Kaufman, Bari, Italy.

Gama J. and Brazdil P. (1995). Characterization of Classification Algorithms. *Proceedings of the seventh Portuguese Conference on Artificial Intelligence, EPIA*, 189-200, Funchal, Madeira Island, Portugal.

Giraud-Carrier Christophe (1998). Beyond Predictive Accuracy: What?. *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, 78–85, Technical University of Chemnitz.

Goel Ashok K. (1996). Meta-Cases: Explaining Case-Based Reasoning. *Proceedings of the Third European Workshop on Case-Based Reasoning*, Published in Advances in Case-Based Reasoning, Lecture Notes in Computer Science, 1168, Springer, New York.

Gordon Diana and Perlis Donald (1989). Explicitly Biased Generalization. *Computational Intelligence*, 5, 67-81.

Gordon Diana F. (1992). Queries for Bias Testing. *Proceedings of the Workshop on Change of Representation and Problem Reformulation.*

Gordon Diana F. (1990). Active Bias Adjustment for Incremental, Supervised Concept Learning. *PhD Thesis, University of Maryland, 1990.*

Holland John, Booker Lashon, Colombetti Marco, Dorigo Marco, Goldberg David, Forrest Stephanie, Riolo Rick, Smith Robert, Lanzi Pier Luca, Stolzmann Wolfgang, and Wilson Stewart (2000). What is a Learning Classifier System?. *Lecture Notes in Artificial Intelligence LNAI 1813*, Springer Verlag, pp. 3-22, 2000.

Holland John (1975). Adaptation in Natural and Artificial Systems. *University of Michigan Press, Ann Arbor (Republished by the MIT Press, 1992.*

Holland John and Reitman J. (1978). Cognitive Systems Based On Adaptive Algorithms. *In D. A. Waterman and F. Hayes Roth, editors, Pattern-directed inference systems*, New York: Academic Press, Springer Verlag, 1978.

Keller Jorg, Paterson Iain, and Berrer Helmutt (2000). An Integrated Concept for Multi-Crieria-Ranking of Data-Mining Algorithms. *Eleventh European Conference on Machine Learning, Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, Barcelona, Spain.

Kohavi Ron (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1137–1143.

Lanzi Pier Luca, Stolzmann Wolfgang, and Wilson Stewart W. (2000). Learning Classifier Systems: From Foundations to Applications. *Lecture Notes in Artificial Intelligence 1813*, Springer-Verlag, New York, 2000.

Li Ming and Vitanyi Paul (1992). Philosophical Issues in Kolmogorov Complexity. *International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, volume 623, Springer Verlag, Berlin, 1992.

Li Ming and Vitanyi Paul (1997). An Introduction to Kolmogorov Complexity and Its Applications. *Springer -Verlag, New York.*

Merz Christopher J. (1995A). Dynamic Learning Bias Selection. *Preliminary papers of the Fifth International Workshop on Artificial Intelligence and Statistics*, 386–395, Florida.

Merz Christopher J. (1995B). Dynamical Selection of Learning Algorithms. *Learning from Data: Artificial Intelligence and Statistics*, D. Fisher and H. J. Lenz (Eds.), Springer-Verlag.

Michie D., Spiegelhalter DJ, and Taylor CC (1994). Machine Learning, Neural and Statistical Classification. *Ellis Horwood, Chichester, England.*

Minton Steve (1993). An Analytic Learning System for Specialized Heuristics. *Proceedings of Thirteenth International Joint Conference on Artificial Intelligence.*

Minton Steve (1989). Explanation Based-Learning: A problem Solving Perspective. *Artificial Intelligence*, 40, 63–118.

Mitchell Tom (1980). The need for biases in learning generalizations. *Technical Report CBM-TR-117*, Computer Science Department, Rutgers University, New Brunswick, NJ 08904.

Mitchell Tom (1997). *Machine Learning.* Ed. MacGraw-Hill

Pfahinger Bernhard, Bensusan Hilan, and Giraud-Carrier Christophe (2000). Meta-Learning by Landmarking Various Learning Algorithms. *Proceedings of the Seventeenth International Conference on Machine Learning*, Stanford, CA.

Prasad M. V. Nagendra and Lesser Victor R. (1997). The Use of Meta-Level Information in Learning Situation-Specific Coordination. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, Nagoya, Japan.

Pratt Lorien and Thrun Sebastian (1997). Second Special Issue on Inductive Transfer. *Machine Learning*, 28.

Pratt Sebastian and Jennings Barbara (1998). A Survey of Connectionist Network Reuse Through Transfer. *Learning to Learn*, Chapter 2, 19–43, Kluwer Academic Publishers, MA.

Prodromidis Andreas L., Chan Philip K., and Stolfo S. (1999). Meta-Learning in Distributed Data Mining Systems: Issues and Approaches. *Advances in Distributed Data Mining*, Book AAAI Press, Kargupta and Chan (eds.).

Prodromidis Andreas L. and Stolfo S. (1999A). A Comparative Evaluation of Meta-Learning Strategies over Large and Distributed Data Sets. *Proceedings of the ICML-99 Workshop on Recent Advances in Meta-Learning and Future Work*, 18–27, Giraud-Carrier Christophe and Pfahringer Bernhard (eds.), Stefan Institute Publisher, Ljubljana.

Prodromidis Andreas L. and Stolfo S. (1999B). Minimal Cost Complexity Pruning of Meta-Classifiers. *Proceedings of AAAI*, Extended Abstract.

Rao, R.B., Gordon, D., and Spears W. (1995). For Every Generalization Action, Is There Really an Equal and Opposite Reaction? Analysis of the Conservation

Law for Generalization Performance. *Proceedings of the Twelfth International Conference on Machine Learning*, 471–479, Morgan Kaufman.

Rendell Larry, Seshu Raj, and Tcheng David (1987A). More Robust Concept Learning Using Dynamically-Variable Bias. *Proceedings of the Fourth International Workshop on Machine Learning*, 66–78, Morgan Kaufman.

Rendell Larry, Seshu Raj, and Tcheng David (1987B). Layered Concept-Learning and Dynamically-Variable Bias Management. *Proceedings of the International Joint Conference of Artificial Intelligence*, 308–314, Milan, Italy.

Ring Mark B. (1998). CHILD: A first Step Towards Continual Learning. *Learning to Learn*, Chapter 11, 261–292, Kluwer Academic Publishers, MA.

Schaffer C. (1994). A Conservation Law for Generalization Performance. *Proceedings of the eleventh International Conference on Machine Learning*, 259–265, San Francisco, Morgan Kaufman.

Schmidhuber Jurgen (1995). Discovering Solutions with Low Kolmogorov Complexity and High Generalization Capability. *Proceedings of the Twelve International Conference on Machine Learning*, 488–49, Morgan Kaufman.

Sutton Richard and Barto Andrew (1995). Reinforcement Learning. *MIT Press, Cambridge Massachusetts.*

Thrun Sebastian and Mitchell Tom (1995). Learning One More Thing. *Proceedings of the International Joint Conference on Artificial Intelligence*, 1217–1223, Morgan Kaufman.

Thrun Sebastian and Lorien Pratt (1998). Learning To Learn: Introduction And Overview. *Learning to Learn*, Chapter 1, 3–17, Kluwer Academic Publishers, MA.

Thrun Sebastian (1998). Lifelong Learning Algorithms. *Learning to Learn*, Chapter 8, 181–209, Kluwer Academic Publishers, MA.

Thrun Sebastian and O'Sullivan Joseph (1998). Clustering Learning Tasks and the Selective Cross-Task Transfer of Knowledge. *Learning to Learn*, Chapter 10, 235–257, Kluwer Academic Publishers, MA.

Todorovski Ljupco and Dzeroski Saso (2000). Combining Multiple Models with Meta Decision Trees. *Eleventh European Conference on Machine Learning, Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, Barcelona, Spain.

Utgoff Paul (1986). Shift of Bias for Inductive Concept Learning. In Michalski, R.S. et al (Ed), *Machine Learning: An Artificial Intelligence Approach Vol. II*, 107–148, Morgan Kaufman, California.

Valiant, L. G. (1984). A Theory Of The Learnable. *Comm. ACM*, 27, 1134-1142.

Vilalta, R. (1998). On the Development of Inductive Learning Algorithms: Generating Flexible And Adaptable Concept Representations. *PhD Thesis, University of Illinois at Urbana-Champaign.*

Vilalta, R. (2001). Research Directions in Meta-Learning: Building Self-Adaptive Learners. *International Conference on Artificial Intelligence*, Las Vegas, Nevada.

Li Ming and Vitanyi Paul (1996). Ideal MDL and its Relation to Bayesianism. *ISIS: Information, Statistics Induction in Science.* World Scientifc, Singapore, pp. 282–291.

Li Ming and Vitanyi Paul (1997). On Prediction by Data Compression. *9th European Conference on Machine Learning.* Lecture Notes in Artificial Intelligence, Springer-Verlag.

Watanabe Satosi (1969). Knowing and Guessing, A Formal and Quantitative Study. *John Wiley & Sons Inc.*

Watanabe Satosi (1985). Pattern Recognition: Human and Mechanical. *John Wiley & Sons Inc.*

Wolpert D. (1992). Stacked Generalization. *Neural Networks*, 5: 241–259.

Wolpert D. (1996). The Lack of a Priori Distinctions Between Learning Algorithms and the Existence of a Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8.