Research Directions In Meta-Learning

Ricardo Vilalta
IBM T.J. Watson Research Center
30 Saw Mill River Rd.,
Hawthorne, NY., 10532 U.S.A.

Youssef Drissi
IBM T.J. Watson Research Center
30 Saw Mill River Rd.,
Hawthorne, NY., 10532 U.S.A.

Proceedings of the International Conference on Artificial Intelligence, 2001.

Abstract ***Meta-learning offers the potential of extending the capabilities of current learning algorithms by making their mechanism flexible according to the domain or task under study. An impediment to move forward in this direction is that no clear consensus exists on the exact meaning of the term meta-learning; different research groups hold different views. This paper proposes a perspective view of meta-learning in which the central goal is to build self-adaptive learners, namely learning algorithms that improve through experience by changing their bias dynamically. We propose a general framework addressing the problem of how to build self-adaptive learners, and show research directions that highlight the challenges lying in front of us to reach such goal.***

keywords: inductive learning, classification, meta-knowledge.

1 Introduction

Inductive learning, or classification, takes place when a learner or classifier (e.g., decision tree, neural network, support vector machine) is applied to some data to produce a hypothesis explaining a target concept; the search for a good hypothesis depends on the *fixed* bias [7] embedded by the learner. The algorithm is said to be able to learn because the quality of the hypothesis normally improves with an increasing number of examples. Nevertheless, since the bias of the learner is fixed, successive applications of the algorithm over the same data always produces the same hypothesis, independently of performance; no knowledge is commonly extracted across domains or tasks [8].

In contrast, meta-learning studies how the hypothesis output by a learner can improve through experience. The goal is to understand how learning itself can become flexible according to the domain or task under study. 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. The goal is to discover ways of dynamically searching for the best learning strategy as the number of tasks increases [11, 9]. Hence, meta-learning advocates the need for continuous adaptation of the learner. If a learner fails to perform efficiently, one would expect the learning mechanism itself to adapt in case the same task is presented again. Learning can then take place not only at the example (i.e., base) level, but also at the across-task (i.e., meta) level.

Despite the promising research direction offered by meta-learning, no apparent consensus exists of what is meant by such term. Examples of different views abound: building a meta-learner of base-learners [2], selecting inductive biases dynamically [3], building meta-rules matching task properties with algorithm performance [4], inductive transfer and learning to learn [8], learning classifier systems [5], etc. After addressing some common views of meta-learning, this paper proposes a perspective view where the main goal is to build a self-adaptive learner that improves its inductive bias through experience. We propose a general framework to build self-adaptive learners, and delineate research directions that address the challenges lying in front of us to reach such goal.

This paper is organized as follows. Section 2 gives definitions and background information on classification. Section 3 provides our own perspective view of the nature and potential avenues of research in meta-learning. Finally, Section 4 ends with a summary 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 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, namely that minimizes a loss function (e.g., zero-one loss) in the input-output space, $\mathcal{X} \times \mathcal{C}$, 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* $\mathcal{H_L}$ until it finds a *hypothesis* $h_L \in \mathcal{H_L}$, that approximates the true function F. Thus a learning algorithm L maps a training set into a hypothesis, $L : \mathcal{T} \rightarrow \mathcal{H_L}$, where $\mathcal{T}$ is the space of all training sets of size m. The selected hypothesis h_L 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 nature and size of the hypothesis space $\mathcal{H_L}$, and it imposes an ordering or ranking Υ over all hypotheses in $\mathcal{H_L}$. 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 considered by L_A is smaller than the size of the hypothesis space considered by L_B (i.e., if $|\mathcal{H}_{\mathcal{L}_A}| \leq |\mathcal{H}_{\mathcal{L}_B}|$). In this case, the bias embedded by L_A conveys more extra-evidential information [12] 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 $F \in \mathcal{H}$). An incorrect bias precludes finding a perfect estimate to target concept F.

3 A Perspective View

In base-learning, the hypothesis space $\mathcal{H_L}$ of a learning algorithm L is fixed. Applying a decision tree, neural network, or a support vector machine over some data produces a hypothesis that depends on the *fixed* bias embedded by the learner. If we represent the space of all possible learning tasks[1] 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 [10, 13]. One may rightly argue that the space of all tasks contains many random instances; failing to learn over those instances carries in fact no negative consequences. For this reason, we will assume R_L belongs to a

[1] Let a learning task be a 3-tuple, (F, m, Φ), comprising a target concept F, a training-set size m, and a sample distribution Φ from which the examples in the training set are drawn.

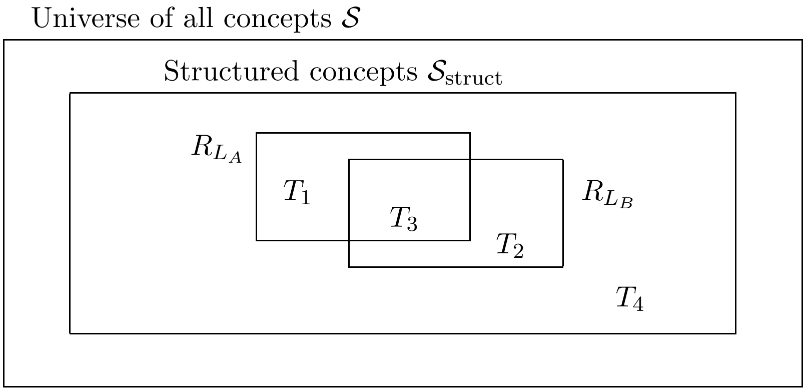
Universe of all concepts $\mathcal{S}$
Structured concepts $\mathcal{S}_{\text{struct}}$
R_{L_A}
T_1
T_3
T_2
R_{L_B}
T_4

Figure 1: Each learning algorithm covers a region of (structured) tasks favored by its bias. Task T_1 is best learned by algorithm L_A, T_2 is best learned by algorithm L_B, whereas T_3 is best learned by both L_A and L_B. Task T_4 lies outside the scope of L_A and L_B.

subset of structured tasks, $S_{\text{struct}} \subset S$, where each task is non-random and can be ascribed a low degree of complexity (e.g., Kolmogorov complexity [6]).

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 embedded in algorithm L and how they interact with each other) that contribute to dominate in R_L. A solution to the problem above would provide guidelines for choosing the right algorithm on a particular task. As illustrated in Figure 1, each task T_i may lie inside or outside the region that favors the bias embedded by a learning algorithm L. In Figure 1, task T_1 is best learned by algorithm L_A because it lies within the region R_{L_A}. Similarly, T_2 is best learned by algorithm L_B, whereas T_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 T_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 T_4 is prone to receive 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 T_4.

3.1 Self-Adaptive Learners

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 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 ac-

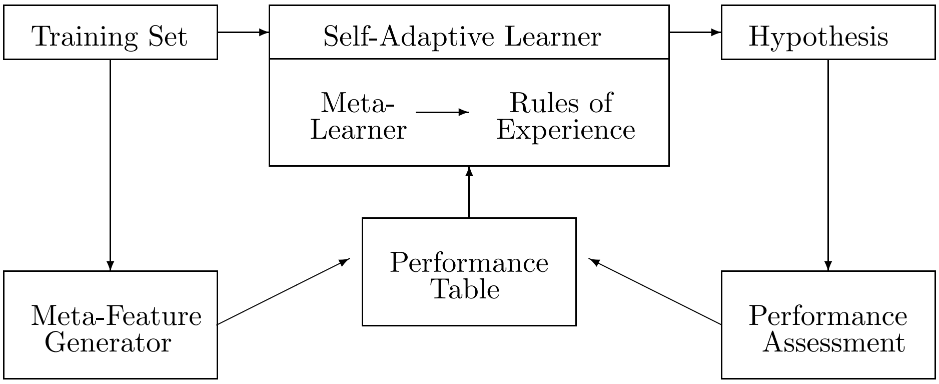

Figure 2: A flow diagram of a self-adaptive learner.

cumulated meta-knowledge to change its own bias according to the characteristics of each task. This is one kind of life-long learning [11].

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 embeds 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 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. We provide our own view of possible research directions addressing these challenges.

3.1.1 The Quality of Bias

First, 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? To answer these questions, assume the bias of learner L, B_L, is fully specified by a two-tuple $B_L = (\mathcal{H}_L, \Upsilon)$, where $\mathcal{H}_L$ is the hypothesis space from which L must select a hypothesis h_L, and Υ specifies an ordering of all hypotheses in $\mathcal{H}_L$ given a training set T_{train}. In base-learning, B_L is fixed: learner L is designed to choose the hypothesis $h_L \in \mathcal{H}_\mathcal{L}$ with best ranking according to Υ after looking at T_{train} (a sub-optimal search strategy might fail to find the hypothesis with best-ranking as a trade-off for efficiency due to the size of $\mathcal{H}_\mathcal{L}$). A self-adaptive learner cannot assume B_L fixed; the bias must be flexible according to the characteristics of the task under study. Hence, an adaptive learner must be able to search among different families of hypothesis spaces $\{\mathcal{H}_i\}$ and different orderings $\{\Upsilon_i\}$.

Now, if we had a way to measure the distance between pairs of hypothesis spaces and pairs of hypothesis orderings, then the quality of the learning bias could be defined as a function of these distance measures. In other words, the quality of the output of an adaptive learner depends on the proximity of the hypothesis space and hypothesis ordering to the true target values. Currently, meta-learning lacks any theory pointing in this direction.

One last observation is necessary. Since the training set is sampled according to a distribution Φ, one may try to obtain the expected values of these distances by averaging over different training sets of size m according to Φ. In addition, if there is a probability distribution from which the hypothesis spaces are drawn, then one may try to average over different hypothesis spaces of same size according to such distribution [1].

To conclude, the challenge lies on defining the distance between pairs of hypothesis spaces and hypothesis orderings. Measures like predictive accuracy or ROC-curves convey almost no information about these distances.

Relevant Meta-Features

Second, how can we characterize a domain in terms of relevant meta-features? Ultimately a task is well characterized by the probability distribution of class labels in the input-output space. Since we assume here a deterministic target function (i.e., zero-Bayes risk), each example is assumed to have a unique class with probability one. In this case it is the distribution Φ from which the examples are drawn that dictates the distribution of class labels[2]. For example, one distribution may produce dense clusters of positive and negative examples separated by regions of (almost) empty space; another distribution may produce clusters with a mixed proportion of classes, namely clusters of class-uniform examples that often intersect, which complicates leaning.

The challenge lies on defining meta-features identifying the nature of Φ. We believe meta-features must be closely related to the mechanism of L, such that no universal way of defining Φ exists. A characterization of Φ relevant to the performance of a learning algorithm L is intimately related to the mechanism of L, i.e., to the bias of L.

For example, assume a target concept F in which the boundaries between regions of class-uniform examples are not linear. We know applying a linear discriminant L over training set T_{train} is prone to produce a poor estimate of F. The question is how can we characterize Φ from T_{train} *and* L to determine the situations in which L will fail giving good approximations to F? To answer this question, let us assume we have a means to measure the distance between two hypothesis. Let h^* be the hypothesis output by L in the space of linear-discriminant hypotheses. Let $d(h^*, F)$, be the distance between h^* and target function F. Our goal is to find meta-features characterizing Φ that show strong correlation with $d(h^*, F)$. We need to know how far or close is L to learning F efficiently. In our example of linear discriminants, a candidate meta-feature can be set to measure how far is a decision boundary from linearity. Such meta-feature by itself may be difficult to define; we believe, however, that the major challenges lie on defining a distance-metric between pairs of hypotheses, and on defining relevant meta-features that depend on Φ as much as L.

Flexibility at the Meta-Level

Finally, one must be aware of a problem related to the flexibility embedded by the self-adaptive learner of Figure 2: whereas the bias is now selected dynamically, the meta-learner is not self-adaptive and employs a fixed form of bias. The meta-learner in Figure 2 takes as input a performance table where each example contains a vector of meta-features and a label or class corresponding to the bias employed by the base algorithm. 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, that is able 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 ex-

[2] A probabilistic estimation of the target function would be necessary if Bayes risk is not zero. In that case an example would be assigned a class with certain probability according to a fixed but unknown probability distribution over the input-output space.

hibit 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.

To solve this problem, we suggest a scenario in which two self-adaptive learners act as meta-learners for each other. We envision a self-adaptive learner working in two modes: the normal mode, in which the learner improves through the accumulation of meta-knowledge as described in Section 3.1 (Figure 2), and a meta-learning mode in which the learner plays the role of a meta-learner for the other learner. At a fixed point in time, the two self-adaptive learners work on different modes: while one works on normal mode, the other must work on meta-learning mode helping the other improve through experience. Each learner would then exhibit full flexibility in the dynamic selection of bias.

Clearly the solution above overlooks an assortment of implementation details. Our goal is simply to point to promising research directions that can bring the construction of self-adaptive learners into reality. The sections described above provide interesting goals that we hope will stimulate the research community to contribute to the field of meta-learning.

4 Conclusions

Despite many different views currently active in meta-learning, no apparent consensus exists of what is meant by such term. This paper outlines a perspective view of meta-learning in which the goal is to build self-adaptive learners that improve their bias dynamically according to the characteristics of the domain under analysis. We believe such view can unify current efforts, leading them in a promising direction.

The construction of self-adaptive learners poses major challenges to the research community. This paper highlights several of those challenges and provides suggestions on possible research avenues. For example, how can we assess if the bias adopted by a learning algorithm suits the domain under analysis?, or how can we construct relevant meta-features to characterize a domain?. Our analysis unveils the importance of having a metric over the space of hypotheses, the space of families of hypotheses, and the space of hypothesis orderings (Section 3.1.1 and Section 3.1.1). Such metrics can tell us how far or close is our final estimation to the true target function. Unfortunately, few has been done in this area [14]; future work will investigate plausible models for this type of metrics.

Finally, a major challenge in the construction of self-adaptive learners is how to incorporate a flexible bias in both the base-learner and the meta-learner (Section 3.1.1). Adding meta-learners on top of existing ones does not eliminate the problem as long as there is at least one meta-learner having a fixed form of bias. Future work will explore how to make two self-adaptive learners serve as meta-learners for each other, in this way ensuring full flexibility in the dynamic selection of bias.

Acknowledgments

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

References

[1] Baxter Jonathan. A Model Of Inductive Bias. *Journal of Artificial Intelligence Research*, 12, 149–198, 2000.

[2] Chan Philip and Stolfo S. On The Accuracy Of Meta-Learning For Scalable Data Mining. *Journal of Intelligent Integration of Information*, 1998.

[3] DesJardins Marie And Gordon Diana. Evaluation And Selection Of Biases In Machine Learning. *Machine Learning*, 20, 5–22, 1995.

[4] Gama J. and Brazdil P. . Characterization Of Classification Algorithms. In *7th Portuguese Conference on Artificial Intelligence, EPIA*, 189–200, 1995.

[5] Lanzi Pier Luca, Stolzmann Wolfgang and Wilson Stewart. Learning Classifier Systems. Lecture Notes in *Artificial Intelligence*, Springer-Verlag, New York, NY, 2000.

[6] Li Ming and Vitányi Paul M. B. . *An Introduction To Kolmogorov Complexity And Its Applications.* Springer Verlag, New York, 1993.

[7] Mitchell Tom. The Need For Biases In Learning Generalizations. Tech. rep. CBM-TR-117, Computer Science Department, Rutgers University, New Brunswick, NJ 08903, 1980.

[8] Pratt Lorien and Thrun Sebastian. Second Special Issue On Inductive Transfer. *Machine Learning*, 28, 1997.

[9] Rendell Larry, Seshu Raj and Tcheng David. Layered Concept-Learning And Dynamically-Variable Bias Management. In *Proceedings of Tenth International Joint Conference on Artificial Intelligence*, 08–314, Milan, Italy, 1987.

[10] Schaffer Cullen. A Conservation Law For Generalization Performance. In *Proceedings of the Eleventh International Conference on Machine Learning*, 259–265, San Francisco: Morgan Kaufmann, 1994.

[11] Thrun Sebastian. Lifelong Learning Algorithms. In Thrun S., and Pratt, L. (Eds.), *Learning To Learn*, Chap. 8 : 181–209. Kluwer Academic Publishers, 1998.

[12] Watanabe Satosi. *Knowing And Guessing.* John Wiley and Sons, New York, 1969.

[13] Wolpert David. The Lack Of A Priori Distinctions Between Learning Algorithms And The Existence Of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8, 1341-142, 1996.

[14] Wolpert David. Any Two Learning Algorithms Are (Almost) Exactly Identical. Unpublished manuscript. NASA Ames Research Center, 2001.